Hiroshi Umeo Shin Morishita
Katsuhiro Nishinari Toshihiko Komatsuzaki
Stefania Bandini (Eds.)

LNCS 5191

# Cellular Automata

8th International Conference on Cellular Automata
for Research and Industry, ACRI 2008
Yokohama, Japan, September 2008, Proceedings

Springer

Lecture Notes in Computer Science 5191

*Commenced Publication in 1973*
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Hiroshi Umeo  Shin Morishita
Katsuhiro Nishinari  Toshihiko Komatsuzaki
Stefania Bandini (Eds.)

# Cellular Automata

8th International Conference on Cellular Automata
for Research and Industry, ACRI 2008
Yokohama, Japan, September 23-26, 2008
Proceedings

Springer

Volume Editors

Hiroshi Umeo
University of Osaka Electro-Communication
572-8530 Neyagawa, Osaka, Japan
E-mail: umeo@cyt.osakac.ac.jp

Shin Morishita
Yokohama National University
240-8501 Yokohama, Japan
E-mail: mshin@ynu.ac.jp

Katsuhiro Nishinari
University of Tokyo
Graduate School of Engineering
113-8656 Tokyo, Japan
E-mail: tknishi@mail.ecc.u-tokyo.ac.jp

Toshihiko Komatsuzaki
Kanazawa University
920-1192 Kanazawa, Japan
E-mail: toshi@t.kanazawa-u.ac.jp

Stefania Bandini
University of Milan–Bicocca
20126 Milano, Italy
E-mail: bandini@disco.unimib.it

Library of Congress Control Number: 2008932925

CR Subject Classification (1998): F.1.1, F.2.2, I.6, C.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-79991-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-79991-7 Springer Berlin Heidelberg New York



Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12447084 06/3180 5 4 3 2 1 0

# Preface

This volume constitutes the proceedings of the 8th International Conference on Cellular Automata for Research and Industry, ACRI 2008, which took place in Yokohama, Japan, September 23-26, 2008. The conference, which was organized by Yokohama National University, was the eighth in a series of conferences inaugurated in 1994 in Rende, Italy, and followed by ACRI 1996 in Milan, Italy, ACRI 1998 in Trieste, Italy, ACRI 2000 in Karlsruhe, Germany, ACRI 2002 in Geneva, Switzerland, ACRI 2004 in Amsterdam, The Netherlands and ACRI 2006 in Perpignan, France.

The ACRI conference has been traditionally focused on challenging problems and new research not only in theoretical but application aspects of cellular automata, including cellular automata tools and computational sciences. It is also concerned with applications and solutions of problems from the fields of physics, engineering, environment science, social science and life sciences. Its primary goal is to discuss problems from a variety of scientific fields, to identify new issues and to enlarge the research fields of cellular automata. Since its inception, the ACRI conference has attracted an ever-growing community and has raised knowledge and interest in the study of cellular automata for both new entrants into the field as well as researchers already working on particular aspects of cellular automata.

First invented by von Neumann, cellular automata models have been popularized and investigated in many areas during the last few decades. They provide a mathematically rigorous framework for a class of discrete dynamical systems that allow complex, unpredictable behavior to emerge from the deterministic local interactions of many simple components operating in parallel and distributed manner.

ACRI 2008 brought together over 100 distinguished mathematicians, computer scientists, and other researchers working in the field of cellular automata theory and its applications. A special interest was devoted to the general concepts, theories, methods and techniques associated with modeling, analysis, and implementation in various systems including biological, physical, ecological, social systems and so on. Cellular automata are classically modeled as a regular grid with synchronicity and homogeneity. ACRI 2008 encouraged recent trends which consider asynchronous, inhomogeneous cellular automata with unstructured environments. In order to highlight the multidisciplinary nature of the cellular automata research area, the Second International Workshop on Crowds and Cellular Automata C&CA 2008 and the Third International Workshop on Natural Computing IWNC 2008 were organized as satellite workshops within the scope of ACRI 2008.

The volume contains 65 refereed papers addressing various important topics in cellular automata, covering theoretical results and highlighting potential applications. A total of 43 papers were presented as oral talks and 22 papers were

presented as posters during the conference by speakers coming from about 23 different countries. These papers were selected among 78 submitted contributions. Each paper was reviewed by at least two members of the scientific committees. We are extremely grateful to these reviewers, who accepted the difficult task of selecting papers. Their expertise and efficiency ensured the high quality of the conference. The volume also contains 11 extended abstracts dealing with crowds and cellular automata, which were presented during the C&CA 2008 workshop.

Four invited speakers of worldwide reputation presented the latest results in the field of cellular automata. We would like to take this opportunity to express our sincere thanks to Debashish Chowdhury from the Indian Institute of Technology, Leon Chua from the University of California, Berkeley, Norio Konno from Yokohama National University, and Andreas Schadschneider from the Universität zu Köln, who kindly accepted our invitation to give plenary lectures.

It should be stressed that this conference would have been impossible without the help and continuous encouragements from a number of people, especially the members of the Steering Committees who strongly supported the organization of ACRI 2008 in Yokohama. First of all, we would like to thank the authors who showed their interest in ACRI 2008 by submitting their papers for consideration. We wish to extend our gratitude to Stefania Bandini, one of the organizers of the C&CA 2008 workshop and Yasuhiro Suzuki, Andrew Adamatzky, and Masami Hagiya, the organizers of IWNC 2008, who helped to introduce the ACRI conference to related scientific communities. A special word of thanks goes to Sara Manzoni for the assistance with C&CA 2008. It is a pleasure to express our sincere thanks to our colleagues of the Organizing Committees. A special word of thanks goes to Toshihiko Shiraishi for his work during the organization of the conference and the assistance he provided to the participants. Finally, the organization of ACRI 2008 was made possible thanks to financial and technical support of the board and several departments of Yokohama National University, the University of Osaka Electro-Communication, the University of Tokyo, Kanazawa University, Kozo Keikaku Engineering Inc., PTV (Germany), and the City of Yokohama.

September 2008

Hiroshi Umeo
Shin Morishita
Katsuhiro Nishinari
Toshihiko Komatsuzaki
Stefania Bandini

# Organization

## ACRI 2008 Organization

| | |
|---|---|
| Conference Chairs | Hiroshi Umeo (Chair)<br>Shin Morishita (Co-chair)<br>Katsuhiro Nishinari (Co-chair) |
| Conference Secretary | Toshihiko Komatsuzaki |
| Local Organizing Committees | Shin Morishita, Toshihiko Shiraishi, Toshihiko Komatsuzaki, Katsuhiro Nishinari, Hiroshi Umeo |
| C&CA Workshop Chairs | Stefania Bandini, Shin Morishita, Katsuhiro Nishinari |
| Natural Computing Workshop Chairs | Yasuhiro Suzuki, Andrew Adamatzky, Masami Hagiya, Hiroshi Umeo |

## International Steering Committee

| | |
|---|---|
| Stefania Bandini | University of Milano-Bicocca (Italy) |
| Bastien Chopard | University of Geneva (Switzerland) |
| Giancarlo Mauri | University of Milano-Bicocca (Italy) |
| Hiroshi Umeo | University of Osaka Electro-Communication (Japan) |
| Thomas Worsch | University of Karlsruhe (Germany) |

## Referees

S. Adachi (Japan)
A. Adamatzky (UK)
F. Bagnoli (Italy)
S. Bandini (Italy)
O. Bandman (Russia)
B. Ben Youssef (Canada)
L. Berec (Czech Republic)
P. P. Chaudhuri (India)
B. Chopard (Switzerland)
A. Deutsch (Germany)
S. Di Gregorio (Italy)
M. Droz (Switzerland)
W. Dzwinel (Poland)
A. El Jai (France)
S. El Yacoubi (France)
G. Fujita (Japan)
S. Gourbiere (France)
A. Hoekstra (The Netherlands)

K. Imai (Japan)
T. Isokawa (Japan)
J. Kaandorp (The Netherlands)
E. Kita (Japan)
A. Kitajima (Japan)
T. Komatsuzaki (Japan)
J. Kroc (Czech Republic)
A. T. Lawniczak (Canada)
J. Lee (Japan)
J.-Q. Liu (Japan)
P. Maji (India)
D. Makowiec (Poland)
N. Matsui (Japan)
G. Mauri (Italy)
M. Meyer-Hermann (Germany)
A. Mingarelli (Canada)
S. Morishita (Japan)
K. Nishinari (Japan)
H. Nishio (Japan)
F. Peper (Japan)
R. Serra (Italy)
T. Shiraishi (Japan)
M. C. Simon (France)
G. Sirakoulis (Greece)
P. Sloot (The Netherlands)
G. Spezzano (Italy)
D. Talia (Italy)
G. Tempesti (Switzerland)
M. Tomassini (Switzerland)
L. Torenvliet (The Netherlands)
A. Uejima (Japan)
H. Uoi (Japan)
H. Umeo (Japan)
R. Vollmar (Germany)
B. Voorhees (Canada)
J. Weimar (Germany)
T. Worsch (Germany)
A. Zomaya (Australia)

## Sponsoring Institutions

Yokohama National University
University of Osaka Electro-Communication
University of Tokyo
Kanazawa University
Kozo Keikaku Engineering Inc.
PTV (Germany)
City of Yokohama

# Table of Contents

## Invited Papers



## Contributed Papers

## Part 1: Tools and Theory

## CA Theory and Implementation

## Computational Theory

## Part 2: CA Applications

### Physical Modeling



### Urban, Environmental and Social Modeling

## Pedestrian and Traffic Flow Modeling

## Crypto and Security



## System Biology

## CA-Based Hardware



## Crowds and Cellular Automata

# From CA to Gene Expression: Machines and Mechanisms

Debashish Chowdhury[1], Ashok Garai[1], Philip Greulich[2], Katsuhiro Nishinari[3], Andreas Schadschneider[2], Tripti Tripathi[1], and Jian-Sheng Wang[4]

[1] Physics Department, Indian Institute of Technology, Kanpur 208016, India
[2] Institute for Theoretical Physics, University of Cologne, 50937 Köln, Germany
[3] Department of Aeronautics and Astronautics, School of Engineering, University of Tokyo, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[4] Department of Physics, National University of Singapore, Singapore 117542, Singapore

**Abstract.** Molecular motors are proteins or macromolecular complexes which use input energy to perform mechanical work. Some of these motors move on filamentous proteins whereas other move on DNA or RNA strands. Often, many such motors move simultaneously on the same track and their collective movement is similar to vehicular traffic on highways. We have developed theoretical models of different types of molecular motor traffic by appropriately extending the totally asymmetric simple exclusion process (TASEP). Thus, our models of molecular motor traffic belong to the broad class of driven-diffusive lattice gas models which have close relations with cellular automata. By drawing analogy with vehicular traffic, we have introduced novel quantities for characterizing the nature of the spatio-temporal organization of molecular motors on their tracks. We show how the mechano-chemistry of the individual motors influence the traffic-like intracellular collective phenomena.

## 1 Introduction

Motility is the hallmark of life. Most of the motions of animals and plants arise from movements at the molecular level which are driven by motor proteins. A key feature of molecular motor transport is that the motor proteins move on filamentous "tracks" [1,2,3,4]. The tracks for motor proteins are made of either proteins or nucleic acids (e.g., DNA or RNA). A common feature of all these motors is that these perform mechanical work by utilizing some other form of input energy and hence the name "motor". All the molecular motors we consider in this paper directly convert chemical energy into mechanical work.

During several biological processes many motors move simultaneously on the same track. The collective movement of the motors under such cicumstances strongly resemble vehicular traffic flow [5,6]. Our aim is to analyze molecular motor traffic from the perspective of vehicular traffic [7,8]. In the "particle-hopping" models of vehicular traffic each vehicle is represented by a particle and the dynamics of the system is formulated in terms of "rules" which are

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 1–10, 2008.

reminiscent of the update rules of cellular automata (CA). To our knowledge, the totally asymmetric simple exclusion process (TASEP) [9] is the simplest model of a system of interacting self-propelled particles on a discrete lattice. It has been extended in several ways to formulate "particle-hopping" models for capturing various interesting aspects of vehicular traffic [7]. Our models of molecular motor traffic may be regarded as biologically motivated extensions of TASEP. In all the examples of molecular motor traffic considered in this paper, the states of the system (both position and chemical states) are represented by discrete variables. In the analytical formulation of our theories, time is treated as a continuous variable, whereas discretization of time is needed not only for the numerical integration of the equations of motion, but also for computer simulations of these models.

## 2 Traffic of Kinesins on Microtubule Track: Change of Lane?

The members of the *kinesin* superfamily of cytoskeletal motors move on microtubules (MT). These motors run on chemical fuel in the sense that the mechanical energy required for their movement is supplied from the energy released when adenosine triphosphate (ATP) is hydrolyzed to adenosine diphosphate (ADP) [4].

We have focussed attention on a family of single-headed kinesins, called KIF1A. Our original model (from now onwards, we shall refer to it as the NOSC model) [10,11] captures the essential steps of the mechano-chemical cycle of individual KIF1A motors as well as steric interactions on the same track. Normally, a single microtubule consists of thirteen protofilaments each of which is analogous to a "lane" for the molecular motors. Recently we have extended the NOSC model [12] by allowing additional processes which correspond to lane changing, i.e., allowing a motor to shift from one protofilament to a neighbouring one on the same MT.

The equispaced binding sites for KIF1A on a given protofilament of the MT are labelled by the integer index $i$ ($i = 1, ..., L$). We use the integer index $j$ to label the protofilaments; the position of each binding site is denoted completely by the pair $(i, j)$. We impose periodic boundary conditions along both the $i$- and $j$-directions.

A single biochemical cycle of a KIF1A motor consists of a sequence of four states, namely, kinesin (K), kinesin bound with ATP (KT), kinesin bound with ADP and phosphate (KDP) and, finally, kinesin bound with only ADP (KD). The motor binds strongly to the MT track in both the states K and KT; the state KDP has very short life time and KD binds weakly to the track. Therefore, at each spatial location in our simplified model, a KIF1A is allowed to exist in one of the two distinct "chemical" states depending on whether it is bound strongly or weakly to the track; these two chemical states are denoted by the symbols $S$ and $W$, respectively.

The allowed transitions and the corresponding rate constants are shown in fig.1. The rate constants $\omega_a$ and $\omega_d$ account for the attachments and detachments of the motors. The rate constant $\omega_b$ corresponds to the unbiased one-dimensional Brownian motion of the motor in the state where it is weakly bound to the

MT track. The rate constant $\omega_h$ is associated with the process driven by ATP hydrolysis which causes the transition of the motor from the state $S$ to the state $W$. The rate constants $\omega_f$ and $\omega_s$, together, capture the Brownian ratchet mechanism of movement of a KIF1A motor [10,11].

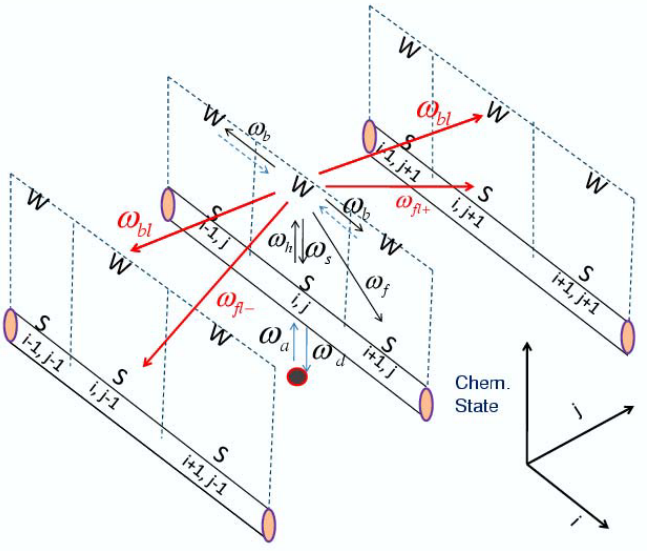



**Fig. 1.** A schematic description of the mechano-chemical cycle of a single-headed kinesin motor KIF1A in our extended model where lane changings are allowed. The equispaced sites labelled by the integers $..., i-1, i, i+1, ...$ denote the binding sites of the motor on a given protofilament of the microtubule (MT) track while the integer index $j$ labels the protofilaments. The symbols $S$ and $W$ denote the two "chemical" states of the motor in which it is, respectively, strongly and weakly bound to the track. The allowed transitions are indicated by the arrows and the symbols accompanying the arrows are the corresponding rate constants.

The rules of time evolution in the extended NOSC model proposed in ref.[12] are identical to those in the NOSC model, except for the following additional lane-changing rules (see fig.1):

A motor weakly-bound (i.e., in state $W$) to the binding site $i$ on the protofilament $j$ is allowed to move to the positions $(i, j+1)$ and $(i, j-1)$

(i) *without* simultaneous change in its chemical state, both the corresponding rates being $\omega_{bl}$;
(ii) *with* simultaneous transition to the chemical state $S$, the corresponding rate constants being $\omega_{fl+}$ and $\omega_{fl-}$, respectively.

As in the earlier TASEP-type models of cytoskeletal motor traffic [13,14,15,16,17,18], none of the lattice sites is allowed to be occupied by more than one motor at a time.

Let $S_i(j,t)$ and $W_i(j,t)$ denote the probabilities for a motor to be in the "chemical" states $S$ and $W$, respectively, at site $i$ on the protofilament $j$. In the extended NOSC model, under mean-field approximation, the master equations for the probabilities $S_i(j,t)$ and $W_i(j,t)$ are given by

$$\begin{aligned}\frac{dS_i(j,t)}{dt} &= \omega_a[1-S_i(j,t)-W_i(j,t)] - \omega_h S_i(j,t) - \omega_d S_i(j,t) + \omega_s W_i(j,t) \\ &+\omega_f W_{i-1}(j,t)[1-S_i(j,t)-W_i(j,t)] \\ &+\omega_{fl+}[W_i(j-1,t)][1-S_i(j)-W_i(j)] \\ &+\omega_{fl-}[W_i(j+1,t)][1-S_i(j)-W_i(j)], \end{aligned} \tag{1}$$

$$\begin{aligned}\frac{dW_i(j,t)}{dt} &= \omega_h S_i(j,t) - \omega_s W_i(j,t) - \omega_f W_i(j,t)[1-S_{i+1}(j,t)-W_{i+1}(j,t)] \\ &-\omega_b W_i(j,t)[2-S_{i+1}(j,t)-W_{i+1}(j,t)-S_{i-1}(j,t)-W_{i-1}(j,t)] \\ &+\omega_b[W_{i-1}(j,t)+W_{i+1}(j,t)][1-S_i(j,t)-W_i(j,t)] \\ &+\omega_{bl}[W_i(j-1,t)+W_i(j+1,t)][1-S_i(j,t)-W_i(j,t)] \\ &-\omega_{bl}W_i(j,t)[2-S_i(j+1,t)-W_i(j+1,t)-S_i(j-1,t) \\ &-W_i(j-1,t)-\omega_{fl+}W_i(j,t)[1-S_i(j+1,t)-W_i(j+1,t)] \\ &-\omega_{fl-}W_i(j,t)[1-S_i(j-1,t)-W_i(j-1,t)]. \end{aligned} \tag{2}$$

Solving these equations analytically, we address a fundamental question: does lane changing increase or decrease flux per lane?

In the steady state under *periodic* boundary conditions, $\tilde{S} = S_i(j,t)$ and $\tilde{W} = W_i(j,t)$, independent of $t$ and irrespective of $i$ and $j$; from eqs.(1) and (2), we get

$$\tilde{S} = \frac{-\tilde{\Omega}_h - \tilde{\Omega}_s - (\tilde{\Omega}_s - 1)K + \sqrt{\tilde{D}}}{2K(1+K)} \tag{3}$$

$$\tilde{W} = \frac{\tilde{\Omega}_h + \tilde{\Omega}_s + (\tilde{\Omega}_s + 1)K - \sqrt{\tilde{D}}}{2K}, \tag{4}$$

where $K = \omega_d/\omega_a$, $\tilde{\Omega}_h = \omega_h/\tilde{\omega_f}$, $\tilde{\Omega}_s = \omega_s/\tilde{\omega_f}$, with $\tilde{\omega_f} = \omega_f + \omega_{fl+} + \omega_{fl-}$, and

$$\tilde{D} = 4\tilde{\Omega}_s K(1+K) + (\tilde{\Omega}_h + \tilde{\Omega}_s + (\tilde{\Omega}_s - 1)K)^2. \tag{5}$$

The average total density of the motors attached to each filament of the MT in the steady state is given by

$$\rho = \tilde{S} + \tilde{W} = \frac{\tilde{\Omega}_h + \tilde{\Omega}_s + (\tilde{\Omega}_s + 1)K - \sqrt{\tilde{D}} + 2}{2(1+K)}. \tag{6}$$

Using the expressions (3) and (4) for $\tilde{S}$ and $\tilde{W}$, respectively, in the expression

$$J = \omega_f \tilde{W}(1 - \tilde{S} - \tilde{W}) \tag{7}$$

for the flux of KIF1A motors per lane of the MT highway, we get

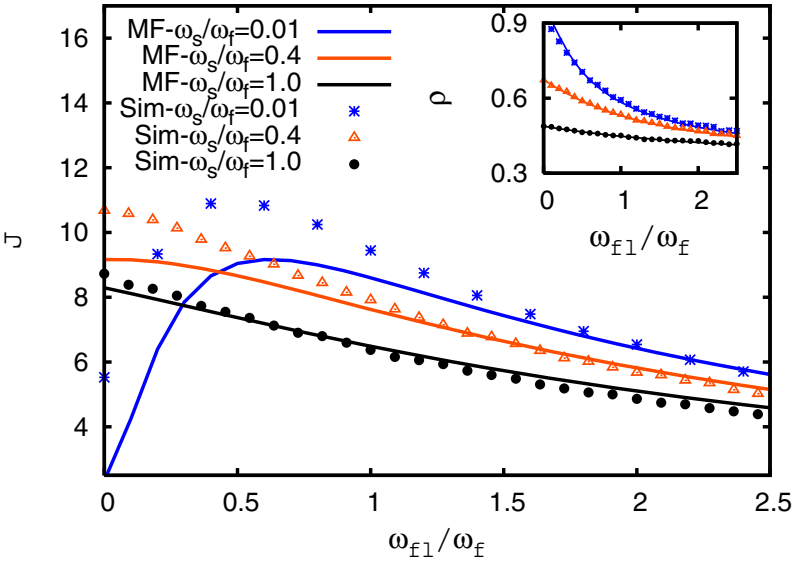



**Fig. 2.** (Color online) Flux per lane (and, in the inset, average density of the motors on each lane) are plotted against $\omega_{fl}/\omega_f$ for a few values of $\omega_s/\omega_f$. Our mean-field predictions (labelled MF) are plotted by lines while the discrete data points (labelled Sim) have been obtained from our computer simulations of the model.

$$J = \frac{\omega_f \left[ K^2 - \left( \tilde{\Omega}_h + (1+K)\tilde{\Omega}_s - \sqrt{\tilde{D}} \right)^2 \right]}{4K(1+K)}. \tag{8}$$

For graphical presentation of our main results, we use the estimates of the rate constants which were extracted earlier [11] from empirical data on single KIF1A (we use $\omega_h = 125\ \mathrm{s}^{-1}$). Since no estimate of $\omega_{fl+}$ and $\omega_{fl-}$ are available, we use $\omega_{fl+} = \omega_{fl-} = \omega_{fl}$ and vary the single parameter $\omega_{fl}/\omega_f$ over a wide range to explore the consequences of different rates of lane changing. We have also carried out computer simulations of the extended NOSC model and computed the flux $J$ of the motors per protofilament as well as the average density $\rho$; a comparison of the predictions of our mean-field theory with these simulation data establishes the level of accuracy of our mean-field theoretical predictions.

In Fig. 2 we plot $J$ (obtained from eqn. (8)) and $\rho$ (given by eqn.(6)) against $\omega_{fl}/\omega_f$ for several different values of $\omega_s/\omega_f$ and compare with the corresponding simulation data. When $\omega_s/\omega_f$ is sufficiently high, the density $\rho$ is small even in the absence of lane changing ($\omega_{fl} = 0$); in that case, the motors feel hardly any steric hindrance. In this regime, increasing $\omega_{fl}/\omega_f$ of $\omega_s/\omega_f$ has very little effect on the average speed of the motors; it is the decrease of density that is responsible for the *monotonic* decrease of $J$ with $\omega_{fl}$.

In sharp contrast, at sufficiently low values of $\omega_s/\omega_f$, $J$ varies *non-monotonically* with $\omega_{fl}/\omega_f$. In this regime of $\omega_s/\omega_f$, at $\omega_{fl} = 0$, the high density of $\rho$ causes steric

hindrances which, in turn, leads to small $J$. When $\omega_{fl}$ is "switched on", $\rho$ decreases with increasing $\omega_{fl}$ and $J$ increases up to a maximum because of the weakening of the hindrance effects. But, beyond a certain range of $\omega_{fl}/\omega_f$, the density of motors becomes so low that the movement of the motors is practically free of mutual hindrance; the decrease of $J$ beyond its maximum is caused by the further reduction of density. Larger difference between the predictions of our approximate analytical calculations and computer simulation data at lower values of $\omega_s/\omega_f$ arises from the fact that the mean-field approximation neglects correlations which increases with increasing densitity of the motors.

## 3 Traffic of RNAP Motors on DNA Tracks

The motor RNA polymerase (RNAP) catalyzes the polymerization of a RNA molecule from the corresponding DNA template [19] and the process is called transcription. To our knowledge, almost all the models of transcription available in the literature [20,21,22,23,24,25,26,27,28,29,30,31] capture only the stochastic mechano-chemistry of the individual RNAP motors. The effects of steric interactions of the RNAPs on the rate of RNA synthesis has been modelled only in a recent work by two of us [32,33].

The model reported originally in ref.[32] is described schematically in fig.3. Each lattice site corresponds to a nucleotide on the DNA template. The elongation of the growing RNA by one nucleotide leads to a forward stepping of the RNAP by one unit. A mechano-chemical cycle of the RNAP during elongation of RNA consists of the following major steps: (i) Nucleoside triphosphate (NTP) binding to the active site of the RNAP, (ii) NTP hydrolysis, (iii) release of pyrophosphate ($PP_i$) (which is produced by the hydrolysis of NTP), and (iv) simultaneous forward stepping of the RNAP. Since $PP_i$-release is known to be the rate-limiting step, we consider only two distinct chemical states $\mu$ of the RNAP; $\mu = 1$ refers to the state in which the RNAP is not bound to any $PP_i$ whereas $\mu = 2$ corresponds to the state with bound $PP_i$.

The processes corresponding to the rate constants $\omega_{12}$ and $\omega_{21}$ correspond to $PP_i$-release and its reverse reaction. The symbol $\omega_{21}^f$ is the rate of addition of one NTP to the elongating RNA whereas $\omega_{12}^b$ is that of the reverse reaction. The remaining four rate constants, namely, $\omega_{11}^f$, $\omega_{11}^b$, $\omega_{22}^f$, and $\omega_{22}^b$ correspond to polymerization/depolymerization of the RNA, by one monomer, unaided by the RNAP. Normally, a single RNAP is large enough to cover $r(> 1)$ successive nucleotides on the track. Therefore, a lattice site is *occupied* by a RNAP if it coincides with the leftmost of the $r$ sites representing that RNAP while the next $r - 1$ sites on its right are *covered* by the same RNAP. However, each RNAP can move forward or backward by only one site in each time step, irrespective of the numerical value of $r$. Moreover, forward or backward step of a RNAP is implemented only if the target site is not already covered by any other RNAP. Suppose, the total number of RNAPs on the DNA template is $N$. Then, $\rho = N/L$ is the *number density* of the RNAPs whereas the *coverage density* $\rho_{cov} = Nr/L$ is the total fraction of the nucleotides covered by all the RNAPs together.

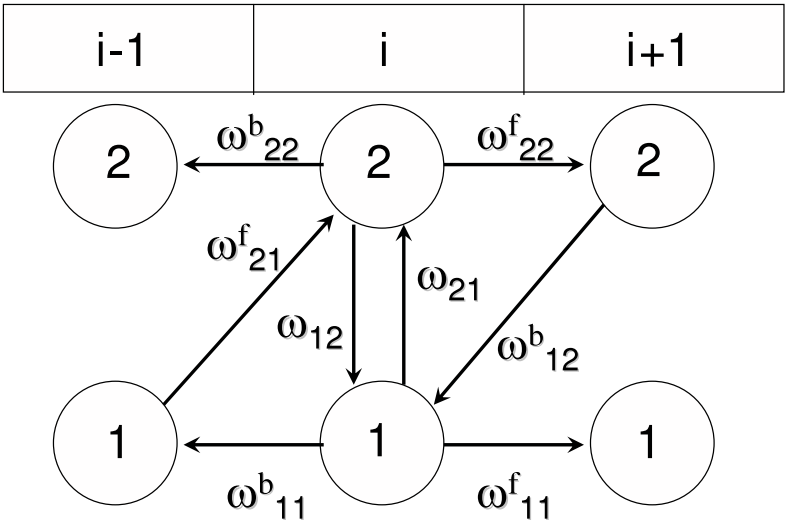



**Fig. 3.** A schematic representation of the mechano-chemical cycle of each RNAP in the model. The equispaced sites labelled by the integers $..., i-1, i, i+1, ...$ denote the nucleotides on the template DNA track. No $PP_i$ is bound to the RNAP in the state 1 whereas the pyrophosphate ($PP_i$)-bound state of the RNAP is labelled by the index 2. The allowed transitions denoted by arrows and the corresponding rate constants are also shown.

Let $P_\mu(i,t)$ be the probability that there is a RNAP at the lattice site $i$ and in the chemical state $\mu$ at time $t$. Let $Q(\underline{i}|j)$ be the conditional probability that, given a RNAP at site $i$, site $j$ is empty. In the mean-field approximation, the master equations for $P_\mu(i,t)$ are given by [32]

$$\begin{aligned}
\frac{dP_1(i,t)}{dt} &= \omega_{11}^f\ P_1(i-1,t)\ Q(\underline{i-1}|i-1+r) \\
&+ \omega_{11}^b\ P_1(i+1,t)\ Q(i+1-r|\underline{i+1}) \\
&+ \omega_{12}^b\ P_2(i+1,t)\ Q(i+1-r|\underline{i+1}) \\
&+ \omega_{12}\ P_2(i,t) - \omega_{21}\ P_1(i,t) \\
&- (\omega_{11}^f + \omega_{21}^f)\ P_1(i,t)\ Q(\underline{i}|i+r) \\
&- \omega_{11}^b\ P_1(i,t)\ Q(i-r|\underline{i})
\end{aligned} \tag{9}$$

$$\begin{aligned}
\frac{dP_2(i,t)}{dt} &= \omega_{22}^f\ P_2(i-1,t)\ Q(\underline{i-1}|i-1+r) \\
&+ \omega_{22}^b\ P_2(i+1,t)\ Q(i+1-r|\underline{i+1}) \\
&+ \omega_{21}^f\ P_1(i-1,t)\ Q(\underline{i-1}|i-1+r) \\
&+ \omega_{21} P_1(i,t) - \omega_{12}\ P_2(i,t) \\
&- (\omega_{22}^b + \omega_{12}^b)\ P_2(i,t)\ Q(i-r|\underline{i}) \\
&- \omega_{22}^f\ P_2(i,t)\ Q(\underline{i}|i+r)
\end{aligned} \tag{10}$$

In the steady state under periodic boundary conditions,

$$P_1 = \left( \frac{\omega_{12} + \omega_{12}^b Q}{\Omega_\updownarrow + \Omega_\leftrightarrow Q} \right) \rho$$
$$P_2 = \left( \frac{\omega_{21} + \omega_{21}^f Q}{\Omega_\updownarrow + \Omega_\leftrightarrow Q} \right) \rho \tag{11}$$

where

$$\Omega_\updownarrow = \omega_{12} + \omega_{21}$$
$$\Omega_\leftrightarrow = \omega_{21}^f + \omega_{12}^b \tag{12}$$

and $Q$ is given by

$$Q(\underline{i}|i+r) = Q(i|\underline{i+r}) = \frac{1-\rho r}{1+\rho-\rho r} \tag{13}$$

The corresponding steady-state flux is given by

$$J = \Omega_1 P_1 Q + \Omega_2 P_2 Q$$
$$= (\Omega_1 P_1 + \Omega_2 P_2) \left( \frac{1-\rho_{cov}}{1+\rho-\rho_{cov}} \right) \tag{14}$$

where

$$\Omega_1 = \omega_{11}^f + \omega_{21}^f - \omega_{11}^b$$
$$\Omega_2 = \omega_{22}^f - \omega_{12}^b - \omega_{22}^b, \tag{15}$$

are two *effective forward hopping rates* from the states 1 and 2, respectively.

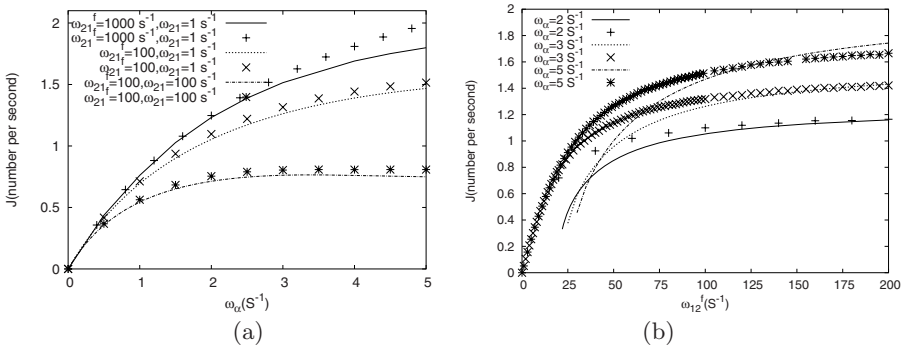



**Fig. 4.** The steady-state flux of the RNAPs, under open boundary conditions, plotted as a function of (a) $\omega_\alpha$, for three sets of values of the pair of parameters [NTP], [PP$_i$]; (b) $\omega_{21}^f$ for three values of the parameter $\omega_\alpha$. The lines correspond to our mean-field theoretic predictions whereas the discrete data points have been obtained from computer simulations.

However, for modeling transcription, open boundary conditions are more realistic than periodic boundary conditions. It is not difficult to write down the counterparts of the equations (10) and (11) under open boundary conditions, but cannot be solved analytically. Solving these equations numerically in the steady state, the numerical estimates of the flux and the average density profile were predicted [32]. These mean-field theoretic estimates are plotted as functions of $\omega_\alpha$ and $\omega_{21}^f$, respectively, in figs.4(a) and (b). In the same figure, the corresponding simulation data are also plotted for comparison. The trends of variation of the flux and the average density profile indicate a transition from the low-density phase to the maximal current phase in fig.4(a) and from the high-density phase to the maximal current phase in fig.4(b) [9].

## 4 Conclusion

In this article we have presented two examples of molecular motor traffic which have been modelled using the language of driven-diffusive lattice gases. Historically, however, first attempt in this direction was made in the context of traffic of ribosomes on a messenger RNA (mRNA) strand during translation of genetic message was developed [34,35,36,37,38,39,40,41,42,43,44,45]. Very recently, such models have been made more realistic by incorporating the mechano-chemistry of individual ribosomes [46,47] in the same spirit in which the models of RNAP traffic have been developed. It would be interesting to develop similar models for traffic-like collective phenomena exhibited by unicellular organisms.

This work is supported (through DC) by a research grant from CSIR (India).

## References

1. Howard, J.: Mechanics of Motor Proteins and the Cytoskeleton. Sinauer Associates (2001)
2. Schliwa, M. (ed.): Molecular Motors. Wiley-VCH, Chichester (2002)
3. Kolomeisky, A.B., Fisher, M.E.: Annual Review of Physical Chemistry. 58, 675 (2007)
4. Hackney, D.D., Tamanoi, F.: The Enzymes, vol. XXIII. Elsevier, Amsterdam (2004)
5. Chowdhury, D., Schadschneider, A., Nishinari, K.: Phys. of Life Rev. 2, 318 (2005)
6. Chowdhury, D., Chakrabarti, B.K., Dutta, A.: Common trends in traffic systems. Physica A 372(1) (2006)
7. Chowdhury, D., Santen, L., Schadschneider, A.: Phys. Rep. 329, 199 (2000)
8. Schadschneider, A.: In: This proceedings
9. Schütz, G.M.: Phase Transitions and Critical Phenomena, vol. 19. Acad. Press (2001)
10. Nishinari, K., Okada, Y., Schadschneider, A., Chowdhury, D.: Phys. Rev. Lett. 95, 118101 (2005)
11. Greulich, P., Garai, A., Nishinari, K., Schadschneider, A., Chowdhury, D.: Phys. Rev. E. 75, 041905 (2007)
12. Chowdhury, D., Garai, A., Wang, J.S.: Phys. Rev. E. 77(R), 050902 (2008)
13. Lipowsky, R., Klumpp, S., Nieuwenhuizen, T.M.: Phys. Rev. Lett. 87, 108101 (2001)

14. Lipowsky, R., Chai, Y., Klumpp, S., Liepelt, S., Müller, M.J.I.: Physica A. 372, 34 and references therein (2006)
15. Parmeggiani, A., Franosch, T., Frey, E.: Phys. Rev. Lett. 90, 86601 (2003); Phys. Rev. E 70, 046101 (2004)
16. Frey, E., Parmeggiani, A., Franosch, T.: Genome Inf. 15, 46 (2004) and references therein
17. Evans, M.R., Juhasz, R., Santen, L.: Phys. Rev. E. 68, 026117 (2003)
18. Popkov, V., Rakos, A., Williams, R.D., Kolomeisky, A.B., Schütz, G.M.: Phys. Rev. E. 67, 066117 (2003)
19. Bai, L., Santangelo, T.J., Wang, M.D.: Annu. Rev. Biophys. Biomol. Str. 35, 343 (2006)
20. Jülicher, F., Bruinsma, R.: Biophys. J. 74, 1169 (1998)
21. Wang, H.Y., Elston, T., Mogilner, A., Oster, G.: Biophys. J. 74, 1186 (1998)
22. Sousa, R.: Trends in Biochem. Sci. 21, 186 (1996)
23. Guajardo, R., Sousa, R.: J. Mol. Biol. 265, 8 (1997)
24. Guo, Q., Sousa, R.: J. Mol. Biol. 358, 241 (2006)
25. Bai, L., Shundrovsky, A., Wang, M.D.: J. Mol. Biol. 344, 335 (2004)
26. Bai, L., Fulbright, R.M., Wang, M.D.: Phys. Rev. Lett. 98, 068103 (2007)
27. Bar-Nahum, G., Epshtein, V., Ruckenstein, A.E., Rafikov, R., Mustaev, A., Nudler, E.: Cell. 120, 183 (2005)
28. Tadigotla, V.R., Maoileidigh, D.O., Sengupta, A.M., Epshtein, V., Ebright, R.H., Nudler, E., Ruckenstein, A.E.: Proc. Nat. Acad. Sci. USA. 103, 4439 (2006)
29. Yamada, Y.R., Peskin, C.S.: arxiv:q-bio.BM/0603012 (2006)
30. Woo, H.J.: Phys. Rev. E 74, 011907 (2006)
31. Voliotis, M., Cohen, N., Molina-Paris, C., Liverpool, T.B.: Biophys. J. 94, 334–348 (2008)
32. Tripathi, T., Chowdhury, D.: Phys. Rev. E. 77, 011921 (2008)
33. Tripathy, T.: Ph.D. Thesis, IIT Kanpur (in preparation)
34. MacDonald, C., Gibbs, J., Pipkin, A.: Biopolymers. 6, 1 (1968)
35. MacDonald, C., Gibbs, J.: Biopolymers. 7, 707 (1969)
36. Lakatos, G., Chou, T.: J. Phys. A. 36, 2027 (2003)
37. Shaw, L.B., Zia, R.K.P., Lee, K.H.: Phys. Rev. E. 68, 021910 (2003)
38. Shaw, L.B., Sethna, J.P., Lee, K.H.: Phys. Rev. E. 70, 021901 (2004)
39. Shaw, L.B., Kolomeisky, A.B., Lee, K.H.: J. Phys. A. 37, 2105 (2004)
40. Chou, T.: Biophys. J. 85, 755 (2003)
41. Chou, T., Lakatos, G.: Phys. Rev. Lett. 93, 198101 (2004)
42. Schönherr, G., Schütz, G.M.: J. Phys. A. 37, 8215 (2004)
43. Schönherr, G.: Phys. Rev. E 71, 026122 (2005)
44. Dong, J.J., Schmittmann, B., Zia, R.K.P.: J. Stat. Phys. 128, 21 (2007)
45. Dong, J.J., Schmittmann, B., Zia, R.K.P.: Phys. Rev. E. 76, 051113 (2007)
46. Basu, A., Chowdhury, D.: Phys. Rev. E 75, 021902 (2007)
47. Basu, A., Chowdhury, D.: Amer. J. Phys. 75, 931 (2007)

# What's New in Wolfram's New Kind of Science?

Leon Chua

University of California, Berkeley,
EECS Department, Berkeley, CA 94720-1770, USA
chua@eecs.berkeley.edu

**Abstract.** Wolframfs monumental best seller entitled gA new kind of scienceh was based almost entirely on brute-force computer simulations. In sharp contrast, this lecture presents a rigorous analytical theory based on attractors from a nonlinear dynamics perspective. New results and concepts to be presented include the partitioning (via Felix Kleinfs Vierergruppe) of all 256 local Boolean rules studied empirically by Wolfram into 88 global equivalence classes, one of which contains 4 topologically-conjugate rules capable of universal computation, and endowed with a 1/f spectrum. Another major result is the rigorous characterization of the time-asymptotic dynamics (attractors) of 112 local rules via an explicit generalized Bernoulli shift formula. Even more surprising, we have discovered the attractors of 170 local rules are blessed with the remarkable property of time-reversality. For such rules, the past evolution in time can be recovered from the future evolutions of a corresponding gtwinh rule. Only 86 local rules exhibit an garrow of timeh. One of our most fascinating discoveries is a new phenomenon, dubbed an gIsle of Eden,h having no counter part in hyperbolic differential equations, which has neither a past, nor a future! In addition to providing a mathematical foundation for brainlike dynamics, the discoveries cited above provide simple dynamical mechanism for mimicking many exotic phenomena from brain science, quantum physics, cosmology, etc.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, p. 11, 2008.

# Quantum Walks and Quantum Cellular Automata

Norio Konno

Graduate School of Engineering,
Yokohama National University,
79-5 Tokiwadai, Hodogaya-ku, Yokohama, 240-8501, Japan
konno@ynu.ac.jp

**Abstract.** The quantum walk is a generalization of classical random walks. Considerable work has been recently investigated on quantum walks in connection with quantum computing. The walk can be considered as a class of quantum cellular automata. There exists a one-to-one correspondence between them in a more general setting. Moreover we consider another class of quantum cellular automata which can be considered as a quantum version of stochastic cellular automata.

**Keywords:** quantum walk, quantum cellular automaton, stochastic cellular automaton.

## 1 Introduction

In recent years, the study of the quantum walk (QW) has gained much interest in the quantum information and computational research areas. The discrete-time QW with coin space can be considered as a quantum version of the random walk (RW). Ambainis et al. [1] studied intensively the behavior of the discrete-time QW. Patel, Raghunathan, and Rungta [10] constructed a QW without the coin space and analyzed the asymptotic behavior of the walk. In fact the QW can be considered as a class of quantum cellular automata (QCA) on the line. There exists a one-to-one correspondence between them in a more general setting. Recent reviews on QWs are found in Kendon [6], Konno [8]. We present another class of QCA which can be considered as a quantum version of stochastic cellular automata (SCA). The SCA is sometimes called the Domany-Kinzel model [2].[1]

## 2 QW

The one-dimensional QW is given by the following unitary matrix:

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{U}(2),$$

[1] In this manuscript we do not discuss a QCA introduced and studied by [4]. This QCA is a generalization of cellular automata.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 12–21, 2008.

where $a, b, c, d \in \mathbb{C}$. Here $\mathbb{C}$ is the set of complex numbers, and $\mathrm{U}(m)$ is the set of $m \times m$ unitary matrices. So we have $|a|^2 + |c|^2 = |b|^2 + |d|^2 = 1$, $a\overline{c} + b\overline{d} = 0$, $c = -\triangle\overline{b}$, $d = \triangle\overline{a}$, where $\overline{z}$ is a complex conjugate of $z \in \mathbb{C}$ and $\triangle = \det U = ad - bc$. The QW is a quantum generalization of the RW with an additional degree of freedom called the chirality. The chirality takes values left and right, and means the direction of the motion of the particle. The evolution of the QW is given by the following rule. At each time step, if the particle has the left chirality, it moves one unit to the left, and if it has the right chirality, it moves one unit to the right. More precisely, the unitary matrix $U$ acts on two chirality states $|L\rangle$ and $|R\rangle$: $|L\rangle \rightarrow a|L\rangle + c|R\rangle$, $|R\rangle \rightarrow b|L\rangle + d|R\rangle$, where $L$ and $R$ refer to the right and left chirality state respectively. In fact, define $|L\rangle = {}^T[1, 0]$, $|R\rangle = {}^T[0, 1]$, where $T$ indicates the transposed operator, so we have $U|L\rangle = a|L\rangle + c|R\rangle$, $U|R\rangle = b|L\rangle + d|R\rangle$. The study of the dependence of some important quantities on initial qubit state is one of the essential parts, so we define the set of initial qubit states as follows:

$$\Phi = \left\{ \varphi = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2 : |\alpha|^2 + |\beta|^2 = 1 \right\}.$$

Let $X_n$ be the QW at time $n$ starting from initial qubit state $\varphi \in \Phi$. It should be noted that $P(X_n = 0) = 1$. To explain $X_n$ briefly, we introduce the following matrices $P$ and $Q$:

$$P = \begin{bmatrix} a & b \\ 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}.$$

Remark that $U = P + Q$. For fixed $l$ and $m$ with $l + m = n$ and $-l + m = x$,

$$\Xi_n(l, m) = \sum_{l_j, m_j} P^{l_1} Q^{m_1} P^{l_2} Q^{m_2} \cdots P^{l_n} Q^{m_n}$$

summed over all $l_j, m_j \geq 0$ satisfying $l_1 + \cdots + l_n = l$, $m_1 + \cdots + m_n = m$ and $l_j + m_j = 1$. The definition of $\Xi_n(l, m)$ gives

$$P(X_n = x) = (\Xi_n(l, m)\varphi)^*(\Xi_n(l, m)\varphi) = \|\Xi_n(l, m)\varphi\|^2,$$

where $n = l + m$, $x = -l + m$ and $*$ means the adjoint operator. The Hadamard walk has been extensively investigated in the study of QWs (see Fig. 1). The unitary matrix $U$ of the walk is defined by the following Hadamard gate:

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

In contrast with the central limit theorem for the RW, Konno [7] gave the following new type of weak limit theorems in $abcd \neq 0$ case:

**Theorem 1.** *If $n \rightarrow \infty$, then $X_n/n \Rightarrow Z$, where $Z$ has a density $f(x) = f(x; \varphi = {}^T[\alpha, \beta])$:*

$$f(x) = \frac{\sqrt{1 - |a|^2}}{\pi(1 - x^2)\sqrt{|a|^2 - x^2}} \left\{ 1 - \left( |\alpha|^2 - |\beta|^2 + \frac{a\alpha\overline{b\beta} + \overline{a\alpha} b\beta}{|a|^2} \right) x \right\},$$

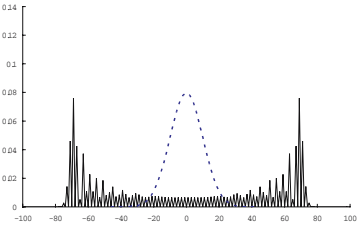



**Fig. 1.** Probability distributions of one-dimensional symmetric RW (dotted line) and Hadamard walk (solid line)

*for* $x \in (-|a|, |a|)$*, and* $f(x) = 0$ *for* $|x| \geq |a|$*, where* $Y_n \Rightarrow Y$ *means that* $Y_n$ *converges weakly to a limit* $Y$*.*

## 3 QCA and QW

We define the dynamics of a one-dimensional QCA including the model investigated by Patel, Raghunathan, and Rungta [10]. Let $\eta_n^{(m)}(x)(\in \mathbb{C})$ be the amplitude of the QCA at time $n \in \mathbb{Z}_+$ and at location $x \in \mathbb{Z}$ starting from $m \in \mathbb{Z}$, that is, $\eta_0^{(m)}(m) = 1$ and $\eta_0^{(m)}(x) = 0$ if $x \neq m$. Here $\mathbb{Z}$ (resp. $\mathbb{Z}_+$) is the set of (resp. non-negative) integers. Moreover, let $\zeta_n^{(m:\pm)}(x) = |\alpha\eta_n^{(m)}(x) + \beta\eta_n^{(m\pm1)}(x)|^2$, and $\zeta_n^{(m:\pm)} = (\zeta_n^{(m:\pm)}(x) : x \in \mathbb{Z})$, where $\alpha, \beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta|^2 = 1$. As we will show later, the $\zeta_n^{(m:\pm)}(x)$ is equivalent to a probability distribution of a QW at time $n$, where the pair $(\alpha, \beta)$ corresponds to the initial qubit state of the QW. The evolution of the QCA on the line is given by $\eta_{n+1}^{(m)} = \overline{U}\eta_n^{(m)}$, where $\overline{U}$ is the unitary matrix:

$$
\overline{U} = \begin{array}{c} \\ \vdots \\ -3 \\ -2 \\ -1 \\ 0 \\ +1 \\ +2 \\ +3 \\ +4 \\ \vdots \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} \dots & -3 & -2 & -1 & 0 & +1 & +2 & +3 & +4 & \dots \end{array} \\
\begin{pmatrix}
\ddots & . & . & . & . & . & . & . & . & \dots \\
\dots & b & a & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\
\dots & a & b & c & d & 0 & 0 & 0 & 0 & \dots \\
\dots & d & c & b & a & 0 & 0 & 0 & 0 & \dots \\
\dots & 0 & 0 & a & b & c & d & 0 & 0 & \dots \\
\dots & 0 & 0 & d & c & b & a & 0 & 0 & \dots \\
\dots & 0 & 0 & 0 & 0 & a & b & c & d & \dots \\
\dots & 0 & 0 & 0 & 0 & d & c & b & a & \dots \\
\dots & 0 & 0 & 0 & 0 & 0 & 0 & a & b & \dots \\
\dots & . & . & . & . & . & . & . & . & \ddots
\end{pmatrix}
\end{array},
$$

with $a, b, c, d \in \mathbb{C}$ and $\eta_n^{(m)}$ is the configuration

$$
\eta_n^{(m)} = {}^T[\dots, \eta_n^{(m)}(-1), \eta_n^{(m)}(0), \eta_n^{(m)}(+1), \dots],
$$

for any $n \in \mathbb{Z}_+$. Let $||u||^2 = \sum_{x=-\infty}^{\infty} |u(x)|^2$. The unitarity of $\overline{U}$ ensures that if $||\eta_0^{(m)}|| = 1$, then $||\eta_n^{(m)}|| = 1$ for any $n \in \mathbb{Z}_+$. Furthermore, if $||\zeta_0^{(m:\pm)}|| = 1$, then $||\zeta_n^{(m:\pm)}|| = 1$ for any $n \in \mathbb{Z}_+$. A little algebra reveals that $\overline{U}$ is unitary if and only if $|a|^2+|b|^2+|c|^2+|d|^2 = 1$, $a\overline{d}+\overline{a}d+b\overline{c}+\overline{b}c = a\overline{c}+b\overline{d} = a\overline{b}+\overline{a}b = c\overline{d}+\overline{c}d = 0$ (Condition C). Here we consider $a, b, c, d$ satisfying the above condition. Trivial cases are "$|a| = 1, b = c = d = 0$", "$|b| = 1, a = c = d = 0$", "$|c| = 1, a = b = d = 0$", and "$|d| = 1, a = b = c = 0$". For other cases, we have the following five types:

Type I: $|b|^2 + |c|^2 = 1$, $b\overline{c} + \overline{b}c = 0$, $bc \neq 0$, $a = d = 0$.
Type II: $|a|^2 + |b|^2 = 1$, $a\overline{b} + \overline{a}b = 0$, $ab \neq 0$, $c = d = 0$.
Type III: $|c|^2 + |d|^2 = 1$, $c\overline{d} + \overline{c}d = 0$, $cd \neq 0$, $a = b = 0$.
Type IV: $|a|^2 + |d|^2 = 1$, $a\overline{d} + \overline{a}d = 0$, $ad \neq 0$, $b = c = 0$.
Type V: $a, b, c, d$ satisfying Condition C and $abcd \neq 0$.

Both Types I and II are also trivial cases. To investigate non-trivial Types III - V, we consider two types of QWs. We see that a direct computation implies that $(a, b, c, d)$ satisfying Condition C has the following form:

$$(a, b, c, d) = e^{i\delta}(\cos\theta\cos\phi, -i\cos\theta\sin\phi, \sin\theta\sin\phi, i\sin\theta\cos\phi), \tag{1}$$

where $\theta, \phi, \delta \in [0, 2\pi)$. From now on, we assume that $(a, b, c, d)$ has the above form. Remark that the case studied by [10] is $\delta = \pi/2$ and $\theta = \phi = \pi/4$, that is, $(a, b, c, d) = (i/2, 1/2, i/2, -1/2)$, which belongs to Type V.

Both one-dimensional A-type and B-type QWs are given by the following unitary matrix:

$$U = \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \in \mathrm{U}(2).$$

For the $j$-type QW ($j = A, B$), let $\Psi_{j,n}(x)$ denote the amplitude at time $n$ at location $x$, where

$$\Psi_{j,n}(x) = \begin{bmatrix} \Psi_{j,n}^L(x) \\ \Psi_{j,n}^R(x) \end{bmatrix} \in \mathbb{C}^2,$$

with the chirality being left (upper component) or right (lower component). For each $j = A$ and $B$, the dynamics of $\Psi_{j,n}(x)$ for the $j$-type QW starting from the origin with an initial qubit state $\varphi \in \Phi$ is presented as the following transformation:

$$\Psi_{j,n+1}(x) = P_j\Psi_{j,n}(x+1) + Q_j\Psi_{j,n}(x-1), \tag{2}$$

where

$$P_A = \begin{bmatrix} a' & b' \\ 0 & 0 \end{bmatrix}, \quad Q_A = \begin{bmatrix} 0 & 0 \\ c' & d' \end{bmatrix} \quad \text{and} \quad P_B = \begin{bmatrix} a' & 0 \\ c' & 0 \end{bmatrix}, \quad Q_B = \begin{bmatrix} 0 & b' \\ 0 & d' \end{bmatrix}.$$

It is noted that $U = P_j + Q_j$ $(j = A, B)$. From (2), we see that the unitary matrix of the system is described as

$$\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & O & P_j & O & O & O & \dots \\ \dots & Q_j & O & P_j & O & O & \dots \\ \dots & O & Q_j & O & P_j & O & \dots \\ \dots & O & O & Q_j & O & P_j & \dots \\ \dots & O & O & O & Q_j & O & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{with} \quad O = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

for $j = A$ and $B$.

To begin with, we investigate a relation between the QCA and the A-type QW. To do so, the unitary matrix of the QCA

$$\overline{U} = \begin{array}{c} \\ \vdots \\ -3 \\ -2 \\ -1 \\ 0 \\ +1 \\ +2 \\ +3 \\ +4 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccccccc} \dots & -3 & -2 & -1 & 0 & +1 & +2 & +3 & +4 & \dots \end{array} \\ \begin{pmatrix} \ddots & . & . & . & . & . & . & . & . & \dots \\ \dots & b & a & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & a & b & c & d & 0 & 0 & 0 & 0 & \dots \\ \dots & d & c & b & a & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & a & b & c & d & 0 & 0 & \dots \\ \dots & 0 & 0 & d & c & b & a & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & a & b & c & d & \dots \\ \dots & 0 & 0 & 0 & 0 & d & c & b & a & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & 0 & a & b & \dots \\ \dots & . & . & . & . & . & . & . & . & \ddots \end{pmatrix} \end{array}$$

is rewritten as

$$\overline{U} = \begin{array}{c} \\ \vdots \\ -1 \\ 0 \\ +1 \\ +2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccc} \dots & -1 & 0 & +1 & +2 & \dots \end{array} \\ \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & T_A & Q_A & O & O & \dots \\ \dots & P_A & T_A & Q_A & O & \dots \\ \dots & O & P_A & T_A & Q_A & \dots \\ \dots & O & O & P_A & T_A & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array},$$

where

$$P_A = \begin{bmatrix} d & c \\ 0 & 0 \end{bmatrix}, \quad T_A = \begin{bmatrix} b & a \\ a & b \end{bmatrix}, \quad Q_A = \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}.$$

We consider a pair $(2x-1, 2x)$ in the QCA as a site $x$ in the A-type QW for any $x \in \mathbb{Z}$. Moreover we observe that $2x-1$ (resp. $2x$) site in the QCA corresponds

to right (resp. left) chirality at a site $x$ in the A-type QW. We call the QCA a *generalized A-type QW*. More precisely,

$$\Psi_{A,n}(x) = \begin{bmatrix} \Psi^R_{A,n}(x) \\ \Psi^L_{A,n}(x) \end{bmatrix}$$

and $\Psi_{A,n+1}(x) = Q_A\Psi_{A,n}(x+1) + T_A\Psi_{A,n}(x) + P_A\Psi_{A,n}(x-1)$. From the above observation, we see that "Type V QCA $\longleftrightarrow$ generalized A-type QW", where "$X \longleftrightarrow Y$" means that there is a one-to-one correspondence between $X$ and $Y$; that is,

$$\Psi^R_{A,n}(x) = \beta\eta^{(-1)}_n(2x-1) + \alpha\eta^{(0)}_n(2x-1), \quad \Psi^L_{A,n}(x) = \beta\eta^{(-1)}_n(2x) + \alpha\eta^{(0)}_n(2x),$$
$$\zeta^{(0:-)}_n(2x-1) = |\Psi^R_{A,n}(x)|^2, \quad \zeta^{(0:-)}_n(2x) = |\Psi^L_{A,n}(x)|^2.$$

We see that Type III is nothing but an A-type QW by interchanging $P_A$ and $Q_A$, and the roles of left and right chiralities with $c = b' = c'$, $d = a' = d'$. That is, "Type III QCA $\longleftrightarrow$ A-type QW".

As in the case of the A-type QW, we study a relation between the QCA and the B-type QW; that is, "Type V QCA $\longleftrightarrow$ generalized B-type QW". To do this, the unitary matrix of the QCA

$$\overline{U} = \begin{array}{c} \\ \vdots \\ -2 \\ -1 \\ 0 \\ +1 \\ +2 \\ +3 \\ +4 \\ +5 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccccccc} \dots & -2 & -1 & 0 & +1 & +2 & +3 & +4 & +5 \dots & \end{array} \\ \begin{pmatrix} \ddots & . & . & . & . & . & . & . & . & \dots \\ \dots & b & c & d & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & c & b & a & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & a & b & c & d & 0 & 0 & 0 & \dots \\ \dots & 0 & d & c & b & a & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & a & b & c & d & 0 & \dots \\ \dots & 0 & 0 & 0 & d & c & b & a & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & a & b & c & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & d & c & b & \dots \\ \dots & . & . & . & . & . & . & . & . & \ddots \end{pmatrix} \end{array}$$

is rewritten as

$$\overline{U} = \begin{array}{c} \\ \vdots \\ -1 \\ 0 \\ +1 \\ +2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccc} \dots & -1 & 0 & +1 & +2 & \dots \end{array} \\ \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & T_B & P_B & O & O & \dots \\ \dots & Q_B & T_B & P_B & O & \dots \\ \dots & O & Q_B & T_B & P_B & \dots \\ \dots & O & O & Q_B & T_B & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array},$$

where

$$P_B = \begin{bmatrix} d & 0 \\ a & 0 \end{bmatrix}, \quad T_B = \begin{bmatrix} b & c \\ c & b \end{bmatrix}, \quad Q_B = \begin{bmatrix} 0 & a \\ 0 & d \end{bmatrix}.$$

We consider a pair $(2x, 2x+1)$ in the QCA as a site $x$ in the B-type QW for any $x \in \mathbb{Z}$. Moreover we observe that $2x$ (resp. $2x+1$) site in the QCA corresponds to left (resp. right) chirality at site $x$ in the B-type QW. We call the QCA a *generalized B-type QW*. As in the case of the A-type QW, it is shown that "Type V QCA $\longleftrightarrow$ generalized B-type QW", that is,

$$\Psi^L_{B,n}(x) = \alpha\eta^{(0)}_x(2x) + \beta\eta^{(1)}_n(2x), \quad \Psi^R_{B,n}(x) = \alpha\eta^{(0)}_n(2x+1) + \beta\eta^{(1)}_n(2x+1),$$
$$\zeta^{(0:+)}_n(2x) = |\Psi^L_{B,n}(x)|^2, \quad \zeta^{(0:+)}_n(2x+1) = |\Psi^R_{B,n}(x)|^2.$$

We see that Type IV becomes a B-type QW with $d = a' = d'$, $a = b' = c'$; that is, "Type IV QCA $\longleftrightarrow$ B-type QW".

Meyer [9] has investigated the B-type QW, which was called a *quantum lattice gas automaton*. His case (for example, (24) in his paper [9]) can be obtained by $\delta \to 3\pi/2$, $\phi \to \rho$, and $\theta \to \pi/2 + \theta$ in (11).

From now on we consider a relation between Type V QCA and two-step QW. The meaning of the "two-step" is that we identify the one-step transition of Type V QCA with the two-step transition of the QW.

First "Type V QCA $\longleftrightarrow$ two-step A-type QW" is given. Next "Type V QCA $\longleftrightarrow$ two-step B-type QW" is also presented. Combining them all, we finally obtain the following relations:
"Type V QCA $\longleftrightarrow$ generalized A-type QW $\longleftrightarrow$ two-step A-type QW"
"Type V QCA $\longleftrightarrow$ generalized B-type QW $\longleftrightarrow$ two-step B-type QW"

Now we present "generalized A-type QW $\longleftrightarrow$ two-step A-type QW" in the following way. A direct computation implies that a generalized A-type QW with

$$P_A = \begin{bmatrix} d & c \\ 0 & 0 \end{bmatrix}, \quad T_A = \begin{bmatrix} b & a \\ a & b \end{bmatrix}, \quad Q_A = \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$$

is equivalent to a two-step A-type QW with

$$P_A(1) = \begin{bmatrix} i\cos\phi\, e^{i\theta_2} & \sin\phi\, e^{i\theta_2} \\ 0 & 0 \end{bmatrix}, \quad P_A(2) = e^{i\delta} \begin{bmatrix} \sin\theta\, e^{-i\theta_2} & -i\cos\theta\, e^{i\theta_1} \\ 0 & 0 \end{bmatrix},$$

and

$$Q_A(1) = \begin{bmatrix} 0 & 0 \\ \sin\phi\, e^{i\theta_1} & i\cos\phi\, e^{i\theta_1} \end{bmatrix}, \quad Q_A(2) = e^{i\delta} \begin{bmatrix} 0 & 0 \\ -i\cos\theta\, e^{-i\theta_2} & \sin\theta\, e^{-i\theta_1} \end{bmatrix},$$

for any $\theta_1, \theta_2 \in [0, 2\pi)$ such that $P_A = P_A(2)P_A(1)$, $Q_A = Q_A(2)Q_A(1)$, $T_A = P_A(2)Q_A(1) + Q_A(2)P_A(1)$. Note that $(a, b, c, d)$ has the form given in (11) and $U(n) \equiv P_A(n) + Q_A(n)$ is unitary for $n = 1, 2$.

In a similar fashion, we show that "generalized B-type QW $\longleftrightarrow$ two-step B-type QW"; that is, a generalized B-type QW with

$$P_B = \begin{bmatrix} d & 0 \\ a & 0 \end{bmatrix}, \quad T_B = \begin{bmatrix} b & c \\ c & b \end{bmatrix}, \quad Q_B = \begin{bmatrix} 0 & a \\ 0 & d \end{bmatrix}$$

corresponds to a two-step B-type QW with

$$P_B(1) = \begin{bmatrix} i\cos\phi\, e^{i\theta_2} & 0 \\ \sin\phi\, e^{i\theta_1} & 0 \end{bmatrix}, \quad P_B(2) = e^{i\delta} \begin{bmatrix} \sin\theta\, e^{-i\theta_2} & 0 \\ -i\cos\theta\, e^{-i\theta_2} & 0 \end{bmatrix},$$

and

$$Q_B(1) = \begin{bmatrix} 0 & \sin\phi\, e^{i\theta_2} \\ 0 & i\cos\phi\, e^{i\theta_1} \end{bmatrix}, \quad Q_B(2) = e^{i\delta} \begin{bmatrix} 0 & -i\cos\theta\, e^{i\theta_1} \\ 0 & \sin\theta\, e^{-i\theta_1} \end{bmatrix},$$

for any $\theta_1, \theta_2 \in [0, 2\pi)$ such that $P_B = P_B(2)P_B(1)$, $Q_B = Q_B(2)Q_B(1)$, $T_B = P_B(2)Q_B(1) + Q_B(2)P_B(1)$. Remark that $P_B(n) + Q_B(n) = P_A(n) + Q_A(n)$ for $n = 1, 2$.

The asymptotic result by [10] can be easily derived from Theorem 1 and the connection between the QCA and the two-step QW. The detailed discussion can be found in [3].

## 4 QCA and SCA

The Domany-Kinzel model is a class of SCA in one dimension with two parameters $(p, q) \in [0, 1]^2$. The model can be identified with the mixed site-bond oriented percolation model on a square lattice for a special case. When $p = 1$, $q = 0$, it becomes Wolfram's rule 90.

In this section, we consider another class of QCA which can be considered as a quantum version of the Domany-Kinzel model. First we introduce the following unitary matrix as in Katori et al. [5]:

$$Q_4 = \begin{bmatrix} a & \cdot & c & \cdot \\ \cdot & c & \cdot & e \\ b & \cdot & d & \cdot \\ \cdot & d & \cdot & f \end{bmatrix} \in \mathrm{U}(4),$$

where we represent 0 by "$\cdot$". The unitarity gives $|a|^2 + |b|^2 = |c|^2 + |d|^2 = |e|^2 + |f|^2 = 1$, $a\overline{c} + b\overline{d} = c\overline{e} + d\overline{f} = 0$.

For examples,

$$Q_4^{(1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ -1 & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & -1 \end{bmatrix}, \qquad Q_4^{(2)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & -1 & \cdot \\ \cdot & -1 & \cdot & 1 \end{bmatrix}, \qquad Q_4^{(3)} = \begin{bmatrix} \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix}.$$

Note that $Q_4^{(3)}$ is equivalent to Wolfram's rule 90.

The local update rule is given by $11 \to 11$ with amplitude $a$ or $01$ with amplitude $b$, $10 \to 10$ with amplitude $c$ or $00$ with amplitude $d$, $01 \to 11$ with amplitude $c$ or $01$ with amplitude $d$, $00 \to 10$ with amplitude $e$ or $00$ with amplitude $f$. The Domany-Kinzel model corresponds to $a = q$, $b = 1 - q$, $c = p$, $d = 1 - p$, $e = 0$, $f = 1$. However, in the case of the Domany-Kinzel model, the amplitude is considered as the probability.

The dynamics of the QCA on $\{1, 2, \ldots, N, N + 1\}$ is determined by $Q$:

$$Q = Q_{N+1,N}\, Q_{N,N-1}\, Q_{N-1,N-2} \cdots Q_{2,1},$$

where $Q_{x+1,x}$ is the operator which defines the amplitude that a site $x$ is occupied at time $n+1$ given the state of QCA at sites $x, x+1$ at time $n$. $Q_{x+1,x}$ is a $2^{(N+1)} \times 2^{(N+1)}$ unitary matrix, however we may also represent it as a $4 \times 4$ unitary matrix $Q_4$.

Next we consider the $N = 4$ case. Then

$$Q_{2,1} = Q_4 \otimes I_2 \otimes I_2 \otimes I_2, \quad Q_{3,2} = I_2 \otimes Q_4 \otimes I_2 \otimes I_2,$$
$$Q_{4,3} = I_2 \otimes I_2 \otimes Q_4 \otimes I_2, \quad Q_{5,4} = I_2 \otimes I_2 \otimes I_2 \otimes Q_4,$$

where $I_n$ is the $n \times n$ identity matrix.

We apply $Q = Q_{5,4}\, Q_{4,3}\, Q_{3,2}\, Q_{2,1}$ to a configuration $\eta = (0,0,1,0,0)$, where we impose a fixed boundary condition:"0" at site $N+1 = 5$.

$$\begin{aligned}
&Q\,(0,0,1,0,0)\\
&= (I_2 \otimes I_2 \otimes I_2 \otimes Q_4)(I_2 \otimes I_2 \otimes Q_4 \otimes I_2)(I_2 \otimes Q_4 \otimes I_2 \otimes I_2)(Q_4 \otimes I_2 \otimes I_2 \otimes I_2)\\
&\qquad \left(\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right)\\
&= (I_2 \otimes I_2 \otimes I_2 \otimes Q_4)\,(I_2 \otimes I_2 \otimes Q_4 \otimes I_2)\,(I_2 \otimes Q_4 \otimes I_2 \otimes I_2)\\
&\qquad \left\{\left(e\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} + f\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right) \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right\}\\
&= (I_2 \otimes I_2 \otimes I_2 \otimes Q_4)\,(I_2 \otimes I_2 \otimes Q_4 \otimes I_2)\,(I_2 \otimes Q_4 \otimes I_2 \otimes I_2)\\
&\left(e\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} + f\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right)\\
&= (I_2 \otimes I_2 \otimes I_2 \otimes Q_4)\,(I_2 \otimes I_2 \otimes Q_4 \otimes I_2)\\
&\left(ce\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} + de\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right)\\
&\left(cf\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} + df\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}\right).
\end{aligned}$$

Therefore we obtain

$$\begin{aligned}
&Q\,(0,0,1,0,0)\\
&= c^2e^2\,(1,1,1,1,0) + c^2ef\,(1,1,1,0,0) + cde^2\,(1,1,0,1,0) + cdef\,(1,1,0,0,0)\\
&+ cde^2\,(1,0,1,1,0) + cdef\,(1,0,1,0,0) + d^2e^2\,(1,0,0,1,0) + d^2ef\,(1,0,0,0,0)\\
&+ c^2ef\,(0,1,1,1,0) + c^2f^2\,(0,1,1,0,0) + cdef\,(0,1,0,1,0) + cdf^2\,(0,1,0,0,0)\\
&+ cdef\,(0,0,1,1,0) + cdf^2\,(0,0,1,0,0) + d^2ef\,(0,0,0,1,0) + d^2f^2\,(0,0,0,0,0).
\end{aligned}$$

The probability that a configuration $(1,1,1,1,0)$ exists is $|c^2e^2|^2$. Moreover, the probability that site 1 is occupied is $|c^2e^2|^2 + |c^2ef|^2 + 2|cde^2|^2 + 2|cdef|^2 + |d^2e^2|^2 + |d^2ef|^2$ (Fig. 2 shows an example for rule $Q_4^{(1)}$ with a fixed boundary condition: "0" at $N+1 = 8$ site).

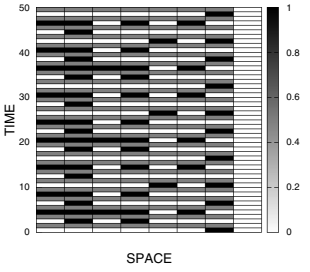



**Fig. 2.** The initial configuration is (0,0,0,0,0,0,1,0). Dark regions denote high probabilities and bright regions denote low probabilities.

## References


1. Ambainis, A., Bach, E., Nayak, A., Vishwanath, A., Watrous, J.: One-Dimensional Quantum Walks. In: Proc. of the 33rd Annual ACM Symposium on Theory of Computing, pp. 37–49 (2001)
2. Domany, E., Kinzel, W.: Equivalence of Cellular Automata to Ising Models and Directed Percolation. Phys. Rev. Lett. 53, 311–314 (1984)
3. Hamada, M., Konno, N., Segawa, E.: Relation between Coined Quantum Walks and Quantum Cellular Automata, RIMS Kokyuroku. No. 1422, 1–11 (2005) quant-ph/0408100
4. Inokuchi, S., Mizoguchi, Y.: Generalized Partitioned Quantum Cellular Automata and Quantization of Classical CA. Int. J. Unconv. Comput. 1, 149–160 (2005); Inui, N., Inokuchi, S., Mizoguchi, Y., Konno, N.: Statistical Properties for a Quantum Cellular Automaton, Phys. Rev. A, 72, 032323 (2005); Inui, N., Nakamura, K., Ide, Y., Konno, N.: Effect of Successive Observation on Quantum Cellular Automaton. J. Phys. Soc. Jpn. 76, 084001 (2007)
5. Katori, M., Konno, N., Sudbury, A., Tanemura, H.: Dualities for the Domany-Kinzel Model. J. Theor. Prob. 17, 131–144 (2004)
6. Kendon, V.: Decoherence in Quantum Walks - A Review. Math. Struct. in Comp. Sci. 17, 1169–1220 (2007)
7. Konno, N.: Quantum Random Walks in One Dimension. Quantum Inf. Process 1, 345–354 (2002); Konno, N.: A New Type of Limit Theorems for the One-Dimensional Quantum Random Walk. J. Math. Soc. Jpn. 57, 1179–1195 (2005)
8. Konno, N.: Quantum Walks, Lecture at the School, Quantum Potential Theory: Structure and Applications to Physics, held at the Alfreid Krupp Wissenschaftskolleg, Greifswald, 26 February - 9 March 2007 (Reihe Mathematik, Ernst-Moritz-Arndt-Universitat Greifswald, Preprint, No.2, 2007). Lecture Notes in Mathematics. Springer, Heidelberg (2007), http://www.math-inf.uni-greifswald.de/algebra/qpt/konno-26june07.pdf
9. Meyer, D.A.: From Quantum Cellular Automata to Quantum Lattice Gases. J. Statist. Phys. 85, 551–574 (1996)
10. Patel, A., Raghunathan, K.S., Rungta, P.: Quantum Random Walks do not need a Coin Toss. Phys. Rev. A. 71, 032347 (2005)

# Modelling of Transport and Traffic Problems

Andreas Schadschneider

Institut für Theoretische Physik, Universität zu Köln, 50937 Köln, Germany
as@thp.uni-koeln.de

**Abstract.** The Asymmetric Simple Exclusion Process (ASEP) is the simplest cellular automaton which captures the essential aspects of most transport and traffic phenomena. It describes the directed motion of particles obeying an exclusion principle. For specific applications, however, various generalizations of the ASEP are necessary. These are discussed for the case of highway traffic, ant trails, pedestrian dynamics and intracellular transport.

**Keywords:** traffic flow, jam, ASEP, ant trails, pedestrian dynamics, intracellular transport.

## 1 Introduction

Cellular automata (CA) have become an important tool for the investigation of traffic systems from both the theoretical as well as practical point of view [1,2,3,4]. The discreteness of all relevant variables (space, time, state) makes them ideally suited for high-performance computer simulations. However, with increasing computer power this advantage will become less important in the future. Instead the fact that the dynamics of CA models is usually based on intuitive rules is an important advantage. Especially in interdisciplinary applications, where the interactions between agents are not based on the fundamental physical forces, rule-based models allow to take into account e.g. psychological aspects in a natural and efficient way.

Here we first discuss the most fundamental CA model which describes traffic and transport problems, the ASEP (Sec. 2). We will see that already this extremely simple model is able to capture the basic properties of such system, but not all. Specific system require specific modifications to improve the realism of the model. This will be discussed for highway traffic (Sec. 3), traffic on ant trails (Sec. 4), pedestrian dynamics (Sec. 5) and intracellular transport (Sec. 6).

## 2 Asymmetric Simple Exclusion Process (ASEP)

Generically all traffic and transport systems belong to the class of nonequilibrium systems in which many fascinating effects can be observed [2,5,6,7]. The simplest model which captures the main features of such systems is the so-called *Asymmetric Simple Exclusion Process (ASEP)*. It is not only the "mother of all traffic models", but also a paradigmatic example of *driven diffusive systems*.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 22–31, 2008.

The ASEP describes the motion of particles which obey an exclusion principle, i.e. the space occupied by a particle is not available for others, on a discrete lattice. The dynamics is rather simple: A particle (•) moves to its empty right neighbour site (◦) with probability $q$ (... • ◦... $\stackrel{q}{\rightarrow}$ ... ◦ •...). In all other cases the particle will not move (... • ◦... $\stackrel{1-q}{\rightarrow}$ ... • ◦... and ... • •... $\stackrel{1}{\rightarrow}$ ... • •...). In physics usually a random-sequential update is used corresponding to continuous time dynamics. Then the hopping probability becomes a hopping rate. For most applications, discrete updates like synchronous (parallel) or sequential ones are more realistic because they provide a timescale for calibration. However, the qualitative behaviour does not change for the different updates [8].

For periodic boundary conditions the exact solution for the stationary state can be derived using various methods (see [1] and references therein). For random-sequential dynamics a site-oriented mean-field theory becomes exact, i.e. the occupations of neighbouring sites are uncorrelated. For parallel dynamics correlations are generated by Garden-of-Eden states that can not be reached by the dynamics [9]. In this case the interparticle distribution function factorizes and the stationary state is described exactly by a car-oriented mean-field theory.

The most important quantitative characterization of traffic systems is the *fundamental diagram* defined as the density-dependence of the flow, $J(\rho)$. For the ASEP the fundamental diagram can be given explicitly (Fig. 1) Due to the particle-hole symmetry of the rules it is symmetric around $\rho = 1/2$.

From a physics as well as a practical point of view open boundary conditions are more interesting. These are usually realized by "reservoirs" at both ends of the chain where particles can enter and exit with probabilities (or rates) $\alpha$ and $\beta$, respectively. This case has been studied extensively in recent years and is now well understood (see e.g. [6,7]). As in the periodic case, the stationary state can be obtained exactly [10,11], e.g. using the matrix-product Ansatz [7].

Fig. 1 shows the phase diagram obtained by varying the boundary rates $\alpha$ and $\beta$. The origin of the three phases can easily be understood. In the low-density phase (A) the current depends only on the input rate $\alpha$. The input is less efficient than the transport in the bulk or the output and therefore dominates the behaviour of the whole system. In the high-density phase (B) the output is the least efficient part. Therefore the current depends only on $\beta$. In the maximal current phase (C), input and output are more efficient than the transport in the bulk. Here the current has reached the largest possible value corresponding to the maximum of the fundamental diagram of the periodic system.

Quantitative predictions for the phase diagram and the boundary-induced phase transitions can be made using a coarse-grained description [12] which remains correct for the more sophisticated models discussed later. It relates the phase boundaries to properties of the periodic system which can be derived from the fundamental diagram, namely the so-called shock velocity $v_s$ and the collective velocity $v_c$. $v_s$ is the velocity of a 'domain-wall' which in nonequilibrium systems denotes an object connecting two possible stationary states. The collective velocity $v_c$ describes the velocity of the center-of-mass of a local perturbation in a homogeneous, stationary background of density $\rho$.

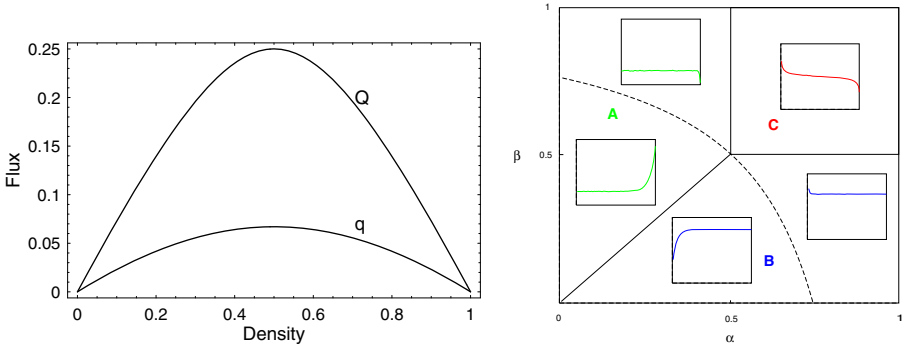



**Fig. 1.** Left: Fundamental diagram of the ASEP for two different hopping probabilities $Q > q$. Right: Phase diagram with low-density (A), high-density (B) and maximal-current phase (C). The insets show the typical shape of the density profiles.

The results for the ASEP show that boundary conditions play an important role in nonequilibrium systems. For most traffic applications periodic boundary conditions are not very realistic, e.g. since the number of vehicles fluctuates strongly due to on- and off-ramps.

## 3 Highway Traffic

### 3.1 Empirical Results

Theoretical results have to be compared with empirical observations on a qualitative or quantitative level. *Qualitative* results are usually related to the occurrence of spatio-temporal structures among like *jams*. *Quantitative results* like fundamental diagrams can be used for calibration of model parameters.

Two types of jams can be distinguished. *Bottleneck-induced jams* occur at locations of reduced capacity (*bottlenecks*) when the inflow is larger than this capacity. For *spontaneous jams* or *phantom jams* this is not true, at least not in an obvious way. Both empirical observations [13] as well as controlled experiments [14] indicate that growing instabilities can lead to jams even in the absence of bottlenecks. At intermediate densities the imperfect driving of human drivers can create a chain reaction where drivers overreact in braking manoevers which become necessary to avoid accidents when approaching the preceding car with large velocity.

In [15] it is argued that *all* jams are created by bottlenecks which are just sometimes not easy to identify. Often jams occur at the same location every day, especially close to ramps, sharp bends etc. However, these jams are not necessarily bottleneck-induced and might occur even though the local capacity has not yet been reached. Probably both mechanisms are relevant in the sense that inhomogeneities increase the probability of spontaneous jamming.

Quantitative results are obtained at many highway locations where empirical data are collected automatically by stationary inductive loops. Flow (current)

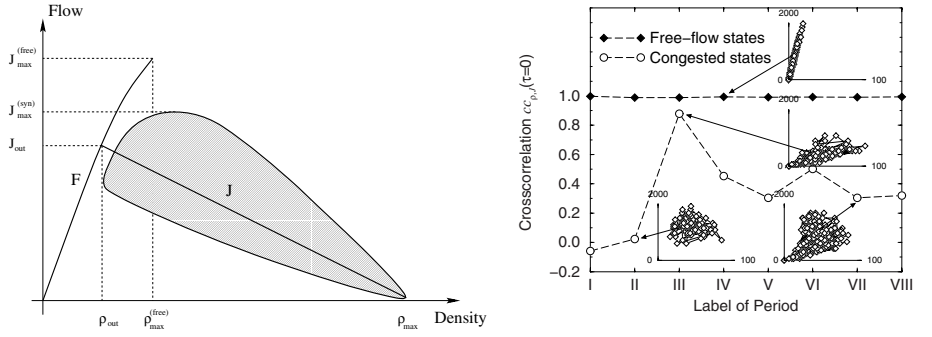



**Fig. 2.** Left: Schematic form of the fundamental diagram. $F$ denotes the free flow branch and $J$ the jam line. Right: Empirical cross-correlation function. Different periods of free-flow and congested traffic are labeled by I through VIII.

$J$ and velocity $v$ can be derived easily but density can not be measured locally and is usually determined using the hydrodynamic relation $J = \rho v$.

Nowadays three different phases of traffic flow are distinguished [16,17]: In *free flow* interactions between vehicles are rare. Cars move with their desired velocity and flow increases linearly with density (Fig. 2). States with flows larger than $J_{\text{out}}$ form the *metastable branch*. All states not of free flow type are called *congested states*. They are characterized by an average velocity smaller than the desired velocity of the drivers. Two congested phases can be distinguished. *Wide jams* are regions of high density and negligible average velocity and flow. The jam front moves upstream (opposite to the driving direction) at typical velocity $v_{\text{Jam}} \approx 15$ km/h. In *synchronized flow* [16] the average velocity is significantly lower than in free flow, but the flow can be much larger than in wide jams. Characteristic is the absence of a functional flow-density relation and data points are spread irregularly over a 2d area (Fig. 2). This leads to a vanishing cross-correlation function [18] between density $\rho$ and flow $J$ (Fig. 2).

Modern detectors provide detailed information about the microscopic structure of traffic flow, e.g. through the distribution of time-headways, optimal-velocity functions, correlation functions, cluster distributions etc. [18,19].

### 3.2 NaSch Model

The ASEP does neither reproduce spontaneous jam formation, which requires a mechanism that creates chain reactions of braking manoevers, nor the observed asymmetry of the fundamental diagram. To obtain a more realistic model an extension of the ASEP to higher velocities is necessary.

The Nagel-Schreckenberg (NaSch) model [20,21] is a probabilistic CA. The state of each car $n$ is characterized by its velocity $v_n = 0, 1, ..., v_{\max}$. The position of the $n$-th vehicle is denoted by $x_n$. Then $d_n = x_{n+1} - x_n - 1$ is its headway, i.e. the number of empty cells in front of it. At each time step $t \to t+1$, all cars are updated *in parallel* according to the following "rules":

*Step 1: Acceleration.*
If $v_n < v_{max}$, velocity is increased by 1, i.e. $v_n^{(1)} = \min(v_n + 1, v_{\max})$.

*Step 2: Deceleration (due to other cars).*
If $d_n < v_n^{(1)}$, velocity is reduced to $d_n$, i.e. $v_n^{(2)} = \min(v_n^{(1)}, d_n)$.

*Step 3: Randomization.*
If $v_n^{(2)} > 0$, velocity is decreased randomly by 1 with probability $p$, i.e.

$$v_n^{(3)} = \begin{cases} \max(v_n^{(2)} - 1, 0) & \text{with probability } p, \\ v_n^{(2)} & \text{with probability } 1 - p. \end{cases}$$

*Step 4: Vehicle movement.*
Each car is moved forward according to its new velocity $v_n = v_n^{(3)}$ determined in Steps 1–3, i.e. $x_n \to x_n + v_n$.

*Step 1* expresses the desire of the drivers to move as fast as possible (or allowed). *Step 2* reflects the interactions between vehicles and guarantees the absence of collisions in the model. *Step 3* incorporates many effects, e.g. natural fluctuations in the driving behaviour. It is responsible for spontaneous jam formation since it can lead to the chain reactions described above. Finally, in *Step 4* all cars will move with their new velocity as determined in the first three steps. This set of rules is minimal in the sense that every subset or change in the order will no longer produce realistic behaviour, e.g. spontaneous jams.

The timescale corresponding to one update step can be estimated in different ways [20]. Typical parameter values lead to timesteps which correspond to approximately 1 sec in real time which is of the same order of magnitude as the smallest relevant timescale in real traffic, the reaction time of the drivers.

The fundamental diagram of the NaSch model consists of a free flow and a congested branch (lines $F$ and $J$ in Fig. 2). However, it does not reproduce neither metastable states nor the synchronized phase.

### 3.3 Extensions of the NaSch Model

A simple modification of the NaSch model which reproduce the metastable states of high flow is the so-called *Velocity-Dependent-Randomization (VDR) model* [22] where the randomization parameter depends on the velocity of the car, $p = p(v)$:

*Step 0: Determination of the randomization parameter.*
The randomization parameter for the $n$-th car is given by $p = p(v_n(t))$.

This step is carried out before the acceleration *Step 1*. Metastable states occur for *slow-to-start rules* [22] where

$$p(v) = \begin{cases} p_0 & \text{for } v = 0, \\ p & \text{for } v > 0, \end{cases} \tag{1}$$

with $p_0 > p$, i.e. cars which have been standing in the previous timestep brake with higher probability $p_0$ than moving cars. This leads to fundamental diagrams

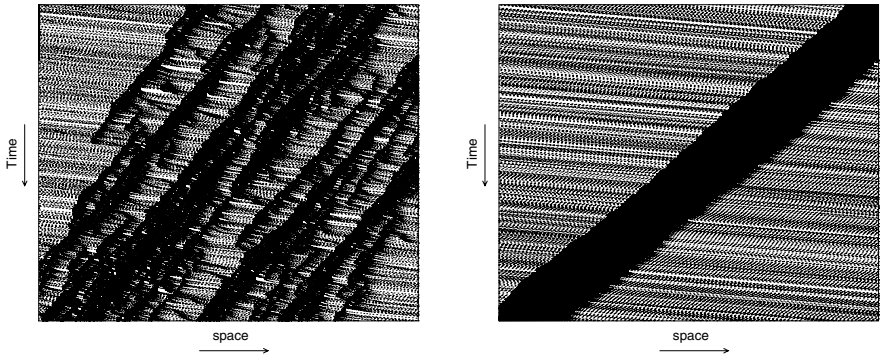



**Fig. 3.** Typical space-time diagrams of the NaSch model (left) and the VDR model for $p \ll p_0$ (right). One can clearly see the spontaneous jam formation in the NaSch model and different structure of the jams in both models.

which consist of the two branches $F$ and $J$ in Fig. 2, including the states with $J > J_{\text{out}}$. However, no synchronized traffic is found in this simple modification.

The macroscopic structure of the congested states is very different from that of the NaSch model [22]. It exhibits phase separation into a free flow region and a large jam which is almost compact for $p \ll 1$ (see Fig. 3). In contrast, in the NaSch model stop-and-go waves are found (Fig. 3). The structure of the free flow branch is very similar to that of the NaSch model. However, for $J > J_{\text{out}}$ the homogeneous free flow states are not stable, but can decay to a congested state through fluctuations or small perturbations.

The dynamics of the NaSch and VDR model is mainly based on the avoidance of accidents. This does not reproduce the synchronized phase and also the agreement with empirical data on a microscopic level is not very satisfactory [23]. Therefore it has been suggested that the desire of the drivers for smooth and comfortable driving is responsible for the occurrence of synchronized traffic [24]. This is realized through "anticipation" of the behaviour of preceding cars which reduces the risk of braking abruptly and allows for smoother driving. Thus the three observed traffic phases correspond to different driving strategies. In free flow, drivers drive as fast as possible and interactions are rare. In the jammed phase, the avoidence of accidents determines the behaviour and in synchronized traffic it is the desire to drive in a smooth and comfortable way.

These aspects are incorporated in the *brake light model* [25] which is able to reproduce all three phases and shows good agreement with detailed empirical single-vehicle data [23]. Anticipation is realized through brake lights which indicate (within an interaction horizon) velocity changes of the preceding car.

Similar ideas are used in the KKW model [26]. Drivers change their behaviour within a synchronization distance to the preceding vehicle where they try to move at the same velocity as the preceding car instead of maximizing their speed. Another approach [27] emphasizes the conflict between human overreaction and limited acceleration and deceleration capabilities as possible origin of congested traffic states. However, this model is no longer intrinsically accident free.

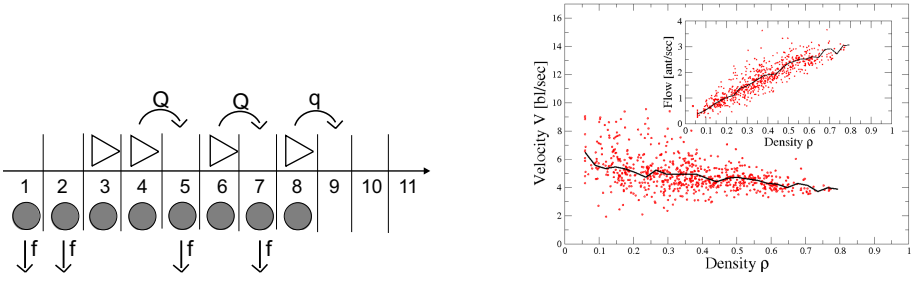



**Fig. 4.** Definition of the ant trail model (left): The symbols correspond to ants (▷) and pheromone marks (●). Fundamental diagrams from empirical observations (right).

## 4 Ant Trails

Ants form transport networks that have many similarities with human highway systems [28]. Essential for the formation and maintainance of these trails is a special form of chemical communication called *chemotaxis*. Ants mark their path by a chemical called pheromone that can be "smelled" by other ants which follow the trace to find food sources etc.

Chemotaxis can be incorporated into an ASEP-based model [29,30,31,32]. Now particles, corresponding to ants, move with two different hopping probabilities $Q$ and $q$ depending whether or not a pheromone is present at the target cell (Fig. 4). To model the trail following, $Q$ in the presence of pheromone should be larger than $q$ in the absence of pheromones. In order to model evaporation of the trace free pheromones (at sites without ant) are removed with probability $f$.

Surprisingly for small evaporation rates $f$ the fundamental diagram is qualitatively different from that of highway traffic. The average velocity is no longer a monotonically decreasing function of the density. Instead in the presence of chemotaxis [29,30,31,32] it can exhibit a maximum at a finite value of the density. This is related to the formation of platoons of ants.

Empirical observations show the absence of a jammed regime in natural trails of the species *Leptogenys processionalis* [33]. The average velocity is almost independent of the density and flow always increases with increasing density (Fig. 4). Also no overtaking was observed. Qualitative observations confirm the platoon formation predicted by the ASEP-based ant trail model [33].

## 5 Pedestrian Dynamics

Due to its generically 2d nature pedestrian motion is more difficult to describe in terms of simple models, but it exhibits many interesting collective effects and self-organization phenomena [34,17]: At large densities various kinds of *jamming* phenomena occur, e.g. when many people try to leave a large room at the same time. In counterflow, when two groups of people move in opposite directions, *lane formation* can occur. Pedestrians self-organize such that (dynamically varying)

lanes of unidirectional flow are formed. This reduces interactions with oncoming pedestrians and allows higher walking speeds. At bottlenecks, e.g. doors, counterflow can lead to *oscillations* of the flow direction. At intersections various collective patterns of motion like short-lived roundabouts can be formed which make the motion more efficient.

Several models for the description of pedestrian dynamics have been suggested [34,17]. The *social force model* [35,17] treats pedestrians as particles subject to long-ranged forces induced by the social behaviour of the individuals. This leads to (coupled) equations of motion similar to Newtonian mechanics.

In [36,37,38] a CA model has been introduced which takes its inspiration from the process of chemotaxis. It is in many respects a two-dimensional variant of the ant trail model of Sec. 4. Moving pedestrians create a "trace" which is, in contrast to chemotaxis, only virtual. Its main purpose is to transform effects of long-ranged interactions (e.g. following people walking some distance ahead) into a local interaction (with the "trace"). This allows for efficient simulations on a computer, especially in complex geometries.

This basic idea is realized through so-called *floor fields*. In one time step each pedestrian can move to one of the nine neighbouring cells in direction $(i, j)$ of a 2d lattice. The transition probabilities now depend on the strength of the floor field in the target cell. A motion in the direction of large fields is preferred.

In fact two different floor fields are used. The *static floor field* $S$ is constant and takes into account the geometry of the system (building). In order to model people leaving a room one uses a static floor which increases with decreasing distance to the exit. The second field, called *dynamic floor field* $D$, is just the virtual trace left by the pedestrians. In contrast to the ant trail model, where only the presence or absence was distinguished, the dynamic floor field can have different strengths. This allows to incorporate diffusion to neighbouring cells which corresponds to the broadening and dilution of the trace.

Finally the transition probabilities depend on the preferred walking direction and speed of each individual. This is encoded in the *matrix of preferences* $M$. Its entries are related to the preferred velocity vector and its longitudinal and transversal standard deviations [36].

The transition probability $p_{ij}$ in direction $(i, j)$ is then determined by

$$p_{ij} = N M_{ij} D_{ij} S_{ij} (1 - n_{ij}). \tag{2}$$

$N$ is a normalization factor to ensure $\sum_{(i,j)} p_{ij} = 1$ where the sum is over the nine possible target cells. The factor $1 - n_{ij}$, where $n_{ij}$ is the occupation number of cell $(i, j)$, takes into account that transitions to occupied cells are forbidden.

The details of the update rules can be found in [36,37,38]. There it is also shown that the floor-field model - despite its simplicity - allows to reproduce the empirically observed phenomena.

## 6 Molecular Motors

Intracellular transport is carried by *molecular motors* which are proteins that can directly convert chemical into mechanical energy required for their movement

along filaments constituting what is known as the *cytoskeleton* [39]. The cytoskeleton has many similarities with human-built road networks. *Microtubules* which are the tracks for long-range transport can be considered the analogues of highways. But in contrast to the latter, motors can enter and leave the microtubule at any location, not just at the ends or at on- or off-ramps. This so-called *Langmuir kinetics* can be incorporated into the ASEP by allowing particle creation and annihilation at *any* site of the lattice [40,41] not only the first an last one. This leads to the existence of novel phases, e.g. the coexistence of low and high density regimes, separated from each other by domain walls. Empirical evidence for such a phase has been found in [42].

For a more detailed discussion of intracellular transport we refer to [43].

## 7 Discussion

The ASEP is the most basic CA model for the description of transport and traffic problems. It captures only two vary basic features, namely the directionality of motion and the exclusion principle. This is already sufficient to reproduce many aspects at least qualitatively.

ASEP-based approaches are flexible enough to allow for simple and intuitive extensions that are able to provide a quantitive description in many situations. This has lead to interesting applications, e.g. traffic forecasting [44].

## References

1. Chowdhury, D., Santen, L., Schadschneider, A.: Phys. Rep. 329, 199 (2000)
2. Schmittmann, B., Zia, R.P.K.: Statistical Mechanics of Driven Diffusive Systems. In: Domb, C., Lebowitz, J.L. (eds.) Phase Transitions and Crititcal Phenomena, vol. 17. Academic Press, London (1995)
3. Maerivot, S., De Moor, B.: Phys. Rep. 419, 1 (2005)
4. Chowdhury, D., Nishinari, K., Schadschneider, A.: Phys. Life Rev. 2, 318 (2005)
5. Privman, V. (ed.): Nonequilibrium Statistical Mechanics in One Dimension. Cambridge University Press, Cambridge (1997)
6. Schütz, G.M.: Exactly Solvable Models for Many-Body Systems. In: Domb, C., Lebowitz, J.L. (eds.) Phase Transitions and Crititcal Phenomena, vol. 19, Academic Press, London (2001)
7. Blythe, R.A., Evans, M.R.: J. Phys. A 40, R333 (2007)
8. Rajewsky, N., Santen, L., Schadschneider, A., Schreckenberg, M.: J. Stat. Phys. 92, 151 (1998)
9. Schadschneider, A., Schreckenberg, M.: J. Phys. A 31, L225 (1998)
10. Derrida, B., Evans, M.R., Hakim, V., Pasquier, V.: J. Phys. A 26, 1493 (1993)
11. Schütz, G., Domany, E.: J. Stat. Phys. 72, 277 (1993)
12. Kolomeisky, A.B., Schütz, G., Kolomeisky, E.B., Straley, J.P.: J. Phys. A 31, 6911 (1998)
13. Treiterer, J.: Ohio State Technical Report No. PB 246 094 (1975)
14. Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S., Yukawa, S.: New J. Phys. 10, 033001 (2008)

15. Daganzo, C.F., Cassidy, M.J., Bertini, R.L.: Transp. Res. A. 33, 365 (1999)
16. Kerner, B.S.: The Physics of Traffic. Springer, Heidelberg (2004)
17. Helbing, D.: Rev. Mod. Phys. 73, 1067 (2001)
18. Neubert, L., Santen, L., Schadschneider, A., Schreckenberg, M.: Phys. Rev. E 60, 6480 (1999)
19. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Phys. Rev. E 65, 056133 (2002)
20. Nagel, K., Schreckenberg, M.: J. Physique I 2, 2221 (1992)
21. Schreckenberg, M., Schadschneider, A., Nagel, K., Ito, N.: Phys. Rev. E 51, 2339 (1995)
22. Barlovic, R., Santen, L., Schadschneider, A., Schreckenberg, M.: Eur. Phys. J. B. 5, 793 (1998)
23. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Phys. Rev. E. 70, 016115 (2004)
24. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Phys. Rev. E 65, 015101(R) (2002)
25. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: J. Phys. A 33, L477 (2000)
26. Kerner, B.S., Klenov, S.L., Wolf, D.E.: J. Phys. A 35, 9971 (2002)
27. Lee., H.K., Barlovic, R., Schreckenberg, M., Kim, D.: Phys. Rev. Lett. 92, 23702 (2004)
28. Hölldobler, B., Wilson, E.O.: The Ants. Belknap Press, Cambridge (1990)
29. Chowdhury, D., Guttal, V., Nishinari, K., Schadschneider, A.: J. Phys. A 35, L573 (2002)
30. Nishinari, K., Chowdhury, D., Schadschneider, A.: Phys. Rev. E 67, 036120 (2003)
31. Kunwar, A., John, A., Nishinari, K., Schadschneider, A., Chowdhury, D.: J. Phys. Soc. Jpn. 73, 2979 (2004)
32. Nishinari, K., Schadschneider, A., Chowdhury, D.: In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, p. 192. Springer, Heidelberg (2004)
33. John, A., Schadschneider, A., Chowdhury, D., Nishinari, K.: Swarm Intelligence 2 (2008)
34. Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., Seyfried, A.: Encyclopedia of Complexity and System Science. Springer, Heidelberg (to appear, 2009) (arxiv.org/abs/0802.1620)
35. Helbing, D., Molnar, P.: Phys. Rev. E51, 4282 (1995)
36. Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J.: Physica A 295, 507 (2001)
37. Kirchner, A., Nishinari, K., Schadschneider, A.: Phys. Rev. E 67, 056122 (2003)
38. Schadschneider, A., Kirchner, A., Nishinari, K.: LNCS, vol. 2492, p. 239. Springer, Heidelberg (2002)
39. Howard, J.: Mechanics of Motor Proteins and the Cytoskeleton. Sinauer Associates (2001)
40. Parmeggiani, A., Franosch, T., Frey, E.: Phys. Rev. Lett. 90, 086601 (2003)
41. Lipowsky, R., Klumpp, S., Nieuwenhuizen, T.M.: Phys. Rev. Lett. 87, 108101 (2001)
42. Nishinari, K., Okada, Y., Schadschneider, A., Chowdhury, D.: Phys. Rev. Lett. 95, 118101 (2005)
43. Chowdhury, D., et al.: These proceedings
44. http://www.autobahn.nrw.de

# Occurrence of Gliders in an Infinite Class of Life-Like Cellular Automata

Susumu Adachi[1], Ferdinand Peper[1,2,⋆], Jia Lee[3], and Hiroshi Umeo[4]

[1] Nano ICT Group
National Institute of Information and Communications Technology, Japan
[2] peper@nict.go.jp
[3] Celartem Technology Inc., Japan
[4] Dept. of Computer Science, Osaka Electro-Communication University, Japan

**Abstract.** The Game of Life (GL), Larger Than Life (LtL), and the Kaleidoscope of Life (KL) are cellular automaton (CA) models with a rich palette of configurations, some of which facilitate universal computation. Common to all these models is that the transition rules by which they are governed are outer-totalistic. The KL distinguishes itself by the striking simplicity of its transition rule, which does not even take into account a cell's state itself for its update. This paper investigates an infinite class of CA, all of which are similar to KL except for their differently sized neighborhoods. Characterized by a discrete parameter $d$, a neighborhood in such a CA consists of the cells at Moore distances 1, 2,..., or $d$ of a cell. We show that signal-carrying configurations ("gliders") occur in infinitely many of these CA models. We also show that the probability of convergence of a random configuration toward a dead cellular space increases with the increase in parameter $d$. These seemingly contradictory results suggest that the presence of gliders are not necessarily a reliable benchmark for the sustainability of Life in cellular space.

## 1 Introduction

Life in CA refers to a class of outer-totalistic models able to sustain computational universality through autonomously behaving configurations in cellular space. Some of these configurations—usually called *gliders*—propagate in cellular space as part of a cyclical metamorphosis process and interact with other gliders or with other more or less stable configurations in well-defined and well-understood ways. The *Game of Life (GL)* [1,2] is at the root of a class of Life-like models, which has extended into the *Larger Than Life (LtL)* class of CA [3,4] and recently has also given birth to the *Kaleidoscope of Life (KL)* [5,3]. The KL is based on a transition rule that uses a very simple decision criterion on whether a cell will be alive in the next generation or not: if the number of living cells at Moore-distance 1 or 2 from a cell equals the number 4, the cell will be alive in

⋆ Corresponding author.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 32–41, 2008.

the next generation. Remarkably, the state of the cell itself is irrelevant in this criterion. The independence of a cell's state from its previous state is shown in [5] to be closely connected to the properties of classical spin systems.

LtL covers a large class of models, part of which has been characterized in [4] by configurations called *bugs*. The KL, on the other hand, sports configurations more closely resembling those in the GL. While the dynamics of KL and GL are similar, the KL appears to be more "long-lived": it takes longer to reach convergence to stable or periodic configurations, probably due to the larger neighborhood used in comparison with the GL. This invites the question of whether models can be constructed that have similar characteristics to the KL and differ only in the size of their cells' neighborhoods.

This paper explores an infinite class of such models, and shows that each member of this class contains gliders. Yet, in most of these models, Life is hard to sustain, since random initial configurations will rapidly converge to a dead cell space, as will be shown. This puts KL at the boundary of a class of CA, which are increasingly unlikely to contain configurations capable of computation as the size of the cell neighborhood increases.

This research may result in novel CA models that are characterized by an extremely simple transition rule, to the extent that physical implementations in terms of magnetic spin systems are within the realm of possibilities. Such implementations may be at the basis of nanocomputer architectures, which have attracted increasingly attention in recent years due to their promise to extend the life time of Moore's law for another couple of decades beyond the decade or so it is still expected to last.

This paper is organized as follows. In Section 2 we define the basic model, followed by some prominent cell configurations in the KL in Section 3. Section 4 describes variations of the KL that have larger neighborhood diameters, and shows the presence of gliders in such models. Simulation results on the convergence of some of these models as well as a probabilistic analysis of the models are shown in Section 5. The paper finishes with conclusions and a discussion.

## 2 An Infinite Class of Life-Like Cellular Automata

The model consists of a 2-dimensional square array of cells, each of which can be in either of the states, 0 (*dead*) and 1 (*alive*). We assume that each cell in the cell space is identified by a unique integer, and that $\sigma_i(t)$ is the state of cell $i$ at time $t$ and $N(i)$ is the neighborhood of cell $i$. This neighborhood consists of the cells at orthogonal or diagonal distances $1, 2, ..., d$ from cell $i$ (Moore neighborhood), giving a total of $4d^2 + 4d$ neighbors, with $d$ denoting the radius of the neighborhood. Fig. 1 shows such a neighborhood for $d = 2$, which corresponds to the basic KL model in [5]. The transition rule of the model is defined in terms of the states of the cells in the neighborhood of cell $i$:

$$\sigma_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j \in N(i)} \sigma_j(t) = k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

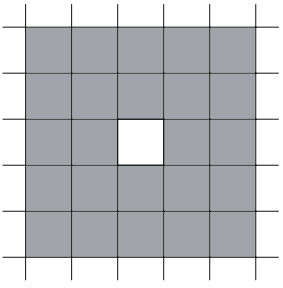


**Fig. 1.** Neighborhood of a cell (center) in the Kaleidoscope of Life (gray cells). The radius of the neighborhood is $d = 2$.

In other words, a cell becomes or stays alive if the number of living cells in its neighborhood is $k$; otherwise the cell dies or remains dead. It is assumed that $k = 2d$. All cells in the cell space undergo transitions simultaneously. Transition rules in which a cell's next state depends on the sum of the states in its neighborhood. are called *outer-totalistic*. If a cell's next state does not depend on the state of the cell itself, the rule is called *inner-independent*. The rule proposed in this paper is thus inner-independent outer-totalistic. A member of the infinite class of these CA is denoted by KL($d$). The traditional Kaleidoscope of Life in [5] is KL(2).

## 3 Useful Configurations in the Kaleidoscope of Life

The basic model of KL(2) supports a wide variety of configurations, of which only a few are used for computation. Signals are encoded in this model by configurations (called gliders) that dynamically propagate along cellular space. Two types of gliders used in computation in KL(2) are shown in Figs. 2 and 3, the first being only of peripheral use and the second type being the main mechanism to encode signals.

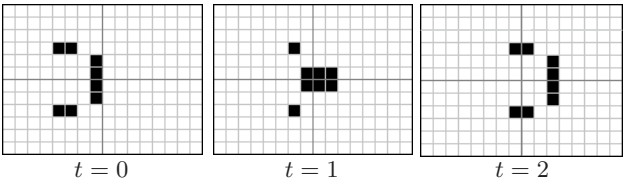



**Fig. 2.** A glider with period of 2 in KL(2). This glider appears in some configurations as an intermediate form, but it is not actively used for computation.

The *Hanabi* (meaning 'fireworks' in Japanese) is a periodical configuration that finds wide use in KL(2) to turn gliders to the left, to convert between the two types of gliders and to eliminate superfluous gliders (Fig. 4). It assumes a

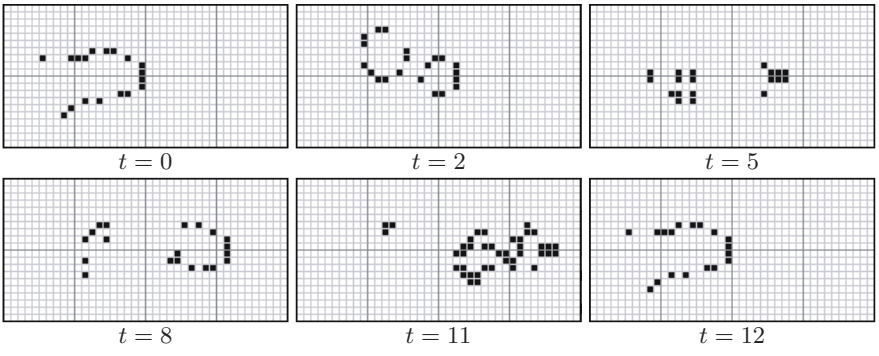



**Fig. 3.** A glider in KL(2) with a period of 12. This glider is used to encode signals on the cellular space.

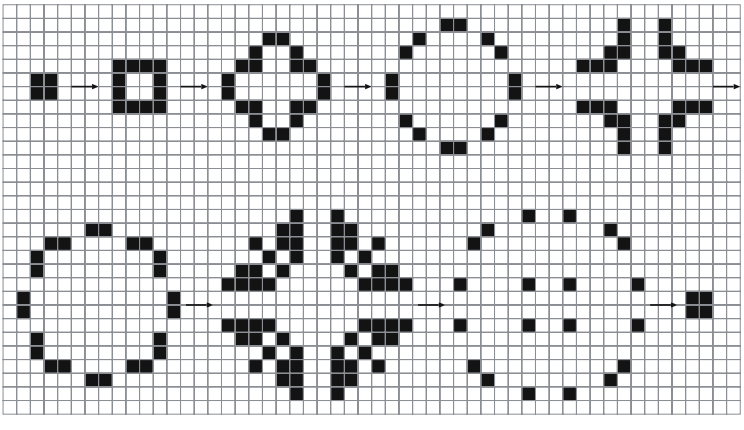


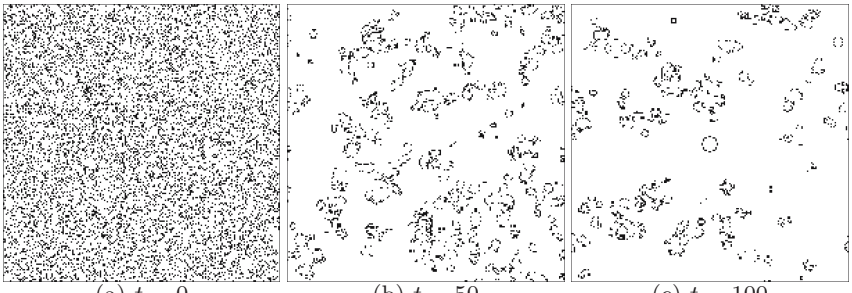
**Fig. 4.** The Hanabi pattern is used for a wide variety of tasks in KL(2), such as turning gliders to the left or right, conversion between glider types, and the elimination of gliders. It has a period of 8.

period of eight generations, and this period as well as the phase of the period is left undisturbed in most cases when the Hanabi interacts with a glider.

A good impression can be obtained from the nature of a Life-like CA by starting it with a random configuration, and let it run for a few hundred generations. This will usually result in some of the standard configurations to emerge, such as gliders and the Hanabi in the case of KL(2). Snapshots of the first 100 generations of KL(2) initialized randomly are shown in Fig. 5.

## 4 Gliders in the Infinite Class of Cellular Automata

The abundance of patterns in the KL(2) model invites the question whether the models in KL($d$) for different $d$ can sustain similar patterns. This section



(a) $t = 0$ (b) $t = 50$ (c) $t = 100$

**Fig. 5.** Evolution from a random initial configuration in KL(2) at times (a) $t = 0$, (b) $t = 50$, and (c) $t = 100$. The probability of a cell being alive in the initial configuration is 0.2.

gives a preliminary (positive) answer by showing that gliders are a very common phenomenon in KL($d$). In fact, we prove that infinitely many models in KL($d$) contain gliders. We focus on the model KL($d$) with the values $d = 4r$ for $r = 2, 3, 4, ....$ It turns out that all these models contain gliders. Examples of gliders in the KL(8) and KL(16) models are given in Figs. 6 and 7. These gliders have period 2 and they have very similar shapes, at time $t = 0$ as well as at time $t = 1$.

It turns out that this shape can be generalized for the case $d = 4r$ with $r = 2, 3, 4, ...$ (Fig. 8). The basic principle is that the glider at $t = 0$ generates two vertical bars and six isolated living cells at $t = 1$. The bar at the front of the signal (right side of the signal) is just sufficient in size to generate a new head of the glider at the right of it at $t = 2$, but a similar structure will not be created at the left of the front bar because of the six isolated cells. Instead a tail is generated at $t = 2$ that slightly bifurcates toward the isolated cells. The vertical bar at the tail at $t = 1$ (left side of the signal) is sufficiently short to have no influence on successive generations, as it dies out. The result is a glider at $t = 2$ that has advanced by $d$ cells in comparison with the glider at $t = 0$.

## 5 Probabilistic Analysis

Since the presence of gliders in all models of the form KL($4r$) suggests the occurence of Life, it makes sense to investigate the characteristics of these models in more detail. If we take a look at a simulation for the case KL(1), we see that an initial random configuration results in configurations that appear just as random as the initial one, and that appear to have an increase in the density of living cells (Fig. 9). Indeed, this model fails to show convergence to a set of standard patterns, as simulations have shown.

The situation is quite different for the KL(3) model. After less than 100 generations the cellular space is virtually dead in most cases (Fig. 10).

Simulations to measure the density of living cells in the cellular space show a very different behavior of KL($d$) in the cases $d = 1$, $d = 2$, and $d = 3$ (Fig. 11).

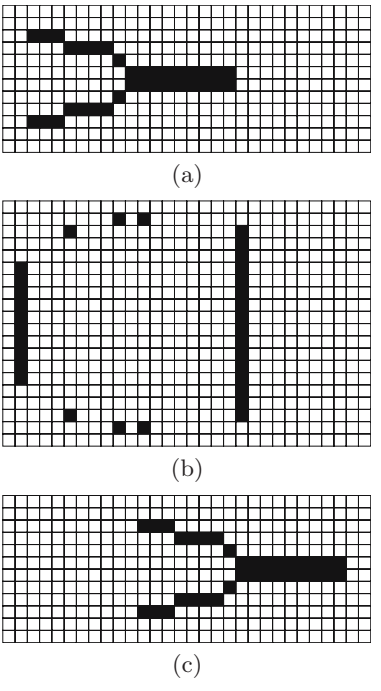



**Fig. 6.** Glider in KL(8) with period 2. (a) $t = 0$, (b) $t = 1$, and (c) $t = 2$.

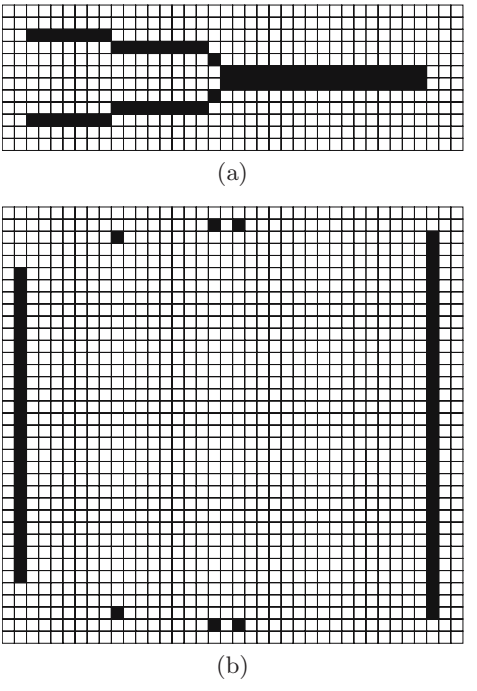



**Fig. 7.** Glider in KL(16) with period 2. (a) $t = 0$ and (b) $t = 1$.

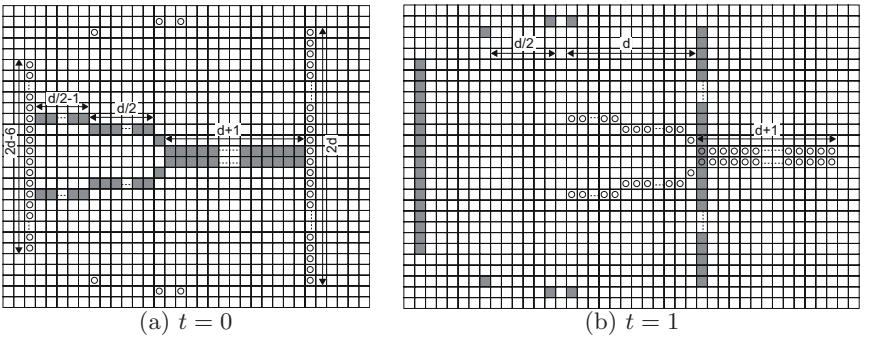
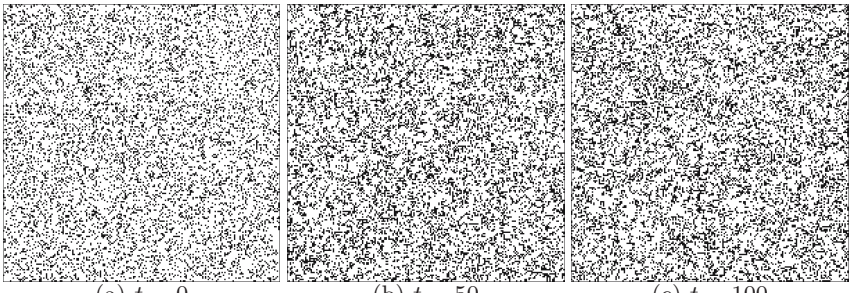


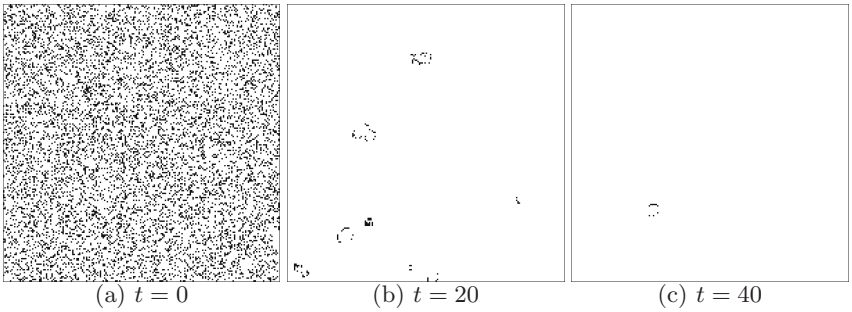

**Fig. 8.** Glider in KL($d$) with period 2 at (a) $t = 0$ and (b) $t = 1$. The gray cells denote the configuration at the indicated time $t$ and the cells containing circles denote the configuration as it would appear at $t + 1$.

KL(1) appears to have a density of living cells that stays at a high level over time, confirming the behavior observed in Fig. 9. KL(2) appears to have a gradual decrease in density to an asymptotic positive value, which indicates a gradual



(a) $t = 0$ (b) $t = 50$ (c) $t = 100$

**Fig. 9.** Evolution from a random initial configuration in KL(1) at times (a) $t = 0$, (b) $t = 50$, and (c) $t = 100$. The probability of a cell being alive in the initial configuration is 0.2.



(a) $t = 0$ (b) $t = 20$ (c) $t = 40$

**Fig. 10.** Evolution from a random initial configuration in KL(3) at times (a) $t = 0$, (b) $t = 20$, and (c) $t = 40$. The probability of a cell being alive in the initial configuration is 0.2.

convergence to a certain set of patterns. KL(3) sees the density of living cells decrease rapidly to 0, which indicates infertile grounds to sustain Life.

How will the behavior of KL($d$) be for higher values of $d$? To obtain a rough estimate of this, we adopt the mean-field approach in [6] and calculate the probability that a cell is alive at time $t = x+1$ given a certain probability of living cells at time $t = x$, which will be denoted by $p(x)$. We emphasize that this estimate has its limitations, since it assumes a random initial state of the cellular space as well as negligible correlations between neighboring cells. Especially the latter assumption fails to hold for CA in general, but the mean-field approximation still gives a useful first impression when comparing the CA models KL($d$) for different values of $d$. The probability $p(x)$ can be expressed as:

$$p(x + 1) = f(p(x)),$$

with

$$f(p) = \binom{n}{m} p^m (1 - p)^{n-m},$$

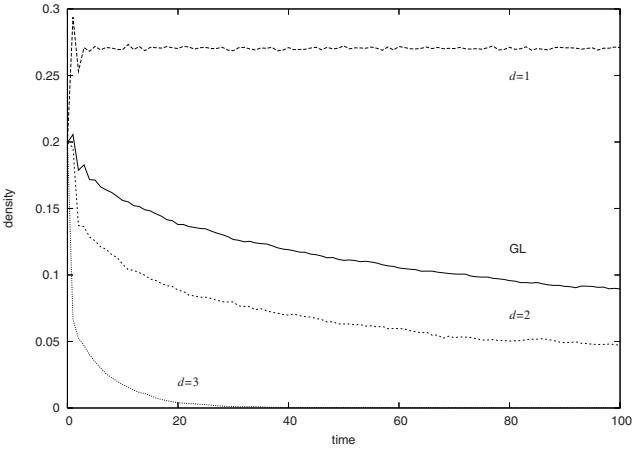
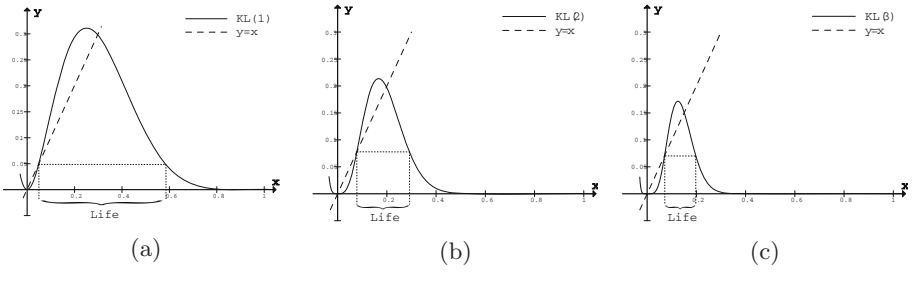



**Fig. 11.** Density of living cells in KL($d$) for $d = 1$, $d = 2$, and $d = 3$ over time. In the KL(1) model the density of living cells remains at a quite high level, which suggests either Life-like behavior or a lack of convergence. For the KL(2) model the density converges to a positive value, confirming the Life-like behavior of the model. In the KL(3) model the density converges rapidly to zero, suggesting that few configurations survive over time. As reference, the density curve of the Game of Life (GL) is also given.




(a) (b) (c)

**Fig. 12.** Mean-field approximations of (a) KL(1), (b) KL(2), and (c) KL(3). The horizontal axis denotes $p(x)$ and the vertical axis $p(x + 1)$. The area indicated by the text *Life* corresponds to the probability interval between which living cells are unlikely to die.

and $n = 4d^2 + 4d$ and $m = 2d$. Fig. 12 gives the graphs of $p(x + 1)$ (vertical) against $p(x)$ (horizontal) for the values $d = 1$, $d = 2$, and $d = 3$. The probability interval in which cells are unlikely to die—indicated by the phrase "Life"—is quite large for $d = 1$, smaller for $d = 2$, and much smaller for $d = 3$. This indicates that cells have a smaller probability of being alive for the higher values of $d$ than for the smaller values. To show that this trend holds in general, we investigate function $f$ in more detail. Its derivative is:

$$f'(p) = \binom{n}{m} p^{m-1}(1-p)^{n-m-1}(m - np)$$

This function has an $(m-1)$-fold root in $p=0$, an $(n-m-1)$-fold root in $p=1$, and a single root in $p=m/n=1/(2d+2)$. This implies that the local maximum of the function lies always between 0 and 1, and that this maximum moves to the left as $d$ increases, and will eventually converge to 0 as $d$ goes to infinity.

What is the length of the interval in which cells are likely to be alive? It is hard to derive an analytical solution of this length, so we give an approximation. The two points between 0 and 1 at which $f''(p)=0$ serve as a reasonable estimate for the start- and end-points of this interval, so to give an approximation of the interval's length we calculate the difference of their $x$-coordinates. It is easy to derive that

$$f''(p) = \binom{n}{m} p^{m-2}(1-p)^{n-m-2}\left((n^2-n)p^2 - 2m(n-1)p + m^2 - m\right)$$

so the roots of this function are, apart from the ones at $p=0$ and $p=1$:

$$p_{12} = \frac{m(n-1) \pm \sqrt{m(n-1)(n-m)}}{n(n-1)}$$

It is easily verified that these roots lie between 0 and 1. The difference of these two roots, expressed in terms of $d$, is:

$$p_2 - p_1 = \sqrt{\frac{2d+1}{(d+1)^2(4d^2+4d-1)}}$$

Obviously, this value decreases monotonously with increasing $d$, converging to 0 as $d$ goes to infinity. Though the mean-field approximation differs from the actual behavior of a CA, we can still draw meaningful conclusions from its asymptotic behavior. The convergence to 0 of the probability interval that sustains life, makes it extremely unlikely that the cellular space as a whole will be a fertile ground for patterns to emerge from initial random configurations. This is of course a probabilistic analysis. In reality, there can be patterns that survive, even in the models with higher values of $d$. Indeed, the previous section shows that gliders are among such patterns. These gliders grow in size, however, as a linear function of $d$, making their emergence from random configurations increasingly unlikely for high values of $d$.

## 6 Conclusions and Discussion

This paper presents an infinite class of inner-dependent outer-totalistic CA that all have gliders among their possible configurations. The discrete-valued parameter $d=4r$ for $r=2,3,4...$, which corresponds to the cell neighborhood's radius, serves as an index for the members KL($d$) of this class. Gliders in KL($d$) have a length of O($d$), which makes their occurrence extremely unlikely in randomly initialized configurations or configurations emerging from them. The proven presence of gliders in KL($d$), however, shows that this probability will never be 0.

In the light of the convergence of the probability of living cells in KL($d$) to 0 as $d$ approaches infinity, we conclude that interesting phenomena may occur in a cellular space that is at first sight—when doing trial-and-error computer simulations—virtually dead. A next logical step in this research would be to identify other living configurations in the infinite class of CA, and to find ways to make these configurations interact in possibly useful ways (such as to support computation).

Our results seem to confirm an observation on the web site [7] that the presence of gliders in CA is more common than expected, and that it appears to be hardly correlated with the classification of CA dynamics in the four classes proposed by Wolfram in [8]. In other words, Life may be more common than expected, yet it may be in unexpected places and spaces!

## References

1. Berlekamp, E., Conway, J., Guy, R.: 25: What is Life. In: Winning Ways for your Mathematical Plays, vol. 2. Academic Press, London (1982)
2. Gardner, M.: The fantastic combinations of john conway's new solitaire game "life". Scientific American 223(10), 120–123 (1970)
3. Griffeath, D.: Self-organization of random cellular automata: four snapshots. In: Probability and Phase Transition. NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci, vol. 420, pp. 49–67. Kluwer Academic, Dordrecht (1994)
4. Evans, K.: Larger than Life: threshold-range scaling of Life's coherent structures. Physica D 183(1-2), 45–67 (2003)
5. Adachi, S., Lee, J., Peper, F., Umeo, H.: Kaleidoscope of Life: a 24-neighborhood outer-totalistic cellular automaton. Physica D 237(6), 800–817 (2008)
6. Schulman, L., Seiden, P.: Statistical mechanics of a dynamical system based on conway's game of life. J. Stat Phys. 19(3), 293–314 (1978)
7. Eppstein, D.: Which "life"-like cellular automata have gliders (2005), http://www.ics.uci.edu/~eppstein/ca/
8. Wolfram, S.: Universality and complexity in cellular automata. Physica D 10(1-2), 1–35 (1984)

# Cellular Automata-Based Structures to Compute the Solutions of Linear Difference Equations*

A. Fúster-Sabater[1], P. Caballero-Gil[2], and O. Delgado[1]

[1] Institute of Applied Physics, C.S.I.C., Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es, oscar.delgado@iec.csic.es
[2] Faculty of Maths, D.E.I.O.C., University of La Laguna, 38271 Tenerife, Spain
pcaballe@ull.es

**Abstract.** A cellular automata-based linear model that computes all the solutions of linear binary difference equations has been developed. Such a model is based on successive concatenations of a basic linear automaton. Different sequential solutions are obtained from different automaton initial states. Many of these solutions are binary sequences of cryptographic utility. In this way, a linear structure based on cellular automata realizes not only difference equation solutions but also generates sequences currently used in cryptography. The model is simple, linear and can be applied in a range of practical cryptographic applications.

**Keywords:** cellular automata, natural computing, difference equation, cryptography.

## 1 Introduction

Cellular Automata (CA) are particular forms of finite state machines that can be investigated by usual analytic techniques ([3], [11], [14]). CA are defined as uniform arrays of identical cells in an $n$-dimensional space. They are characterized by different parameters [14] (cellular geometry, neighborhood specifications, number of states per cell, transition rules, etc). In this work, our interest is concentrated on one-dimensional binary CA with three site neighborhood and *linear* transition rules. In addition, CA here considered will be *hybrid* (different cells evolve under different transition rules) and *null* (cells with null content are adjacent to the automaton extreme cells).

On the other hand, stream ciphers are the fastest among the encryption procedures so they are implemented in many practical applications e.g. the algorithms A5 in GSM communications [9] or the encryption system E0 in Bluetooth specifications [2]. From a short secret key (known only by the two interested parties) and a public algorithm (the sequence generator), stream cipher procedure consists in generating a long sequence of seemingly random bits, that is a

* This work has been developed in the frame of the project HESPERIA (http://www.proyecto-hesperia.org) under programme CENIT and supported by Centro para el Desarrollo Tecnológico Industrial (CDTI) as well as by the companies: Soluziona, Unión Fenosa, Tecnobit, Visual-Tools, BrainStorm, SAC and TechnoSafe.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 42–49, 2008.

pseudo-random sequence. In cryptographic terms, such a sequence is called the keystream sequence.

In the literature, there are different families of pseudo-random sequences currently based on Linear Feedback Shift Registers (LFSRs) [8]. The output sequences of such linear registers are combined by means of nonlinear functions in order to produce keystream sequences of cryptographic application. They can be generated in different ways:

1. By a LFSR controlled by another LFSR (which may be the same one) e.g. multiplexed sequences [10], clock-controlled sequences [1], cascaded sequences [7], shrinking generator sequences [4] etc.
2. By one or more than one LFSR and a feed-forward nonlinear function e.g. Gold-sequence family, Kasami (small and large set) sequence families, GMW sequences, Klapper sequences, No sequences etc. See [8] and the references cited therein.

In the present work, it is showed that one-dimensional linear CA based on rules 90/150 generate all the solutions of linear difference equations with binary constant coefficients. Some of these solutions correspond to the sequences produced by the previous keystream generators. In this way, we have simple CA that not only generate all the solutions of a kind of difference equations but also they are linear models of nonlinear cryptographic sequence generators. Due to the linearity of the CA transition rules, modelling these CA-based structures is simple and efficient.

## 2 Fundamentals and Basic Notation

In this section, the two basic structures considered within this paper (linear difference equations and one-dimensional linear hybrid CA) are briefly introduced.

### 2.1 Linear Difference Equations with Binary Coefficients

Troughout this work, the following kind of linear difference equations with binary coefficients will be considered:

$$(E^r \oplus \sum_{j=1}^{r} c_j \, E^{r-j}) \, a_n = 0, \qquad n \geq 0 \tag{1}$$

where $a_n \in GF(2)$ is the *n-th* term of a binary sequence $\{a_n\}$ that satisfies the previous equation. $E$ is the shifting operator that operates on the terms $a_n$ of a solution sequence (i.e. $E^j a_n = a_{n+j}$). The coefficients $c_j$ are constant binary coefficients $c_j \in GF(2)$, $r$ is an integer and the symbol $\oplus$ represents the XOR logic operation. The $r$-degree characteristic polynomial of the equation (1) is:

$$P(x) = x^r + \sum_{j=1}^{r} c_j \, x^{r-j} \tag{2}$$

and specifies the linear recurrence relationship of the sequence $\{a_n\}$. This means that its *n-th* term, $a_n$, can be written as a linear combination of the previous terms:

$$a_n \oplus \sum_{j=1}^{r} c_j \; a_{n-j} = 0, \qquad n \geq r. \tag{3}$$

If $P(x)$ is a primitive polynomial [12] and $\alpha$ one of its roots, then

$$\alpha, \; \alpha^2, \; \alpha^{2^2}, \ldots, \; \alpha^{2^{(r-1)}} \in GF(2^r) \tag{4}$$

are the $r$ different roots of such a polynomial, see [13]. In this case, it can be proved [12] that the solutions of the equation (1) are sequences of the form:

$$a_n = \sum_{j=0}^{r-1} A^{2^j} \; \alpha^{2^j n}, \qquad n \geq 0 \tag{5}$$

where $A$ is an arbitrary element in $GF(2^r)$. That is to say, $\{a_n\}$ is a Pseudo-Noise sequence (*PN*-sequence) [8] of characteristic polynomial $P(x)$ and period $2^r - 1$ whose starting point is determined by the value of $A$. If $A = 0$, then the solution of the equation (1) is the identically null sequence.

### 2.2 One-Dimensional Linear Hybrid CA

Now, our attention is focussed on one-dimensional binary linear hybrid CA with three site neighborhood. In fact, there are eight of such transition rules among which only two (rule 90 and rule 150) lead to non trivial structures. Both rules are defined as follows [11]:

| Rule 90 | Rule 150 |
|---|---|
| $b_{n+1}^k = b_n^{k-1} \oplus b_n^{k+1}$ | $b_{n+1}^k = b_n^{k-1} \oplus b_n^k \oplus b_n^{k+1}$ |

Indeed, at time $n+1$ the content of the *k-th* cell, $b_{n+1}^k \in GF(2)$, depends on the content at time $n$ of either two different cells (rule 90) or three different cells (rule 150), with $k = 1, ..., L$, where $L$ is the length of the automaton. Moreover, the state of the automaton at time $n$ is the binary content of the $L$ cells at such an instant. A natural form of representation for this kind of automata is given by an $L$-tuple $\Delta_L = (d_1, d_2, ..., d_L)$ where $d_k = 0$ if the *k-th* cell verifies rule 90 while $d_k = 1$ if the *k-th* cell verifies rule 150. In addition, $\Delta_k = (d_1, d_2, ..., d_k)$ for $k = 1, ..., L$ denotes the corresponding sub-automaton of length $k$.

Given a primitive polynomial $Q(x)$, the Cattell and Muzio synthesis algorithm [3] provides us with a pair of reversal linear 90/150 CA whose characteristic polynomial is $Q(x)$. Therefore, a one-dimensional binary linear 90/150 cellular automaton of primitive characteristic polynomial $P(x)$ given by (2) will generate the *PN*-sequence defined by equation (5), see [6]. As an example, Table 1 depicts in bold the *PN*-sequence obtained either:

1. As a solution of the equation (1) with characteristic polynomial $P(x) = x^3 + x^2 + 1$, $r = 3$ and $A = 1$ ($A \in GF(2^3)$)

$$a_n = 1 \; \alpha^n \oplus 1 \; \alpha^{2n} \oplus 1 \; \alpha^{4n}, \qquad n \geq 0 \tag{6}$$

**Table 1.** The same *PN*-sequence obtained either as a difference equation solution or as a sequence generated by two reversal linear CA

| Differen. eq. sol. | CA: 150 90 90 | | | 90 90 150 | | |
|---|---|---|---|---|---|---|
| **1** | **1** | 0 | 1 | **1** | 1 | 0 |
| **1** | **1** | 0 | 0 | **1** | 1 | 1 |
| **1** | **1** | 1 | 0 | **1** | 0 | 0 |
| **0** | **0** | 1 | 1 | **0** | 1 | 0 |
| **1** | **1** | 1 | 1 | **1** | 0 | 1 |
| **0** | **0** | 0 | 1 | **0** | 0 | 1 |
| **0** | **0** | 1 | 0 | **0** | 1 | 1 |

2. As the sequence generated by the pair of reverse CA, e.g. (150 90 90) and (90 90 150), starting at the initial states (1,0,1) and (1,1,0), respectively. At the remaining CA cells, shifted versions of the same *PN*-sequence are generated.

## 3 Generalization to More Complex Difference Equations

Let us generalize the difference equations of the subsection 2.1 to a more complex kind of linear difference equations whose roots have a multiplicity greater than 1. In fact, we are going to consider equations of the form:

$$(E^r + \sum_{j=1}^{r} c_j \; E^{r-j})^p \; a_n = 0. \qquad n \geq 0 \tag{7}$$

$p$ being an integer $> 1$. The characteristic polynomial $P_M(x)$ of this kind of equations is:

$$P_M(x) = P(x)^p = (x^r + \sum_{j=1}^{r} c_j \; x^{r-j})^p \tag{8}$$

In this case, the roots of $P_M(x)$ are the same as those of the polynomial $P(x)$, that is $(\alpha, \alpha^2, \alpha^{2^2}, \ldots, \; \alpha^{2^{(r-1)}})$, but with multiplicity $p$. Now the solutions of (7) are:

$$a_n = \sum_{i=0}^{p-1} (\binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j} \; \alpha^{2^j n}) \tag{9}$$

where $A_i \in GF(2^r)$ and the $\binom{n}{i}$ are binomial coefficients modulo 2. According to (5), the term $\sum_{j=0}^{r-1} A_i^{2^j} \; \alpha^{2^j n}$ represents the *n-th* element of the same *PN*-sequence as before whose starting point is determined by $A_i$. Thus, $\{a_n\}$ is just the bitwise sum of $p$ times the same *PN*-sequence starting at different points and weighted by different binomial coefficients. In fact, each binomial coefficient defines a succession of binary values with a constant period $p_i$. Table 2 shows the sequences and values of $p_i$ for the first coefficients $\binom{n}{i}$.

**Table 2.** Binomial coefficients, binary values and periods $p_i$

| Binomial coeff. | Binary values | $p_i$ |
|---|---|---|
| $\binom{n}{0}$ | 1,1,1,1,1,1,1,1,1,1, ... | $p_0 = 1$ |
| $\binom{n}{1}$ | 0,1,0,1,0,1,0,1,0,1, ... | $p_1 = 2$ |
| $\binom{n}{2}$ | 0,0,1,1,0,0,1,1,0,0, ... | $p_2 = 4$ |
| $\binom{n}{3}$ | 0,0,0,1,0,0,0,1,0,0, ... | $p_3 = 4$ |
| $\binom{n}{4}$ | 0,0,0,0,1,1,1,1,0,0, ... | $p_4 = 8$ |
| $\binom{n}{5}$ | 0,0,0,0,0,1,0,1,0,0, ... | $p_5 = 8$ |
| $\binom{n}{6}$ | 0,0,0,0,0,0,1,1,0,0, ... | $p_6 = 8$ |
| $\binom{n}{7}$ | 0,0,0,0,0,0,0,1,0,0, ... | $p_7 = 8$ |

Remark that the choice of $A_i$ determines the characteristics of the sequences $\{a_n\}$ that are solutions of the equation (7). Indeed, the period $T$ of $\{a_n\}$ depends on the periods $T_i = p_i \cdot (2^r - 1)$ of the $p$ sequences that are summed in (9). The linear complexity $LC$ of $\{a_n\}$ is related with the number of roots with their corresponding multiplicities weighted by $A_i$ that appear in (9). The number $N$ of different sequences $\{a_n\}$ is related with the number of different $p$-tuples of values of $A_i$.

This kind of difference equations given by (7) is crucial because many binary sequences of cryptographic application (those ones referred in the introduction) have characteristic polynomials given by the equation (8). Consequently, many cryptographic sequences are solutions of linear difference equations. In this way, it would be very convenient to have a simple CA-based linear model able to compute all the solutions of these difference equations, among them we could find a great variety of cryptographic sequences. Next section tackles this problem.

## 4 Realization of the Linear Difference Equation Solutions by Means of CA

Since the characteristic polynomial of the considered equations is $P_M(x) = P(x)^p$, it seems quite natural to construct the solutions of such equations by concatenating $p$ times the basic automaton of characteristic polynomial $P(x)$. The following result is a concrete formalization of this idea.

**Theorem 1.** *Let $C$ be a linear 90/150 cellular automaton of length $L$, binary codification $(d_1, d_2, \ldots, d_{L-1}, d_L)$ and characteristic polynomial $P(x)$. Let $\tilde{C}$ be the reversal version of $C$, with binary codification $(d_L, d_{L-1}, \ldots, d_2, d_1)$, and the same length and polynomial as those of $C$. Then, the 2L-tuple defined by $(d_1, d_2, \ldots, \overline{d_L}, \overline{d_L}, \ldots, d_2, d_1)$ represents the linear 90/150 cellular automaton of length $2L$ and characteristic polynomial $P(x)^2$.*

The proof of this theorem is based on the recurrence relationship for the characteristic polynomials of the successive sub-automata of a given automaton [3].

The result can be iterated a number of times for successive polynomials and rule vectors:

$$\begin{array}{lcl} P(x) & \longleftrightarrow & \Delta_L = (d_1, d_2, ..., d_L) \\ P(x)^2 & \longleftrightarrow & \Delta_{2L} = (d_1, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, d_1) \\ P(x)^{2^2} & \longleftrightarrow & \Delta_{2^2 L} = (d_1, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, \overline{d_1}, \overline{d_1}, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, d_1) \\ \vdots & \longleftrightarrow & \vdots \qquad\qquad\qquad\qquad \vdots \end{array}$$

Notice that the basic automaton is concatenated with its reversal version after the complementation of the last rule. Successive applications of this result provide us with CA whose characteristic polynomials are: $P(x)^2$, $P(x)^{2^2}$, $P(x)^{2^3}$, $\ldots, P(x)^{2^q}$ of lengths $2L$, $2^2L$, $2^3L$, $\ldots, 2^qL$, respectively. Remark that for every $P(x)$ there are two reverse basic automata that may be used in the concatenation procedure. Consequently, for $2^{q-1} < p \leq 2^q$ the two automata built as in Theorem 1 will produce for different CA initial states all the sequences $\{a_n\}$ with characteristic polynomial $P(x)^p$ that satisfy the difference equation.

### 4.1 Illustrative Example

Let us see a simple example illustrating the previous sections. Consider a pair of reverse CA $\Delta_5 = (1, 0, 0, 0, 0)$ and $\Delta_5^* = (0, 0, 0, 0, 1)$ of length $L = 5$ associated to the characteristic polynomial $P(x) = x^5 + x^4 + x^2 + x + 1$. If $P_M(x) = P(x)^p$ with $p = 4$, then one of the CA obtained by concatenation will be: $\Delta_{20} = (1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1)$. Now different choices of $A_i$ (not all null) will allow us to generate all the solutions of the difference equation (7) with the previous parameters.

1. If $A_0 \neq 0$ and $A_i = 0 \ \ \forall\ i > 0$, then the cellular automaton will produce $N_0 = 1$ sequences, that is a unique $PN$-sequence of period $T_0 = 31$, linear complexity $LC_0 = 5$ and characteristic polynomial $P(x)$. In addition, the automaton cycles through doubly symmetric states of the form:

$$(a_0, a_1, a_2, a_3, a_4, a_4, a_3, a_2, a_1, a_0, a_0, a_1, a_2, a_3, a_4, a_4, a_3, a_2, a_1, a_0)$$

   with $a_i \in GF(2)$. The 31 doubly symmetric states are concentrated into the same cycle.
2. If $A_0 \in GF(2^5), A_1 \neq 0$ and $A_i = 0 \ \ \forall\ i > 1$, then the cellular automaton will produce $N_1 = 16$ different sequences of period $T_1 = 62$, linear complexity $LC_1 = 10$ and characteristic polynomial $P(x)^2$. Moreover, the automaton cycles through symmetric states of the form:

$$(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_9, a_8, a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$$

   with $a_i \in GF(2)$. In fact, there are $2^{10} - 32 = 992$ symmetric states distributed in 16 cycles of 62 states each of them.

3. If $A_0, A_1 \in GF(2^5), A_2 \neq 0$ and $A_i = 0 \ \ \forall\, i > 2$, then the cellular automaton will produce $N_2 = 256$ different sequences of period $T_2 = 124$, linear complexity $LC_2 = 15$ and characteristic polynomial $P(x)^3$. Moreover, the automaton cycles through several repetitive states of the form:

$$(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9)$$

with $a_i \in GF(2)$.
4. If $A_0, A_1, A_2 \in GF(2^5), A_3 \neq 0$, then the cellular automaton will produce $N_3 = 8192$ different sequences of period $T_3 = 124$, linear complexity $LC_3 = 20$ and characteristic polynomial $P(x)^4$. In addition, the automaton cycles through the states not included in the previous cycles.

In brief, a simple linear structure based on CA allows us by successive concatenations to compute in a natural way all the sequences that are the solutions of linear binary difference equations.

## 5 Conclusions

All the sequential solutions of linear binary coefficient difference equations can be realized by means of linear models based on 90/150 cellular automata. Different sequential solutions come from different automaton initial states. Some of these solutions have a straight cryptographic application in stream ciphers. In this way, very popular cryptographic sequence generators conceived and designed as nonlinear generators can be linearized in terms of cellular automata. The linearization procedure is simple and can be applied to cryptographic examples in a range of practical application. The implementation of these 90/150 linear models is easy and very adequate for FPGA logic. This characteristic makes it suitable for developments where time execution is relevant as in stream ciphers and in communication systems with high transmission rates.

## References

1. Beth, T., Piper, F.: The Stop-and-Go Generator. In: EUROCRYPT 1984. LNCS, vol. 228. Springer, Heidelberg (1985)
2. Bluetooth, Specifications of the Bluetooth system, http://www.bluetooth.com/
3. Cattell., K., et al.: 2-by-n Hybrid Cellular Automata with Regular Configuration: Theory and Application. IEEE Trans. Computers 48(3), 285–295 (1999)
4. Coppersmith, D., Krawczyk, H., Mansour, Y.: The Shrinking Generator. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
5. Fúster-Sabater, A., de la Guía-Martínez, D.: Linealization of Stream Ciphers. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 612–622. Springer, Heidelberg (2004)
6. Fúster-Sabater, A., Caballero-Gil, P.: Cellular Automata in Cryptanalysis of Stream Ciphers. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 611–616. Springer, Heidelberg (2006)

7. Gollmann, D., Chambers, W.G.: Clock-Controlled Shift Register. IEEE J. Selected Areas Commun. 7(4), 525–533 (1987)
8. Gong, G.: Theory and Applications of q-ary Interleaved Sequences. IEEE Trans. Information Theory 41(2), 400–411 (1995)
9. GSM, Global Systems for Mobile Communications, http://cryptome.org/gsm-a512.htm
10. Jennings, S.M.: Multiplexed Sequences: Some Properties. In: EUROCRYPT 1982. LNCS, vol. 149. Springer, Heidelberg (1983)
11. Kari, J.: Theory of cellular automata: A survey. Theoretical Computer Science 334, 3–33 (2005)
12. Key, E.L.: An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators. IEEE Trans. Informat. Theory 22(6), 732–736 (1976)
13. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications. Cambridge University Press, Cambridge (1986)
14. Wolfram, S.: Random Sequence Generation by Cellular Automata. Adv. in Appl. Math. 7, 127–169 (1986)

# Computing by Swarm Networks

Teijiro Isokawa[1], Ferdinand Peper[1,2], Masahiko Mitsui[1],
Jian-Qin Liu[3], Kenichi Morita[4], Hiroshi Umeo[5],
Naotake Kamiura[1], and Nobuyuki Matsui[1]

[1] Division of Computer Engineering, University of Hyogo, Japan
isokawa@eng.u-hyogo.ac.jp
[2] Nano ICT Group,
National Institute of Information and Communications Technology, Japan
[3] Biological ICT Group,
National Institute of Information and Communications Technology, Japan
[4] Dept. of Information Engineering, Hiroshima University, Japan
[5] Dept. of Computer Science, Osaka Electro-Communication University, Japan

**Abstract.** Though the regular and fixed structure of cellular automata greatly contributes to their simplicity, it imposes a strict limitation on the applications that can be modeled by them. This paper proposes *swarm networks*, a model in which cells, unlike in cellular automata, have irregular neighborhoods. Timed asynchronously, a cell in this model acts like an *agent* that can dynamically interact with a varying set of other cells under the control of transition rules. The configurations in which cells are organized according to their neighborhoods can move around in space, following simple mechanical laws. We prove computational universality of this model by simulating a circuit consisting of asynchronously timed circuit modules. The proposed model may find applications in nanorobotic systems and artifical biological systems.

## 1 Introduction

Nanorobots, Artificial Cells, Smart Dust—all these models have in common a large number of distributed units that interact with each other to conduct certain tasks. Like in Cellular Automata (CA), the underlying units are relatively simple—usually being nothing more than Finite State Machines—but unlike in CA the units are more dynamic in the way they interact. They form swarms of agents that communicate with each other through dynamic networks of interconnections. How can we characterize the functionality of such swarms? Will their less-rigid communication structures cause a loss of functionality relative to CA models with comparable complexity? Or will swarm networks be more powerful through their more flexible way of interaction?

A useful measure of power—useful at least in the world of computer scientists–is whether a model is computationally universal. This measure basically separates "interesting" models from the "uninteresting" ones, forming the major motivation in the last century to characterize models in terms of computability. Universal Turing Machines [1] are well-known in this context, but other models

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 50–59, 2008.

have been proposed too [2]. In the context of CA, computational universality is often proved by embedding logic circuits on cellular space. This requires a CA to simulate a universal set of primitives, like the AND-gate and the NOT-gate, as well as to simulate signals being propagated between these primitives. Some well-known computationally universal CA are found in [3,4,5,6] when the timing model is synchronous—implying the simultaneous update of all cells at each step. In an asynchronous model, on the other hand, update of cells is randomly timed. Universality of asynchronous CA is proven in a similar way as with synchronous CA, i.e., by laying out circuits on cellular space, be it that different sets of primitives are used to compensate for the lack of clock signals [7,8,9,10,11].

Swarms have been researched in contexts varying from insects [12] to swarm robotics [13], and from distributed sensor networks [14] to groupware systems [15]. Stevens [16] has proposed a swarm-based system that is able to replicate itself. Simulated on a computer, the agents in this model are divided in different types, each with a different functionality. There are agents that conduct boolean operations, but also agents that exert forces in certain directions, and so on, and the agents move in 2-dimensional continuous space according to Newton-like laws. Each agent has four terminals through which it can be connected to and exchange integer values with other agents. Through these connections, agents can be organized in certain configurations, that act like a kind of "organisms", which have more complicated functionalities than individual agents.

In this paper we present a swarm network model in which all agents are identical, like the cells in CA but unlike in Stevens' model. The functionalities of the agents are determined by their states as well as by the patterns by which they are mutually interconnected. Based on these agents, we construct two circuit elements that form a universal set of primitives in the class of delay-insensitive circuits [17], i.e. circuits robust to delays in their wires and primitives. This result implies that any arbitrary delay-insensitive circuit can be constructed from the agents, proving the computational universality of the model.

This research promises applications in which simplicity of agents is important, while the cooperative actions of the agents are sufficiently powerful to result in interesting behavior. Nanorobotics is one particular application that comes to mind: the tiny robots in such an application face severe restrictions in their complexity; yet, combined in swarms of nanorobots, they should have a certain minimal functionality to be of use. Sensor networks may be another application, in which sensor agents derive added functionality from the mutual cooperation in their sensing behavior, for example to measure gradients in certain physical observables.

## 2 Computational Elements

A few decades ago Priese [17] proposed circuit elements from which arbitrary delay-insensitive circuits can be constructed. Called the E-element and K-element [17], these elements—schematically shown in Figs. 1 and 2—are universal, forming a base for the construction of a sequential automaton. The circuits constructed from

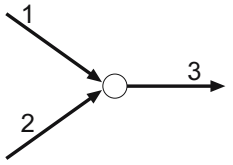



**Fig. 1.** K-element

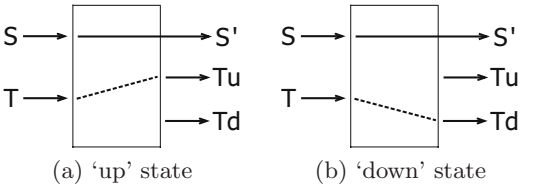



**Fig. 2.** E-element in (a) 'up' state and (b) 'down' state

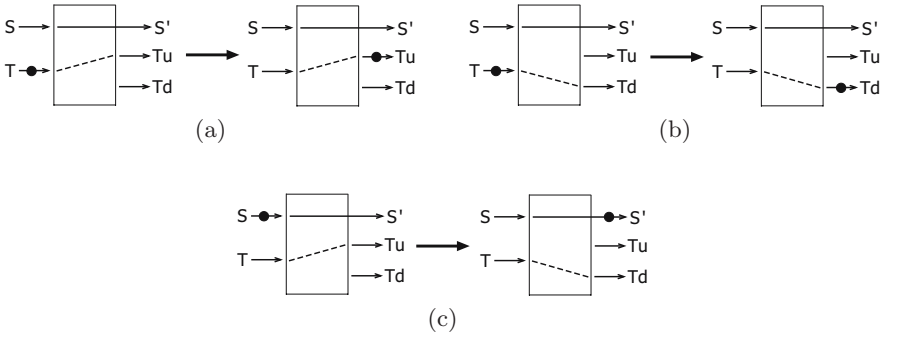



**Fig. 3.** Operations of an E-element: (a) when in the 'up' state, (b) when in the 'down' state, and (c) changing state upon receiving an input signal on wire S. A token (blob) on a wire denotes a signal.

E-elements and K-elements have in common that they employ only one signal at a time. Though inefficient, this is sufficient to guarantee universality.

The K-element has two input wires and one output wire and it accepts a signal coming from either input wire and outputs it to the output wire.

The E-element is an element with two input wires (S and T) and three output wires (S′, $T_u$, and $T_d$), as well as two internal states ('up' or 'down'). Input from wire T will be redirected to either of the output wires $T_u$ or $T_d$, depending on the internal state of the element: when this state is 'up' (resp. 'down'), a signal on the input wire T flows to the output wire $T_u$ (resp. $T_d$) as in Fig. 3(a) (resp. Fig. 3(b)). By accepting a signal from input wire S, an E-element changes its internal state from 'up' to 'down' or from 'down' to 'up', after which it outputs a signal to output wire S′, as shown in Fig. 3(c).

## 3 Model of the Agents

Contained in two-dimensional space, agents in the proposed swarm networks model have a circular shape, the outside of which has six terminals attached at identical distances from each other (Fig. 4). Agents are connected to each other via these terminals, which are used to exchange input and output between agents. The terminal colored black in Fig. 4 indicates that it forms a connection

with another agent. Each agent is assumed to be a Mealy-type finite automaton, with an internal state denoted by a symbol in it (Fig. 4). The functionality of an agent is determined—apart from the agent's state—by the connection pattern of the agent to other agents. So, an agent being connected to two agents at opposite terminals, for example, has a different functionality than an agent connected to three agents via adjacent terminals.

An agent's functionality is mostly expressed in terms of logical transitions, but it may also contain a physical component. For certain patterns at which an agent is connected to other agents, the agent may experience a force exerted from a certain terminal, pulling it in a certain direction. The space containing the agents satisfies simple mechanical laws. Apart from the abovementioned forces, there are forces between terminals interconnected to each other. Modeled as springs, the connections between terminals exert a repulsive force between terminals very near to each other, and an attractive force between terminals more remote from each other. So, connections are elastic. Communication is not only limited to terminals connected to each other, but may also take place between two terminals that are unconnected but very near to each other, to the extent that the two terminals are at a distance that is less than the distance between two adjacent terminals within an agent. The states and output of the agent are determined by a transition function. This function has as its domain the agent's internal state, the input values from the I/O terminals, and the connection pattern of the I/O terminals. The output domain of the transition function covers the agent's internal state, the output values to the I/O terminals, and whether a force is exerted to the agent. Formally, the transition function $f$ is defined as:

$$f(q, \boldsymbol{i}, \boldsymbol{c}) \rightarrow (q', \boldsymbol{o}, m) \tag{1}$$

where $q$ and $q'$ are the internal states before and after the transition, respectively, $\boldsymbol{i} = \{i_0, \cdots, i_5\}$ is the set of the input values on the I/O terminals, $\boldsymbol{o} = \{o_0, \cdots, o_5\}$ is the set of values output to the I/O terminals. The connection pattern is denoted by $\boldsymbol{c} = \{c_0, \cdots, c_5\}$. The value of $m$ in the output of the transition function determines whether a force is exerted upon the agent. We assume that the transition rules are rotation symmetric, i.e., one transition rule exists in six varieties, which are rotated analogues of each other. An illustrative transition rule is depicted in Fig. 5, where the direction of the exerted force is indicated by the dotted arrow.

## 4 Building Circuits by Swarm Networks

To establish the computational universality of the model, we show how the K- and E-elements can be constructed by groups of agents. An agent in the model has one of two states ($q_1$ or $q_2$) and each of its I/O terminals inputs and outputs a number from the set $\{1, 2\}$. There is also another type of agent, called *wall agent*, that is passive. Wall agents are lined up into structures that form boundaries between which configurations of the normal agents may move around. Represented as black circles in the figures, wall agents constitute the isolating walls

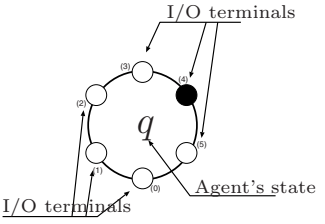



**Fig. 4.** Individual Agent. The terminals are labeled by the numbers between brackets.

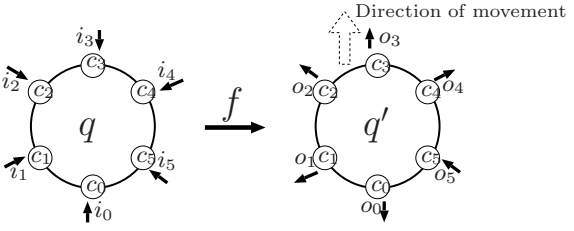



**Fig. 5.** Transition rule for an agent

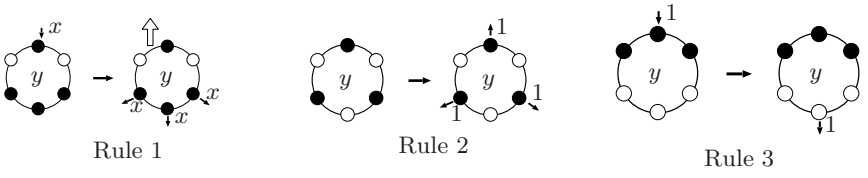



**Fig. 6.** Transition rules for a signal: rule 1 facilitates the movement of a signal, rule 2 provides a constant source of 1-values driving the processes in the signal, and rule 3 provides 1-outputs at the tail of the signal to be sensed by the switching bar in the E-element.

of the wires in the circuits to be constructed. Signals travel in the circuit along these wires.

The behavior of agents is governed by transition rules, which are applied to the agents according to an asynchronous updating mode (random timing). Fig. 6 shows three transition rules that are used in the operation of a signal, the configuration of which is shown in Fig. 7. The symbol $x$ in rule (1) denotes an input and output value being either 1 or 2. When, for example, the input value at terminal 3 of an agent is 1, the same value 1 is output at terminals 0, 1, and 5. This rule also facilitates the exertion of a force such that the agent moves to the north. There is one agent in a signal that is governed by rule (1): it is at the inner part of the signal, and denoted by the symbol (b) in the configuration constituting the signal (Fig. 7). Transition rule (2) applies to another agent in the inner part of the signal, which is labeled by the symbol (a) in Fig. 7. This agent provides a constant stream of 1-values output to other agents in the signal, such as the above agent labeled by (b). These 1-values being output are also received by agents that behave according to rule (3), and these agents respond with a 1-value output to their opposite terminal. This 1-value being output will in its turn be transmitted to the E-element when the signal passes through it, as a result of which the E-element's state will be flipped, as we will see later.

The turn of a wire resp. crossing of two wires can be implemented in a straightforward way, i.e., by appropriate configurations of wall agents, which guide the

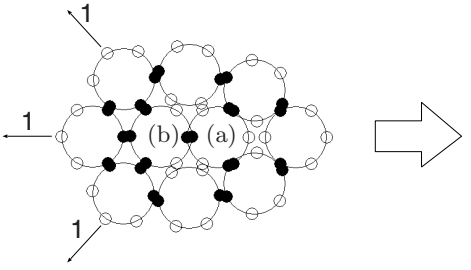
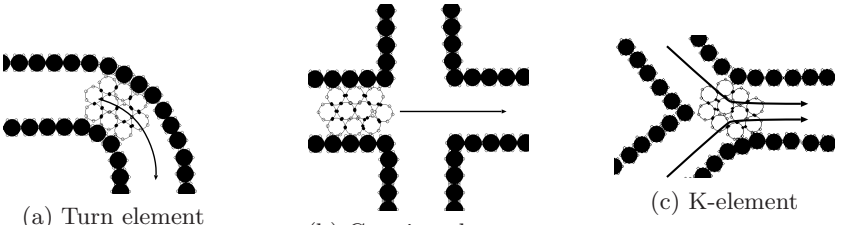



**Fig. 7.** A configuration of a signal. The big arrow right of the configuration denotes the direction of the signal.



(a) Turn element (b) Crossing element (c) K-element

**Fig. 8.** Configurations of circuit element by wall agents

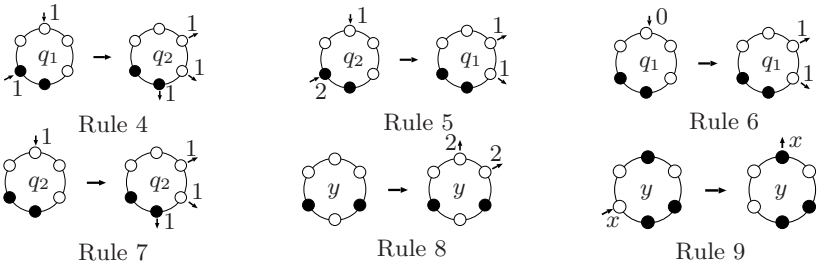



**Fig. 9.** Transition rules for operating an E-element

signals in accordance with the wall's shape (Figs. 8(a) resp. 8(b)). The temporary absence of wall structures at the time of a signal's crossing does not affect the proper passing of the signal through the crossing, since the force exerted on a signal pulls it across this momentary lapse of the walls. The K-element can be constructed by wall agents in a similar way as the turn and crossing elements (Fig. 8(c)): a signal from either input wire will be guided to the output wire by the wall agents.

For the construction of an E-element six more transition rules are required, which are shown in Fig. 9. The agents to which rules (4), (5), (6), and (7) apply all have the same connection patterns, but each of these rules applies to different patterns of input and output values from other agents. The symbol 0 at terminal

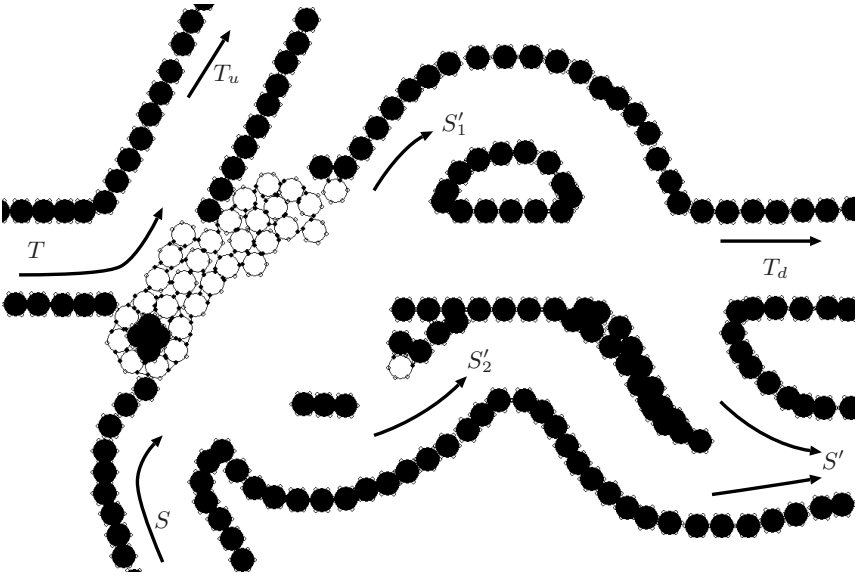



**Fig. 10.** A configuration of agents representing an E-element

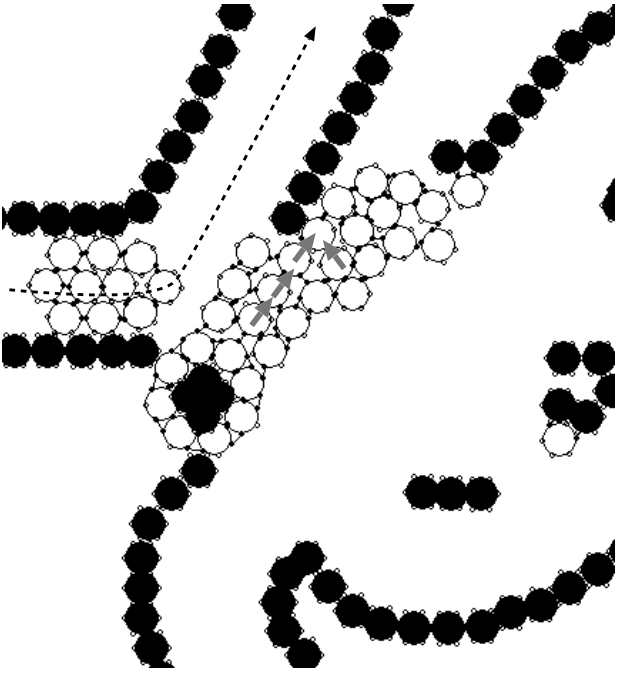


**Fig. 11.** Flow of a signal in an E-element in the state 'up'

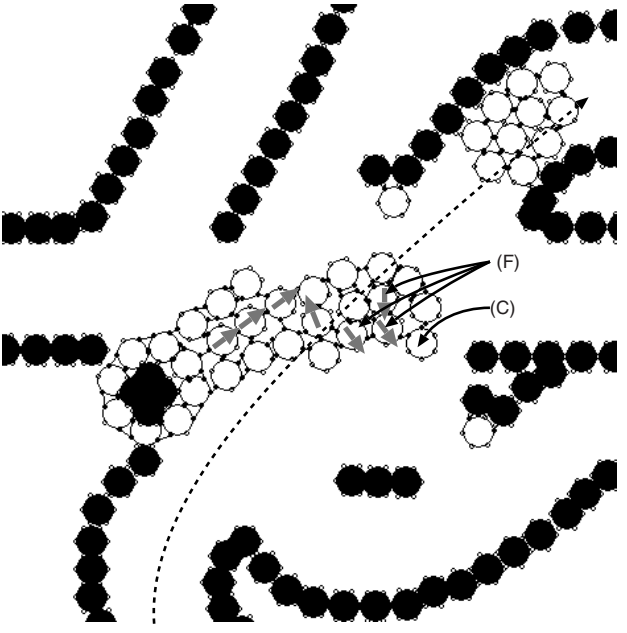



**Fig. 12.** Switching of an E-element from the state 'up' to the state 'down' as the result of a signal being input from terminal $S$

3 in rule (6) denotes the condition that no values from another agent are input to this terminal.

Fig. 10 shows a realization of an E-element by agents. Signals in this element flow according to the direction of the arrows. The rotation bar in this configuration, which hinges around a fixed post of wall agents, indicates the state of the E-element by its position. In Fig. 10 the state is 'up'. A signal from input wire $T$ exits from either output wire $T_u$ or $T_d$, depending on the direction of the rotation bar, without changing the state of the E-element. A signal input to wire $S$ also passes through the E-element and leaves from either output wire $S'_1$ or $S'_2$, before eventually being merged into one output wire $S'$. In this case, the state of the E-element changes as the 1-outputs emanating from a signal's tail result in the rotation bar being flipped.

Fig. 11 shows a signal passing through an E-element in state 'up' in more detail, with the directions of the forces exerted on the agents inside of the rotation bar indicated by gray arrows. Though the signal passing through the E-element touches the rotation bar, and thus exerts yet another force on it, the integrity of the rotation bar (and thus its shape) is maintained as a result of the pulling forces inside the bar. Switching of the E-element takes only place if the signal's tail touches the tip of the rotation bar, but in this case this will not happen, since there is a wall in between to prevent this.

When a signal enters the E-element from input terminal $S$, like in Fig. 12, its tail will touch the tip of the rotation bar on the signal's exit. This results

in the transmission of a 1-value from the signal to agent (C) in the bar's tip in Fig. 12. The chain of reactions caused by this between agents in the rotation bar will then effectuate state changes in some of these agents. Accompanying the state changes is an increase in the number of agents on which downward forces are exerted. To be exact, three agents will experience a downward force, and these agents are indicated by the symbol (F) in Fig. 12. The rotation bar will then rotate downward as a result of these forces. The opposite of this process—switching the E-element's state back to state 'up'—is accomplished in a similar way, by inputing a signal to $S$, which will then pass $S_2'$ in Fig. 10, in the process touching the rotation bar's tip. The opposite will then happen: the downward forces will be switched off and the bar will move upwards again.

## 5 Conclusions and Discussion

This paper presents a model of swarm networks and shows how to conduct universal computation by groups of agents in these networks. An agent is a two-state Mealy-type finite automaton with six input/output terminals, some of which are connected to other agents' terminals. The state of an agent, the output values of its terminals, and the connectivity of each agent determine the functionality of the agent; these variables are thus directly reflected by the transition rules in the model. Agents are similar to cells in cellular automaton models, except that the interconnection structure between agents is irregular. Universal computation is achieved in this model by nine transition rules, through the simulation of the asynchronously timed K and E circuit elements, as well as through the simulation of signal propagation between these elements. Simulations on a computer reveal that the proposed model behaves in a way that somehow resembles biological phenomena. The elastic nature of the connections between the agents appears an important ingredient in this context, as it results in an efficient distribution of the pulling forces among agents connected to each other.

The proposed model may be useful for the realization of computational devices based on biological mechanisms or other physical nanometer-scale interactions. The agents could for example be implemented in terms of proteins. This includes motor proteins, i.e., proteins that facilitate the transport of certain chemical substances inside organisms. It is well-known that proteins can be bound to other proteins, like with our agents, and that such bindings result in new properties and behavior of the formed components [18,19]. A protein can be thought of as being in a certain state through the addition of a phosphorus molecule: when such a molecule is present, we speak of a *phosphorylated* protein, otherwise of a *dephosphorylated* protein. The state of a protein can be influenced by the state of other proteins in its vicinity, according to so-called *domain-specific reactions*, a domain in a protein corresponding to a terminal in an agent. These reactions tend to be strongly dependent on the bindings of the protein to other proteins, in a similar way as the interconnection pattern of an agent with other agents influences the agent's functionality. Though space does not allow us to give specific biological implementations of the agents, the richness of interactions between proteins provide ample inspiration toward the realization of this.

## References


1. Turing, A.: On computable numbers, with an application to the entscheidungsproblem. Proc. London Math. Soc. 2(42), 230–265 (1936)
2. Church, A.: An unsolvable problem of elementary number theory. American Journal of Mathematics 58(2), 345–363 (1936)
3. Neumann, J.V.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966)
4. Banks, E.: Universality in cellular automata. In: IEEE 11th Ann. Symp. on Switching and Automata Theory, pp. 194–215 (1970)
5. Codd, E.F.: Cellular Automata. Academic Press, Inc., Orlando (1968)
6. Serizawa, T.: Three-state Neumann neighbor cellular automata capable of constructing self-reproducing machines. Syst. and Comput. in Japan 18(4), 33–40 (1987)
7. Adachi, S., Peper, F., Lee, J.: Computation by asynchronously updating cellular automata. Journal of Statistical Physics 114(1/2), 261–289 (2004)
8. Lee, J., Adachi, S., Peper, F., Morita, K.: Embedding universal delay-insensitive circuits in asynchronous cellular spaces. Fundamenta Informaticae 58(3/4), 295–320 (2003)
9. Lee, J., Adachi, S., Peper, F., Mashiko, S.: Delay-insensitive computation in asynchronous cellular automata. Journal of Computer and System Sciences 70(2), 201–220 (2005)
10. Peper, F., Lee, J., Adachi, S., Mashiko, S.: Laying out circuits on asynchronous cellular arrays: a step towards feasible nanocomputers? Nanotechnology 14(4), 469–485 (2003)
11. Peper, F., Lee, J., Abo, F., Isokawa, T., Adachi, S., Matsui, N., Mashiko, S.: Fault-tolerance in nanocomputers: A cellular array approach. IEEE Trans. Nanotechnology 3(1), 187–201 (2004)
12. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press, USA (1999)
13. Bayindir, L., Sahin, E.: A review of studies in swarm robotics. Turk J. Elec. Engin. 15(2), 115–147 (2007)
14. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
15. ter Beek, M., Ellis, C., Kleijn, J., Rozenberg, G.: Synchronizations in team automata for groupware systems. Comput. Supported Coop. Work 12(1), 21–69 (2003)
16. Stevens, W.: Simulating self-replicating machines. Journal of Intelligent and Robotic Systems 49(2), 135–150 (2007)
17. Priese, L.: Automata and Concurrency. Theoretical Computer Science 25(3), 221–265 (1983)
18. Krauss, G.: Biochemistry of signal transduction and regulation. Wiley-VCH, Weinheim (2008)
19. Liu, J.Q., Shimohara, K.: Biomolecular computation for bionanotechnology. Artech House, Boston (2007)

# On a Membrane Formation in a Spatio-temporally Generalized Prisoner's Dilemma

Yuji Katsumata and Yoshiteru Ishida

Department of Knowledge-Based Information Engineering,
Toyohashi University of Technology
Tempaku, Toyohashi 441-8580, Japan
http://www.sys.tutkie.tut.ac.jp

**Abstract.** A spatio-temporal generalization involves not only conventional temporal strategies that determine the player's action based on the opponent past actions but also spatial strategies based on the neighbor players' current actions and configurations. This framework allows the model to be dealt as a second order cellular automaton. With this involvement of the spatial strategies, we have observed a membrane formation which protects the cooperating clusters from being corroded by defecting intruders.

**Keywords:** spatial prisoner's dilemma, generalized prisoner's dilemma, second order cellular automata, membrane formation, maintenance of cooperating clusters.

## 1 Introduction

It has been long debated and discussed that the core mechanism that allows cooperation to evolve in social, biological, or ecological systems in spite of seemingly more advantageous strategy of the defection [1,2,3,4]. After a genius work of spatial framework by Nowak and May [5,6], the maintenance and protection of cooperators' cluster can be regarded as a problem of cellular automaton. Many possible mechanisms for the maintenance and protection of cooperators' cluster have been proposed [7,8,9]. In a spatio-temporal generalization of Prisoner's Dilemma (PD) [10], we observed a membranous phenomenon where a membrane in a perimeter of cooperators' cluster and protect the cluster from the invasion by defectors where the cooperators' cluster would be invaded otherwise.

Prisoner's Dilemma has been providing motivations in many fields not only international politics but evolutionary biology since a seminal work by Axelrod. Spatial prisoner's dilemma invented by Nowak and May also provides another dimension that these originally game theoretic studies can be related to the field of cellular automata. In the spatio-temporal generalization of dilemmaDcomes even more obvious that the generalized model is a sort of CA (cellular automata): a second order CA where the strategy first determines the rule based on the neighbors' configuration and the then rule in turn determines the next action.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 60–66, 2008.

One could point out that the involvement of player's benefit (expressed by a payoff matrix as in Table 1) is crucial but it is still mathematically (with a language of Mappings) regarded as a second order CA.

Section 2 states definitions and notations used in this note. Section 3 presents the main results of conditions for the membrane formation. Conditions are stated without proof. Formal discussions will be presented elsewhere.

## 2 Definitions and Notations of Generalized Prisoner's Dilemma

We have studied a spatial version of PD, and proposed a generalized TFT (Tit-for-Tat; it would defect only when the adversary defects, and would cooperate otherwise) such as *k1C, k2D* and their combination *k1C-k2D*, where *k1* is a parameter indicating generosity and *k2* contrariness. Dynamics of these spatial strategies in a two-dimensional lattice has been also studied in a noisy environment.

The PD is a game played just once by two players with two actions (cooperation, C, or defect, D). Each player receives a payoff ($R$, $T$, $S$, $P$) where $T > R > P > S$.

In IPD (Iterated PD), each player (and hence the strategy) is evaluated with further constraint: $2R > T + S$. In Spatial Prisoner's Dilemma (SPD), each site in a two-dimensional lattice corresponds to a player. Each player plays PD with the neighbors (8 adjacent players as in Fig.1), and changes its action by the total score it received.

Our model generalized SPD by introducing spatial strategy. Each player placed at each lattice of the two-dimensional lattice. Each player has an action and a strategy, and receives a score. Spatial strategy determines the next action dependent upon the spatial pattern of actions in the neighbors. Each player plays PD with the neighbors, and changes its strategy to the strategy that earns the highest total score among the neighbors. Table 1 is the Payoff matrix of PD. In our simulations, $R$, $S$, $T$, and $P$ are respectively set to $1, 0, b$ ($1 < b < 2$, a bias for defectors) and 0 in simulations below following the Nowak-May's simulations [6].

**Table 1.** The Payoff Matrix of the Prisoner's Dilemma Game. *R, S, T, P* are payoff to the player 1. ($1 < b < 2$)

| | | Other | |
|---|---|---|---|
| | | C | D |
| Player | C | $R(1)$ | $S(0)$ |
| | D | $T(b)$ | $P(0)$ |

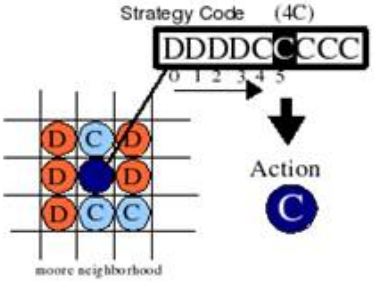



**Fig. 1.** A Strategy Code for Spatial Strategies

Our SPD is done with spatial strategies: the next action will be determined based on the pattern of neighbors' actions. Score is calculated by summing up all the scores received from PD with 8 neighbor players and itself. After $q$ (strategy update cycle) steps of interactions with neighbors and self, the strategy will be chosen from the strategy with the highest score among the neighbors including the self. Thus, the strategy will be updated at every $q$ (set to be 1 throughout this paper) steps. In an evolutionary framework, strategy will be also changed by a mutation rate (set to be 0 throughout this paper) where mutation is operated on the string of the strategy code below.

To specify a spatial strategy, actions of the eight neighbors in the neighborhood radius $r = 1$ (i.e., the *Moore* neighbors as in Fig.1. When $r = 2$, it would be 24 neighbors.) and the player itself must be specified (Fig.1), hence $2^9$ rules are required. For simplicity, we restrict ourselves on a "totalistic spatial strategy" that depend on the number of D (defect) action of the neighbor, not on their positions.

This $k$-D can be regarded as a spatial version of TFT where $k$ indicates the spatial version of the generosity [11] (how many D actions in the neighbor are tolerated.).

## 3 Membrane Formation

In studying a mechanism that allows cooperators clusters to be preserved, we are studying a spatial version of generosity: how many defections in the neighborhood are tolerated rather than how many previous defections of the opponent in the spatio-temporal generalized context. In interactions between All-D v.s. $k$-D instead of All-D v.s. All-C (as in Nowak-May's SPD), we found that clusters of $k$-D form membrane of action D protecting the inner cluster of action C (Note that $k$-D can take both C or D depending on the number of Ds in the neighborhood). We observed that this membrane formation occurs as in Fig. 2 when a certain parameter scope of $k$ (spatial generosity), $r$ (neighborhood radius) and $b$ (bias for defectors). We will focus on the condition of the membrane formation with respect to these parameters. Throughout this note, simulations are conducted in a square lattice with periodic boundary condition with the following parameters listed in Table 2.

**Table 2.** List of Parameters for Simulations

| Name | Description | Value |
|---|---|---|
| $L \times L$ | Size of the space | 1,500 × 1,500 |
| $N$ | Number of the players | 2,250,000 |
| $T$ | Number of steps | 2,500 |
| $N_{k-D(0)}$ | Initial number of the $k$-D with C state | All-D and $k$-D are randomly assigned with equal probability. Similarly to C/D. |
| $r$ | Neighborhood radius | 1, 2, 3 |

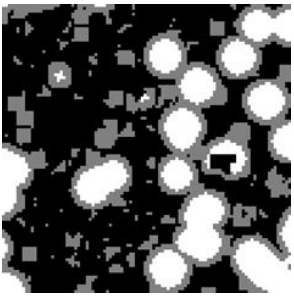


**Fig. 2.** Membrane formation in a generalized PD. Black cells are All-D; white and gray cells are C and D state of $k$-D, respectively. In this snapshot, $k = 6$ and 6-D strategy players are allocated in a random positions initially.

The parameter $b$ is set to be a minimal value that allows All-D to expand. Fig. 3 shows a part of the square lattice where C and D players are indicated white and black cells respectively. For All-D in the corner (indicated by a red circle) to gain the profit higher than the cooperators, b must satisfy $5b > 9$ since the highest profit of the cooperators is 9 when $r = 1$ (similarly, $16b > 25$ when $r = 2$, and $33b > 49$ when $r = 3$).

After the membranes are formed, the following three phenomena are observed depending on the $k$ value.

1. $k$ is too small: The membrane will grow toward the center of $k$-D cluster and will corrode the C state of $k$-D.
2. $k$ is small: The $k$-D cluster covered by the membrane will stay stable.
3. $k$ is large: The $k$-D cluster covered by the connected membrane will expand.
4. $k$ is too large: The $k$-D cluster covered by the broken membrane will expand and the cluster will eventually collapse.

Since we are interested in conditions for cooperators to be preserved, we focus on the conditions on the cases 2 and 3 above. The spatial generosity $k$ increases as the case proceeds downward from 1 to 4.

For the membrane formation, the spatial generosity $k$ must exceed a certain value formulated by the neighborhood radius $r$:

$$k \geq (2r + 1)(r + 1) - r.$$

Otherwise, the membrane will grow inside toward the center of the $k$-D cluster as in case 1.

For the cluster protected by the membrane to expand, the spatial generosity $k$ must further exceed a larger threshold:

$$k \geq (2r + 1)(r + 1).$$

Otherwise, the cluster does not expand although it is indeed protected by the membrane.

In the case 4, membrane is broken if $k$ exceeds a threshold (Fig. 4)

$$k \geq (2r+1)^2 - \sum_{i=0}^{i<\frac{-1+\sqrt{8r+17}}{2}} P(\{-\frac{i(i+1)}{2} + (2+r)\}),$$

where $P(x) = x$ when $x > 0$ and $P(x) = 0$ otherwise, and the k-D clusters with the broken membrane will collapse when they contact with each other in the expansion.

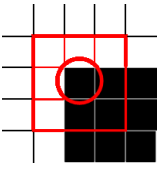


**Fig. 3.** C and D players are indicated white and black cells respectively. For All-D in the corner (indicated by the circle) to gain the profit higher than the cooperators, b must satisfy $5b > 9$ since the highest profit of the cooperators is 9 when $r = 1$ (similarly, $16b > 25$ when $r = 2$, and $33b > 49$ when $r = 3$).

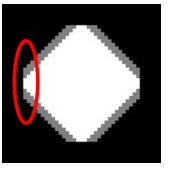


**Fig. 4.** Broken membrane indicated by the oval. The minimal cluster of 7-D is set in the center of the sea of All-D. All-D is indicated by black cell, while C state and D state of 7-D is indicated by white and gray cell, respectively. The figure shows a snapshot after 16 steps starting from $3 \times 3$ square of $k$-D in the center.

Fig. 5 plots the time evolution the fraction of $k$-D when the simulation starts from a random configuration stated with parameters as stated in Table 2. The fraction of 6-D is highest among other $k$-Ds, since the membrane protects the expanding 6-D clusters. The fractions of 7-D and 8-D are lower than 6-D because the membranes are broken in these $k$-Ds. The fraction of 5-D is the lowest, since the 5-D cluster does not expand although the cluster is protected by the membrane.

## 4 Discussions

We proposed yet another mechanism for preserving and protecting the cluster of cooperation in the spatio-temporally generalized Prisoner's Dilemma. After the membrane is formed, it can protect the cluster of cooperators from being invaded by the defectors. The condition for the membrane formation can be formulated by the parameter indicating spatial generosity. If the spatial generosity is too large, the membrane will be broken, while the membrane will develop into inside the cluster eradicating the cooperators if the spatial generosity is too small.

We also observed that several different polygons will form depending on the parameters and lattice topology. This phenomenon will be related to crystal

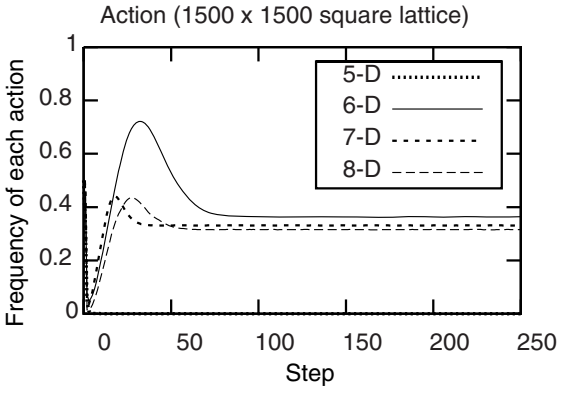



**Fig. 5.** Time evolution of the fraction of $k$-D when $r = 1$ and $b = 1.800010$. The fraction of 6-D is highest among other $k$-Ds, since the membrane protects the expanding 6-D clusters. The fractions of 7-D and 8-D are lower than 6-D because the membranes are broken in these $k$-Ds. The fraction of 5-D is the lowest, since the 5-D cluster does not expand although the cluster is protected by the membrane.

formation of different shapes in physical phenomena, while the membrane formation is related to biological phenomena where clusters of cells and chemical substances must be preserved for a certain amount of time and space.

**Acknowledgment.** This work was supported by The Global COE Program Frontiers of Intelligent Sensing, from the ministry of Education, Culture, Sports, Science and Technology. This work was also supported in part by Grants-in-Aid from Toyohashi University of Technology.

## References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
2. Axelrod, R.: The Evolution of Strategies in the Iterated Prisoner's Dilemma. In: Davis, L. (ed.) Genetic Algorithms and Simulating Annealing, Pitman, pp. 31–41 (1987)
3. Axelrod, R., Dion, D.: The Further Evolution of Cooperation. Science 242, 1385–1390 (1988)
4. Boyd, R.: Mistakes Allow Evolutionary Stability in the Repeated Prisoner's Dilemma Game. J. theor. Biol. 136, 47–56 (1989)
5. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. Nature 359, 826–829 (1992)
6. Nowak, M.A., May, R.M.: The spatial dilemmas of evolution. Int. J. Bifurc. Chaos 3, 35–78 (1993)
7. Hauert, C., Doebeli, M.: Spatial structure often inhibits the evolution of cooperation in the spatial Snowdrift game. Nature 428, 643–646 (2004)
8. Nowak, M.A., Sasaki, A., Taylor, C., Fudenberg, D.: Emergence of cooperation and evolutionary stability in finite populations. Nature 428, 646–650 (2004)

9. McNamara, J.M., Barta, Z., Houston, A.I.: Variation in behaviour promotes cooperation in the Prisoner's Dilemma game. Nature 428, 745–748 (2004)
10. Ishida, Y., Mori, T.: Spatial Strategies on a Generalized Spatial Prisoner's Dilemma. J. of Artificial Life and Robotics 9(3), 139–143 (2005)
11. Grim, P.: The greater generosity of the spatialized prisoner's dilemma. J. theor. Biol. 173, 353–359 (1995)

# An Asynchronous Cellular Automaton Implementing 2-State 2-Input 2-Output Reversed-Twin Reversible Elements

Jia Lee[1], Ferdinand Peper[2], Susumu Adachi[2], and Kenichi Morita[3]

[1] Celartem Technology Inc., Japan
lijia@celartem.com
[2] Nano ICT Group,
National Institute of Information and Communications Technology, Japan
[3] Dept. of Information Engineering, Hiroshima University, Japan

**Abstract.** Reversible computers usually work in a synchronous mode, i.e., in the presence of clock signals, but in the light of the asynchronous nature of microscopic physical phenomena this may be an anomaly. The alternative, an asynchronous mode of operation, has therefore attracted attention from researchers, witness the proposal of a reversible circuit element in (Morita 2001) that works in such a mode. Simplicity of circuit elements like this is an important design objective since it correlates positively with the efficiency by which they may be realized physically. In this paper, we present two mutually inverse logic elements that compare favorably to other circuit elements in terms of their number of states and their number of input and output lines. We show that the proposed circuit elements can perform universal computation by embedding circuits made of them in asynchronous cellular automata.

## 1 Introduction

Reversible logic has its origins in computing schemes that achieve near-zero power consumption by preventing entropy loss in computations. It has been extensively studied [1,2,3,4,5,6], but always under the assumption that timing is synchronous, i.e., that all logic elements switch simultaneously in accordance with a central clock. For example, the Fredkin gate [4], a well-known reversible gate, fails to work correctly if all its input signals would arrive at different times. Asynchronous systems have virtually been unexplored for reversible computing, probably since the randomness by which events in them are timed appears incompatible with the backward determinism of reversible computing. This lack of interest may be hard to defend in the light of the existence of microscopic physical interactions that are both asynchronous and reversible. As with reversibility, asynchrony tends to reduce power consumption, be it for different reasons: logic elements in an asynchronous system need not be active in the absence of signals, unlike in synchronous systems, in which idle logic elements may engage in dummy switching events triggered by the continuous arrival of clock signals [7].

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 67–76, 2008.

To achieve low-power computing in practice, it makes sense to investigate the combination of asynchrony and reversibility and particularly to find asynchronous reversible logic elements to be used as the basic building blocks to construct universal circuits. Intuitively, the less complex a logic element is, the easier it can be implemented physically—a reason to look for elements with as few input lines, output lines, and internal states as possible. Unsuitable for use in an asynchronous framework are the reversible logic elements typically employed in synchronous circuits, since they lack the timing functionality to make up for the absence of a clock. A straightforward measure of a circuit element's complexity is the ease by which it can be implemented in cellular automata: for, a complex functionality usually translates in an increased number of cell states and transition rules.

Patra and Fussell [8] has studied asynchronous reversible systems in the context of *Delay-Insensitive (DI) circuits*. A delay-insensitive circuit (e.g. see [7]) is an asynchronous circuit in which signals may be subject to arbitrary delays without this being an obstacle to the circuit's correct operation. The circuits constructed in [8] are not reversible in the strict sense, since his constructions require a *Merge-element*—an irreversible element that merges two input streams of signals into a single output stream of indistinguishable signals.

Morita [9], has proposed a DI reversible logic element, called a *Rotary Element* (RE), from which computationally universal models can be constructed, including a reversible Turing machine. The RE has four input lines, four output lines, and two internal states. An improved element with three input lines, three output lines, and two internal states is proposed in [10], and more such elements are investigated in [11]. In both references [10,11] it is proven that each of the proposed elements can be used as a basis into which the RE can be decomposed, which implies that the elements are universal. In [12] an asynchronous reversible cellular automaton is proposed that implements the RE. The construction of the reversible Turing machine from REs in [9] implies the universality of this cellular automaton.

This paper proposes a pair of DI reversible logic elements each of which has two input lines, two output lines, and two internal states. These elements are each other's functional reverses, i.e., running signals backwards through one element produces the equivalent of the other element. We prove that the two elements as a set are universal by constructing an RE from them. The asynchronicity of these two elements combined with their mutually reversed functionalities enable efficient implementation in a special type of asynchronous cellular automata, called *Self-Timed Cellular Automata* (STCA) [13], such that merely five transition rules are required.

Section 2 defines the proposed reversible elements in detail. In section 3 an RE is constructed from the proposed elements; this result implies the universality of the elements, since a universal reversible Turing machine can be constructed from RE elements [9]. Implementation of these two elements and circuits based on them in terms of an asynchronous cellular automaton are described in section 4. The paper finishes with conclusions and a short discussion.

## 2 Reversible Logic Elements

A *reversible sequential machine* [9] is a system defined as $N = (Q, \Sigma, \Delta, \delta)$, where $Q$ is a finite set of states ($Q \neq \emptyset$), $\Sigma$ and $\Delta$ are sets of input and output symbols, respectively. The transition function $\delta : Q \times \Sigma \rightarrow Q \times \Delta$ is bijective. A reversible sequential machine is a special type of Mealy machine [14].

A *reversible serial module* is a system defined as $M = (I, O, N)$, where $I$ and $O$ are two sets of input and output lines, respectively ($I \cap O = \emptyset$). $N = (Q, \Sigma, \Delta, \delta)$ is a reversible sequential machine with $I$ in one-to-one correspondence with $\Sigma$, and $O$ in one-to-one correspondence with $\Delta$.

Signals used for inputs and outputs of a reversible serial module are considered particles. The binary signals 1 and 0 are encoded by the presence or absence, respectively, of a particle on a line. Let $\mu : I \rightarrow \Sigma$ be the bijective function between $I$ and $\Sigma$, and $\nu : O \rightarrow \Delta$ be the bijective function between $O$ and $\Delta$. A reversible serial module $M$ is said to be in state $q (\in Q)$ if the reversible sequential machine $N$ of $M$ is in state $q$. Assume $a \in I$, $b \in O$ and $q, q' \in Q$ such that $\delta(q, \mu(a)) = (q', \nu^{-1}(b))$. Then if a particle arriving on input line $a$ is received by $M$ in state $q$, $M$ operates on this particle such as to transfer it from $a$ to output line $b$, and to change $M$'s state from $q$ to $q'$. The operation of $M$ is undefined for simultaneous input signals on its input lines. In other words, a reversible serial module can only process at most one input particle at any time. The operation of $M$ is reversible, in that from the current state and output, the previous state and input can be uniquely determined due to the bijective transition function $\delta$.

We present two reversible serial modules, which have inverse functionalities. One of the modules, called the *Reading Toggle* (RT) element, is defined as $(\{S, T\}, \{T_A, T_B\}, N_{RT})$, where $N_{RT} = (\{A, B\}, \Sigma_{RT}, \Delta_{RT}, \delta_{RT})$ (see Fig. 1). Let $\mu : \{S, T\} \rightarrow \Sigma_{RT}$ be the bijective function between $\{S, T\}$ and

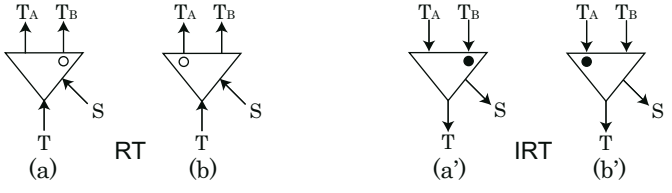



**Fig. 1.** RT element in (a) state A, and (b) state B. IRT element in (a') state A, and (b') state B.

$\Sigma_{RT}$, and $\nu : \{T_A, T_B\} \rightarrow \Delta_{RT}$ be the bijective function between $\{T_A, T_B\}$ and $\Delta_{RT}$. The RT element operates such that a particle arriving on input line $T$ is transferred to output line $T_A(T_B)$ if the RT is in state A(B); in this case, the state changes to B(A) (upper row of Fig. 2(a)). A particle arriving on input line $S$ is merely transferred to output line $T_A(T_B)$ if the RT is in state A(B), without the state being changed (lower row of Fig. 2(a)).

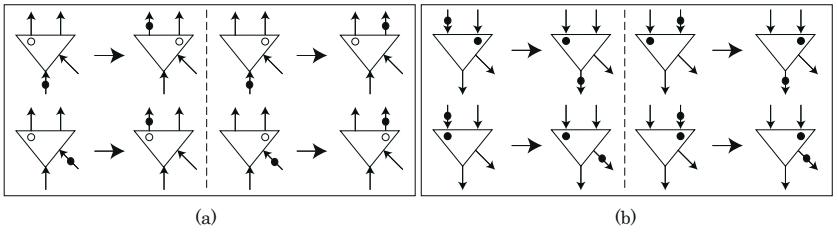



**Fig. 2.** (a) RT and (b) IRT operating on a particle arriving on one of its input lines. The particle is denoted by a token on a line. The two transition rules of (a) in the upper resp. lower row describe the RT's operation in case of a particle arriving on line $T$ resp. $S$. In addition, the two transition rules of (b) in the upper resp. lower row describe the IRT's operation in case of a particle being output to line $T$ resp. $S$.

The other reversible serial module, the *Inverse Reading Toggle* (IRT) element, is defined as $(\{T_A, T_B\},\ \{S, T\},\ N_{IRT})$ where $N_{IRT}$ is defined as $N_{IRT} = (\{A, B\},\ \Sigma_{IRT},\ \Delta_{IRT},\ \delta_{IRT})$. Let $\mu' :\ \{S, T\} \rightarrow \Sigma_{IRT}$ be the bijective function between $\{T_A, T_B\}$ and $\Sigma_{IRT}$, and $\nu' :\ \{T_A, T_B\} \rightarrow \Delta_{RT}$ be the bijective function between $\{S, T\}$ and $\Delta_{IRT}$. The IRT element operates such that a particle arriving on input line $T_A(T_B)$ is transferred to output line $T$ if the IRT is in state B(A); in this case, the state changes to A(B) (upper row of Fig. 2(b)). A particle arriving on input line $T_A(T_B)$ is merely transferred to output line $S$ if the IRT is in state A(B), without the state being changed (lower row of Fig. 2(b)). Simultaneous particles on the input lines of RT or IRT are not allowed. Obviously, both RT and IRT are reversible, and they are each other's inverse.

## 3 Construction of RE by the Reversible Elements

Any reversible Turing machine (for more details on such machines see [1,15]) can be constructed by using a network of REs, in which at most one particle moves around at any time [9]. Since delays in any of the REs or lines do not affect the correctness of the computing process in the circuit, this circuit is DI. Such reversible computers consisting of REs need no central clock signal to drive the operations of each RE [9], i.e., they are asynchronous.

We construct an RE from RT and IRT elements to show the universality of the RT and IRT elements in an asynchronous mode of operation. An RE is a reversible serial module that is defined as $(\{n, e, s, w\}, \{n', e', s', w'\}, N_{RE})$ with $N_{RE} = (\{H, V\},\ \Sigma_{RE},\ \Delta_{RE},\ \delta_{RE})$ (see Fig. 3). Let $\hat{\mu} :\ \{n, e, s, w\} \rightarrow \Sigma_{RE}$ be the bijective function between $\{n, e, s, w\}$ and $\Sigma_{RE}$, and $\hat{\nu} :\ \{n', e', s', w'\} \rightarrow \Delta_{RE}$ be the bijective function between $\{n', e', s', w'\}$ and $\Delta_{RE}$. The RE operates such that if a particle comes from a direction parallel to the rotating bar of an RE, it passes straight through to the opposite output line, without changing the direction of the bar (the state of the RE), as in Fig. 4(a); if the particle comes from a direction orthogonal to the rotating bar, it is deflected to the right, and the bar rotates by 90 degrees (Fig. 4(b)). An RE remains in its state if no particle

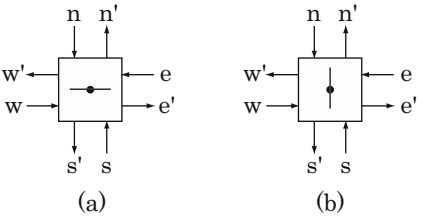



**Fig. 3.** An RE in (a) the $H$-state, and (b) the $V$-state, displayed as respectively horizontal and vertical bars in the RE

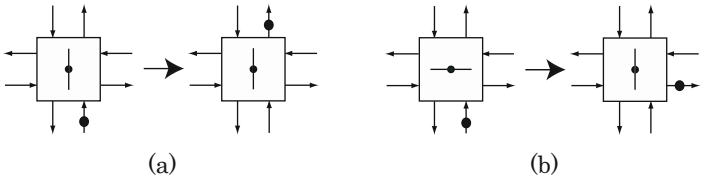



**Fig. 4.** REs operating on an input particle in (a) the parallel case, and (b) the orthogonal case

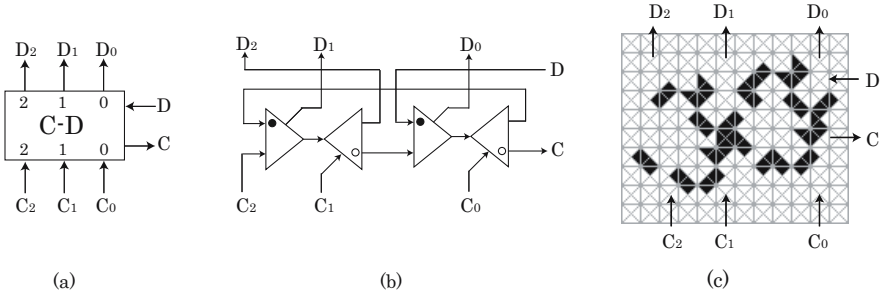



**Fig. 5.** (a) A C-D module. (b) The realization of a C-D module in state 0 from RT and IRT elements. (c) Construction of C-D module in the STCA (see Section 4 for details).

arrives on any of its input lines. Simultaneous particles on any pair of input lines of an RE are not allowed.

In [9] is was shown that the RE element is universal by composing a circuit of RE modules that can simulate a universal Turing machine. To show that any reversible Turing machine can be realized from RT and IRT elements, it suffices to construct an RE out of these elements. We first construct an intermediate module from the RT and IRT elements to simplify the construction of the RE. This module, called a *Coding-Decoding* (C-D) module [10] (see Fig. 5), has four input lines $\{C_0, C_1, C_2, D\}$, four output lines $\{D_0, D_1, D_2, C\}$, and three states $\{0, 1, 2\}$. If the C-D module is in state 0, an input particle arriving on input line $C_i$ ($i \in \{0, 1, 2\}$) changes the state of the C-D module from 0 to $i$, and the particle is transferred to output line $C$. Then, a subsequent particle coming

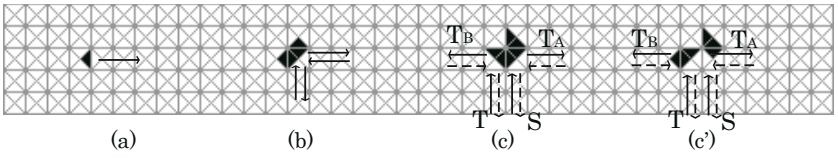



**Fig. 6.** (a) A signal. A subcell in state 1 is denoted by a filled triangle, while a subcell in state 0 is denoted by a blank. (b) A left or right turn element. (c) The configuration representing an RT or IRT element in state A. The direction of the input and output signals can be both ways: one way, indicated by the solid arrows, makes the configuration work as an RT element, whereas the other way, indicated by dashed arrows, corresponds to an IRT element. (c') Configuration representing an RT or IRT element in state B. The arrows indicate the directions of signal propagation for the RT and the IRT elements.

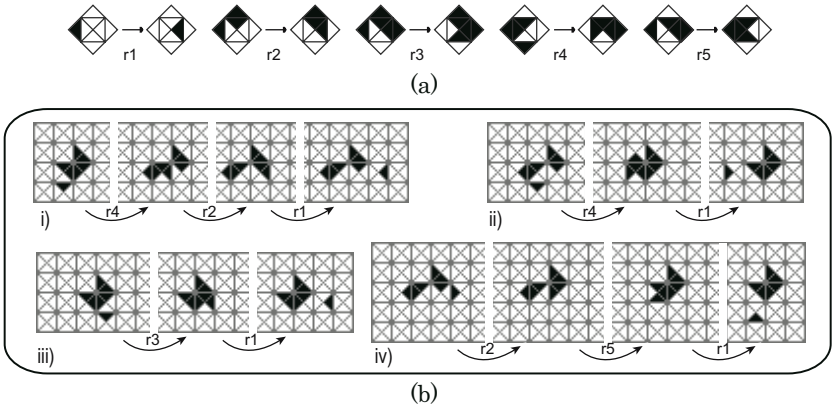



**Fig. 7.** (a) Transitions rules of STCA. The rotational and reflective equivalents of the rules are omitted. (b) RT and IRT elements operating on input signals: An RT element receiving a signal from input path $T$ when it is in state i) $A$ or ii) $B$. iii) The RT receiving an input signal from path $S$ when it is in state $A$. iv) An IRT receiving a signal from input path $T_A$ when it is in state $B$. Each arrow indicates one transition step of cells, whereby its label refers to the corresponding transition rule. It can be verified that the RT (or IRT) element here will fail to work on a signal arriving on its input path $S$ (resp. $T_B$) when it is in state $B$. Implementation of the full functionalities of RT and IRT elements is possible, but this tends to increase the number of rules and result in more complicated cellular configurations as compared to those in Fig. 6.

from input line $D$ is transferred to output line $D_i$ if the C-D module is in state $i$, and the state is reset to 0. The C-D module is unable to receive input from lines $C_0, C_1$, or $C_2$ if it is in states 1 or 2. We apply the C-D module in the construction of the RE such that an input particle arriving on an input line of the C-D module is always followed by an output particle on an output line, before a new particle is input to an input line.

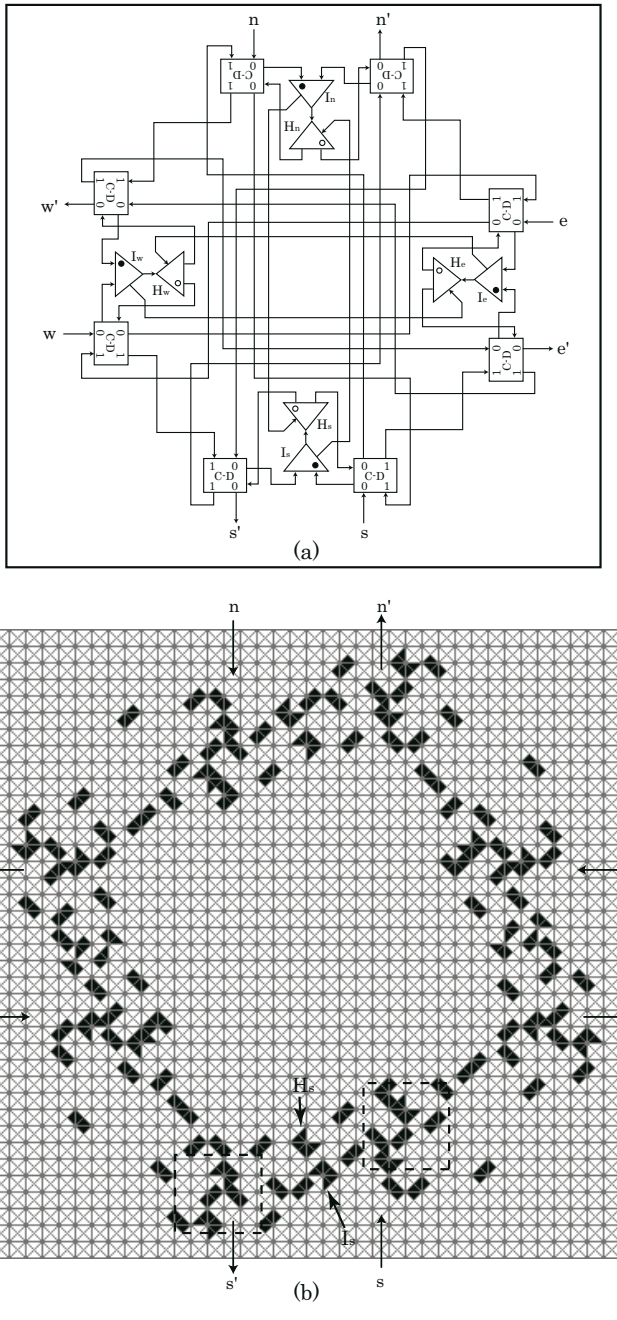



**Fig. 8.** (a) Realization of an RE in state $V$ from RT and IRT elements, in which all C-D modules are in state 0 initially. (b) Construction of this RE in the STCA. Dashed boxes are put around the areas in (b) in which C-D modules adjacent to the elements $H_s$ and $I_s$ are placed according to the construction in (a).

Figure 8(a) shows in detail the realization of an RE from RT and IRT elements, whereby the four subcircuits of RT and IRT elements, indicated by $(H_s, I_s)$, $(H_e, I_e)$, $(H_n, I_n)$, and $(H_w, I_w)$, are in states $(A, A)$, $(B, B)$, $(A, A)$, and $(B, B)$, respectively; this represents the RE being in the $V$-state. The same subcircuits being in states $(B, B)$, $(A, A)$, $(B, B)$, and $(A, A)$, respectively, represent the RE being in the $H$-state. This result implies that we can construct a circuit from RT and IRT elements that simulates a reversible Turing machine, in which at most one particle moves around the entire circuit [9]. Thus, the RT and IRT elements are logically universal, and can work in asynchronous mode, i.e. without their operations having to be driven by a central clock.

Finally, from the constructions in Figs. 5(b) and 8(a), it can be observed that each RT (or IRT) element never receives a signal from input line $S$ (resp. $T_B$) when it is in state $B$. This implies that the functionalities of the RT and IRT can be further simplified, which tends to benefit their implementations, of these two elements on asynchronous cellular automata, as the next section shows.

## 4 Embedding Reversible Elements in STCA

A Self-Timed Cellular Automaton is a two-dimensional array of identical cells. Each cell is partitioned into four subcells in one-to-one correspondence with its four nearest neighboring cells, and each subcell takes only one of two states: 0 or 1. Each cell undergoes state transitions via transition rules that operate on the cell itself along with the nearest subcells of each of its four neighbors. Moreover, the update of cells are timed randomly and independently of each other, and hence, are asynchronous.

Figure 6 shows some fundamental patterns used in the STCA. In particular, the pattern in Fig. 6(a) represents a signal that will be transferred to the right; whereas the pattern in Fig. 6(b) is used to change the direction of a signal to the left or right. Moreover, both the local configurations in Fig. 6(c) and (c') represent an RT or an IRT element. Their difference corresponds to the two internal states: $A$ and $B$, of an RT or IRT element, respectively. The update of all the patterns in Fig. 6 are controlled by the five transition rules given in Fig. 7(a), for example, as demonstrated in Fig. 7(b).

Following the construction in Fig. 5(b), we are able to lay out an C-D module in our asynchronous cellular automaton by the configuration illustrated in Fig. 5(c). Furthermore, in accordance with the circuit scheme in Fig. 8(a), we lay out the configuration of an RE on the STCA in Fig. 8(b). This implies that a Turing machine can be constructed on the cellular automaton, provided the cellular space is sufficiently large.

## 5 Conclusions and Discussion

This paper proposes two asynchronous reversible logic elements that are each others' mutually reverse. Called RT and IRT, the elements have two input lines, two output lines, and two states. The elements are universal, because they can

be used to construct Morita's RE element [9], which is universal. Moreover, the elements are less complex than the RE. Following the line of thought in [9], we can construct a universal reversible computer from RTs and IRTs, in which at most one particle moves around at any time. This allows the RT and IRT elements to conduct their computational tasks asynchronously without needing a central clock signal to drive their operations.

The implementation of the proposed circuit elements on the cellular automaton requires five transition rules, which is one rule more than the implementation of the RE [12]. This indicates that the circuit elements may be slightly more complex in functionality than the RE, even though they require less input and output lines. A reason for the greater complexity could be the symmetry of the RE, as opposed of the lack thereof of the proposed elements, as well as the fact that all functionality of the RE is concentrated in one module, as opposed to the two modules required in this paper. Still, the number of five rules required here lies closely to the four rules for the RE model, implying that both models are on par with each other. Other implementations—possibly physical—may lead to a different outcome, favoring the proposed elements over the RE.

We have seen that the circuits constructed from the proposed elements allow merely one particle to be present at a time. To realize circuits with multiple particles, we need to combine the elements with a so-called *Join* element [8]—an element with two input lines and one output line, which requires input particles to be present on both input lines in order to produce output, whereas a single input to the Join is just kept pending until a second input arrives. Further research is needed, however, to rigorously define such multiple-signal asynchronous reversible circuits, as pointed out in [10]. The implementation of the Join on an STCA is likely to require at least one more transition rule.

## References

1. Bennett, C.: Logical reversibility of computation. IBM Journal of Research and Development 17(6), 525–532 (1973)
2. Vos, A.D., Raa, B., Storme, L.: Generating the group of reversible logic gates. Journal of Physics A: Mathematical and General 35(33), 7063–7078 (2002)
3. Frank, M., Vieri, C., Ammer, M., Love, N., Margolus, N., Knight Jr., T.F.: A scalable reversible computer in silicon. In: Calude, C.S., Casti, J., Dinneen, M.J. (eds.) Unconventional Models of Computation, pp. 183–200. Springer, Singapore (1998)
4. Fredkin, E., Toffoli, T.: Conservative logic. International Journal of Theoretical Physics 21(3-4), 219–253 (1982)
5. Margolus, N.: Physics-like models of computation. Physica D 10(1/2), 81–95 (1984)
6. Merkle, R.: Reversible electronic logic using switches. Nanotechnology 4(1), 21–40 (1993)
7. Hauck, S.: Asynchronous design methodologies: an overview. Proc. IEEE 83(1), 69–93 (1995)
8. Patra, P., Fussell, D.: Conservative delay-insensitive circuits. In: Workshop on Physics and Computation (PhysComp 1996) (1996)

9. Morita, K.: A simple universal logic element and cellular automata for reversible computing. In: Margenstern, M., Rogozhin, Y. (eds.) MCU 2001. LNCS, vol. 2055, pp. 102–113. Springer, Heidelberg (2001)
10. Lee, J., Peper, F., Adachi, S., Mashiko, S.: On reversible computation in asynchronous systems. In: Quantum Information and Complexity, pp. 296–320. World Scientific, Singapore (2004)
11. Tanaka, K.: Universal reversible logic elements with 3 inputs, 3 outputs, and 2 states. Master Thesis, Hiroshima University (in Japanese) (2003)
12. Lee, J., Peper, F., Adachi, S., Morita, K., Mashiko, S.: Reversible computation in asynchronous cellular automata. In: Calude, C.S., Dinneen, M.J., Peper, F. (eds.) UMC 2002. LNCS, vol. 2509, pp. 220–229. Springer, Heidelberg (2002)
13. Peper, F., Isokawa, T., Kouda, N., Matsui, N.: Self-timed cellular automata and their computational ability. Future Generation Computer Systems 18(7), 893–904 (2002)
14. Mealy, G.: A method for synthesizing sequential circuits. Bell System Technical Journal 34(5), 1045–1079 (1955)
15. Morita, K., Shirasaki, A., Gono, Y.: A 1-tape 2-symbol reversible Turing machine. Trans. IEICE Japan E-72, 223–228 (1989)

# Game 'Life' with Anticipation Property

Alexander Makarenko[1], Boris Goldengorin[2], and Dmitry Krushinsky[2]

[1] Institute for Applied System Analysis at National Technical University of Ukraine (KPI), 37 Pobedy Avenue, Kiev, 03056, Ukraine
makalex@i.com.ua
[2] University of Groningen, Faculty of Economics and Business, P.O. Box 800 9700AV, Groningen, The Netherlands
{b.goldengorin,d.krushinsky}@rug.nl

**Abstract.** Mechanisms of endowing cellular automata (CA) models with anticipation property are considered. As a background for investigation a widely known Conway's game of 'Life' was chosen. Even within this simple model, introduction of anticipation has a crucial impact on the behavior of the system, resulting in emergence of multiple co-existing solutions. The inherent behavioral templates of classical 'Life' (fixed point, cycle, chaos) are valid for its anticipatory version; however, they develop at a level of sets of configurations. Extensive computational simulations were held and the most eloquent results are presented.

**Keywords:** Cellular automata, anticipation, game 'Life'.

## 1 Introduction

Nowadays, perhaps, all those dealing with computers are familiar with Conway's computer game 'Life' [1]. In its simplest variant this game may be considered as a computer implementation of algorithms from a class of so-called cellular automata (CA) (see, for example, [2], [3], [4]). A very brief description of the simplest algorithm for a two-dimension case (numerous modifications may be found in literature on cellular automata) is given below in order to make subsequent material more clear.

Let $Z_2$ be a 2D grid; S is a finite set of states of elements (cells) of the grid, $s_i$ from S is a state of *i*-th cell from $Z_2$. A set of all the cells states at a given moment of time will be denoted as a configuration upon a grid $Z_2$. All the possible configurations make up a configuration space C on $Z_2$. Let $t \in \{0, 1, 2, \ldots\}$ – discrete moments of time and C(t) – configuration at time t.

Local rule updating the state of a cell *k* from $Z_2$ is a predefined transform $T_k$, that turns the present state of a cell to the state at next moment of time.
The rules of state change are local. This means that the state of *k*-th cell depends on the states of cells from its local neighborhood $N_k$.

In "classical" Conway's game 'Life' [1] every cell has only two states (denoted as 0 and 1, S={0,1}); a neighbourhood used in local rules is a Moore's neighbourhood consisting of 8 immediate neighbours.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 77–82, 2008.

In a majority of 'Life' descriptions cell states of 0 and 1 are treated according to the biological interpretation as "dead" and "living" cell. Thus, cells state change may be defined by the following rules:

1. If a living cell has two or three living neighbours then it stays alive at the next moment of time; otherwise it dies.
2. If a dead cell has exactly three living neighbors then it turns to a "living" state at the next moment of time.
3. All the births and deaths occur almost simultaneously.

It should be noted, that both game 'Life' and cellular automata in general were studied by a number of researchers involving various approaches: analytical, quantitative and qualitative. Development of theory and applications resulted in what is now called a "CA approach". In order to illustrate modeling capacity of CA one may mention car and pedestrian traffic, spread of epidemics, applications in hydrodynamics, biology, quantum mechanics and astrophysics. Moreover, several hypotheses suggesting that cellular automata may be considered as a separate paradigm clarifying the structure of the Universe have emerged recently (see, for example, [4]).

However, as far as the origins of CA approach are considered, one should mention research on living organisms, especially their collective activity in human brain (see, for example, McCulloc, Pitts et al.) and, in parallel, cybernetics in the form of theory of automata and Turing machines [5], logical investigations and attempts to simulate intelligent behavior; historical survey on CA [6] is especially interesting. Thus, the very origins of emergence and development of cellular automata are strongly related to research on living systems. Besides, understanding that individual behavior accounting in large social systems and, especially, accounting of their mental characteristics may bring about a revelation of unexpected properties (at least in models [7]) have appeared recently.

A set of these characteristics is enormous [8] and they are being gradually engaged in research on CA [9]. However, there is a mental characteristic that may have far-reaching consequences if being taken into account – anticipation property. The former implies that, while evolving to the next time step, cells take into account their predicted data about the future state of their neighborhood [10].

One way in development of CA theory is based on an attempt to adequately map mental properties, especially anticipation, within the CA framework. So, further part of the paper presents some results of corresponding research and illustrates to what a substantial change in behavior of a CA taking this property into account may lead. In particular, results on the game 'Life' endowed with anticipation property (we have called this model 'LifeA', 'Life' with anticipation) are given.

## 2 'LifeA' Model: A Brief Description

### 2.1 Next State Function

Let's consider rules driving the classical 'Life'. To formalize them, we have introduced two auxiliary functions $f_0(x)$ and $f_1(x)$, their values are given below.

**Table 1.** Auxiliary functions $f_0(x)$ and $f_1(x)$

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| $f_0(x)$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $f_1(x)$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Suppose that out of the interval [0;8] both these functions are zero-valued. Moreover, if the argument is not an integer value, it is rounded to the nearest integer.

Under the introduced notions next state function for *k*-th cell will be as follows:

$$F_k = F(S_k) = \begin{cases} f_0(S_k), C_k = 0 \\ f_1(S_k), C_k = 1 \end{cases}, F_k \in \{0,1\} \ , \tag{1}$$

where $S_k$ – number of (non-zero) neighbors of *k*-th cell, $C_k \in \{0;1\}$ (0 if *k*-th cell is dead, 1 - otherwise).

Thus, the evolution of an automaton with N cells may be denoted as:

$$C_k^{t+1} = F_k^t = F(S_k^t), k = 1..N \ . \tag{2}$$

### 2.2 Anticipation

To endow a model with anticipation property one has to design a next step function that depends on the next (expected) cell states [10]. In this case, next state function of classical 'Life' defined as:

$$F_k^t = F(S_k^t) \tag{3}$$

was replaced by the following two functions:

$$F_k^t = F((1-\alpha)\cdot S_k^t + \alpha \cdot S_k^{t+1}), \alpha \in [0;1] \tag{4}$$

$$F_k^t = F(S_k^t + \alpha \cdot S_k^{t+1}), \alpha \in IR \tag{5}$$

Further, these variants of a next state function will be referred to as weighted and additive, correspondingly. Cellular automata driven by either of these functions were implemented.

### 2.3 Topology of the Example

Like that of classical 'Life', neighborhood of a cell consists of its 8 neighbors, i.e. this is a Moore's neighborhood with "punctured" center. The value of this center cell ($C_k$) is taken into account immediately in the next state function.

In order to avoid irregularities related to boundary conditions, a widely used periodic boundary was involved: if cells are enumerated as it given in Fig. 2 (see below), then cell #2 has the following eight neighbors: 1,3, 5,6,7, 13,14,15.

| 12 | 15 | 14 | 13 | 12 | 15 |
|---|---|---|---|---|---|
| 0 | 3 | 2 | 1 | 0 | 3 |
| 4 | 7 | 6 | 5 | 4 | 7 |
| 8 | 11 | 10 | 9 | 8 | 11 |
| 12 | 15 | 14 | 13 | 12 | 15 |
| 0 | 3 | 2 | 1 | 0 | 3 |

**Fig. 1.** Numeration of a 16-cell automaton used in simulations

## 3 Simulation Results

In order to perform preliminary study of model's properties, special software was created. It was used in computational experiments for different values of α parameter and different initial configurations. Test results are presented in the graphs embedding two types of data representation.

The first one is a tree of solutions obtained from step to step. This type of data representation has discrete time (steps) along the X-axis and numbers corresponding to automaton configurations along the Y-axis (automaton grid has 16 cells, each of them may contain 0 or 1, therefore, each automaton configuration is unambiguously defined by a 16-bit number in a range of 0000..FFFF).

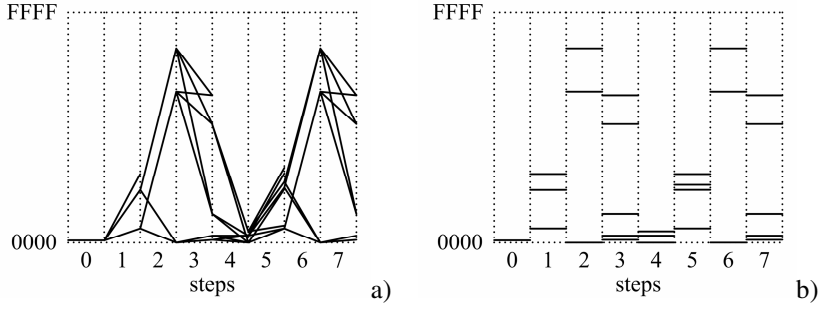



**Fig. 2.** Emergence of several configurations in a 'LifeA' model (additive next state function, α = 1.0, initial state is 0007)

From the figure it can be seen that at the first step 3 solutions (i.e. 3 coexistent configurations) are obtained. At the second step two of them bifurcate while the third appears to deadlock; and so on. This type of data representation has a drawback: the scale along the Y-axis is ranged between 0 and FFFF (decimal 65535), thus making some solutions to merge. Figure 3b represents the same data as 3a but without the generating links between configurations.

Second type of data representation reflects the evolution of number of living cells from step to step. It has, like the previous case, steps along the X-axis and a number of living cells along the Y-axis. A corresponding example graph is given below.

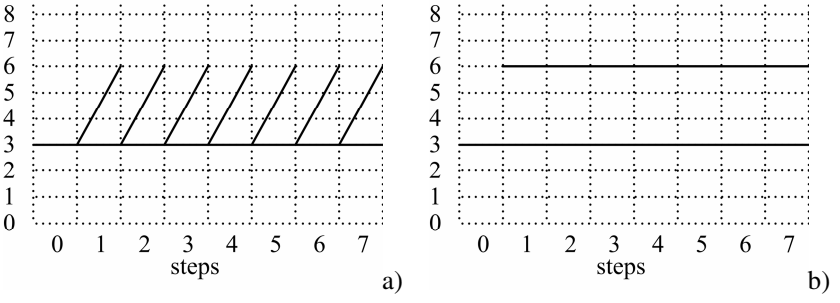



**Fig. 3.** Two coexisting configurations. Input data are the same as for Fig. 3.

Here it can be seen that at every step solutions with 3 and 6 living cells are generated, however, the one with 6 living cells deadlocks. Taking into account the previous graph (Fig. 3) depicting the periodicity of solutions, one may conclude that the number of living cells remains constant with time and only their mutual disposition in the grid is evolving.

The common for all the experiments held was that after several steps the number of solutions reaches its maximum value (depending on input data) and remains constant (see Fig 5). So, a cellular automaton reaches its steady state, but stationarity here proves at a level of sets of solutions that either change periodically (see Fig. 3) or remain constant from step to step.

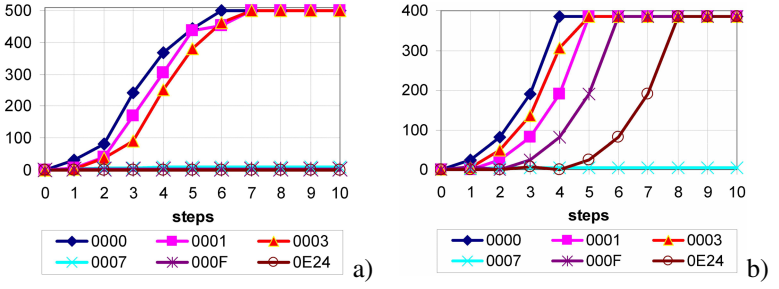



**Fig. 4.** Evolution of number of solutions with time for different initial configurations (additive next state function, a) $\alpha = 0.5$, b) $\alpha = 1.0$ )

*Remark:* data presented in the graphs were obtained using an additive next state function. Weighted function gives similar results, so function choice for a particular problem is defined by a specific application (sometimes it is better to have a bounded parametric set).

## 4 Discussion

The major peculiarity that was discovered during the simulations of game ‘LifeA’ is emergence and performance of state multivaluedness in the system. This new behavior

lets to re-consider already studied problems for game 'Life' (and for cellular automata in general), though, if being studied in-depth, this is the subject for further research and separate reports. In fact, even well-established concepts of the CA theory may be subjected to generalization and reconsideration (especially those related to interpretation of solutions).

Here we would like to outline only few possible prospects. Firstly, certain objects that may be referred to (analogously to the classical case) as "cycles of a definite period", "chaos", "complex behavior", etc. can be marked out. For example, this sole aspect demands broadening of mathematical definitions. On the other hand, this will help to formulate relevant problems in self-organization theory. Moreover, these models involving CA may be useful for a variety of applications (first of all for simulation of drivers and pedestrians behavior [9] and for economics). However, one may suppose the main stream of such a research to deal with possible impacts of these phenomena in biology.

**Acknowledgements.** The authors would like to thank the University of Groningen (SOM Department) for partial financial support of this work.

## References

1. Gardner, M.: The fantastic combinatika of John Conway's new solitaire game 'Life'. Scientific American 223(4), 120–123 (1970)
2. von Neumann, J.: Theory of self-reproducing automata. In: Burks, A. (ed.), Univ. of Illinois, Urbana (1966)
3. Toffoli, T., Margolis, N.: Cellular automata computation. Moscow, Mir (in Russian) (1991)
4. Wolfram, S.: A new kind of science. Wolfram Media Inc., USA (2002)
5. Turing, A.: The chemical basis of morphogenesis. Phil. Trans. Roy. Soc. of London. Ser. B 237, 37–72 (1952)
6. Vollmar, R.: John von Neumann and cellular automata. Invited Lecture at ACRI 2004, Amsterdam (2004), http://www.science.uva.nl/research/scs/events/ACRI2004/movie_VollmarLA.html
7. Makarenko, A.: Anticipating in modeling of large social systems - neuronets with internal structure and multivaluedness. International Journal of Computing Anticipatory Systems 13, 77–92 (2002)
8. Gilbert, N., Troitzsch, K.: Simulation for the social scientist. Open University Press, Surrey (1999)
9. Goldengorin, B., Makarenko, A., Smilianec, N.: Some applications and prospects of cellular automata in traffic problems. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 532–537. Springer, Heidelberg (2006)
10. Makarenko, A., Stashenko, A.: Some two- steps discrete-time anticipatory models with 'boiling' multivaluedness. In: Daniel, M.D. (ed.) AIP Conference Proceedings, USA, vol. 839, pp. 265–272 (2006)

# On the Representation of Gliders in Rule 54 by De Bruijn and Cycle Diagrams

Genaro J. Martínez[1], Andrew Adamatzky[1], and Harold V. McIntosh[2]

[1] Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol, United Kingdom
{genaro.martinez,andrew.adamatzky}@uwe.ac.uk
http://uncomp.uwe.ac.uk/genaro/
http://uncomp.uwe.ac.uk/adamatzky/

[2] Departamento de Aplicación de Microcomputadoras, Instituto de Ciencias, Universidad Autónoma de Puebla, Puebla, México
mcintosh@servidor.unam.mx
http://delta.cs.cinvestav.mx/~mcintosh/

**Abstract.** Rule 54, in Wolfram's notation, is one of elementary yet complexly behaving one-dimensional cellular automata. The automaton supports gliders, glider guns and other non-trivial long transients. We show how to characterize gliders in Rule 54 by diagram representations as de Bruijn and cycle diagrams; offering a way to present each glider in Rule 54 with particular characteristics. This allows a compact encoding of initial conditions which can be used in implementing non-trivial collision-based computing in one-dimensional cellular automata.

## 1 Preliminaries

Amongst one-dimensional cellular automata (CA) studied by Wolfram in [15,16], one can find a few evolution rules supporting gliders (particles or mobile self-localizations); rules 110 and 54 exhibit particularly rich and somewhat complex behaviour.[1] Interaction between gliders can be employed to execute logical operations, and thus ultimately to perform universal computation [1]. Collision-based computing schemes are very sensitive to initial configurations of gliders, even a shift in a glider phase can completely destroy a cascade of logical gates. There is a need for a compact and uniform description of glider types and glider interactions. In the present paper, taking Rule 54 as example, we discuss how such a description can be reached by de Bruijn and cycle diagrams.

Following, Wolfram's notation — a one-dimensional elementary CA has two parameters $(k, r)$, number of states $k$ and cell neighborhood radius $r$ – Rule 54 is a CA with parameters $(2, 1)$, i.e. two cell-states and three cell neighborhood (a central cell, its left and right neighbors). The local transition function $f$ is determinated as follows: $111 \rightarrow 0$, $110 \rightarrow 0$, $101 \rightarrow 1$, $100 \rightarrow 1$, $011 \rightarrow 0$, $010 \rightarrow 1$, $001 \rightarrow 1$ and $000 \rightarrow 0$. The binary sequence 00110110 in decimal notation represents the evolution rule 54.

[1] http://uncomp.uwe.ac.uk/genaro/Rule54.html

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 83–91, 2008.

A detailed study of Rule 54 was initiated in [2], and some basic characteristics of gliders, or particles, were initially derived by Hanson and Crutchfield in [4]. They constructed a subset diagram capable of identifying sequences of gliders in Rule 54. No complete characterization of gliders was provided however. This is a reason why we became interested in representing each glider in Rule 54 by a well-defined way as de Bruijn and cycle diagrams [8,12,14,17].

In a previous paper [5] we demonstrated self-organization in Rule 54 developing a 'genealogical tree' of gliders, where every glider can be derived from collisions between other ones, and also classifying all possible scenarios of binary and ternary collisions between gliders. In the present paper we advance our understanding of glider dynamics by de Bruijn and cycle diagram representations.

Results of the present paper are based on de Bruijn diagrams [8,11,12,13], cycle diagrams [14,9] and computing based on interaction between gliders [1].

## 2 Gliders in Rule 54

A glider is a compact group of non-quiescent states traveling along cellular automata lattice. Rule 54 automaton exhibits relatively a small number of glider types, which makes it particularly attractive for discretization and formal representation. Thus we look tools to characterize these gliders and control collisions in Rule 54 from its initial condition.[2]

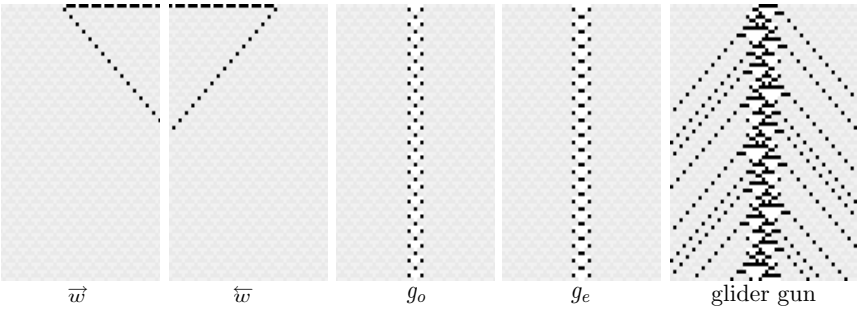



**Fig. 1.** Gliders in Rule 54 (time goes down). Cell-state 1 is shown by black pixels, also a filter is represented for clarity.

To represent gliders in Rule 54 we follow Boccara's *et al.* notation [2]. Thus we can display every glider together with a list of its properties (dynamic, names, speed and periodic width), see examples in Fig. 1 and Tab. 1.

We study automata with periodic boundary conditions, let $e_1$ and $e_2$ represent glider phases in the periodic background. Thus we have four primitive gliders — $w^{\rightarrow}, w^{\leftarrow}, g_o, g_e$ — and a compound glider — the glider gun. Speed $v_g$ of a glider

[2] Single gliders in Rule 54 or packages and extensions of them also can be consulted in http://uncomp.uwe.ac.uk/genaro/rule54/glidersRule54.html

**Table 1.** Properties of gliders in Rule 54

| structure (glider) | $v_g$ | periodic width |
|---|---|---|
| $e_1$ | $2/2 = 1$ | 4 |
| $e_2$ | $2/2 = 1$ | 4 |
| $w^{\rightarrow}$ | $2/2 = 1$ | 2 |
| $w^{\leftarrow}$ | -2/2 = -1 | 0-4 |
| $g_o$ | $0/4 = 0$ | 6-2 |
| $g_e$ | $0/4 = 0$ | 7-3 |
| glider gun | $0/32 = 0$ | 14-4 |

is evaluated using the period between displacements. Therefore $w^{\rightarrow}$ glider moves with positive slope, $w^{\leftarrow}$ glider moves with negative slope and $g_o$, $g_e$ and the glider gun move with velocity zero (as still live configurations in one dimension).

### 2.1 De Bruijn Diagrams Calculating Gliders in Rule 54

For a one-dimensional cellular automaton of order $(k, r)$ and a given finite alphabet $K$, its de Bruijn diagram is defined as a directed graph with $k^{2r}$ vertexes and $k^{2r+1}$ edges. The vertexes are labeled with the elements of the alphabet of length $2r$. An edge is directed from vertex $i$ to vertex $j$, if and only if, the $2r-1$ final symbols of $i$ are the same as $2r-1$ initial symbols in $j$ forming a neighborhood of $2r+1$ states represented by $i \diamond j$. In this case, the edge connecting $i$ to $j$ is labeled by $f(i \diamond j)$ (the value of the neighborhood defined by the local function $f$) [12,13].

Thus de Bruijn diagram of any one-dimensional CA can be constructed as follow:

$$M_{i,j} = \begin{cases} 1 \text{ if } j = ki, ki+1, \ldots, ki+k-1 \pmod{k^{2r}} \\ 0 \text{ in other case} \end{cases} \quad (1)$$

Modulo $k^{2r} = 2^2 = 4$ represents the number of vertexes in the de Bruijn diagram and $j$ takes values from $k * i = 2i$ to $(k * i) + k - 1 = (2 * i) + 2 - 1 = 2i + 1$. The vertexes (indexes of $M$) are labeled by fractions of neighborhoods originated by 00, 01, 10 and 11, the overlap determines each connection. This way Fig. 2 displays Rule 54's matrix evolution and de Bruijn diagram of Rule 54 respectively.

Paths in the de Bruijn diagram may represent chains, configurations or classes of configurations in the evolution space. Also fragments of the diagram itself are useful in discovering periodic blocks of strings, pre-images, codes, and cycles [11,13].

After the de Bruijn diagram is completed, we can calculate an extended de Bruijn diagram [11,7]. An extended de Bruijn diagram takes into account more significant overlapping of neighborhoods. Thus, we represent $M^{(2)}_{R54}$ by indexes $i = j = 2r * n$, where $n \in \mathbb{Z}^+$. Moreover the de Bruijn diagram grows exponentially, order $k^{2r^n}$, for each $M^{(n)}_{R54}$; consequently basic de Bruijn diagram is obtained for $n = 1$.

$$M_{R54} = \begin{bmatrix} 0 & 1 & . & . \\ . & . & 1 & 0 \\ 1 & 1 & . & . \\ . & . & 0 & 0 \end{bmatrix}$$

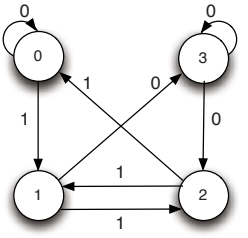



**Fig. 2.** de Bruijn diagram for Rule 54

Let us calculate de Bruijn diagrams for gliders $w^{\rightarrow}$ and $w^{\leftarrow}$ with periodic background. Tab. 1 shows that the gliders translate two cells in a time. Hence the extended de Bruijn diagram of order $M^{(2)}_{R54}$ can be used to extract a cyclic structure of gliders.[3] The constructed diagrams show all possible relations but, we will focus on cycles, or periodic strings generated by local-transition functions.

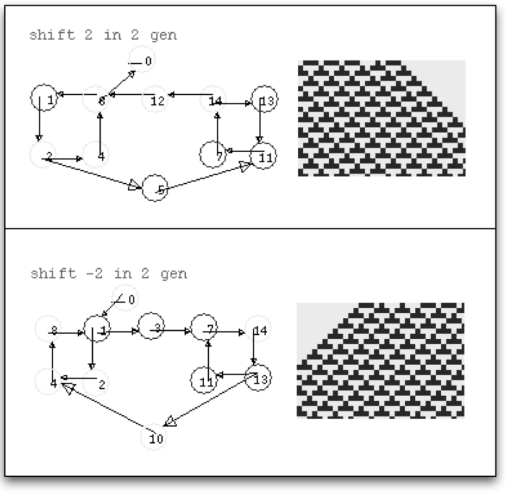



**Fig. 3.** de Bruijn diagrams corresponding to gliders $w^{\rightarrow}$ (up) and $w^{\leftarrow}$ (down)

Fig. 3 demonstrates the de Bruijn diagram when strings are translated two sites to the right (+) or to the left (−). In this case, Rule 54 offers an easy case to extract gliders from their cyclic representation in de Bruijn diagrams. A glider can be identified as a cycle and the glider interactions with regard or their phase changes are reflected in bigger cycles containing two or more cycles. The

[3] Extended de Bruijn diagrams are calculated with NXLCAU21 software developed by Harold McIntosh. Application and source code are available at http://delta.cs.cinvestav.mx/~mcintosh/oldweb/software.html

first diagram displays periodic strings moving two cells to the right in two times, i.e., period and displacement in the periodic background defining the glider $w^{\rightarrow}$. This diagram has a positive orientation of cycles and also shows that relations of vertexes $(1, 2, 4, 6)$ and $(13, 11, 7, 14)$ representing all possible phases where glider $w^{\rightarrow}$ can be placed. However, the existence of this glider is related to both cycles of the diagram. Thus the periodic background in phase one represents the string 1000 and in phase two the string 1110. So, to represent glider $w^{\rightarrow}$ in its two different phases we must make a transition from one cycle to another, or to concatenate the strings $(1000)^*$-00-$(1110)^*$.

Rule 54 has a particular characteristic where gliders can travel between two or three different backgrounds. Fig. 3 show four cycles, three of them self-contained and one cycle starts with the nil state. Fragments of evolutions in the same picture show what types of gliders are more likely to be defined by these cycles. We can see a large cycle represented by the vertexes $(1, 2, 5, 11, 13, 14, 12, 6)$. This cycle is equivalent to the periodic string 10111000, which produces an evolution space covered with just a pair of $w^{\rightarrow}$ gliders. Finally, a fourth cycle, represented by the cycle 0, determines a transition between two different patterns, "fuse configurations." The periodic background is formed by a cycle of length four and the existence of gliders is determined by other cycles. Therefore, we see that the problem of representing gliders by de Bruijn diagrams is reduced to the classification of cycles in the diagrams.

The advantage of using a de Bruijn diagram is that many problems concerning automata are thereby transformed into known problems regarding of the tracing of paths through a graph. For instance, no loop can be longer than the total number of nodes in the graph without repeating some segment; but then there must exist still other loops in which the repeated segment is traversed an arbitrary number of times. For example, a binary automaton depending upon nearest neighbors has eight distinct neighborhoods, representable as eight links connecting four nodes, it follows that no static configuration can be more than four cells long without repeating some two-cell partial neighborhood. Thus the static configurations are rather severely constrained.

Sometimes the de Bruijn diagram reveals information about localized aspects of a configuration. For example if an acceptable path terminates at a node in which all the outgoing links are acceptable, it does not need to continue. Likewise, if all the incoming links are acceptable, the path may begin just as it has been part of a loop. Thus semi infinite structures may be located, or even finite ones if both ends have such universal terminations. This leads to the phenomenon of membranes and macrocells which Wolfram noticed during the course of his investigations. That is, an automaton may have patches which are isolated from one another by static regions, whose evolutions procede quite independently [11].

### 2.2 Cycle Diagrams Calculating Gliders in Rule 54

De Bruijn diagrams have demonstrated their power in representing periodic strings based in gliders in elemental CA [7]. Some other tools could also be used to derive these sequences, e.g. the cycle diagrams (Fig. 4). Of course, we

should remember that de Bruijn diagrams grow exponentially, so our potential for computational classification of the diagrams is fairly limited. Thus cycle diagrams can help at least calculating attractors as was extensively studied by Wuensche in [14,9] and thus precisely represent some selected periodic patterns.

There are two ways to obtain the cycles for a given automaton. The first is to enumerate all the rings of the desired length, and follow up the evolution of each. In doing so task, various shortcuts can be taken, such as generating the configurations in Gray code order so that only a single cell changes state from one to the next. Still lifes can be detected very quickly in this way. Numerical comparison of successive generations means that whenever the new generation is smaller, it has been already examined and there is no need for further exploration.

The second way is more systematic and is worth the bookkeeping effort involved. A graph whose links are determined by evolution is prepared, following which a path enumerating procedure is followed to locate all the loops, whose lengths will give the periods of all the cycles of that length. Cycles of length up to ten can be obtained easily, twenty with effort, but passing thirty requires dedication; for binary automata it is slightly easier, increasingly more difficult for others [11].

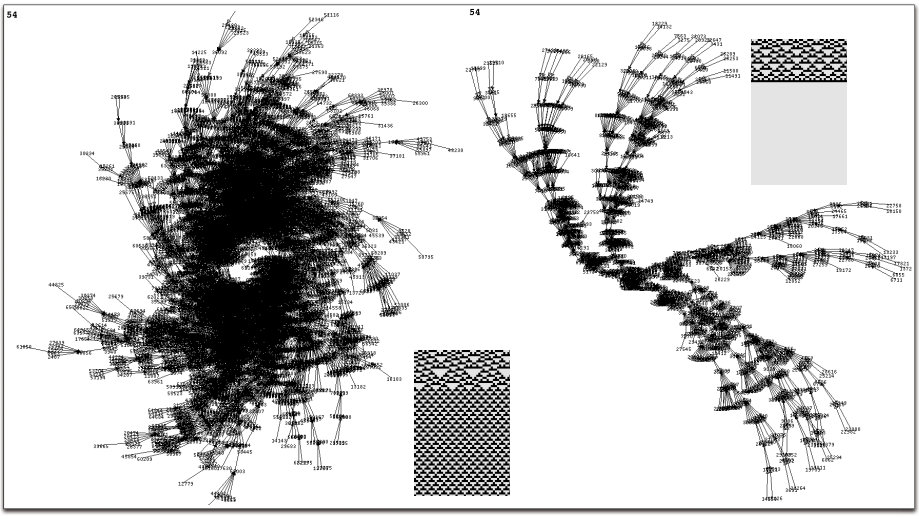



**Fig. 4.** Cycle diagrams calculating their two possible periodic and stable background in Rule 54 from their attractors

Also cycle diagrams represent very well and easily the concept of global configurations and Garden of Eden configurations. The first involucre a global state of a CA in each vertex therefore each global state represents an instantaneous description of the evolution space and also is guaranteed that each global state has an image. The second concept is represented with configurations without ancestors, i.e., leaves as vertexes that have not images from another global configurations, these configurations are called Garden of Eden configurations.

Finally we must mention that the evolution space in Rule 54 can be characterized as well by means of tiles as it was studied in Rule 110 [10,7]. Although in this paper it is not extended the analysis, several of them are classified from their de Bruijn and cycle diagrams.

Thus we can enumerate some important characteristics to determine gliders from their cycle diagrams as follow:

**Table 2.** Cycle diagrams calculating gliders in Rule 54

| cycle length | cycle | total vertexes | structures |
|---|---|---|---|
| 4 | 4 | 4 | $T_3$ and $T_2$ tiles |
| 6 | 4 | 5 | $g_e$ glider |
| 8 | 4 | 14 | $g_e$ gliders joined |
| | 6 | 28 | $g_e$ glider with a $T_2$ |
| 9 | 4 | 44 | $g_e$-$g_o$ gliders joined |
| | 27 | 45 | $T_4$ transporting a $w^{\leftarrow}$ (extensible as a $T_5$ in R110) |
| 10 | 30 | 90 | two $T_4$ tiles joined |
| 11 | 4 | 125 | $g_o$ glider with a $T_6$ tile |
| | 11 | 55 | packages of $T_4$ tiles |
| | 99 | 231 | meta-glider with $w^{\rightarrow}$-$T_5$-$T_6$-$T_4$-$T_2$ tiles |
| 12 | 10 | 124 | periodic background composed of $2T_6$-$2T_3$-$T_2$ tiles |
| | 12 | 102 | $2w^{\rightarrow}$ gliders |
| 13 | 4 | 406 | $(g_e$-$g_o)^*$ gliders concatenated |
| | 169 | 1274 | meta-glider with $T_8$-$2T_4$-$T_2$ and $w^{\rightarrow}$ gliders |
| 14 | 112 | 805 | meta-glider with $T_8$-$3T_4$-$T_2$ tiles |
| 15 | 330 | 7680 | meta-glider with $T_5$-$2T_6$-$T_4$-$T_2$ tiles |
| 16 | 6 | 116 | periodic background composed of $T_6$-$T_2$ tiles |
| | 8 | 8 | $w^{\rightarrow}$ gliders |
| | 14 | 944 | meta-glider with $w^{\rightarrow}$-$g_o$-$w^{\leftarrow}$ gliders |
| | 16 | 2896 | $2w^{\rightarrow}$ gliders |
| | 40 | 1246 | meta-glider with $T_8$-$5T_6$-$2T_2$-$3T_4$-$T_5$ tiles |

In this way we can list a number of periodic strings derived from the de Bruijn and cycle diagrams. In this case, every primitive glider (see Fig. 1) can be reproduced from different cycles as shows Tab. 2. The concept of meta-glider refers to a synchronization of several gliders preserving their form, reactions and structure periodically as was also reported in Rule 110 in [6].

## 3 Coding Rule 54

Quickly we must show how to code Rule 54 to solve some problems reported in [5]. We can select a production by gliders or sequences and code its respective initial condition.[4] If we select a production by gliders, a number of them are

[4] See demos of encoding collision between single gliders and their trains in `http://uncomp.uwe.ac.uk/genaro/rule54/collisionsRule54.html`

important, e.g. if we want to produce a $w^{\rightarrow}$ glider, we need to collide a $g_o$ with a $w^{\leftarrow}$ glider. The encoding can be described as follows:

(a) $w^{\rightarrow} = e_1^+$-$g_o$(A,f$_1$)-$e_1^+$-$w^{\leftarrow}$(f$_2$)-$e_2^+$
(b) $w^{\rightarrow} = e_1^+$-$g_o$(B,f$_1$)-$e_1^+$-$w^{\leftarrow}$(f$_2$)-$e_2^+$
(c) glider gun $= e_1^+$-$g_e$(A,f$_1$)-$g_e$(B,f$_1$)-4$e_1$-$w^{\leftarrow}$(f$_1$)-$e_2^+$
(d) double glider gun $= e_1^+$-2$w^{\rightarrow}$(f$_1$)-8$e_1$-2$g_e$(A,f$_1$)-2$e_1$-2$g_e$(A,f$_1$)-8$e_1$-2$w^{\leftarrow}$(f$_1$)-$e_1^+$ (see fig. 5).

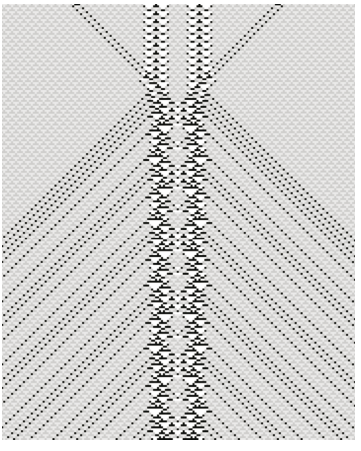


**Fig. 5.** Double glider gun in Rule 54 produced from multiple collisions between gliders

## 4 Conclusions

The rows of the period-cycle table can be found from de Bruijn diagrams in the same way that the cycles can be found from the evolution diagram; since $2r + 1$ cells are needed to deduce a generation of evolution, only about half as many periods as cycles can be worked out for a given amount of effort. This anomaly is really an artifact of the way $r$ parameterizes the neighborhood, and would disappear if half-integral increments were taken for $r$.

Similar theoretical conclusions are possible, since the periods are taken from a subset of the de Bruijn diagram. A $2r$-stage de Bruijn diagram for $k$ symbols has $k^{2r}$ nodes; $k$ times as many links. Once this number of links has been used up in constructing a path through the diagram, one of them would have to be repeated. Thus there is also an exponential upper bound in the rows of the period-cycle table. For example, if an automaton has a cycle of period 2, it must already show up in some short ring; if it has not appeared in rings below a certain limit, it will never appear in longer rings [11].

Finally a way to code Rule 54 is needed to obtain a complete description of gliders. With this system we would be able to construct arbitrary initial conditions in Rule 54 to control simple o complicated reactions with dozen or

hundred of gliders. Potentially this feature will provide a powerful tool to develop spatiotemporal solutions in Rule 54 such as: Rule 54 objects, solitons, eaters and so on. Thus a complete regular glider-based language in Rule 54 with de Bruijn diagrams will be presented in our next paper.

## Acknowledgement

Our research was partly supported by Engineering and Physical Sciences Research Council (EPSRC), United Kingdom, grant EP/F054343.

## References

1. Adamatzky, A. (ed.): Collision-Based Computing. Springer, Heidelberg (2003)
2. Boccara, N., Nasser, J., Roger, M.: Particle like structures and their interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular automaton rules. Physical Review A 44(2), 866–875 (1991)
3. Golomb, S.W.: Shift Register Sequences. Holden-Day, San Francisco (1967)
4. Hanson, J.E., Crutchfield, J.P.: Computacional Mechanics of Cellular Automata: An Example. Physics D 103(1-4), 169–189 (1997)
5. Martínez, G.J., Adamatzky, A., McIntosh, H.V.: Phenomenology of glider collisions in cellular automaton Rule 54 and associated logical gates. Chaos, Solitons and Fractals 28, 100–111 (2006)
6. Martínez, G.J., McIntosh, H.V., Seck Tuoh Mora, J.C., Chapa Vergara, S.V.: Rule 110 objects and other collision-based constructions. J. Cellular Automata 2(3), 219–242 (2007)
7. Martínez, G.J., McIntosh, H.V., Seck Touh Mora, J.C., Chapa Vergara, S.V.: Determining a regular language by glider-based structures called phases $f_i\_1$ in Rule 110. J. Cellular Automata (in press)
8. McIntosh, H.V.: Linear cellular automata via de Bruijn diagrams (1991), http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html
9. McIntosh, H.V.: Commentaries on: The Global Dynamics of Cellular Automata (by Wuensche, A., Lesser, M.:) (1993), http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html
10. McIntosh, H.V.: Rule 110 as it relates to the presence of gliders (1999), http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html
11. McIntosh, H.V.: One Dimensional Cellular Automata (by publish)
12. Voorhees, B.H.: Computational analysis of one-dimensional cellular automata. World Scientific Series on Nonlinear Science, Series A, vol. 15 (1996)
13. Voorhees, B.H.: Remarks on Applications of De Bruijn Diagrams and Their Fragments. Journal of Cellular Automata (in press)
14. Wuensche, A., Lesser, M.: The Global Dynamics of Cellular Automata. Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley Publishing Company, Reading (1992)
15. Wolfram, S.: Theory and Aplications of Cellular Automata. World Scientific Press, Singapore (1986)
16. Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign (2002)
17. Wuensche, A.: Classifying Cellular Automata Automatically. Complexity 4(3), 47–66 (1999)

# A Cellular Automaton Model for Tribological Problems

Michael Müller and Georg-Peter Ostermeyer

Braunschweig University of Technology (Germany),
Institute of Dynamics and Vibrations

**Abstract.** Many tribological systems are characterized by an interface dynamics determined by the growth and destruction of hard thin and smooth structures, so called 'patches'. These patches transmit the main part of the friction power and protect the softer pad ingredients from wear. This behavior, which is explained with the example of a brake system, can be interpreted as a kind of self-organization process. Hereby rather simple local balance equations, considering the aspects of temperature, particle flow, patch dynamics and transmitted friction power, result in a complex global tribological character. In this paper it is shown that the Cellular Automaton is a very helpful method to describe the above standing items and their correlation. In order to avoid certain numerical instabilities, the respective boundary conditions are discussed.

**Keywords:** Tribology, Brake Systems, Self-Organization, Friction, Wear, Boundary Conditions.

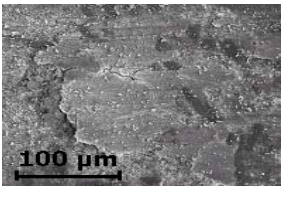
## 1 Introduction

The basic problem of modelling tribological systems with respect to a prediction or reproduction of measured friction and wear phenomena is the huge lengthscale gap. This is due to the fact that, on one hand side friction is caused by an interaction between atoms, and on the other hand side one tries to gain some macroscopic properties of bodies in contact. There exist numerous Molecular Dynamics Simulations, for example [1], because of hardware reasons (memory and processing) their informational value is still limited to some micrometers and nanoseconds. In principle, they are helpful to understand the formation of the so called 'third body' - a very thin layer between two asperities - but conclusions towards a macroscopic behavior are not really possible. Macroscopic models, for instance FE-models, e.g. [2], usually contain macroscopic input data, such as characteristic diagrams for the friction coefficient, but they often neglect the effect of wear and plastic deformations. These all are the reasons why neither friction coefficients nor wear rates can be predicted to date.

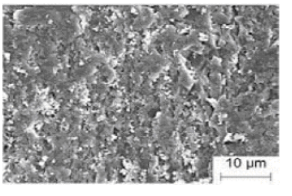
One can observe a specific effect for a certain group of tribological systems which are determined by a process on a mesoscopic (some hundred microns up to some millimeters) lengthscale. This certain group follows a kind of self-organization in terms of characteristic surface patterns arising and degrading

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 92–99, 2008.

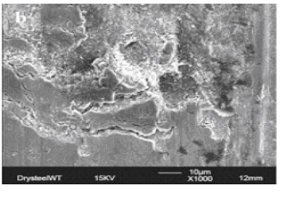
during the friction process. Fig. 1 shows some representatives of this group. These are dry-friction systems such as brakes (a) [3], clutches (b) [4], railroad (c) [5], gear box (d) [6], bearing cage (e) [7], chalk on blackboard (f). There exist many more, even lubricated systems, the patterns shown here originate solely from the Proceedings of the World Tribology Congress III (2005). Although these systems are very different at the first sight, they have a special thing in common: their respective surface patterns cause a significant reduction of wear. Since the gap between this mesoscopic and the macrosopic world could be closed in the future, this approach can be a promising way to get a better understanding for these tribological systems. The self-organizing character of these systems is to be found in the fact, that, during the tribological process, an original stochastic or even chaotic boundary layer turns into a layer with regular, stable structures. For this tribo-group it seems that this behavior is given by nature and causes a high resistance of the system against wear.



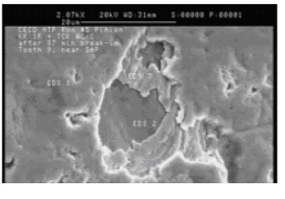

(a) brake pad



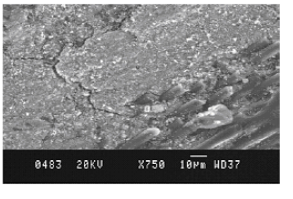

(b) clutch



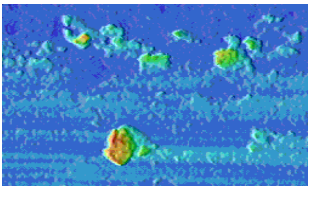
(c) rail



(d) gear wheel




(e) bearing cage



(f) chalk onblackboard

**Fig. 1.** Microscopy pictures of different tribological systems

Basically, the ideal tool to describe systems determined by self organization is a Cellular Automaton model. The way how a model, considering the growth and destruction of the above mentioned structures (so called 'patches') and the respective interdependencies with respect to heat and wear, can be derived is illustrated in the following sections. First of all the fundamental process of formation and degradation of patches is explained by means of the interface of brake systems, since for this tribological system the process is well investigated by measurements.

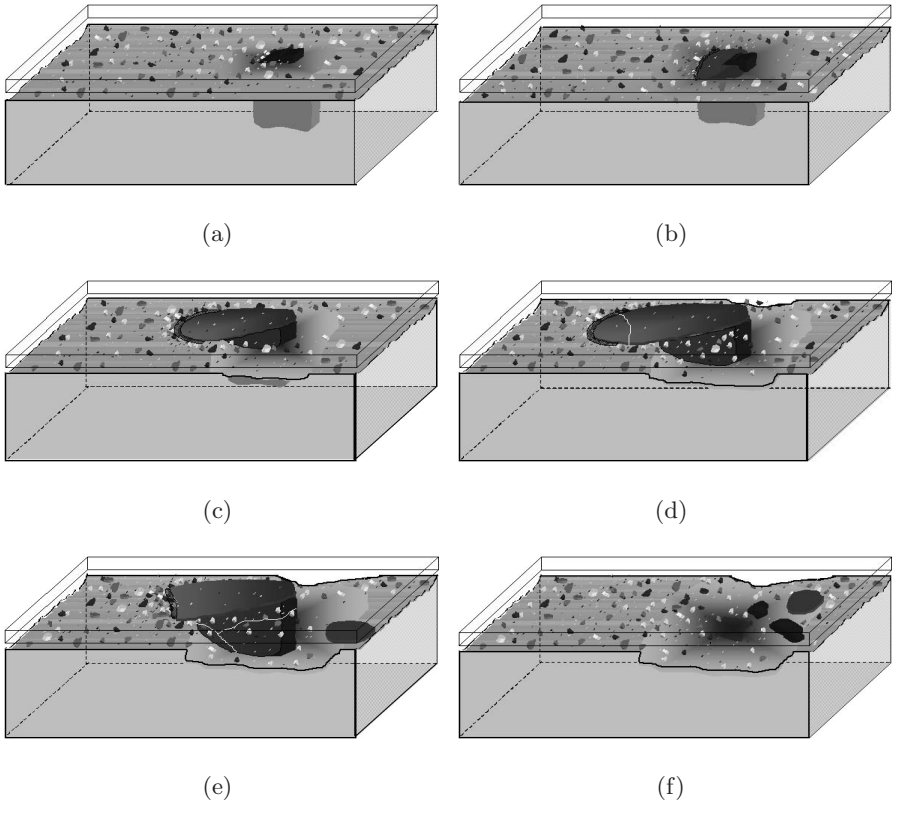



**Fig. 2.** Patch-growth and patch-destruction

## 2 Friction and Wear in the Interface of Brake Systems

In principle, the composition of a brake lining consists of hard particles, for example metallic fibres, aramide fibres, quartz particles and further fillers which are embedded into a rather soft polymer matrix of phenolic resin. If there is contact between the grey cast iron disk and the polymer matrix, wear particles are detached from the matrix material. The situation changes fundamentally when a hard pad ingredient reaches the surface (see Fig. 2a). This hard inhomogeniety (in this case a quartz) wears much less than the soft resin so that the local normal load increases and it is pressed into the matrix [3], [8]. Since a higher local normal force is correlated with a higher rate of energy dissipation, the temperature rises there rapidly. It can reach values of more than 1000 °C. On the other hand the wear particle stream is disturbed in so far that wear particles are either deflected at the quartz or agglomerate in front of it (Fig. 2b). The last mentioned debris is compacted and as a consequence of the high temperatures, a hard and thin oxide layer develops on top of the compacted debris (Fig. 2c). The

oxide layer's hardness is in the range of the inhomogeniety's hardness. Therefore it protects the softer matrix material below from wear. It can be concluded that, originated by the hard particle a hard structure, a patch, grows perpendicular and reverse to the particle flow direction. The patch transmits a major part of the friction power because the load concentrates on it. Due to the high mechanical and thermal load and its size, at a certain point the patch reaches his limit of stability and cracks arise (Fig. 2d). Consequently parts of the patch will break and be transported out of the system (Fig. 2e), or this part agglomerates in front of another patch. Caused by the further wear of the matrix material the quartz will lose its foothold and will then be detached as well (Fig. 2f).

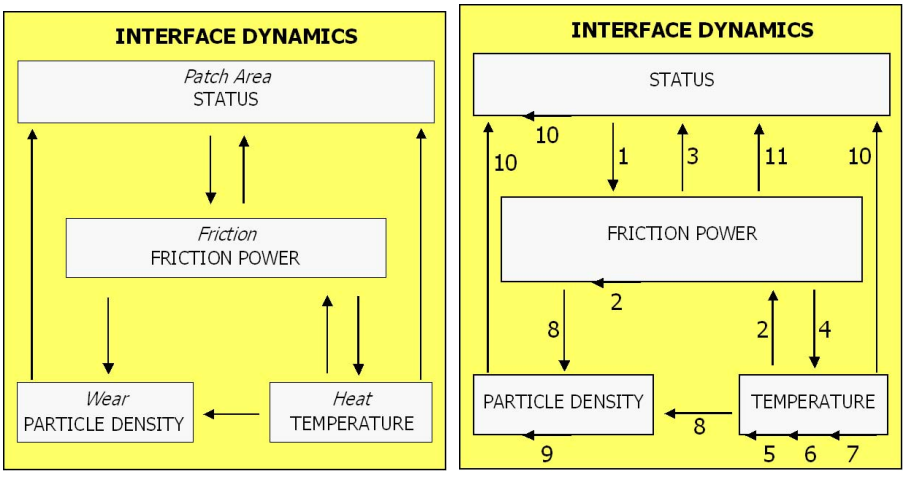
These processes can be interpreted as an equilibrium of flow of patches growing and becoming destroyed and thereby determining the global friction and wear behavior of the system. That is the reason why the authors believe that, models taking into account the dynamics of this mesoscopic scale (for brake systems some hundred microns), have the potential to gain a fundamental understanding towards these tribological systems. A respective Cellular Automaton model is described next.

## 3 A Cellular Automaton Model to Describe the Interface Dynamics

As shown in the previous section, there exist several interactions between the surface topography in terms of the patch area and the aspects of friction, wear, and heat.These interactions can be depicted in the diagram illustrated in Fig. 3a. In order to achieve a detailed description of the interface processes, the set of inner variables has to regard those items and interdependencies. Therefore, the Cellular Automaton model contains a set of 4 inner variables for each cell:

- *status*: The *status* of a cell can be 'patch' or 'not patch' depending on the question, whether the cell belongs to a patch or not. The patches in brake systems have a size of some hundred microns. Thus the cell size is defined 10 microns x 10 microns and for representativeness reasons the total grid size is chosen 150 x 150 cells.
- *friction power*: The *friction power* of a cell describes the product of friction force and relative velocity on the cell. This value can firstly be discretionary, so strictly seen this term injures the definition of Cellular Automata which premises a finite set of inner variables. As it will be shown later, this claim can be evaded with respect to the model used here.
- *temperature*: For each cell a *temperature* is computed. As for the *friction power*, the *temperature* is not limited by a finite number of states.
- *particle density*: The wear volume, which is currently on the cell, is measured with a *particle density*. Also this value can theoretically be any real number.

The set of transition functions is shown in Fig. 3b. Since they are extensively explained in [9] and [10], the 11 rules are described only briefly in the following:




(a) Interactions (b) Set of transition functions

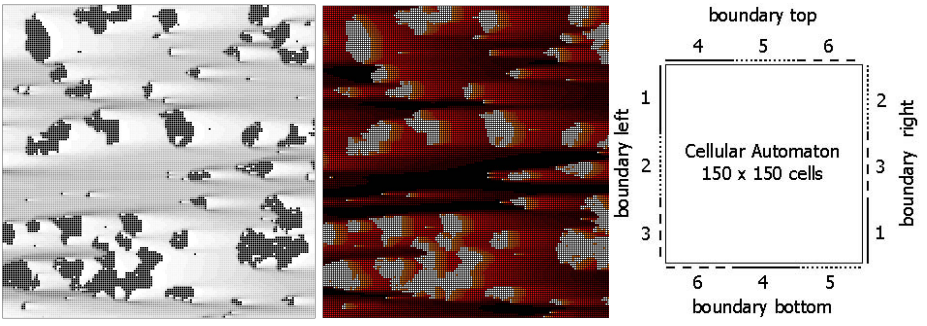
**Fig. 3.** (a) Interactions between patch area, load, wear and heat, (b) resulting set of transition functions

1. *Friction power distribution*: The friction power on a cell equals the product of local normal load (computed with an approach based on Boussinesq's formula for elastic halfspaces), local relative velocity (which depends on the distance between the rotation centre and the cell position) and local friction coefficient.
2. *Global coefficient of friction*: The friction coefficient is correlated with the patch area, the temperatures and the normal loads, since these 3 criteria determine the third body character and therefore the friction coefficient.
3. *'Patch-birth'*: Depending on the brake pad composition, the friction power and the time increment, a probability for an inhomogeneity reaching the surface can be computed, and with this probability the belonging cell spontaneously changes its status from 'not patch' to 'patch'.
4. *Heat generation*: It is assumed that 95 percent of the energy, dissipated on the cell during the time increment, is transformed into heat energy. With the knowledge of the material's specific heat capacity and the density, this energy portion can be used to calculate the cell's temperature increase.
5. *Heat conduction*: The basis for the heat conduction algorithm is the respective partial differential equation. Neighbored cells exchange energies, whereby the belonging portion depends on the temperature difference between the cells before the step, the conductivity of the material and the time increment. It could have been shown that this approach always satisfies the first law of thermodynamics (energy conservation) and for short enough time increments also the second law of thermodynamics (entropy increase). It has been proved that this procedure results in temperatures close to the analytical solution.

6. *Heat transport by the disk's rotation*: In connection with the heat generation and the heat conduction this rule describes the moving of the heat source. So, there is computed a temperature transfer from a cell to its neighbor into the moving direction of the heat source.
7. *Mean disk temperature*: The heat flux into the third dimension can be treated by regarding a mean disk temperature. This term accumulates this energy and vice versa influences the temperature of the discretized interface layer. If the mean disk temperature (represents the deeper disk layers) is high, the gradient towards the surface and therefore also the heat flux in the third dimension decreases.
8. *Wear volume generation*: The implemented generated wear volume on a cell is proportional to the friction power transmitted by the cell. Additionally the high impact of the cell temperature towards the wear generation is taken into account.
9. *Particle stream*: The wear particles are transported by the disk - so their velocity is known. Thus there exists a time increment which stands for their moving by one cell-width. Assuming that the disk moves from left to right, this means that the wear volume of a cell distributes in the cell itself and its 3 right neighbors. In order to regard the disturbance of the wear particle stream by the patches, the belonging portions depend on the statuses of the respective cells, so for each of these 16 possible combinations a distribution key is implemented.
10. *'Patch-growth'*: When a cell has the status 'not patch' and if the cell has at least one neighbor with the status 'patch', the cell changes its status to 'patch' with a certain probability. This probability is a function of the cell's wear particle density , the cell's temperature, the normal load on the cell, the number of neighbors with the status 'patch' and material-characterizing parameters.
11. *'Patch-destruction'*: A set of connected cells with the status 'patch' is stored as a collective. This collective (all affected cells) changes the status to 'not patch' with a certain probability. This probability is a function of patch-size, patch-age, normal load on the patch, temperature on the patch and a parameter regarding the material.

It is obvious that the philosophy of Cellular Automata fits very good with all items concerning the status-changes, the wear particles (because both issues are determined by neighborhood-interactions) and the global values (friction coefficient, disk temperature, because these are integral values). The heat flux and the normal load distribution are usually computed with continuum-approaches. The Cellular Automaton model is used here as a kind of explicit finite difference method. The convergence of the method is linked with the spatial discretization, the occuring maximal gradients and the chosen time increments. The time-increments used in the existing simulations are 2 magnitudes of order lower than convergence-critical time-increments for the maximum gradients in the system. So, numerical instabilities are not a problem in this context and thus the simulation always generated stable and consistent results. A strong proof of that will

be given elsewhere. Fig. 4 illustrates how the computed fields of wear particle density (left) and temperature (middle) look like, also see [11]. Hereby the pad stands still while the disk moves from left to right.




(a) wear particle density, black = status 'patch'

(b) temperature, white= status 'patch'

(c) boundary conditions

**Fig. 4.** Computed fields of (a) wear particles (dark=high density), (b) temperature (bright=high temperature), (c) specific boundary conditions

In Fig. 4a one can clearly observe that the wear particle density has its maximum values in front of (left) and adjacent (above and below) to the patches, whereas there are hardly recognizable densities behind (right) the patches. This behavior is in accordance with the expectations concerning the stream disturbance. The patches transmit the major part of the friction power, thus the temperature (see Fig. 4b )is maximal on and near patches. Since the disk transports the heat, generated on the patches, there arise 'temperature-tails' behind the patches.

With the help of Fig. 4 there can be pointed out one more very important issue, the boundary conditions. The section of the brake pad area simulated with the Cellular Automaton is 1.5 mm x 1.5 mm. It should be a representative for a much bigger surface area (the total brake pad area amounts roughly 40 $cm^2$), so periodic boundary conditions have to be chosen [12]. In order to avoid numerical instabilities, a specific one with an offset of a third of the lattice size is defined. Fig. 4c clarifies this point. So, for instance the boundary at the left middle part equals the right top boundary (boundary 2), and the top left boundary equals the bottom middle boundary (boundary 4).

This procedure is due to the following reason. A patch is a heat source. The heat generated on the patch is transported by the disk to the right boundary. Using classical periodic boundary conditions the heat would enter the Automaton on the left opposite and finally reaches the patch where it was produced. The temperature again steers the growth of the patch and thereby increases the heat generation and so on. Similar effects are to be found for the wear particle density. This kind of excitation is numerically caused and has to be avoided. With the

one-third-offset this kind of instability can only be achieved when there exists a special configuration with pattern-regularaties with a periodicity of one third of the lattice size. These regularities are very improbable so that the usage of these specific boundary conditions minimize those effects.

## 4 Summary, Conclusions and Outlook

It has been shown how the method of Cellular Automata can be applied for a special group of tribological systems, namely those which are determined by the growth and destruction of characteristic hard structures. The set of transition functions connects the issues of patch-size, heat, wear and load. In order to guarantee that the lattice represents a characteristic section of the entire surface, periodic boundary conditions have to be formulated. For this kind of problem numerical instabilities can occur using classical boundary conditions, so specific periodic boundary conditions are used here. Until now the focus was on the investigations of brake systems. Future works will focus on the overall character of the mentioned group of tribological systems. The authors believe that the process of self-organization with respect to the result that wear rates are reduced is a fundamental key for the comprehension of the nature of tribology.

## References

1. Fu, X.-Y., Falk, M.L., Rigney, D.A.: Sliding Behaviour of metallic glass Part II. Computer simulations. WEAR 250, 420–430 (2001)
2. Chung, C., Donley, M.: Mode coupling phenomenon of brake squeal dynamics. SAE-paper 2003-01-1624 (2003)
3. Ostermeyer, G.-P.: On the dynamics of the friction coefficient. WEAR 254, 852–858 (2003)
4. Zum Gahr, K.H., Litzow, U., Poser, K.: Tribological behaviour of advanced ceramics. In: Proceedings of World Tribology Congress III, Washington (2005)
5. Asthana, P., Liang, H.: Effects of contact modes on wear of railroad. In: Proceedings of World Tribology Congress III, Washington (2005)
6. Hua, D.Y., Zhang, J.: Degradation of W-DLC Coatings in rolling / sliding applications. In: Proceedings of World Tribology Congress III, Washington (2005)
7. Surappa, M.K., Shivakumar, K.N.: Sliding wear and friction properties of stitched RTM base carbon-carbon composites. In: Proceedings of World Tribology Congress III, Washington (2005)
8. Ostermeyer, G.-P., Müller, M., Abendroth, H., Wernitz, B.: Surface topography and wear dynamics of brake pads. SAE Paper 2006-01-3202 (2006)
9. Ostermeyer, G.-P., Müller, M.: New Insights into the tribology of brake systems. Journal of Automobile Engineering, Special Issue on Automotive brakes (to appear, 2008)
10. Müller, M.: Zur Topographie- und Grenzschichtdynamik in Bremssystemen. Shaker, Aachen (2007)
11. Müller, M., Ostermeyer, G.-P.: A Cellular Automaton Model to describe the three-dimensional friction and wear mechanism of brake systems. WEAR 263, 1175–1188 (2007)
12. Weimar, J.: Simulation with Cellular Automata. Logos, Berlin (1997)

# Stabilizing and Destabilizing Effects of Embedding 3-Node Subgraphs on the State Space of Boolean Networks

Chikoo Oosawa[1,*], Michael A. Savageau[2], Abdul S. Jarrah[3], Reinhard C. Laubenbacher[3], and Eduardo D. Sontag[4]

[1] Department of Bioscience and Bioinformatics, Kyushu Institute of Technology, Fukuoka, Japan
[2] Department of Biomedical Engineering, University of California, Davis, California, USA
[3] Virginia Bioinformatics Institute, Department of Mathematics, Virginia Polytechnic Institute and State University, Virginia, USA
[4] Department of Mathematics, Rutgers, The State University of New Jersey, New Jersey, USA

**Abstract.** We demonstrate the effects of embedding subgraphs in a Boolean network, which is one of the discrete dynamic models for transcriptional regulatory networks. After comparing the dynamic properties of networks embedded with seven different subgraphs including feedback and feedforward subgraphs, we found that complexity of the state space increases with longer lengths of attractors, and the number of attractors is reduced for networks with more feedforward subgraphs. In addition, feedforward subgraphs can provide higher mutual information with lower entropy in a temporal program of gene expression. Networks with the other six subgraphs show opposite effects on network dynamics. This is roughly consistent with Thomas's conjecture. These results suggest that feedforward subgraph is favorable local structure in complex biological networks.

**Keywords:** Boolean networks; subgraph; feedback; feedforward; mutual information; entropy; transcriptional regulatory networks; Thomas's conjecture.

## 1 Introduction

Complex networks of interacting elements arising in biological, sociological, and physical areas can often be abstracted to graphs or networks. Recent studies of networks [1,2], including transcriptional regulatory networks in cells, have revealed at least two statistical properties: power-law connectivity distributions having a small number of highly connected nodes; highly clustered connections among adjacent nodes [3,4]. The last local structures, called subgraphs or motifs, consist of a few nodes and edges among the nodes that are statistically significant, and can be regarded as functional modules [4]. Since feedback and

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 100–107, 2008.

feedforward subgraphs are basic and ubiquitous circuits in man-made systems, one can expect that transcriptional regulatory networks also have both feedback and feedforward subgraphs. However, only the feedforward subgraphs prevail [4]. Other biological networks such as signal transduction and neuronal networks have similar tendencies. This suggests that feedforward subgraphs are favored in complex biological networks. In general, although the accumulated data of complex networks underlies the statistical significance, it is unclear why feedforward subgraphs are advantageous over other subgraphs in biological systems. We therefore constructed Boolean networks and embedded subgraphs in them to investigate their effects on both network structure and dynamics.

## 2 Model and Method

### 2.1 Boolean Network

The dynamics of the Boolean networks [5,6] is determined by the equation

$$X_i(t+1) = B_i\left[\mathbf{X}(t)\right] \quad (i = 1, 2, ..., N), \tag{1}$$

where $X_i(t)$ is a binary state, either 0 or 1, of node $i$ at time $t$, $B_i(\cdot)$ are Boolean functions [see Table 1] used to update the state of node $i$, and $\mathbf{X}(t)$ is a binary vector that gives the states of the $N$ nodes in the network. After assigning the initial states $\mathbf{X}(0)$ to the nodes, their successive states are updated by input states coming from upstream nodes and their Boolean functions. The dynamic behavior of these networks is represented by a time series of binary states. The time course follows a transient phase from its initial state until it establishes a periodic pattern, called an attractor [Fig. 2].

**Table 1.** 16 Boolean functions with indegree $K_{in} = 2$. In this paper, we used only No. 1, 2, 4, and 8 Boolean functions shown below, because of biological bias for the Boolean functions [7,8,9,10] and the feasibility of computation.

| Inputs | | Output | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Type No. | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

### 2.2 Numerical Condition

To investigate the effects of embedding subgraphs on the Boolean network dynamics, we randomly constructed many networks with varying numbers of independent subgraphs. Seven subgraphs consisted of three nodes and more than three directed edges [Fig. 1 and Table 2]. After embedding the specified number of subgraphs, the rest of the directed edges were assigned at random.

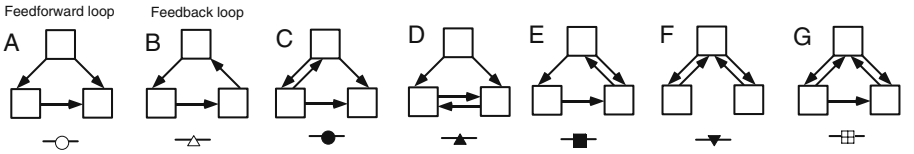



**Fig. 1.** Seven different subgraphs (local structures), The subgraphs consist of 3 nodes (squares) and more than 3 directed edges (arrows). Subgraphs A and B have three internal edges. Subgraphs C – F have four and G has five internal edges. The number of cyclic loops increases with the number of internal edges. The seven subgraphs can be divided into two groups; subgraphs C, D, E, and G comprise subgraph A, feedforward loop; subgraphs B and F do not.

**Table 2.** Numerical condition: All constructed networks consist of the same amount of network resources, i.e., nodes, directed edges, and Boolean functions. Please note that the difference among the generated networks lies in the style of their connections.

| | |
|---|---|
| Size of networks, $N$ | 128 nodes |
| Connectivity | For all nodes, $K_{in} = K_{out} = 2$ |
| Boolean function | Only AND style [Table 1] |
| Types of subgraph | Seven different subgraphs [Fig. 1] |
| Number of embedded subgraphs | 0, 10, 20, 30, and 40 |
| Number of edges | 256 |
| Number of initial states | 2000 per network |
| Number of realizations | More than $3 \times 10^3$ in each condition |

### 2.3 Path Length

To obtain the structural changes of the propagating pathway of the state variables after embedding subgraphs, we measured the average path length [11], which is the average number of the path lengths for all the nodes [Fig. 3b], where the path length is the average number of directed edges in the shortest path from a node to all reachable nodes.

### 2.4 Complexity of State Space

We use two measures to characterize the complexity of state space from the initial states [See Fig. 2]:

1. Basin entropy [11,12]:

$$H_{Basin} = -\sum_i p(i) \log_2 p(i) \tag{2}$$

where, the $p(i)$ satisfies $\sum_i p(i) = 1$, $p(i) = \frac{a_i}{2000}$, and $a_i$ is the number of initial states that reached the i-th attractor [Fig. 2].

2. Sum of the length of attractors: Each network may contain a different number of attractors, and their lengths may vary. This measure defines the total length of attractors in state space.

The two measures indicate the complexity of state space from the initial states. According to the definitions, the larger values of two characteristics sgnify a higher complexity of the state space [Figs. 2 and 4].

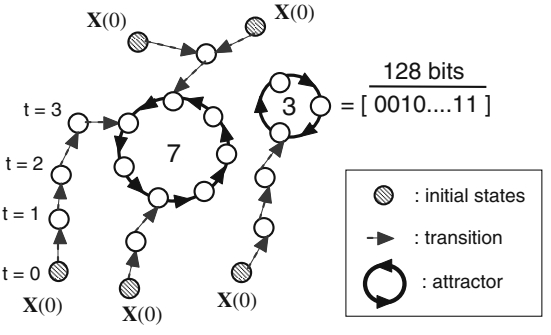



**Fig. 2.** Example of a state space, There are $2^{128}(\sim 10^{39})$ different states (shown as circles) in the space of each network. The numbers 3 and 7, inside the attractors, indicate the attractor lengths. We applied 2000 different initial states (shaded circles) to each network [Table 2]. t = 0, 1,...., indicate time steps of Eq. (1).

### 2.5 Entropy and Mutual Information

We measured the entropy (randomness) and mutual information (correlation) of state variables to characterize the temporal structure of state variables in the Boolean networks [6,13,14]. Both dynamic properties are obtained from attractors [Figs. 2 and 5].

## 3 Results

Since Fig. 3a shows that our method was successful in embedding subgraphs in Boolean networks, the horizontal axis of Figs. 4, 5a, and 5b are the specified number of embedded subgraphs. Because the number of inter-subgraph edges decrease as the number of internal edges increase, the average path length prolongs as the number of embedded subgraphs increases [Fig. 1 and 3b].

We examined the effects of embedding subgraphs on the graphic and dynamic properties. Figure 4 shows the complexity of state space structures. Feedforward has opposite effects from the other six subgraphs. Note that the number of attractors, basin entropy, and the sum of the lengths of attractors decrease slightly with feedforward subgraphs. In general, the complexity of state space increases as the number of internal edges increase [See Fig. 1]. The temporal structure of

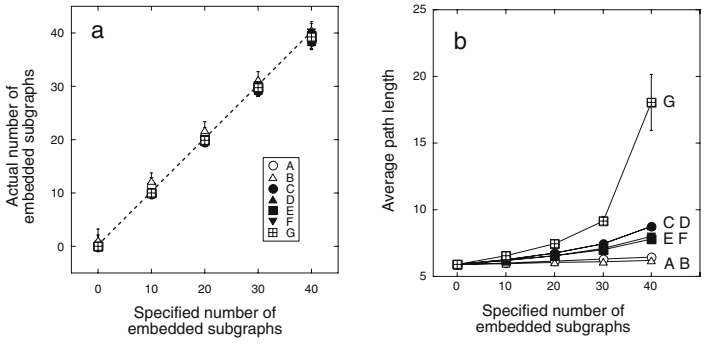



**Fig. 3.** Structural properties of embedded networks. a: Relationship between the specified number of embedded subgraphs and the actual number of embedded subgraphs. Dashed line is given by $y = x$. b: Relationship between the number of embedded subgraphs and average path length. Symbols indicate mean and error bars show SD. Different capital alphabets indicate different subgraphs [See Fig. 1].

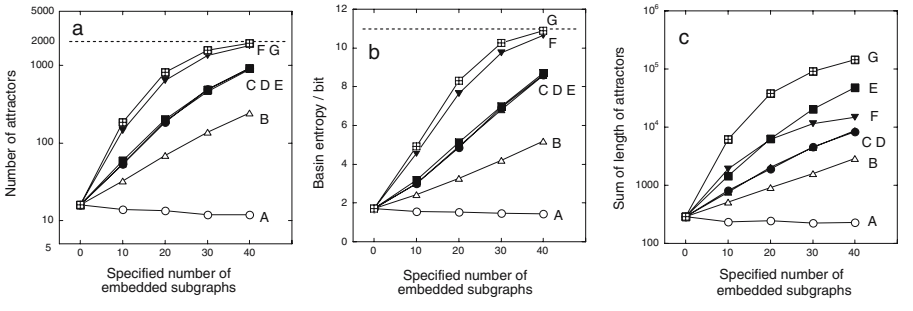



**Fig. 4.** Complexity of state space structure. The relationship between the number of embedded subgraphs and the number of attractors (a), basin entropy (b) [See Eq.(2)], and sum of the length of attractors (c). The symbols indicate mean values. Different capital alphabets indicate different subgraphs [Fig. 1]. Dashed lines indicate the maximum value based on our condition [Table 2]. The curves approaching the dashed lines indicate underestimates of the numbers of attractors.

state variables is shown in Fig. 5. The entropies in Fig. 5a exhibit similar tendencies to those shown in Fig. 4. Unlike Figs. 4 and 5a, the amount of mutual information increases with an increase in the number of embedded subgraphs [Fig. 5b]. Figure 5c shows rearranged results from Fig. 5a and 5b, indicating the productivity of mutual information (correlation) from entropy (randomness).

## 4 Discussion

Based on resultant dynamics of networks as shown in Figs. 4 and 5, the effects of embedding subgraphs in Boolean networks can be divided into two groups.

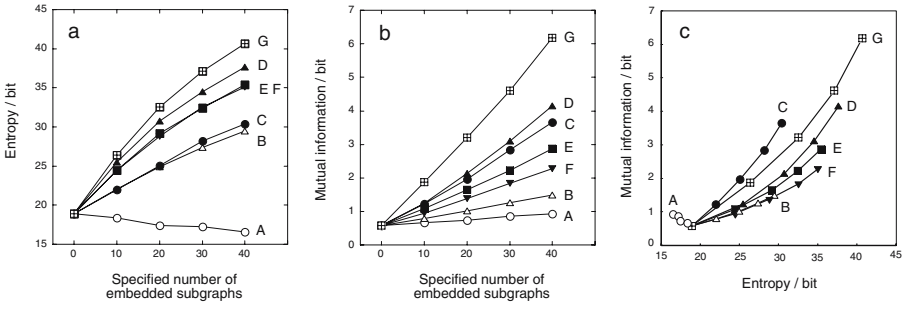



**Fig. 5.** Temporal structure of state variables. The relationship between the number of embedded subgraphs and the amount of entropy (a), the amount of mutual information (b), and rearranged data from a and b, (c). The symbols indicate mean values. Different capital alphabets indicate different subgraphs [Fig. 1].

Networks with more subgraphs, excluding feedforward, show a larger number of attractors and greater entropy and mutual information, demonstrating that the six subgraphs increase the complexity of the state space of the networks. In other words, the subgraphs behave as destabilizers of state space, pattern generators of a temporal program of gene expression, or entropy generators. The resultant mutual information (correlation) is driven by entropy as shown in the six positive slopes in Fig. 5c. These results are consistent with the Thomas's and Sontag's conjecture [15,16,17] because the reciprocal edges in a subgraph contribute to increasing of cyclic loops in the subgraph [Fig. 1].

On the other hand, networks with more feedforward subgraphs show a smaller number of attractors and less entropy, but greater mutual information. This indicates that feedforward subgraphs stabilize the state space as well as organize temporal patterns with less entropy, as shown by the negative slope in Fig. 5c. Detailed analyses with differential equations [18,19] show feedforward loops are robust to variations in biochemical parameters and work as a low-pass filter. Together with our numerical results and related work [11,17], this suggests that feedforward loops are favorable local structure in complex biological networks.

Actual complex biological networks are established based on emergence and evolutionary processes, and the resultant structure has many statistical features. Here we concentrate on the effects of embedding subgraphs in Boolean networks. The constructive approach [6,11,13,14,17] promises to provide insight into the prediction of relationships between network structures, behaviors, and functions.

### 4.1 Control Parameters for Boolean Networks

The control parameters for the dynamics of Boolean networks are input connectivity, $K_{in}$, the size of network, the bias of the Boolean functions, and output connectivity distributions [6,20,21]. In this report, we change only the connection style, while maintaining the same amount of network resources [Table 2].

Figures 4 and 5 demonstrate that the internal connection style, such as the number of reciprocal edges or cyclic loops, may well be regarded as a novel control parameter for the dynamics of Boolean networks.

### 4.2 Differences in Correlation Productivity

In general, complex adaptive systems, including biological systems, perform their functions correctly when certain appropriate communications among nodes are established, because such systems need to add or delete nodes, or change the connectivity strength to adapt to exogeneous inputs optimally. Therefore, the productivity of correlation among nodes is the critical factor for the networks. However, the dynamics of our results are collective behavior of the interaction of single kind of subgraphs. The lowest correlation productivities in Fig. 5c can be seen for subgraphs B and F, these two subgraphs do not involve feedforward structures [Fig. 1]. The other subgraphs, excluding subgraph A, show greater productivity. In fact subgraph C and D have the largest freqencies in signal transduction, neuronal networks [3,4].

## Acknowledgements

This work was supported by a Grant-in-Aid for Young Scientists (B) No. 18740237 from MEXT, Japan (C.O.).

## References

1. Barabási, A.-L., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Genetics 5, 101–112 (2004)
2. Newman, M., Barabási, A.-L., Watts, D.J.: The Structure and Dynamics of Networks. Princeton University Press, Princeton (2006)
3. Alon, U.: An Introduction to Systems Biology. Chapman & Hall/CRC (2006)
4. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science 298, 824–827 (2002)
5. Kauffman, S.A.: The Origins of Order. Oxford University Press, Oxford (1993)
6. Oosawa, C., Savageau, M.A.: Effects of alternative connectivity on behavior of randomly constructed Boolean networks. Physica D 170, 143–161 (2002)
7. Harris, S., Sawhill, B., Wuensche, A., Kauffman, S.: A model of transcriptional regulatory networks based on biases in the observed regulation rules. Complexity 7, 23–40 (2002)
8. Raeymakers, L.: Dynamics of Boolean Networks Controlled by Biologically Meaningful Functions. J. Theor. Biol. 218, 331–341 (2002)
9. Setty, Y., Mayo, A.E., Surette, M.G., Alon, U.: Detailed map of cis-regulatory input function. Proc. Natl. Acad. Sci. 100, 7702–7707 (2003)
10. Nikolajewa, S., Friedel, M., Wilhelm, T.: Boolean networks with biologically relevant rules show ordered behavior. BioSystems 90, 40–47 (2007)
11. Oosawa, C., Takemoto, K., Savageau, M.A.: Effects of feedback and feedforward loops on dynamics of transcriptional regulatory model networks. In: Proceedings of the 13th International Symposium on Artificial Life and Robotics, pp. 885–890, http://arxiv.org/abs/0711.2730v2 [arXiv:0711.2730]

12. Krawitz, P., Shmulevich, I.: Basin Entropy in Boolean Ensembles. Phys. Rev. Lett. 98, 158701 (2007)
13. Oosawa, C., Takemoto, K., Matsumoto, S., Savageau, M.A.: Local cause of coherence in Boolean networks. In: Proceedings of the 12th International Symposium on Artificial Life and Robotics, pp. 621–626 (2007), http://arxiv.org/abs/nlin/0611049 [nlin/0611049]
14. Oosawa, C.: Roles of hubs in Boolean networks. In: Proceedings of the 15th IEEE International Workshop on Nonlinear Dynamics of Electrical Systems, pp. 245–248 (2007), http://arxiv.org/abs/nlin/0703033 [nlin/0703033]
15. Thomas, R., Thieffry, D., Kaufman, D.: Dynamical behaviour of biological regulatory networks I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. Bulletin of Mathematical Biology 57, 247–276 (1995)
16. Sontag, E.D.: Monotone and near-monotone biochemical networks. Syst. Synth. Biol. 1, 59–87 (2007)
17. Sontag, E.D., Veliz-Cuba, A., Laubenbacher, R., Jarrah, A.S.: The effect of negative feedback subgraphs on the dynamics of Boolean networks, http://arxiv.org/abs/0707.3468v2 [arXiv:0707.3468]
18. Mangan, S., Alon, U.: Structure and function of the feed-forward loop network motif. Proc. Natl. Acad. Sci. 100, 11980–11985 (2003)
19. Balázsi, G., Barabśi, A.-L., Oltvai, Z.N.: Topological units of environmental signal processing in the transcriptional regulatory network of *Escherichia coli*. Proc. Natl. Acad. Sci. 102, 7841–7846 (2005)
20. Moreira, A.A., Amaral, L.A.N.: Canalizing kauffman networks: Nonergodicity and its effect on their critical behavior. Phys. Rev. Lett. 94, 218702 (2005)
21. Aldana, M., Cluzel, P.: A natural class of robust networks. Proc. Natl. Acad. Sci. 100, 8710–8714 (2003)

# About 4-States Solutions to the Firing Squad Synchronization Problem

Hiroshi Umeo[1], Jean-Baptiste Yunès[2], and Naoki Kamikawa[1]

[1] University of Osaka Electro-Communication
Neyagawa-shi, Hatsu-cho, 18–8. 5172–8530, Osaka, Japan
umeo@cyt.osakac.ac.jp
[2] LIAFA - Université Paris 7 Denis Diderot,
175, rue du chevaleret. 75013 Paris - France
Jean-Baptiste.Yunes@liafa.jussieu.fr

**Abstract.** We present some elements of a new family of time-optimal solutions to a less restrictive firing squad synchronization problem. These solutions are all built on top of some elementary algebraic cellular automata. Thus, this gives a very new insight on the problem and a more general way of computing on cellular automata.

## 1 The Firing Squad

As an old inverse problem, the Firing Squad Synchronization Problem (FSSP for short – see [6]) proposes to design a linear cellular automata $\mathcal{A} = (\mathcal{Q}, \delta)$, with a finite set of states $\mathcal{Q} = \{\bullet, \mathtt{A}, \mathtt{B}, \ldots, x\}$, and the transition function $\delta : \{\$\} \cup \mathcal{Q} \times \mathcal{Q} \times \{\$\} \cup \mathcal{Q} \to \mathcal{Q}$ ($\$ being an external state) which is able to *synchronize* **any** chain of $n$ cells so that there exists a distinguished state $f \in \mathcal{Q}$ so that:

- from the **starting configuration** at time 0, which is of the form $\mathtt{A}\bullet^{n-1}$;
- it reaches the **synchronized configuration** $\mathtt{f}^n$ after $\mathcal{T}(n)$ transitions;
- so that the state $\mathtt{f}$ never appears before the $\mathcal{T}(n)$-th step.

Besides the fact that the original problem has been studied for a long time (see [1,4,5,11]), it is interesting to note that recent results may give a very new insight on the problem (see [7,8,9,10,13,14,15]). First and independently, Yunès (see [13,14]), Settle & Simon (see [7]) and Umeo *et al.* (see [9]) designed non minimal-time solutions with very few states; since Balzer (see [1] in is known that there is no optimum-time solution to the original problem with 4-states, and since Mazoyer (see [4]) we know that there exists a 6-states solution and wether there exists a 5-states solution).

And a recent promising line of research, started by Umeo and Yunès (see [8,10,15]) is to look for solutions able to synchronize an infinite number but not all lines. That less restrictive problem is very surprisingly solvable with only 4-states (no gap remains as Yunès proved that there is no 3-states solution to that less restrictive problem). What is remarkable is that with 5 states (see [10]) we can use common construction as signal and collision schema, but 4-states solutions are all built using some elementary algebra.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 108–113, 2008.

Here we present some of these solutions in only 4-states which synchronizes every line whose length is a power of 2

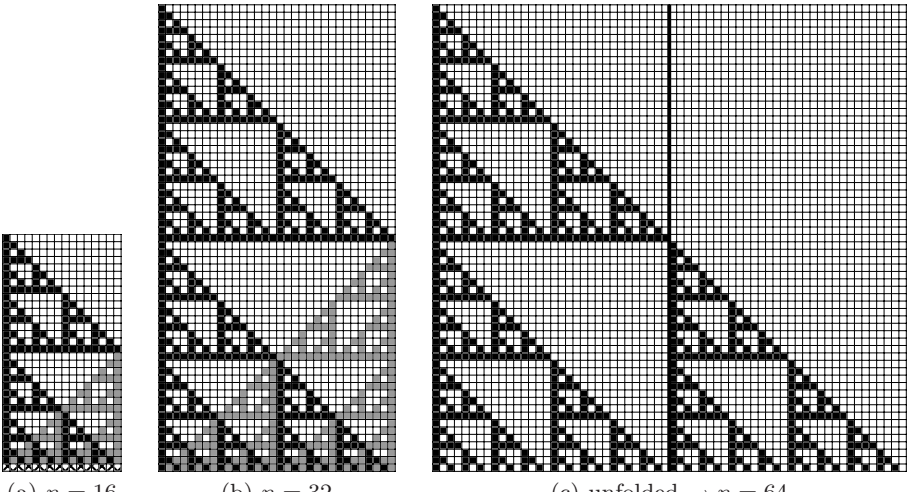
## 2 The Rule 60 Based Solution

A first construction is based on Wolfram's rule 60 (see [12], page 1035). Wolfram stated that running rule 60 (Pascal's triangle modulo 2) on a configuration of length a power of 2 where the left end cell is 1 leads to something that looks like a synchronization. But one can easily note that rule 60 is not a solution to the problem and we will see that we can use it to construct something able to synchronize every power of 2.

The key idea is to use a simple folding of the space-time diagram of rule 60 running on a line of length $2^{n+1}$, which gives a space-time diagram of a 4-states cellular automata running on a line of length $2^n$. That cellular automata exploits an interleaving property of rule 60 configurations such that a synchronization is obtained at the right time ($2n$ for a line of length $n$).

The figure 1 illustrates how the construction is made. The figure 1(a) shows a full run of the solution on a line of length 16, and figures 1(b)and 1(c) exhibits how the *folding* is used. Table 1(a) contains the transition function of our solution.

For a proof of correctness the reader must refer to [15].

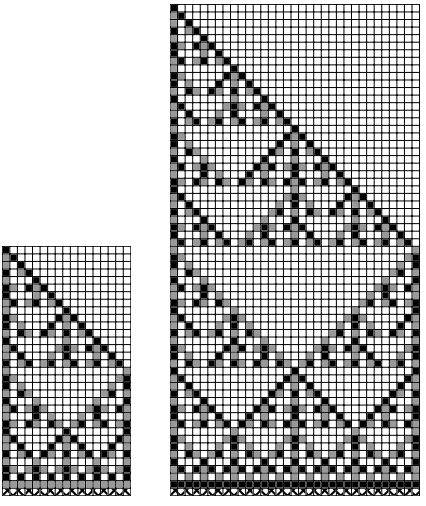


(a) $n = 16$ (b) $n = 32$ (c) unfolded $\rightarrow n = 64$

**Fig. 1.** Folding of a Pascal's triangle modulo 2

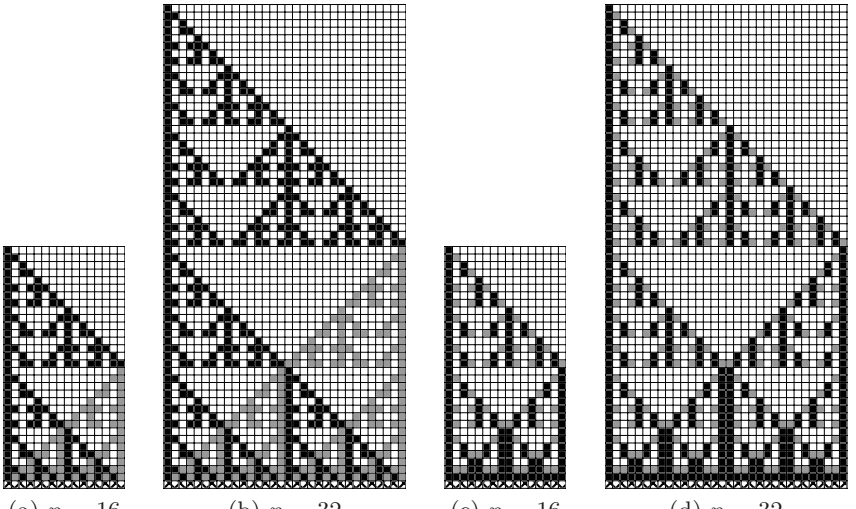
## 3 The Rule 150 Based Solutions

All the following constructions use Wolfram's rule 150 as a base for their design.



(a) $n = 17$ (b) $n = 33$

**Fig. 2.** Strict optimum-time solution



(a) $n = 16$ (b) $n = 32$ (c) $n = 16$ (d) $n = 32$

**Fig. 3.** Rule 150 based solution

### 3.1 Strict Optimum-Time Solution

The figure 2 illustrates how that solution works and the table 1(b) gives the transition function of it. The reader must note that this construction synchronizes every line of length $N = 2^n + 1$ at time $2N - 1$ ($2N - 2$ steps), so this is a strict

**Table 1.** The transition functions

(a) Rule 60. Synchronizes lengths $2^n$

| • | $ | • | A | B |
|---|---|---|---|---|
| $ | | • | | |
| • | • | • | • | B |
| A | A | A | A | |
| B | | • | | B |

| A | $ | • | A | B |
|---|---|---|---|---|
| $ | | A | A | C |
| • | | A | A | |
| A | B | • | • | B |
| B | | A | A | C |

| B | $ | • | A | B |
|---|---|---|---|---|
| $ | | | | |
| • | B | B | B | • |
| A | C | | C | A |
| B | B | B | B | • |

(b) Rule 150. Synchronizes lengths $2^n + 1$

| • | $ | • | A | B |
|---|---|---|---|---|
| $ | | | | |
| • | • | • | A | B |
| A | B | A | B | • |
| B | | B | • | A |

| A | $ | • | A | B |
|---|---|---|---|---|
| $ | | B | C | A |
| • | B | B | | • |
| A | C | | C | |
| B | A | • | | A |

| B | $ | • | A | B |
|---|---|---|---|---|
| $ | | A | B | C |
| • | A | A | • | |
| A | B | • | B | |
| B | C | | | C |

(c) Rule 150. Synchronizes lengths $2^n$

| • | $ | • | A | B |
|---|---|---|---|---|
| $ | | • | | |
| • | • | • | A | B |
| A | A | A | • | |
| B | | B | | • |

| A | $ | • | A | B |
|---|---|---|---|---|
| $ | | A | A | C |
| • | | A | • | |
| A | B | • | A | B |
| B | | A | A | C |

| B | $ | • | A | B |
|---|---|---|---|---|
| $ | | | | |
| • | B | B | B | • |
| A | C | | C | A |
| B | B | • | B | B |

(d) Rule 150. Synchronizes lengths $2^n$

| • | $ | • | A | B |
|---|---|---|---|---|
| $ | | • | | |
| • | • | • | B | A |
| A | B | B | • | |
| B | | A | | • |

| A | $ | • | A | B |
|---|---|---|---|---|
| $ | | A | C | A |
| • | A | A | A | • |
| A | C | A | C | A |
| B | A | • | A | A |

| B | $ | • | A | B |
|---|---|---|---|---|
| $ | | | | |
| • | | | • | |
| A | A | • | | A |
| B | | | A | |

optimum-time solution, as can be opposed to the previous construction which synchronizes in $2N - 1$ steps - one more extra step.

Here the key idea is also to use a folding of some space-time diagram, but such that states `A` and `B` used in the unfolded part are respectively projected to `B` and `A` in the folded part.

### 3.2 Simple Folding Variants of Rule 150 Based Solution

The figure 3 illustrates two variants of the preceding solution and the tables 1(c) and 1(d) contain their respective transition functions. These solutions are also based on Wolfram's rule 150 as one can observe that the *shadow* of their space-time diagram is exactly the rule 150. What we call the shadow of a space-time diagram is what is obtained identifying, in a given space-time diagram, all non quiescent states into a single one. The picture obtained is something which uses only two states (the original quiescent and the shadow state). Of course, in the general case, the shadow diagram is not a 2-states cellular automata space-time

diagram, but it can be in some particular cases (as the cases illustrated here). By example, identifying black and grey states in the upper part of figures 2(a), 2(b), 3(c), and 3(d) give the space-time diagram of rule 150 (upper part of figures 3(a) and 3(b)).

The first solution is built as the first solution of this paper (rule 60 based solution). The space-time diagram for a line of length $2^n$ is obtained by a simple folding of the original space-time diagram of Wolfram's rule 150 running on a line of length $2^{n+1}$.

The second solution is also built using our simple folding but combined with a parity cell position coloring. That coloring doesn't perturb the property used to obtain the synchronization.

## 4 Conclusion

We were able to construct different 4-states solution to the firing squad which synchronizes every line of length $2^n$ or $2^n + 1$. As their are all based on some simple algebraic computation, we think that it would be possible to find a simple formal proof of their correctness as it has already been done to the first of them (see [15]).

All theses solutions have a very low Kolmogorov complexity, the three first presented solutions use only 33 transitions and the last uses only 30 transitions.

One can note that the number of non quiescent cells is not in the order of $n.log(n)$ for a line of length $n$, neither it is in the order of $n^2$, but something in between. That number is of course directly related to the Hausdorff dimension of the fractal built.

These solutions all belong to a new family of solutions based on algebraic computation, may be a new way of **programming on cellular automata** (algebraic cellular automata have already been studied but in the field of discrete dynamical systems).

Moreover, if these constructions doesn't synchronize every line, it is worth notice that a recent reconstruction of Goto's first FSSP solution (see [3,16]) can naturally be based on.

## References

1. Balzer, R.: An 8-State Minimal Time Solution to the Firing Squad Synchronization Problem. Information and Control 10, 22–42 (1967)
2. Gerken, H.-D.: Über Synchronisations - Probleme bei Zellularautomaten. Diplomarbeit, Institut für Theoretische Informatik. Technische Universität Braunschweig (1987)
3. Goto, E.: A minimum time solution of the firing squad synchronization problem. Courses Notes for Applied Mathematics 298, 52–59 (1962)
4. Mazoyer, J.: A Six-State Minimal-Time Solution to the Firing Squad Synchronization Problem. Theoretical Computer Science 50, 183–238 (1987)
5. Minsky, M.: Computation: Finite and Infinite Machines. Prentice-Hall, Englewood Cliffs (1967)

6. Moore, E.F.: The Firing Squad Synchronization Problem. In: Moore, E.F. (ed.) Sequential Machines. Selected Papers, pp. 213–214. Addison-Wesley, Reading (1964)
7. Settle, A., Simon, J.: Non-Minimal Time Solutions for the Firing Squad Synchronization Problem. Technical Report 97-08, University of Chicago (1997)
8. Umeo, H., Kamikawa, N.: A 4-states solution to the firing squad based on Wolfram's rule 150. Private communication (2007)
9. Umeo, H., Maeda, M., Hongyo, K.: A Design of Symmetrical Six-State 3n-Steps Firing Squad Synchronization Algorithms and Their Implementations. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 157–168. Springer, Heidelberg (2006)
10. Umeo, H., Yanagihara, T.: A smallest five-state solution to the firing squad synchronization problem. In: Durand-Lose, J., Margenstern, M. (eds.) MCU 2007. LNCS, vol. 4664, pp. 291–302. Springer, Heidelberg (2007)
11. Waksman, A.: An Optimum Solution to the Firing Squad Synchronization Problem. Information and Control 9, 66–78 (1966)
12. Wolfram, S.: Cellular Automata and Complexity: Collected Papers. Westview Press (2002)
13. Yunès, J.-B.: Seven States Solutions to the Firing Squad Synchronization Problem. Theoretical Computer Science 127(2), 313–332 (1994)
14. Yunès, J.-B.: An Intrinsically non Minimal-Time Minsky-like 6-States Solution to the Firing Squad Synchronization Problem. RAIRO ITA/TIA 42(1), 55–68 (2008)
15. Yunès, J.-B.: A 4-states Algebraic Solution to Linear Cellular Automata Synchronization. International Processing Letters (2008), doi: 10.1016/j.ipl.2008.01.009
16. Yunès, J.-B.: Goto's construction and Pascal's triangle: new insights into cellular automata synchronization. In: Durand, B. (ed.) JAC 2008, Uzès, France, April 21–25, pp. 195–203. MCCME Publishing House Moscow (2008)

# Evaluating Cellular Automata Models by Evolutionary Multiobjective Calibration

Maria Vittoria Avolio[1], Donato D'Ambrosio[2], Salvatore Di Gregorio[2], Valeria Lupiano[1], Rocco Rongo[1], William Spataro[2], and Giuseppe A. Trunfio[3]

[1] Department of Earth Sciences, University of Calabria, 87036 Rende (CS), Italy
[2] Department of Mathematics, University of Calabria, 87036 Rende (CS), Italy
[3] DAP, University of Sassari, 07041 Alghero (SS), Italy

**Abstract.** This paper proposes a multi-objective approach for Cellular Automata (CA) calibration. The method exploits the available temporal sequences of spatial data in order to produce CAs which are non-dominated (i.e. Pareto optimal) with respect to multiple objectives representing the disagreement between the simulated and real dynamics. A preliminary application, based on a parallel multi-objective Genetic Algorithm, showed that the proposed approach can provide significant insights about potentialities and limits of a CA model.

## 1 Introduction

In most Cellular Automata (CA) models of real complex systems (e.g. [1,2]): (*i*) each cell represents a portion of space (e.g. a location on the Earth's surface); (*ii*) the states of the cell correspond to spatial characteristics which are important to the model (e.g. slope, temperature); (*iii*) the transition function models some local interactions among the system components; (*iv*) each CA step corresponds to an interval of time. In addition, the transition function is often dependent on some scalar parameters representing properties common to all systems' components (e.g. [3,2,1]).

The reliability of such parameterized CA models is maximized by some standard procedures, such as *model calibration*. The calibration process consists of finding the values of the model parameters that are not exactly known and cannot be directly measured, in such a way that the model itself better mimics the observed dynamic of the system under consideration. Since the search space is usually large, automated methods have been developed by defining calibration as an optimization problem in which the solution in terms of parameter values must minimize an objective function (i.e. the error measure) [1,2]. In the automatic calibration process, the match between the simulated and the observed phenomenon is usually quantified by a single error measure. However, in many applications the ability of the CA model of reproducing different aspects of the system's dynamic is crucial. In that case, a specific error function is associated to each aspect in order to measure the corresponding level of misfit between the simulated patterns and the real ones. Calibration to multiple objectives is

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 114–119, 2008.

typically achieved by combining them into a single overall objective measure of comparison (e.g. by a weighted linear combination or by multiplying the different objectives). Nevertheless, the particular combination of multiple error measure in a single objective, which usually affects significantly the calibrated parameter values, is largely arbitrary and can be difficult because of the difference of scale of the single measures. In addition, it precludes a full evaluation of the model performances in the space defined by the objectives, since such space is collapsed in a mono-dimensional one. As a suitable alternative to the objective-combination approach, a full multi-objective optimization can be conducted that identifies the entire optimal set of non-dominated or Pareto solutions [4]. In this paper such a multiobjective approach, based on a GA, is suggested with the purpose of optimizing CA models.

## 2 Multiobjective Optimization of Cellular Automata

One of the most common CA-based modelling approach is based on a generalization of the classical CA formulation, where the states of the cell are represented by the values of numeric variables. Moreover, in many applications of the method the transition function depends on a vector of scalar parameters $\mathbf{p} = [p_1, \ldots, p_r]^T$ (e.g. [3,2,1]. In such CA models, all of the local transition functions, acting simultaneously on each cell, can be thought as an overall parameter-dependent transition function $\Phi$ which acts on the entire automaton and gives the global configuration $\boldsymbol{\Omega}^{(t+1)}$ at the step $t+1$ (i.e. the set of all cell states) as:

$$\boldsymbol{\Omega}^{(t+1)} = \Phi(\boldsymbol{\Omega}^{(t)}, \mathbf{p}) \tag{1}$$

Thus, the iterative application of function $\Phi$ to the successive configurations, starting from an initial one $\boldsymbol{\Omega}^{(0)}$, leads to the dynamic process

$$\boldsymbol{\Omega}^{(0)} \xrightarrow{\Phi} \boldsymbol{\Omega}^{(1)} \xrightarrow{\Phi} \cdots \xrightarrow{\Phi} \boldsymbol{\Omega}^{(t)} \quad \Longrightarrow \quad \boldsymbol{\Omega}^{(t)} = \Phi^t(\boldsymbol{\Omega}^{(0)}, \mathbf{p}) \tag{2}$$

where the dependence of the automaton configuration at the time step $t$ on both the initial configuration and the parameters is explicit, with the other automaton characteristics (i.e. the model structure) being fixed. With respect to the parameters $\mathbf{p}$, the model can be optimized to minimise the misfit, expressed by proper measures, between the simulated patterns and the real ones.

To formalize the problem, let us suppose the existence of spatio-temporal datasets collecting some automaton configurations, which come from experiments of the real system behaviour. Let each dataset $\bar{\mathcal{V}}$ be composed by a series of $q$ automaton configurations:

$$\bar{\mathcal{V}} = \{< \bar{\boldsymbol{\Omega}}^{(0)}, \tau_0 >, < \bar{\boldsymbol{\Omega}}^{(1)}, \tau_1 >, \ldots, < \bar{\boldsymbol{\Omega}}^{(q)}, \tau_q >\} \tag{3}$$

where the attribute $\tau_i$ indicates the instant of time in which the configuration $\bar{\boldsymbol{\Omega}}^{(i)}$ is known. Starting from the configuration $\bar{\boldsymbol{\Omega}}^{(0)}$, and given a vector $\mathbf{p}$ of

parameters, the process (2) can be executed for the computation of the $q-1$ automaton configurations:

$$\mathcal{V} = \{\boldsymbol{\Omega}^{(1)}, \ldots, \boldsymbol{\Omega}^{(q)}\} \tag{4}$$

where $\boldsymbol{\Omega}^{(j)} = \Phi^j(\bar{\boldsymbol{\Omega}}^{(0)}, \mathbf{p})$. Thus, the optimization process consists of the determination of a proper value of the vector $\mathbf{p}$, which leads to the best agreement between the real spatio-temporal sequence (3) and the simulated one (4). As mentioned before, in many applications it is desirable that the model reproduces different aspects of the real system evolution in space and time. Thus, in general, $g$ measures of error are defined:

$$\theta_i = \Theta_i\left(\bar{\mathcal{V}}, \mathcal{V}\right) = \Theta_i\left(\bar{\mathcal{V}}, \mathbf{p}\right) \qquad i = 1, \ldots, g \tag{5}$$

Each function $\Theta_i$ corresponds to an optimization objective and accounts for a specific aspect of the disagreement between simulated and real patterns. As pointed out in Section 1, the typical approach of optimizing to multiple targets by combining the multiple objectives into a single one presents some drawbacks.

The alternative proposed in this paper is to conduct a full multiobjective optimization that identifies a set of non-dominated or Pareto solutions [4] within a single optimization run. In particular, the optimization is done using a multiobjective Genetic Algorithm (GA) [5]. The GA is used to evolve a randomly initialized population, whose generic *chromosome* is a vector encoding an $r$-dimensional vector of parameters $\mathbf{p}$. The $i$-th element of the chromosome is obtained as the binary encoding of the parameter $p_i$, using a suitable number of bits and its set of definition $\mathcal{P}(p_i)$. Each chromosome can be decoded in a vector of parameters $\mathbf{p}$ and, through performing a CA simulation, the objective functions can be computed. In the multiobjective GA, to avoid the aggregation of multiple objectives, the comparison of two candidate solutions, with respect to different objectives, is achieved through the concepts of Pareto optimality and dominance. In particular, we say that a solution $\mathbf{p}^*$ (strongly) *dominates* the solution $\mathbf{p}$ if:

$$\forall\, i:\ \Theta_i\left(\bar{\mathcal{V}}, \mathbf{p}^*\right) \leq \Theta_i\left(\bar{\mathcal{V}}, \mathbf{p}\right) \quad \wedge \quad \exists\, j\ : \Theta_j\left(\bar{\mathcal{V}}, \mathbf{p}^*\right) < \Theta_j\left(\bar{\mathcal{V}}, \mathbf{p}\right) \tag{6}$$

In other words, $\mathbf{p}^*$ dominates $\mathbf{p}$ if $\mathbf{p}^*$ is better or equivalent to $\mathbf{p}$ with respect to all objectives, and better in at least one objective. A non-dominated solution can be considered optimal in the Pareto sense (i.e. no criterion can be improved without worsening at least one other criterion). Rather than a single solution, a search based on Pareto-optimality produces a set of non-dominated solutions. In this work, the adopted multiobjective GA is the well known NSGA-II, which has been extensively investigated and is described in [6].

## 3 An Application Example

The multiobjective optimization approach has been applied in conjunction with SCIARA-fv, which is the last release of a family of CA models for lava flows

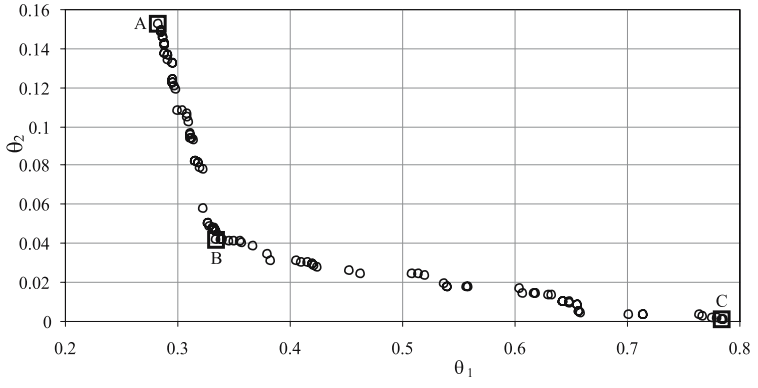



**Fig. 1.** Estimated Pareto front for the calibration example. Marked points correspond to the best spatial matching (A), possible trade-off solution (B) and best temporal matching (C).

simulation [7]. The model, which is described in detail in [1], is based on a hexagonal grid and accounts for the relevant physical processes involved in the macroscopic phenomenon.

In SCIARA-fv the transition function depends on the following seven scalar parameters, invariant in time and space: $p_s$, the time corresponding to a CA step; $p_{Tsol}$, the lava temperature at solidification; $p_{adv}$, the lava adherence at the crater (it represents the amount of lava that, given its current rheological resistance, cannot flow out from the central cell towards any neighbouring ones); $p_{adsol}$, the lava adherence at solidification; $p_{cool}$, the "cooling parameter", which depends on lava rheology and regulates the thermal energy loss due to lava surface irradiation; $p_{Tv}$, the lava temperature at the crater; $p_a$ the cell apothem. Once that the input to the model has been provided, SCIARA-fv can simulate the lava invasion process. The simulation stops when the lava fluxes fall below a threshold value.

Before using the model for predictive applications, the parameters must be optimized for a specific area and lava type using at least one real eruptive event. To this end, in previous applications either a single error measure or the product of some selected measures have been used for measuring the disagreement between the data describing the real event and the simulation outcomes [1]. Instead, in the present application SCIARA-fv has been coupled with the NSGA-II algorithm and two error measures have been selected for conducting a full multiobjective optimization. In particular, the first error measure took into account only the comparison between the areal extensions of the real and simulated events; it was defined as:

$$\theta_1 = 1 - \sqrt{\frac{m(R \cap S)}{m(R \cup S)}} \tag{7}$$

where $R$ and $S$ represent the areas affected by the real and simulated event, respectively, while $m(A)$ denotes the measure of the set $A$. Note that $\theta_1 \in [0,1]$;

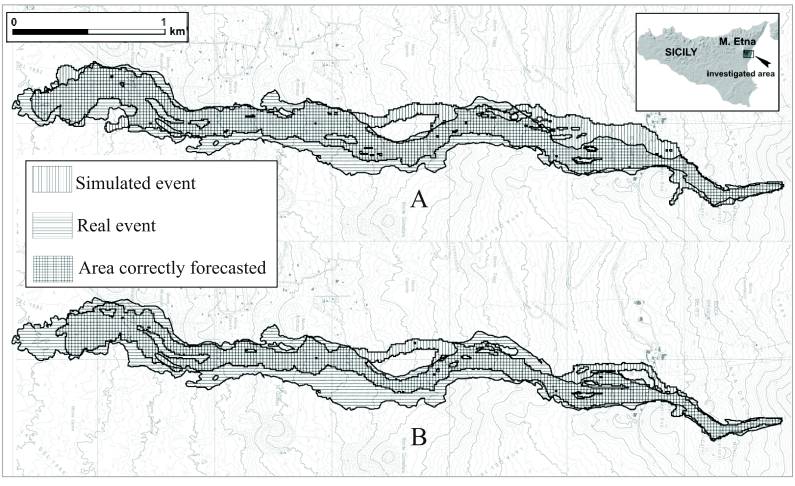



**Fig. 2.** Spatial comparison between real and simulated events for the selected solutions $A$ and $B$ in Figure 1

its value is 1 if the real and simulated events are completely disjoint, being $m(R \cap S)$=0; it is 0 in case of perfect overlap, being $m(R \cap S) = m(R \cup S)$. The second objective function measures the relative error between the duration of the real event and the simulate one.

$$\theta_2 = \frac{\mid \Delta t_{real} - n_{step} \times p_s \mid}{\Delta t_{real}} \tag{8}$$

where $\Delta t_{real}$ is the duration of the real event and $n_{step}$ is the number of CA steps executed until the simulation reaches the stop condition mentioned above. A value $\theta_2 = 0$ implies that the simulated time is exactly equal to the real event duration. Thus, while $\theta_1$ is related to the correspondence between real and simulated spatial patterns, $\theta_2$ indicates how the model is capable of reproducing the real phenomenon in the temporal domain.

The multiobjective calibration exercise refers to an eruptive episode, known as the 2001 Nicolosi event, which started at 3:00 AM on July 18-th, 2001, from the the southern flank of Mt Etna (Sicily, Italy), at 2100 m a.s.l. The event was fed by a lava flow rate of about $7m^3/s$ and after 10 days of activity, it reached its maximum extension, which was almost 6 Km in terms of run-out. In this case, calibration is needed only for the first five parameters since parameter $p_{Tv}$ is set to a value of $1373K$, which corresponds to the typical temperature of Etnean lava flows at craters, and parameter $p_a$ was also prefixed to the value of $5m$ corresponding to the detail of the considered topographic data.

The non-dominated set is represented in Figure 1. It shows that $\Theta_1$ ranges in the interval $[0.283, 0.786]$ while the relative error $\Theta_2$ in the interval $[0.12\%, 15.3\%]$. The shape of the Pareto front suggests that the spatial error $\Theta_1$ can be reduced up to about 0.34 (i.e. point $B$ in Figure 1) without significant loss of accuracy in terms

of the relative error on the event duration $\Theta_2$, which remains below 4%. In other words, along the branch $CB$, a minor relaxation of the performance in simulating the phenomenon duration implies a significant improvement in the quality of the simulated spatial patterns. Following the estimated Pareto front from point $B$ to point $A$, a further reduction of spatial error can be achieved at the price of a significant increment of the event-duration error. In particular, in point $A$ the model attains the best spatial fitting ($\Theta_1 = 0.283$) but the simulated duration presents a difference of about 15.3% with respect to the real one. Clearly, since for most applications the spatial fitting is more important than an exact prediction of the phenomenon duration, a typical trade-off solution could be chosen from the points of the estimated Pareto front which are comprised between point $A$ and point $B$ (see Figure 1). To give an idea of the spatial implications that such a trade-off may imply, in Figure 2 the two maps corresponding to the points $A$ and $B$ are depicted and compared with the map of the real eruptive event.

## 4 Conclusions

In the examined case there are solutions offering a good compromise among all objectives. This might not always be the case. Indeed, significant trade-offs in fitting two or more objectives may indicate an error in the model structure (e.g. a relevant physical process may not be accounted for). In this sense the multi-objective approach can provide useful information about the model and the set of non-dominated solutions offers a representation rich in valuable information concerning the CA model, thereby potentially leading to better optimizations and improved understanding of the model potentialities and limits.

## References

1. Spataro, W., D'Ambrosio, D., Rongo, R., Trunfio, G.A.: An evolutionary approach for modelling lava flows through cellular automata. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 725–734. Springer, Heidelberg (2004)
2. Clarke, K., Hoppen, S., Gaydos, L.: A self-modifying cellular automaton model of historical urbanization in the san francisco bay area. Environment and Planning B-Planning and Design 24, 247–261 (1997)
3. Di Gregorio, S., Serra, R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. Future Generation Computer Systems 16, 259–271 (1999)
4. Pareto, V.: Cours d'Economie Politique, vol. I, II. F. Rouge, Lausanne (1896)
5. Coello, C.C., Veldhuizen, D.V., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, Dordrecht (2002)
6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comp. 6, 182–197 (2002)
7. Crisci, G., Rongo, R., Di Gregorio, S., Spataro, W.: The simulation model SCIARA: the 1991 and 2001 lava flows at mount Etna. Journal of Volcanology and Geothermal Research 132, 253–267 (2004)

# Entropy and Chaos in a Lattice Gas Cellular Automata

Franco Bagnoli[1] and Raúl Rechtman[2]

[1] Dipartimento di Energetica, Università di Firenze,
Via S. Marta, 3 I-50139 Firenze, Italy
Also CSDC and INFN, sez. Firenze
franco.bagnoli@unifi.it
[2] Centro de Investigacíon en Energía, UNAM,
62580 Temixco, Morelos, Mexico
rrs@cie.unam.mx

**Abstract.** We find a simple linear relation between the thermodynamic entropy and the largest Lyapunov exponent (LLE) of an discrete hydrodynamical system, a deterministic, two-dimensional lattice gas automaton (LGCA). This relation can be extended to irreversible processes considering the Boltzmann's $H$ function and the expansion factor of the LLE. The definition of LLE for cellular automata is based on the concept of Boolean derivatives and is formally equivalent to that of continuous dynamical systems.

## 1 Introduction

The relation between thermodynamics and the underlying chaotic properties of a system is of great relevance in the foundations of statistical mechanics [1] and has attracted much interest. For a family of models of simple liquids, including Lennard–Jones, a simple relation between the Kolmogorov–Sinai entropy and the thermodynamic entropy exists [2]. The connection between chaotic dynamical properties and transport coefficients has been established for Lorentz gases [3].

Investigations on extended or multi-body systems are difficult, and most of previous studies were restricted to non-interacting particles on some disordered configuration of scatteres. Moreover, continuous or partially continuous systems require coarse-graining that introduce another degree of arbitrariety in the analysis. These difficulties have lead to the study of rather abstract models (maps) [4].

In this paper we investigate if and to what extent similar relations apply to completely discrete systems like cellular automata, and in particular to lattice gas cellular automata (LGCA).

LGCA are simple models with hydrodynamical behavior [6]. The $D2Q9$ LGCA is a two dimensional model with nine velocities and is one of the simplest models where temperature can be defined [7]. In this paper we find that the thermodynamic entropy density is proportional to the largest Lyapunov exponent (LLE) of a deterministic $D2Q9$ LGCA. Furthermore, in a simple irreversible process,

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 120–127, 2008.

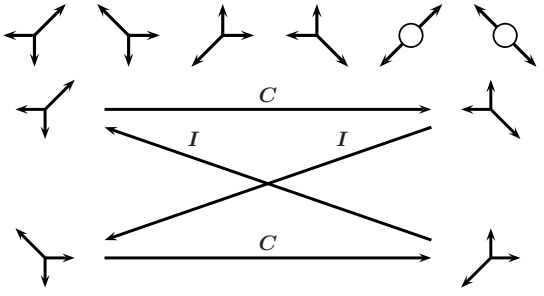



**Fig. 1.** (Top) All these states have the same number of particles, momentum and energy. An arrow represents the presence of a particle with the velocity of the arrow, the open circle a particle at rest. (bottom) Example of entries in a reversible collision table.

there is a linear relation between Boltzmann's $H$ function and the expansion factor of the LLE.

The sensitivity to initial conditions for discrete dynamical systems like cellular automata are has to be understood with care. One cannot perform here the usual limit of vanishing initial distance between configurations. Nevertheless, Lyapunov exponents can be defined and cellular automata which exhibit complex space time patterns have a positive LLE [8]. This definition is closely related to the one used in continuous maps [9].

## 2 Model

The $D2Q9$ model is defined on a two dimensional square lattice. The evolution is in discrete time steps and unit mass particles at every site $\boldsymbol{r}$ can move with one of nine velocities $\boldsymbol{c}_0 = (0,0)$, $\boldsymbol{c}_1 = (1,0)$, $\boldsymbol{c}_2 = (0,1)$, $\boldsymbol{c}_3 = (-1,0)$, $\boldsymbol{c}_4 = (0,-1)$, $\boldsymbol{c}_5 = (1,1)$, $\boldsymbol{c}_6 = (-1,1)$, $\boldsymbol{c}_7 = (-1,-1)$, $\boldsymbol{c}_8 = (1,-1)$. The state of the automaton is given by the set of occupation numbers $\boldsymbol{s}(t) = \{s_k(\boldsymbol{r},t)\}$, where $s_k(\boldsymbol{r},t) = 1$ (0) indicates the presence (absence) of a particle with velocity $\boldsymbol{c}_k$ at site $\boldsymbol{r}$ and time $t$. An exclusion principle forbids the presence of more than one particle in a given site, at a given time with a given velocity.

The time evolution of the system is given by collision and streaming operations. In the collison operation, particles at a given site collide conserving mass and momentum. In the streaming operation particles move to neighboring sites according to their velocities.

Since the number of states for a given site is finite ($2^9$), the local collision operator $C$ is generally implemented as a look-up table. Given a local configuration, the conservation constraints may not define the outgoing state completely as we show in Fig. 1 (top). One may assume that the $C$ look-up table has several columns for all the possible output states. In order to make the automaton deterministic, we assign at the beginning an integer random number $\eta(\boldsymbol{r})$ to each

site, to be used in choosing the column from which the output state is chosen (quenched disorder). That is

$$s_k(\boldsymbol{r}, t+1) = C_k(s_0(\boldsymbol{r}, t), \ldots, s_8(\boldsymbol{r}, t), \eta(\boldsymbol{r})). \tag{1}$$

The choice of the quenched disorder is analogous to the random disposition of scatterers in a wind-tree or other similar Lorentz gases [10]. We have checked in the various conditions studied in this paper that the system is self-averaging: a simulation on a large enough system gives the "average" value over disorder. This is illustrated for instance in Figure 2 (left): the plot is a collection of single-run simulations for different values of a parameter, using different realizations of the quenched disorder. One can notice that fluctuations are quite small.

Moreover, the collisions can be made reversible, an interesting feature for the discussion of irreversibility. To do so, states $s$ must satisfy the condition that in every column $s = \boldsymbol{ICIC}(s)$ holds where $\boldsymbol{I}$ denotes the operator that inverts the velocities. This means that a collision $\boldsymbol{C}$ followed by an inversion $\boldsymbol{I}$ and another collision and inversion leaves the state unchanged when it is taken from the same column in the collision table as we show in Fig. 1 (bottom).

## 3 Lyapunov Exponents

We recall here the salient points of the definition of largest Lyapunov exponent for cellular automata [8]. Let us simplify the notation by using the index $n$ to indicate both the position $\boldsymbol{r}$ and the velocity $k$, with $n = 1, \ldots, 9L$, where $L$ is the number of sites of the automata. The evolution of the state $s_n(t)$ of the LGCA can be seen as an application of a set of Boolean functions

$$s_n(t+1) = F_n(\boldsymbol{s}(t)). \tag{2}$$

The functions $F_n$ and differ in the velocity index $k$ and quenched disorder $\eta(\boldsymbol{r})$. However, since the distribution of the disorder is uniform, and, as shown in the following, the correlations among variables decay very fast, the system is translationally invariant at a mesoscopic level.

Let $\boldsymbol{s}(0)$ and $\boldsymbol{x}(0)$ be two initially close configurations, for example all the components of $\boldsymbol{x}(0)$ may be equal to those of $\boldsymbol{s}(0)$ except for one. We define the bitwise difference between these two configurations with the term "damage". The smallest possible damage is one and the damage vector $\boldsymbol{v}(0)$ is one in the component where $\boldsymbol{s}(0)$ and $\boldsymbol{x}(0)$ are different and zero in all the others. If this damage grows in average during time, the trajectory is unstable with respect to the smallest perturbation. However, due to the discrete nature of LGCA, defects may annihilate during time evolution, altering the measure of instability of trajectories. The correct way of testing for instability is that of considering all possible ways of inserting the smallest damage in a configuration, using as many replicas as the number of components of the configuration, and letting them evolve for one time step counting if the number of damages has grown or diminished. The ensemble of all possible replicas with one damage each is the equivalent of the tangent space for discrete systems.

The task of computing the evolution in tangent space is clearly daunting, but by exploiting the concept of Boolean derivatives [8,11], it is possible to develop a formula very similar to the one used in continuous systems. The Boolean derivative is defined by

$$\frac{\partial F_n(\boldsymbol{s})}{\partial s_p} = F_n(\dots, s_p, \dots) \oplus F_n(\dots, 1 - s_p \dots), \tag{3}$$

with $n, p = 1, \dots, 9L$ and where $\oplus$ represents the sum modulo two (XOR). This quantity measures the sensitivity of the function $F_n$ with respect to a change in $s_p$. The Jacobian matrix $J(\boldsymbol{s})$ has components $J_{np} = \partial F_n(\boldsymbol{s})/\partial s_p$.

We now consider the map

$$\boldsymbol{v}(t+1) = J(\boldsymbol{s}(t))\boldsymbol{v}(t), \tag{4}$$

with $\boldsymbol{v}(0)$ as mentioned above. It is easy to check that $|\boldsymbol{v}(t)| = \sum_i v_i(t)$ is the number of different paths along which a damage may grow in tangent space during time evolution, *i.e.*, with the prescription of just one defect per replica [8]. If there is sensitivity with respect to initial conditions, one expects that $|\boldsymbol{v}(T)|/|\boldsymbol{v}(0)| \sim \exp(\lambda_{_T} T)$ for large $T$ where $\lambda_{_T}$ is the largest finite time Lyapunov exponent. It then follows that

$$\lambda_{_T} = \frac{1}{T} \sum_{t=1}^{T-1} \log u(t) = \langle \log u \rangle_{_T}, \tag{5}$$

where $u(t) = |\boldsymbol{v}(t)|/|\boldsymbol{v}(t-1)|$ is the expansion factor of the LLE. The definition should include the limit when $T \to \infty$ but in practice we always evaluate the finite time LLE. The LLE depends in principle on the initial configuration $\boldsymbol{s}(0)$, initial damage $\boldsymbol{v}(0)$ and quenched disorder $\eta$, but in practice it assumes the same value for all trajectories corresponding to the same macroscopic observables when $T$ is sufficiently large. The LLE of CA as defined above has been used to classify elementary and totalistic Boolean cellular automata [8,12].

## 4 Entropy

We define the entropy $S$ as

$$S = -\sum_{\boldsymbol{s}} p(\boldsymbol{s}) \log(p(\boldsymbol{s})), \tag{6}$$

where $\boldsymbol{s}$ denotes a possible configuration of the system and $p(\boldsymbol{s})$ is the corresponding probability, computed averaging over a set of replicas (statistical ensemble), analogous to the configuration used for computing the LLE. In equilibrium, $S$ is just the the logarithm of the number of possible configurations satisfying the constraints of constant mass and energy.

It is possible to extend the ensemble definition of entropy also to non-equilibrium conditions, $S = S(t)$.

The single particle velocity distribution functions $f_k(\boldsymbol{r},t)$ are defined as the average number of particles at site $\boldsymbol{r}$ with velocity $\boldsymbol{c}_k$ at time $t$ over $R$ samples that share the same quenched disorder $\eta$ and macroscopic constraints with different microscopic initial configurations.

Since the model satisfies Fermi statistics and correlations are unimportant we approximate the probability distribution by

$$p(\boldsymbol{s}) \simeq \prod_{\boldsymbol{r},k} f_k(\boldsymbol{r})^{s_k(\boldsymbol{r})}\big(1-f_k(\boldsymbol{r})\big)^{1-s_k(\boldsymbol{r})}, \tag{7}$$

and the entropy can be approximated by the Boltzmann function $H$, defined as

$$H = -\sum_{\boldsymbol{r},k} f_k(\boldsymbol{r})\log\big(f_k(\boldsymbol{r})\big)+\big(1-f_k(\boldsymbol{r})\big)\log\big(1-f_k(\boldsymbol{r})\big). \tag{8}$$

Defining the connected correlation function $C_{kl}(\boldsymbol{\eta})$ as

$$C_{kl}(\boldsymbol{\eta}) = \frac{1}{L}\sum_{\boldsymbol{r}} s_k(\boldsymbol{r},t)s_l(\boldsymbol{r}+\boldsymbol{\eta},t) - \left(\frac{1}{L}\sum_{\boldsymbol{r}} s_k(\boldsymbol{r},t)\right)^2, \tag{9}$$

we checked that the single-site two-particle correlation function $C_{kl}(0)$ factorizes into the product of single particle distributions before and after the collision, both during the relaxation (out of equilibrium) phase and in the stationary (equilibrium) conditions. In equilibrium, the correlation function $C_{kl}(\boldsymbol{\eta})$ decays to zero for $\boldsymbol{\eta} = 1$. Out of equilibrium, starting with very different configurations in the two halves of the system, the correlation function, still being quite small, exhibits a correlation length of some lattice spacings for a short time. This corresponds to the coherent motion of particles in a shock wave, where the local density is near to zero or nine. However, this correlation quickly disappears; although the motion is correlated at a macroscopic level, as soon as the local density of particles is different from the extremes (for which the collision table has few output configurations) the velocities quickly decorrelate.

In equilibrium, the distribution functions do not depend on $\boldsymbol{r}$ or $t$ and we may write

$$n = \sum_{k=0}^{8} f_k, \qquad\qquad e = \sum_{k=0}^{8} \epsilon_k f_k, \tag{10}$$

where $n$ and $e$ are the number and energy densities respectively, $\epsilon_0 = 0$, $\epsilon_{1,2,3,4} = 1/2$, $\epsilon_{5,6,7,8} = 1$, $0 \le n \le 9$, and $0 \le e \le 6$, Also, $f_1 = f_2 = f_3 = f_4$ and $f_5 = f_6 = f_7 = f_8$.

There is a lower and an upper bound on $e$: the Fermi energy per site $e_F$ and the maximum energy per site $e_M$. For example, if $n = 4$, $e_F = 3/2$ that corresponds to a configuration where at every site there is a particle at rest and the other three have velocities along the axes. On the other hand, $e_M = 4$ corresponds to a state where every site is occupied by four particles with velocities along the diagonals. The existence of these bounds has a deep impact on the entropy of the model: it cannot be an always increasing function of $e$ since the system is

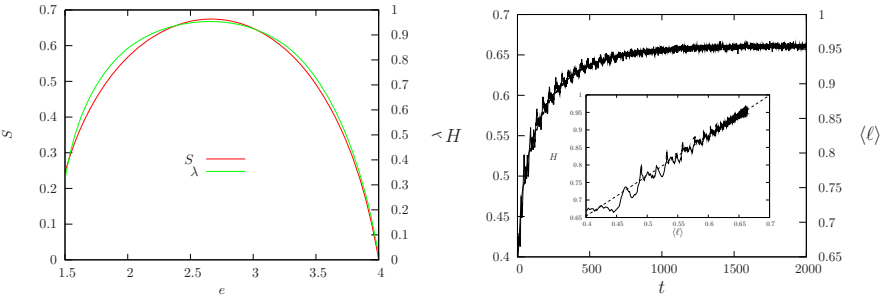



**Fig. 2.** (Left) Entropy per site $S$ (solid line) and LLE $\lambda$ (dashed line) for $n = 4$, simulations with $R = 40$ in a $40 \times 40$ lattice. The simulations are performed for different values of $e$ from $e_{\rm F}(n)$ to $e_{\rm M}(n)$. The LLE $\lambda$ is roughly proportional to $S$. (right) Boltzmann function $H$ (thick line) and Lyapunov expansion factor $\langle \ell \rangle$ (oscillating thin line) versus time $t$, simulations with $n_L = 7.2$, $e_L = 4.8$, $n_R = 1.8$, $e_R = 1.2$, and $R = 40$ on a $40 \times 40$ lattice. The inset shows that, disregarding the oscillations of $\langle \ell \rangle$, there is a linear relation between these quantities. The dashed line is the best fit $H = 1.15\langle \ell \rangle + 0.2$.

ordered both for small and large $e$. The three unknown distribution functions $f_0, f_1$ and $f_5$ satisfy the conservation of mass and energy expressed in Eqs. (10) and maximize the entropy which gives an additional constraint [13]

$$f_1^2(1 - f_0 - f_5) = f_0 f_5(1 - 2f_1). \tag{11}$$

## 5 Results and Discussion

In Fig. 2 (left) we show the entropy density $s = S/L$ and the LLE $\lambda$ as a function of $e$ for $n = 4$. The entropy density grows for $e_F \leq e \leq e^*$ and then decreases for $e^* \leq e \leq e_M$ with $e^* = 2n/3$. The largest Lyapunov exponent $\lambda$ shows the same behavior and is rougly proportional to the entropy $s$. There are small systematic deviations as shown in the inset of the Fig. 2 (left), where $\lambda$ is plotted as a funciton of $s$ for several values of $n$.

This proportionality can understood with a simple stochastic model, in which the mesoscopic dynamics (coarse-grained in space or averaged over quenched disorder) can be approximated by a Markov chain. We assume that for given macroscopic observables, there are $M$ compatible configurations, and that the probability of observing a configuration $x$ is $P(x) = 1/M$ (microcanonical distribution). The entropy is therefore $S = \log(M)$. The (chaotic) dynamics is approximated by a Markov matrix $W(x'|x)$ with $\alpha M$ equal entries per row and column with value $1/\alpha M$, and we assume that $W$ is irreducible (the asymptotic state is unique). The Kolmogorov-Sinai entropy $K$ per unit time is

$$K = -\sum_x P(x) \sum_{x'} W(x'|x) \log W(x'|x) = \log(\alpha M).$$

In the thermodynamic limit, the dominating term is $\alpha \log M$. On the other hand, $K$ is roughly proportional to the largest exponent $\lambda$ and if we assume that the form of the Lyapunov spectrum does not change dramatically when the macroscopic variables change, $K \simeq S + \log(\alpha)$.

The value of the LLE is related to the number of ones in the Jacobian matrix $J$ defined by Eq. (3). This Jacobian matrix contains the linearized effects of the streaming and collision operators. Streaming corresponds to a scrambling of the components of the tangent vector $\boldsymbol{v}(t)$ and therefore does not alter its norm. This is left to collisions, when more than one output configurations are possible. The number of equivalent output configurations in the collision table is small for the extreme values of number and energy densities, and larger for intermediate values. Similar considerations apply to the number of equivalent configurations for a given macroscopic distribution of density and velocities, and constitute the microscopic origin of the proportionality between statistical and dynamical quantities.

We now discuss an irreversible process where a square lattice is initially in an equilibrium state with the left and right sides having different number and energy densities $n_L$, $n_R$, $e_l$, and $e_R$. The system evolves toward equilibrium by means of dumped travelling waves. Boltzmann's $H$ function is defined in the same way as the entropy, Eq. (8), with a minus sign and distribution functions $f_k(\boldsymbol{r}, t)$ that are found as averages over $R$ replicas. In the same numerical simulation, the average Lyapunov expansion factor $\langle \ell \rangle = (1/R) \sum_{i=1}^{R} \log u^{(i)}$ is calculated. As shown in Fig. 2 (right), the two quantities exhibit similar behavior. The Lyapunov expansion factor exhibits more marked oscillations, indicating that this quantity is more sensible to the local variations in density. The inset of Fig. 2 (right) shows that, disregarding oscillations, $\langle \ell \rangle$ is linearly related to $H$ for all the relaxation phase.

The computation of $\langle \ell \rangle$ is performed using a set of tangent vectors, Eq. (4), and these vectors constitute a sort of local memory of the past state. In systems with local variations of density, as in our system in the presence of travelling waves, statistical quantities like $H$ depend on the instantaneous state of the system, while dynamical ones like $\langle \ell \rangle$ depend also on the variations of this state. This factor may be the origin of the different relation between statistical and dynamical quantities in equilibrium and during the relaxation phase.

The $D2Q9$ reversible LGCA model we have discussed exhibits hydrodynamical and thermodynamical behavior, and is therefore one of the more "realistic" models for which a simple relation can be established between its chaotic dynamical properties and its macroscopic behavior in irreversible processes and in equilibrium.

## Acknowledgments

The authors thank Stefano Ruffo for helpful discussions. Partial economic support from CONACyT projects U41347 and 2516, and from the Coordinación de la Investigación Científica de la UNAM is acknowledged.

## References


1. Lebowitz, J.: Phys.Today, p. 32 (September 1973); Zaslavsky, G. M.: Phys. Today, p. 39 (August 2004)
2. Dzugutov, M., Aurell, E., Vulpiani, A.: Phys. Rev. Lett. 81, 1762 (1998)
3. Gaspard, P., Nicolis, G.: Phys. Rev. Lett. 65, 1693 (1990); Gaspard, P., Dorfman, J. R.: Phys. Rev. E 52, 3525 (1995); Evans, D.J., Morris, G.P.: Statistical Mechanics of Nonequilibrium Liquids. Academic Press, London (1990); Hoover, W.G.: Computational Statistical Mechanics. Elsevier, Amsterdam (1991); Dorfman, J.R., van Beijeren, H.: Physica A 240, 12 (1997)
4. Falcioni, M., Palatella, L., Vulpiani, A.: Phys. Rev. E 71, 016118 (2005); Latora, V., Baranger, M.: Phys. Rev. Lett. 82, 520 (1999)
5. d'Humiéres, D., Lallemand, P., Frisch, U.: Europhys. Lett. 2, 291 (1986); Chopard, B., Droz, M.: Phys. Lett. 126A, 476 (1988); Chen, S., Lee, M., Zhao, K., Doolen, G.D.: Physica B 37, 42 (1989)
6. Hardy, J., Pomeau, Y., de Pazzis, O.: J. Math. Phys. 14, 1746, (1973); Frisch, U., Hasslacher, B., Pomeau, Y.: Phys. Rev. Lett. 56, 1505 (1986); Frisch, U., d'Humiéres, D., Hasslacher, B., Lallemand, P., Pomeau, Y., Rivet, J.P.: Complex Systems 1, 649 (1987). Reprinted in Lecture Notes on Turbulence, Herring, J.R., McWilliams, J.C.(eds.): p. 649. World Scientific, Singapore (1989) .
7. Bagnoli, F., Rechtman, R., Zanette, D.: Rev. Mex. Fís. 39, 763 (1993)
8. Bagnoli, F., Rechtman, R., Ruffo, S.: Phys. Lett. A, 172, 34 (1992)
9. Ott, E.: Chaos in Dynamical Systems. Cambridge University Press, Cambridge (1993)
10. Ehrenfest, P., Ehrenfest, T.: Enzyclopädie der Mathematischen Wissenschaften, vol. 4 (Teubner, Leipzig, 1911) English translation: Moravcik, M. J.: Conceptual Foundations of Statistical Approach in Mechanics. Cornell University Press, Ithaca (1959). Ernst, M. H., van Velzen, G. A.: J. Phys. A: Math. Gen. 22, 4611 (1989)
11. Bagnoli, F.: Int. J. Mod. Phys. 3, 307 (1992)
12. Bagnoli, F., Rechtman, R., Ruffo, S.: Lyapunov Exponents for Cellular Automata. In: López de Haro, M., Varea, C. (eds.) Lectures on Thermodynamics and Statistical Mechanics, p. 200. World Scientific, Singapore (1994)
13. Salcido, A., Rechtman, R.: Equilibrium Properties of a Cellular Automaton for Thermofluid Dynamics. In: Cordero, P., Nachtergaele, B. (eds.) Nonlinear Phenomena in Fluids, Solids and Other Complex Systems, p. 217. Elsevier, Amsterdam (1991)

# Analysis of 90/150 Two Predecessor Nongroup Cellular Automata*

Sung-Jin Cho[1], Un-Sook Choi[2], Han-Doo Kim[3],
Yoon-Hee Hwang[4], and Jin-Gyoung Kim[5]

[1] Division of Mathematical Sciences, Pukyong National University
Busan 608-737, Korea
sjcho@pknu.ac.kr
[2] Department of Multimedia Engineering, Tongmyong University
Busan 626-847, Korea
choies@tu.ac.kr
[3] School of Computer Aided Science, Institute of Basic Science
Inje University, Gimhae 621-749, Korea
mathkhd@inje.ac.kr
[4] Department of Information Security, Graduate School, Pukyong National University
Busan 608-737, Korea
yhhwang@pknu.ac.kr
[5] Division of Mathematical Sciences, Pukyong National University
Busan 608-737, Korea
kimjg@pknu.ac.kr

**Abstract.** Many researchers have been studied synthesis method of 90/150 group CA. However, there is a lack of researches for synthesis method of 90/150 nongroup CA. In this paper we propose an algorithm for finding 90/150 Two Predecessor Cellular Automata. Using the proposed algorithm we analyze 90/150 two predecessor CA. Especially, we analyze 90/150 TPSACA and TPMACA which are useful to study hashing. Also we analyze two types of 90/150 two predecessor CA. One is two predecessor CA for the minimal polynomial whose type is of the form $xp(x)$ which is useful to study two predecessor CA whose depth is 1. Another is two predecessor CA for the minimal polynomial whose type is of the form $x(x+1)p(x)$ which is useful to study pseudorandom number generation based on 90/150 two predecessor CA, where $p(x)$ is some primitive polynomial.

## 1 Introduction

Cellular Automata(abbreviately, CA) have been introduced by Von Neumann and Ulam as models of self-organizing and self-reproducing behaviors ([1], [2]). A CA is a discrete time dynamical system, which consists of a uniform array of memories called cells. The states of cells in the array are updated according to

* This work was supported by grant No. (R01-2006-000-10260-0) from the Basic Research Program of the Korea Science and Engineering Foundation.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 128–135, 2008.

a rule : the state of a cell at a given time depends only on its own state and the states of its nearby neighbors at the previous step. A CA is a necessity in many application areas such as test pattern generation, pseudorandom number generation, cryptography, error correcting codes and signature analysis([3] ∼ [10]). The analysis of the state-transition behavior of group CA was studied by many researchers ([6] ∼ [15]). Although the study of nonsingular linear machines has received considerable attention from researchers, the study of the class of machines with singular state-transition matrix has not received due attention. The state-transition matrix of group CA is nonsingular. But the state-transition matrix of nongroup CA is singular. Recently some interesting properties of nongroup CA have been employed in several applications([3], [5], [16] ∼ [20]). Especially, in ([3], [16]) they investigated a special class of nongroup CA denoted as D1*CA. Based on this investigation, D1*CA has been proposed as an ideal test machine which can be efficiently embedded in a finite state machine to enhance the testability of the synthesis design. Also in [5] they investigated 90/150 two predecessor CA whose minimal polynomial is of the form $x(x+1)p(x)$, where $p(x)$ is primitive. The use of these CA configurations simplifies the hardware implementations and avoids several precomputations to obtain the matrix associated to a quadratic function. Thus they studied several cases for different CA lengths. But they didn't show that there exists an $n$-cell 90/150 two predecessor CA for each $n \geq 6$. In this paper, using our algorithm for finding 90/150 two predecessor CA, we analyze 90/150 two predecessor CA. Especially, we analyze $n$-cell 90/150 TPSACA (whose minimal polynomial is $x^n$) and $n$-cell TPMACA (where minimal polynomial is $x^{n-1}(x+1)$) which are useful to study hashing [16]. Also we analyze two types of 90/150 two predecessor CA. One is two predecessor CA for the minimal polynomial whose type is of the form $xp(x)$ which is useful to study 90/150 two predecessor CA like D1*CA [3] whose depth is 1. The proposed $n$-cell 90/150 two predecessor CA has a maximum-length cycle whose length is $2^{n-1}-1$ which is larger than that of D1*CA. Another is 90/150 two predecessor CA for the minimal polynomial whose type is of the form $x(x+1)p(x)$ which is useful to study pseudorandom number generation based on 90/150 nongroup CA [5], where $p(x)$ is primitive.

## 2 CA Preliminaries

A CA consists of a number of interconnected cells arranged spatially in a regular manner [2], where the state-transitions of each cell depends on the states of its neighbors. If the next-state function of a cell is expressed in the form of a truth table, then the decimal equivalent of the output is conventionally called the rule number for the cell [2].

| Neighborhood state : | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 | |
|---|---|---|---|---|---|---|---|---|---|
| Next state: | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | (rule 90) |
| Next state: | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | (rule 150) |

**Definition 2.1.** ([16], [18] ∼ [20])

i) *Group CA*: A CA is called a *group CA* if all the states in its state-transition diagram lie on cycles, otherwise it is referred to as a *non-group CA*.

ii) *Attractor*: A state having a self-loop is referred to as an *attractor*. An attractor can be viewed as a cyclic state with unit cycle length.

iii) *Depth*: The maximum number of state transitions required to reach the nearest cyclic state from any non-reachable state in the CA state-transition diagram is defined as the *depth* of the non-group CA.

iv) *Multiple-attractor CA(MACA)*: The non-group CA for which the state-transition diagram consists of a set of disjoint components forming (inverted) tree-like structures rooted at attractors are referred to as *multiple-attractor CA*. Single attractor CA(SACA) is a MACA whose the number of attractors is just one.

v) *TPMACA*: *TPMACA* is a MACA such that every reachable state in the state-transition diagram has only two predecessors. *TPSACA* is a SACA such that every reachable state in the state-transition diagram has only two predecessors. The minimal polynomial of an $n$-cell TPSACA is $x^n$.

## 3 An Algorithm for Finding 90/150 Two Predecessor CA

In this section we introduce an algorithm for finding 90/150 two predecessor CA.

Let $U$ be the following upper triangular matrix.

$$U = \begin{pmatrix} 1 & a_1 & * & \cdots & * & * & * \\ 0 & 1 & a_2 & \cdots & * & * & * \\ 0 & 0 & 1 & \cdots & * & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_{n-2} & * \\ 0 & 0 & 0 & \cdots & 0 & 1 & a_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

And let $T$ be the following 90/150 tridiagonal matrix.

$$T = \begin{pmatrix} d_1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & d_2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & d_3 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & d_{n-1} & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & d_n \end{pmatrix}$$

(Hereafter we write $T$ by $T =< d_1, d_2, \cdots, d_n >$, where $d_i \in \{0, 1\}$.)

Moreover, let $f(x) = x^n + c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1x + c_0$, where $c_i \in GF(2)$. Then the following $n \times n$ matrix $C$ is said to be the *companion matrix* of $f(x)$.

$$C = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & c_0 \\ 1 & 0 & 0 & \cdots & 0 & c_1 \\ 0 & 1 & 0 & \cdots & 0 & c_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & c_{n-1} \end{pmatrix}$$

**Definition 3.1.** ([21]) For a given $n$-vector $x$ and $n \times n$ matrix $M$, let

$$K(M, x) = (x; Mx; M^2x; \cdots; M^{n-1}x)$$

We call $K(M, x)$ the Krylov matrix and $x$ is called a *seed vector*.

**Theorem 3.2.** ([15]) Let $T =< d_1, d_2, \cdots, d_n >$ and $C$ be the companion matrix of the characteristic polynomial of $T$. Let $U$ be the upper triangular matrix as the above form satisfying $TU = UC$. Then we obtain the following equation:

$$\begin{cases} d_1 = a_1 \\ d_2 = a_1 \oplus a_2 \\ d_3 = a_2 \oplus a_3 \\ \quad \vdots \\ d_{n-1} = a_{n-2} \oplus a_{n-1} \\ d_n = a_{n-1} \oplus c_{n-1} \end{cases} \tag{3.1}$$

Let $f(x)$ be a polynomial corresponding to a 90/150 two predecessor CA, then $f(x)$ is called a 90/150 two predecessor CA polynomial.

**Theorem 3.3.** ([15]) Let $B$ be the $n \times n$ matrix obtained by reducing the $n$ polynomials

$$x^{i-1} + x^{2i-1} + x^{2i} \pmod{f(x)} \ (i = 1, 2, \cdots, n) \tag{3.2}$$

where $f(x)$ is a reducible polynomial. And let the set $\{v | Bv = (0, \cdots, 0, 1)^t\}$ be nonempty set, then the elements of $\{v | Bv = (0, \cdots, 0, 1)^t\}$ become seed vectors for the Krylov matrix, where $A^t$ is the transpose of $A$.

**Theorem 3.4.** Let the Krylov matrix in Theorem 3.3 have an $LU$ factorization. Then $f(x)$ in Theorem 3.3 is a 90/150 two predecessor CA polynomial.

The following algorithm is an algorithm for finding a 90/150 two predecessor CA for the given reducible polynomial.

**Algorithm.** SynthesisOf90/150TPNCA
Input : Polynomial $f(x)$
Output : 90/150 two predecessor CA
Step 1 : Make the matrix $B$ from (3.2).
Step 2 : Solve the equation $Bv = (0, \cdots, 0, 1)^t$. If there doesn't exist a solution, then STOP.
Step 3 : Construct a Krylov matrix $H = K(C^t, v)$ by the seed vector $v$ which is a solution of the equation in Step 2.

Step 4 : If $H$ doesn't have an $LU$ factorization, then STOP.
Step 5 : Compute the LU factorization $H = LU$.
Step 6 : Compute 90/50 two predecessor CA for $f(x)$ by the matrix $U$ using (3.1).

## 4 Analysis of 90/150 Two Predecessor CA

In this section we analyze 90/150 two predecessor CA.

**Theorem 4.1.** Let $\Delta_{2m}$ be the characteristic polynomial of

$$< d_1, d_2, \cdots, d_m, d_m, \cdots, d_2, d_1 >$$

Then the following equation hold.

$$\Delta_{i+1}\Delta_{2m-i-1} + \Delta_i\Delta_{2m-i-2} = \Delta_{i+2}\Delta_{2m-i-2} + \Delta_{i+1}\Delta_{2m-i-3},$$

where $i = 1, \cdots, 2m-2$, $\Delta_{-1} = 0$ and $\Delta_0 = 1$.

**Theorem 4.2.** Let $\Delta_{2m}$ be the characteristic polynomial of

$$< d_1, d_2, \cdots, d_m, d_m, \cdots, d_2, d_1 >$$

and let $f(x)$ be the characteristic polynomial of $< d_1, d_2, \cdots, d_m \oplus 1 >$. Then the following holds:

$$\Delta_{2m} = \{f(x)\}^2$$

**Theorem 4.3.** Let $\mathbf{C_S}^k =< d_1, \cdots, d_k >$ be a $k$-cell 90/150 TPSACA. Then the following hold:
(i) $\mathbf{C_S}^{2k} =< d_1, \cdots, d_k \oplus 1, d_k \oplus 1, \cdots, d_2, d_1 >$ is a $2k$-cell TPSACA with the minimal polynomial $x^{2k}$.
(ii) $\mathbf{C_S}^{2k+1} =< d_1, \cdots, d_k, 0, d_k, \cdots, d_1 >$ is a $(2k+1)$-cell TPSACA with the minimal polynomial $x^{2k+1}$.

**Theorem 4.4.** Let $N(T_m) = \{(a_1, a_2, \cdots, a_m)^t | a_1, a_2, \cdots, a_m \in \{0,1\}\}(:= [(a_1, \cdots, a_m)^t])$ be the null space of the state-transition matrix $T_n$ of an $n$-cell 90/150 TPSACA. Then the following hold:
(i) If $n = 2m (m \in \mathbf{N})$ and $N(T_m) = \{(a_1, a_2, \cdots, a_m)^t | a_1, a_2, \cdots, a_m \in \{0,1\}\}(:= [(a_1, a_2, \cdots, a_m)^t])$, then $N(T_n) = [(a_1, a_2, \cdots, a_m, a_m, \cdots, a_2, a_1)^t]$.
(ii) If $n = 2m+1 (m \in \mathbf{N})$ and $N(T_m) = [(a_1, a_2, \cdots, a_m)^t]$, then
$N(T_n) = [(a_1, a_2, \cdots, a_m, 0, a_m, \cdots, a_2, a_1)^t]$.

**Example 4.5.** Since $< 0, 0, 0 >$ is a 3-cell 90/150 TPSACA, $< 0, 0, 1, 1, 0, 0 >$ is a 6-cell 90/150 TPSACA and $< 0, 0, 0, 0, 0, 0, 0 >$ is a 7-cell 90/150 TPSACA.

**Theorem 4.6.** Let $\mathbf{C}_S^n =< d_1, \cdots, d_n >$ be an $n$-cell 90/150 TPSACA. Then $\mathbf{C}_M^{2n+1} =< d_1, \cdots, d_n, 1, d_n, \cdots, d_1 >$ is a $(2n+1)$-cell 90/150 TPMACA with the minimal polynomial $x^{2n}(x+1)$.

**Table 1.** TPSACA and TPMACA

| n | TPSACA | $N(T_S)$ | TPMACA | $N(T_M)$ | $N(T_M \oplus I)$ |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 |
| 2 | 11 | 11 | | | |
| 3 | 000 | 101 | 010 | 101 | 111 |
| 4 | 1001 | 1111 | | | |
| 5 | 11011 | 11011 | 11111 | 11011 | 10101 |
| 6 | 001100 | 101101 | | | |
| 7 | 0000000 | 1010101 | 0001000 | 1010101 | 1101011 |
| 8 | 10000001 | 11111111 | | | |
| 9 | 100101001 | 111101111 | 100111001 | 111101111 | 101111101 |
| 10 | 1101001011 | 1101111011 | | | |
| 11 | 11011011011 | 11011011011 | 11011111011 | 11011011011 | 10111111101 |
| 12 | 001101101100 | 101101101101 | | | |
| 13 | 0011000001100 | 1011010101101 | 0011001001100 | 1011010101101 | 1101011101011 |

**Remark.** For the case $n$ is even, there does not exist $n$-cell 90/150 TPMACA whose minimal polynomial is $f(x) = x^n + x^{n-1}$.

**Theorem 4.7.** Let $N(T_m) = [(a_1, \cdots, a_m)^t]$ be the null space of the state-transition matrix $T_m$ of an $m$-cell 90/150 TPSACA $\mathbf{C_S^m}$. Then the null space of the $(2m+1)$-cell 90/150 TPMACA $\mathbf{C_M^{2m+1}}$ derived from $\mathbf{C_S^m}$ is

$$N(T_{2m+1}) = [(a_1, \cdots, a_{m-1}, a_m, 0, a_m, a_{m-1}, \cdots, a_1)^t])$$

In Table 1, $N(T_S)$ means the null space of $n$-cell 90/150 TPSACA and $N(T_M)$ means the null space of $n$-cell 90/150 TPMACA. Also $N(T_M \oplus I)$ means the set of all attractors for each $n$-cell 90/150 TPMACA. 101 means $[(1, 0, 1)^t]$.

Chattopadhyay[22] presented an algorithm for finding MACA with all linear rules $(60, 90, 102, 150, 170, 204, 240)$, but in this paper we present a method which synthesize TPMACA using rule 90 and rule 150.

**Theorem 4.8.** Let $f(x) = xp(x)$, where $p(x)$ is a polynomial of degree $n-1$. Then there exists a primitive polynomial $p(x)$ such that $f(x)$ is the minimal polynomial corresponding to the $n$-cell 90/150 two predecessor CA.

**Theorem 4.9.** Let $f(x) = x(x+1)p(x)$, where $p(x)$ is a polynomial of degree $n-2 (n \geq 6)$. Then there exists a primitive polynomial $p(x)$ such that $f(x)$ is the minimal polynomial corresponding to the $n$-cell 90/150 two predecessor CA.

Table 2 shows that there exists an $n$-cell 90/150 two predecessor CA for the 90/150 two predecessor CA polynomial of the form $xp(x)$ ($p(x)$ is some primitive polynomial) for each $n \geq 4$. Also Table 3 shows that there exists an $n$-cell 90/150 TPMACA for the 90/150 TPMACA polynomial of the form $x(x+1)p(x)$ ($p(x)$ is some primitive polynomial) for each $n \geq 6$.

**Table 2.** 90/150 CA for $xp(x)$
(In this table, 320 stands for the polynomial $x^3 + x^2 + 1$.)

| $n$ | $p(x)$ | CA Configuration | $n$ | $p(x)$ | CA Configuration |
|---|---|---|---|---|---|
| 4 | 320 | 0111 | 13 | 12,10,9,8,6,2,0 | 1011101001000 |
| 5 | 430 | 00010 | 14 | 13,8,5,3,0 | 01100110101000 |
| 6 | 520 | 001001 | 15 | 14,11,9,7,0 | 100010001010000 |
| 7 | 65320 | 0011111 | 16 | 15,12,4,3,0 | 1000010010101010 |
| 8 | 740 | 00000011 | 17 | 16,15,12,10,0 | 11011110100010001 |
| 9 | 86520 | 000010001 | 18 | 17,3,0 | 100011101011110001 |
| 10 | 95320 | 0000100100 | 19 | 18,7,0 | 0001110111000101000 |
| 11 | 10,3,0 | 01011111110 | 20 | 19,10,9,3,0 | 01010100110000000010 |
| 12 | 11,2,0 | 011101000110 | 21 | 20,3,0 | 001001010110100100100 |

**Table 3.** 90/150 CA for $x(x+1)p(x)$
(In this table, 210 stands for the polynomial $x^2 + x + 1$.)

| $n$ | $p(x)$ | CA Configuration | $n$ | $p(x)$ | CA Configuration |
|---|---|---|---|---|---|
| 4 | 210 | 1100 | 13 | 11,9,7,5,2,1,0 | 1111101110111 |
| 6 | 410 | 100110 | 14 | 12,10,2,1,0 | 01000110010010 |
| 7 | 53210 | 0100101 | 15 | 13,12,10,5,2,1,0 | 000101010001101 |
| 8 | 610 | 00001110 | 16 | 14,12,10,1,0 | 1100110011010011 |
| 9 | 73210 | 010000000 | 17 | 15,12,9,1,0 | 00000111110100111 |
| 10 | 85310 | 0001001001 | 18 | 16,14,12,1,0 | 101100100110001101 |
| 11 | 95410 | 10000110011 | 19 | 17,13,12,1,0 | 0100101010011011100 |
| 12 | 10,7,6,5,2,1,0 | 001111010101 | 20 | 18,17,12,10,9,1,0 | 00111100100000111000 |

## 5 Conclusion

In this paper we proposed an algorithm for finding 90/150 two predecessor CA. Using the proposed algorithm we analyzed 90/150 two predecessor CA. Especially, we analyzed 90/150 TPSACA and 90/150 TPMACA which are useful to study hashing. Also we analyzed two types of 90/150 two predecessor CA. One is two predecessor CA for the minimal polynomial whose type is of the form $xp(x)$. Another is two predecessor CA for the minimal polynomial whose type is of the form $x(x+1)p(x)$.

## References

1. Von Neumann, J.: The theory of self-reproducing automata. In: Burks, A.W. (ed.). University of Illinois Press, Urban (1966)
2. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601–644 (1983)
3. Chakraborty, S., Chowdhury, D.R., Chaudhuri, P.P.: Theory and application of nongroup cellular automata for synthesis of easily testable finite state machines. IEEE Trans. Computers 45(7), 769–781 (1996)
4. Chattopadhyay, S., Chaudhuri, P.P.: Theory and application of nongroup cellular automata in pattern classification, IEEE Trans. Computers, communicated

5. De la Guia Martinez, D., Peinado Dominguez, A.: Pseudorandom number generation based on nongroup cellular automata. In: Security Technology, 1999, Proceedings, IEEE 33rd Annual 1999 International Carnahan Conference, vol. 45, pp. 370–376 (1999)
6. Das, A.K., Chaudhuri, P.P.: Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation. IEEE Trans. Comput. 42, 340–352 (1993)
7. Serra, M., Slater, T., Muzio, J.C., Miller, D.M.: The analysis of one dimensional linear cellular automata and their aliasing properties. IEEE Trans Computer-Aided Design 9, 767–778 (1990)
8. Hortensius, P.D., McLeod, R.D., Card, H.C.: Parallel random number generation for VLSI systems using cellular automata. IEEE Trans. Computers 38, 1466–1473 (1989)
9. Hortensius, P.D., McLeod, R.D., Miller, D.M., Card, H.C.: Cellular automata based pseudorandom number generations for built-in self test. IEEE Trans. on CAD of Integrated Circuits ans Systems 8, 842–859 (1989)
10. Tsalides, P., York, T.A., Thanailakis, A.: Pseudorandom number generators for systems based on linear cellular automata. IEE Proc(Part E) Computers Digital Techniques 138, 241–249 (1991)
11. Pries, W., Thanailakis, A., Card, H.C.: Group properties of cellular automata and VLSI applications. IEEE Trans. Computers 35, 1013–1024 (1986)
12. Nandi, S., Kar, B.K., Chaudhuri, P.P.: Theory and application of cellular automata in cryptography. IEEE Trans. Computers 43, 1346–1357 (1994)
13. Cattell, K., Muzio, J.C.: Synthesis of one-dimensional linear hybrid cellular automata. IEEE Trans. on Computer Aided Design of Circuits and Systems 15-3, 325–335 (1996)
14. Cho, S.J., Choi, U.S., Hwang, Y.H., Pyo, Y.S., Kim, H.D., Kim, K.S., Heo, S.H.: Computing phase shifts of maximum-length 90/150 cellular automata sequences. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 31–39. Springer, Heidelberg (2004)
15. Cho, S.J., Choi, U.S., Kim, H.D., Hwang, Y.H., Kim, J.G., Heo, S.H.: New synthesis of one-dimensional 90/150 linear hybrid group cellular automata. IEEE Trans. Comput-Aided Des. Integr. Circuits Syst. 26(9), 1720–1724 (2007)
16. Chaudhuri, P.P., Chowdhury, D.R., Nandy, S., Chattopadhyay, C.: Additive cellular automata theory and applications, vol. 1. IEEE Computer Society Press, California (1997)
17. Cho, S.J., Choi, U.S., Hwang, Y.H., Kim, H.D.: Analysis of hybrid group cellular automata. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 222–231. Springer, Heidelberg (2006)
18. Cho, S.J., Choi, U.S., Kim, H.D.: Analysis of complemented CA derived from a linear TPMACA. Computers Math. Applic. 45, 689–698 (2003)
19. Cho, S.J., Choi, U.S., Kim, H.D.: Behavior of complemented CA whose complement vector is acyclic in a linear TPMACA. Math. Comput. Modelling 36, 979–986 (2002)
20. Cho, S.J., Choi, U.S., Hwang, Y.H., Kim, H.D., Choi, H.H.: Behaviors of single attractor cellular automata over Galois Field $GF(2^p)$. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 232–237. Springer, Heidelberg (2006)
21. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
22. Chattopadhyay, S.: Some studies on theory and application of additive cellular automata, PhD thesis, I.I.T., Kharagpur, India (1995)

# Analysis of Linear Group $GF(2^p)$ Cellular Automata*

Un-Sook Choi[1], Sung-Jin Cho[2,**], Yoon-Hee Hwang[3], and Han-Doo Kim[4]

[1] Department of Multimedia Engineering, Tongmyong University
Busan 626-847, Korea
choies@tu.ac.kr
[2] Division of Mathematical Sciences, Pukyong National University
Busan 608-737, Korea
sjcho@pknu.ac.kr
[3] Department of Information Security, Graduate School, Pukyong National University, Busan 608-737, Korea
yhhwang@pknu.ac.kr
[4] School of Computer Aided Science, Institute of Basic Science
Inje University, Gimhae 621-749, Korea
mathkhd@inje.ac.kr

**Abstract.** Cellular Automata(CA) has been used as modeling and computing paradigm for a long time. And CA has been used to model many physical systems. While studying the models of such systems, it is seen that as the complexity of the physical system increase, the CA based model becomes very complex and difficult to track analytically. Also such models fail to recognize the presence of inherent hierarchical nature of a physical system. In this paper we give the characterization of linear group $GF(2^p)$ CA. Especially we analyze the relationship between characteristic polynomial and transition rule of linear group $GF(2^p)$ CA.

## 1 Introduction

Cellular Automata(CA) was first introduced by Von Neumann [1] for modeling biological self-reproduction. Wolfram [2] pioneered the investigation of CA as mathematical models for self-organizing statistical systems and suggested the use of a simple two-state, three-neighborhood CA with cells arranged linearly in one dimension. Das et al. ([3] $\sim$ [5]) developed a matrix algebraic tool capable of characterizing CA. Cho et al. [6] proposed a new method for the synthesis of one-dimensional 90/150 linear hybrid group CA for CA-polynomials. And Cho et al. ([3] $\sim$ [9]) and many researchers ([6], [10] $\sim$ [12]) analyzed CA to study hash function, data storage, cryptography and so on.

* This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD)(KRF-2006-331-D00458).

** Author to whom all correspondence should be addressed.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 136–143, 2008.

CA has been used as modeling and computing paradigm for a long time. And CA has been used to model many physical systems. While studying the models of such systems, it is seen that as the complexity of the physical system increase, the CA based model becomes very complex and difficult to track analytically. Also such models fail to recognize the presence of inherent hierarchical nature of a physical system.

To overcome these problems Sikdar et al. [12] and Cho et al. [13] studied $GF(2^p)$ CA.

In this paper, by using the results in [6] we give the characterization of linear group $GF(2^p)$ CA. Especially we analyze the relationship between characteristic polynomial and transition rule of linear group $GF(2^p)$ CA.

## 2 Linear $GF(2^p)$ CA Preliminaries

A $GF(2^p)$ CA can be viewed as an extension of $GF(2)$ CA. It consists of an array of cells, spartially interconnected in a regular manner, each cell being capable of storing an element of $GF(2^p)$.

Under three neighborhood restriction, the next state of the $i$th cell is given by a function of the weighted combination of the present states of the $(i-1)$th, $i$th and $(i+1)$th cells, the weights being elements of $GF(2^p)$. Thus if $q_i(t)$ is the state of the $i$th cell at the $t$th instant, then

$$q_i(t+1) = \phi(w_{i-1}q_{i-1}(t), w_iq_i(t), w_{i+1}q_{i+1}(t))$$

where $\phi$ denotes the local transition function of the $i$th cell and $w_{i-1}$, $w_i$ and $w_{i+1} \in GF(2^p)$ specify the weights of interconnections.

The transition rule for a three neighborhood $GF(2^p)$ CA cell is represented by a vector of length 3, $< w_{i-1}, w_i, w_{i+1} >$. Here $w_{i-1}$ indicates the weight of dependence of the cell on its left neighborhood, while $w_i$ and $w_{i+1}$ indicate the weighted dependency on itself and its right neighborhood respectively. If the same transition rule vector is applied to all the cells of a $GF(2^p)$ CA, the CA is called an *uniform* $GF(2^p)$ CA, otherwise it is called a *hybrid* $GF(2^p)$ CA.

An $n$ cell $GF(2^p)$ CA can be characterized by an $n \times n$ state transition matrix $T = (t_{ij})$ as follows:

$$t_{ij} = \begin{cases} w_{ij}, & \text{if the next state of the } i\text{th cell depends on the present} \\ & \text{state of the } j\text{th cell by a weight } w_{ij} \in GF(2^p), \\ 0, & \text{otherwise.} \end{cases}$$

For example, let the state transition matrix of a 3-cell $GF(2^2)$ CA be the following:

$$T = \begin{pmatrix} 0 & \alpha^2 & 0 \\ \alpha^2 & \alpha & \alpha^2 \\ 0 & \alpha^2 & 1 \end{pmatrix}$$

where $\alpha$ is a generator of $GF(2^2) = \{0, 1, \alpha, \alpha^2\}$. $\alpha$ is a solution of the generator polynomial $g(x) = x^2 + x + 1$ and the generating matrix $M$ is as the following form:

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

The next state $X'$ of the present state $X$ of an $n$-cell $GF(2^p)$ CA with state transition matrix $T$ is given by $X' = TX$. Here $T$ is an $n \times n$ matrix and $X$ and $X'$ are $n \times 1$ vectors.

For the vectors $X$ and $X'$ we need a vector representation of each $\alpha^i$. Each of the vectors $X$ and $X'$ consists of a string of elements $\alpha^i \in GF(2^p)$. Therefore we need a binary representation of each of these $\alpha^i$. The last column vector of $M^i$ is used as the vector representation of $\alpha^i$.

The addition and multiplication operations follow the additive and multiplicative rules of the underlying $GF(2^2)$ as noted in Table 1.

**Table 1.** Multiplication and addition over $GF(2^2)$

| $\times$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 |
| 2 | 0 | 2 | 3 | 1 |
| 3 | 0 | 3 | 1 | 2 |

| $+$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 1 | 0 | 3 | 2 |
| 2 | 2 | 3 | 0 | 1 |
| 3 | 3 | 2 | 1 | 0 |

In the above example $M^i$ $(i = 2, 3)$ and $\alpha^i$ $(i = 1, 2, 3)$ are as the following form:

$$M^2 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad M^3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\alpha =< 10 >= 2, \alpha^2 =< 11 >= 3, \alpha^3 =< 01 >= 1$$

The *characteristic polynomial* $\Delta(x)$ of the state transition matrix $T$ of a $GF(2^p)$ CA is $\Delta(x) = |T + xI|$. In the above example the characteristic polynomial of $T$ is $\Delta(x) = x^3 + 2x^2 + 3x + 3$. This polynomial is a primitive polynomial on $GF(2^2)$ and thus its period is 63.

Let $\mathbf{C}$ be a $GF(2^p)$ CA whose state transition matrix is $T$. If $\det(T) \neq 0$, then $\mathbf{C}$ is called a *group* $GF(2^p)$ CA, otherwise it is called a *nongroup* $GF(2^p)$ CA.

## 3 Characterization of Linear Group $GF(2^p)$ CA

In the state transition matrix $T_n$ of an $n$-cell $GF(2^p)$ CA $\mathbf{C}$ let the weight of the right state and the weight of the left state be the same. Then this $GF(2^p)$ CA is the natural extension of 90/150 $GF(2)$ CA. Therefore the $T_n$ is as the following:

$$T_n = \begin{pmatrix} d_1 & i & 0 & \cdots & 0 & 0 \\ i & d_2 & i & \cdots & 0 & 0 \\ 0 & i & d_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & i & d_n \end{pmatrix}$$

where $i \in \{0, 1, 2, \cdots, 2^p - 1\}$ is the weight.

**Remark.** We denote $T_n$ by $T_n =< d_1, d_2, \cdots, d_j, \cdots, d_n >_i$, where $d_j \in GF(2^p)$.

The following lemma can be proved by mathematical induction.

**Lemma 3.1.** Let $T_n =< d_1, d_2, \cdots, d_n >_i$. Then we obtain the following equation.

$$|T_{-1}| = 0,\ |T_0| = 1,\ |T_n| = d_n|T_{n-1}| + i^2|T_{n-2}|,$$

where $|T|$ is the determinant of $T$.

Following two theorems give conditions for the uniform group $GF(2^p)$ CA.

**Theorem 3.2.** Let $\mathbf{C}$ be an $n$-cell uniform $GF(2^p)$ CA with the state transition matrix $T_n =< 0, \cdots, 0 >_i$. If $n$ is even (resp. odd), then $\mathbf{C}$ is a group(resp. nongroup) $GF(2^p)$ CA.

*Proof.* Since $d_j = d = 0$ for $j = 1, 2, \cdots, n$, $|T_n| = i^2|T_{n-2}|$ by Lemma 3.1. Since $|T_2| = i^2|T_0| = i^2, |T_n| = i^2|T_{n-2}| = i^n$ where $n$ is even. Therefore for even $n$, $\mathbf{C}$ is a group $GF(2^p)$ CA. Since $|T_1| = i^2|T_{-1}| = 0, |T_n| = i^2|T_{n-2}| = 0$ where $n$ is odd. Therefore for odd $n$, $\mathbf{C}$ is a nongroup $GF(2^p)$ CA.

**Theorem 3.3.** Let $\mathbf{C}$ be an $n$-cell uniform $GF(2^p)$ CA with the state transition matrix $T_n =< i, \cdots, i >_i$. If $n(mod 3) \neq 2$ (resp. $n(mod 3) = 2$), then $\mathbf{C}$ is a group(resp. nongroup) $GF(2^p)$ CA.

*Proof.* By Lemma 3.1 $|T_0| = 1, |T_1| = i$ and $|T_2| = 0$. Since

$$\begin{aligned} |T_{3k+2}| &= i|T_{3k+1}| + i^2|T_{3k}| \\ &= i(i|T_{3k}| + i^2|T_{3k-1}|) + i^2|T_{3k}| \\ &= i^3|T_{3k-1}| = i^3|T_{3(k-1)+2}|, \end{aligned}$$

$$|T_n| = \begin{cases} (i^3)^k|T_0|, & n = 3k \\ (i^3)^k|T_1|, & n = 3k+1 \\ (i^3)^k|T_2|, & n = 3k+2 \end{cases}$$

Hence $\mathbf{C}$ is a group $GF(2^p)$ CA for $n(mod 3) \neq 2$.

Following two theorems give conditions for the hybrid group $GF(2^p)$ CA.

**Theorem 3.4.** Let $\mathbf{C}$ be an $n$-cell hybrid $GF(2^p)$ CA with the state transition matrix $T_n =< 0, d, 0, d, \cdots >_i$. If $n$ is even(resp. odd), then $\mathbf{C}$ is a group(resp. nongroup) $GF(2^p)$ CA.

*Proof.* i) $n = 2m + 1$ :
Since $|T_{2m+1}| = 0 \cdot |T_{2m}| + i^2|T_{2m-1}| = i^2|T_{2(m-1)+1}|$ by Lemma 3.1, $|T_{2m+1}| = (i^2)^m|T_1| = 0$.
ii) $n = 2m$ :
Since $|T_{2m}| = d \cdot |T_{2m-1}| + i^2|T_{2m-2}|$ by Lemma 3.1 and $|T_{2m-1}| = 0$ by i), $|T_{2m}| = i^2|T_{2m-2}|$. Therefore $|T_{2m}| = (i^2)^m|T_0| = i^{2m}$. Hence $\mathbf{C}$ is a group(resp. nongroup) $GF(2^p)$ CA for even(resp. odd) $n$.

**Theorem 3.5.** Let $\mathbf{C}$ be an $n$-cell hybrid $GF(2^p)$ CA with the state transition matrix $T_n =< d, 0, d, 0, \cdots >_i$. If $n(mod4) \neq 3$ (resp. $n(mod4) = 3$), then $\mathbf{C}$ is a group(resp. nongroup) $GF(2^p)$ CA.

*Proof.* Sine $|T_0| = 1$ and $|T_3| = 0$ by Lemma 3.1, we obtain the following equations.

$$\begin{aligned} |T_{4k+3}| &= d \cdot |T_{4k+2}| + i^2|T_{4k+1}| \\ &= d \cdot \{0 \cdot |T_{4k+1}| + i^2|T_{4k}|\} + i^2\{d \cdot |T_{4k}| + i^2|T_{4k-1}|\} \\ &= i^4|T_{4k-1}| = i^4|T_{4(k-1)+3}| \\ &= (i^4)^k|T_3| = 0 \end{aligned} \tag{3.1}$$

$$\begin{aligned} |T_{2m}| &= 0 \cdot |T_{2m-1}| + i^2|T_{2m-2}| = i^2|T_{2(m-1)}| \\ &= (i^2)^m|T_0| = i^{2m} \end{aligned} \tag{3.2}$$

By (3.1) and (3.2),

$$\begin{aligned} |T_{4k+1}| &= d \cdot |T_{4k}| + i^2|T_{4k-1}| \\ &= d \cdot i^{4k} + i^2|T_{4(k-1)+3}| \\ &= d \cdot i^{4k}. \end{aligned}$$

This completes the proof.

## 4 The Relationship between Characteristic Polynomial and Transition Rule of Linear $GF(2^p)$ CA

In this section we analyze the relationship between characteristic polynomial and transition rule of linear $GF(2^p)$ CA. The following theorem can be proved by mathematical induction.

**Theorem 4.1.** Let $\mathbf{C}$ be an $n$-cell $GF(2^p)$ CA with the state transition matrix $T =< d_1, d_2, \cdots, d_n >_i$ and with the characteristic polynomial $\Delta_n$. Then we obtain the following equation.

$$\begin{aligned} \Delta_{-1} &= 0 \\ \Delta_0 &= 1 \\ \Delta_k &= (x + d_k)\Delta_{k-1} + i^2\Delta_{k-2} \end{aligned} \tag{4.1}$$

where $\Delta_k$ is the characteristic polynomial of $< d_1, d_2, \cdots, d_k >_i, k = 1, 2, \cdots, n$.

Theorem 4.1 provides an efficient algorithm to compute the characteristic polynomial of a $GF(2^p)$ CA. Initially, $\Delta_{-1}$ and $\Delta_0$ are set to zero and one, respectively. Equation (4.1) is applied to obtain $\Delta_1$. It is then reapplied to $\Delta_0$ and $\Delta_1$ to calculate $\Delta_2$. Continuing, the polynomials $\Delta_3, \Delta_4, \cdots, \Delta_n$ are computed. Since $\Delta_n$ is the characteristic polynomial of $T$, the calculation of the characteristic polynomial is completed.

The following is an example of the calculation of the characteristic polynomial of the $GF(2^p)$ CA with the rule vector $< 0, 1, 2, 1 >_2$.

**Example 4.2.** Let $\mathbf{C}$ be a $GF(2^2)$ CA with the rule vector $< 0, 1, 2, 1 >_2$.

$$\begin{aligned}
\Delta_{-1} &= 0 \\
\Delta_0 &= 1 \\
\Delta_1 &= (x + d_1)\Delta_0 + 2^2\Delta_{-1} \\
&= (x + 0) \cdot 1 + 2^2 \cdot 0 \\
&= x \\
\Delta_2 &= (x + d_2)\Delta_1 + 2^2\Delta_0 \\
&= (x + 1) \cdot x + 2^2 \cdot 1 \\
&= x^2 + x + 3 \\
\Delta_3 &= (x + d_3)\Delta_2 + 2^2\Delta_1 \\
&= (x + 2) \cdot (x^2 + x + 3) + 2^2 \cdot x \\
&= x^3 + 3x^2 + 2x + 1 \\
\Delta_4 &= (x + d_4)\Delta_3 + 3^2\Delta_2 \\
&= (x + 1) \cdot (x^3 + 3x^2 + 2x + 1) + 2^2 \cdot (x^2 + x + 3) \\
&= x^4 + 2x^3 + 2x^2 + 3
\end{aligned} \tag{4.2}$$

This recurrence relation forms the basis for the synthesis of $GF(2^p)$ CA. Initially, we show how recurrence (4.1) satisfies the division algorithm for polynomials. Then we demonstrate that the repeated application of the recurrence relation is a reverse GCD computation.

We now show that repeated application of the division algorithm reverses the computation of the characteristic polynomial of a $GF(2^p)$ CA. Suppose that $\Delta_n$ and $\Delta_{n-1}$ are known. By the division algorithm, $x + d_n$ and $\Delta_{n-2}$ are uniquely determined and easily calculated. If the division algorithm is then applied to $\Delta_{n-1}$ and $\Delta_{n-2}$, it will calculate $x + d_{n-1}$ and $\Delta_{n-3}$. We may continue this process until we have computed $x + d_1$ and $\Delta_{-1} = 0$.

**Example 4.3.** Let $\mathbf{C}$ be a 4-cell $GF(2^2)$ CA with $\Delta_4 = x^4 + 2x^3 + 2x^2 + 3$ and $\Delta_3 = x^3 + 3x^2 + 2x + 1$.

| *dividend* | *divisor* | *quotient* | *remainder* | $GF(2^2)$ *CA byte* |
|---|---|---|---|---|
| $\Delta_4$ | $\Delta_3$ | $x + 1$ | $2^2(x^2 + x + 3)$ | 1 |
| $\Delta_3$ | $x^2 + x + 3$ | $x + 2$ | $2^2x$ | 2 |
| $x^2 + x + 3$ | $x$ | $x + 1$ | $2^2 \cdot 1$ | 1 |
| $x$ | 1 | $x + 0$ | $2^2 \cdot 0$ | 0 |

(4.3)

From the calculation, we see that the divisor column is the same as the dividend column shifted up one position and the remainder column is a shift of the $i^2$ times with the divisor column. Comparing (4.2) to (4.3), we see that the sequence of polynomial in (4.3) is the reverse of the sequence of intermediate polynomials in

the characteristic polynomial calculation. Furthermore, (4.3) yields the sequence of quotients

$$[x+0, x+1, x+2, x+1]$$

By taking the constant terms of these quotients and reversing, we obtain the rule vector $< 0, 1, 2, 1 >_2$.

In Example 4.3 let $\Delta_3 = x^3+3$. Then we obtain the rule vector $< 3, 1, 2, 2 >_3$. Also let $\Delta_3 = x^3$. Then we obtain the rule vector $< 0, 0, 0, 2 >_3$.

If $\mathbf{C}$ is an $n$-cell $GF(2)$ 90/150 CA with the primitive polynomial as the characteristic polynomial, then there exist two $\Delta_{n-1}$. But the $\Delta_{n-1}$ are several in the Example 4.3.

By Theorem 4.1 we can obtain a $GF(2^p)$ CA with $\Delta_n$ and $\Delta_{n-1}$. But the method for finding $\Delta_{n-1}$ does not exist until now.

**Theorem 4.4.** Let $\mathbf{C}$ be an $n$-cell $GF(2^p)$ CA with the state transition matrix $T =< d_1, d_2, \cdots, d_n >_i$. And let $p(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0$ be the primitive polynomial which is the characteristic polynomial of $T$. For the nonsingular upper tridiagonal matrix $U$ and for the companion matrix $C$ of $p(x)$, let $U$ and $C$ be as the following:

$$U = (u_{ij}) = \begin{cases} u_i,\ i = j \\ a_i,\ i = j-1 \\ 0,\ i > j \\ x_{ij} \in GF(2^p),\ otherwise \end{cases} \qquad C = (s_{ij}) = \begin{cases} 1,\ i = j+1\ (j < n) \\ c_{i-1},\ j = n \\ 0,\ otherwise, \end{cases}$$

where $c_i$ is the coefficient of $p(x)$. Then we obtain the following equation.

$$\begin{cases} d_1 = u_1^{-1} a_1 \\ d_k = u_{k-1}^{-1} a_{k-1} + u_k^{-1} a_k\ (1 < k < n) \\ d_n = u_{n-1}^{-1} a_{n-1} + c_{n-1} \end{cases} \tag{4.4}$$

*Proof.* Since the characteristic polynomials and the minimal polynomials of $T$ and $C$ are the same, $T$ and $C$ are similar. So $TU = UC$. Then we obtain the following:

$$\begin{cases} a_1 = u_1 d_1 \\ a_k = i a_{k-1} + u_k d_k\ (1 < k < n) \\ c_{n-1} u_n = i a_{n-1} + u_n d_n \\ u_{i+1} = i u_i \end{cases} \tag{4.5}$$

Since $i = u_{k-1}^{-1} u_k$, we obtain the following required result

$$\begin{cases} d_1 = u^{-1} a_1 \\ d_k = u_{k-1}^{-1} a_{k-1} + u_k^{-1} a_k\ (1 < k < n) \\ d_n = u_{n-1}^{-1} a_{n-1} + c_{n-1} \end{cases} \tag{4.6}$$

## 5 Conclusion

In this paper we analyzed linear $GF(2^p)$ CA. Especially, we proposed transition rules of linear group $GF(2^p)$ CA. Also we analyzed the characterization of linear

group $GF(2^p)$ CA. Especially we analyzed the relationship between characteristic polynomial and transition rule of linear group $GF(2^p)$ CA. Our results and Cho et al.'s results [6] will be helpful for the development of the synthesis of linear $GF(2^p)$ CA.

## References


1. Von Neumann, J.: The Theory of Self-Reproducing Automata. In: Burks, A.W. (ed.), University of Illinois Press, Urbana (1966)
2. Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev. Mod. Phys. 55, 601–644 (1983)
3. Das, A.K.: Additive Cellular Automata: Theory and Applications as a Built-In Self-Test Structure, Ph.D Thesis, I.I.T. Kharagpur, India (1990)
4. Das, A.K., Chaudhuri, P.P.: Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation. IEEE Trans. Comput. 42, 340–352 (1993)
5. Das, A.K., Chaudhuri, P.P.: Efficient characterization of cellular automata. Proc. IEE(Part E) 137(1), 81–87 (1990)
6. Cho, S.J., Choi, U.S., Hwang, Y. H., Kim, H.D., Kim, J.G., Heo, S.H.: New synthesis of one-dimensional 90/150 linear hybrid group cellular automata. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 26(9), 1720–1724 (2007)
7. Cho, S.J., Choi, U.S., Hwang, Y.H., Pyo, Y.S., Kim, H.D., Heo, S.H.: Computing Phase Shifts of Maximum-Length 90/150 Cellular Automata Sequences. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 31–39. Springer, Heidelberg (2004)
8. Cho, S.J., Choi, U.S., Kim, H.D.: Analysis of complemented CA derived from a linear TPMACA. Computers Math. Applic. 45, 689–698 (2003)
9. Cho, S.J., Choi, U.S., Kim, H.D.: Behavior of complemented CA whose complement vector is acyclic in a linear TPMACA. Mathl. Comput. Modelling 36, 979–986 (2002)
10. Nandi, S., Kar, B.K., Chaudhuri, P.P.: Theory and applications of cellular automata in cryptography. IEEE Trans. Computers 43, 1346–1357 (1994)
11. Paul, K.: Theory and application of GF($2^p$) Cellular automata, Ph.D Thesis, Departmen of Computer Science and Technology, Bengal Engineering College (A Deemed University) (2002)
12. Sikdar, B.K., Majumder, P., Mukherjee, M., Ganguly, N., Das, D.K., Chaudhuri, P.P.: Hierarchical Cellular automata as an on-chip test pattern generator. In: VLSI Design, Fourteenth International Conference on 2001, pp. 403–408 (2001)
13. Cho, S.J., Choi, U.S., Hwang, Y.H., Kim, H.D., Choi, H.H.: Behaviors of single attractor cellular automata over Galois field $GF(2^p)$. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 232–237. Springer, Heidelberg (2006)

# On the Collision-Propagation and Gather-Update Formulations of a Cellular Automata Rule

Bastien Chopard[1], Jean-Luc Falcone[1], Ranaivo Razakanirina[1], Alfons Hoekstra[2], and Alfonso Caiazzo[2]

[1] University of Geneva, Switzerland
bastien.chopard@cui.unige.ch
[2] University of Amsterdam, The Netherlands

**Abstract.** We consider two formulations of a cellular automata: the first one uses a gather-update paradigm and the second one a collision-propagation paradigm. We show the equivalence of both descriptions and, using the latter paradigm, we propose a simple way to define a Cellular Automata on a graph with arbitrary topology. Finally, we exploit the duality of formulation to reconsider the problem of characterizing invertible cellular automata.

## 1 Introduction

Traditionnally, a 1D, nearest neighbors Cellular Automata (CA) defined on a discrete spatial domain $D$ is expressed as [3,9]

$$s(r, t+1) = F(s(r-1, t), s(r, t), s(r+1, t)) \qquad \forall r \in D \tag{1}$$

where $F$ is the update rule, $s(r, t)$ the state of site $r \in D$ at time $t = 0, 1, 2, \ldots$. The extension to higher dimensions or larger neighborhood is straightforward. Eq. (1) refers to what we call here a *gather-update* (GU) formulation. The values $s(r-1, t)$, $s(r+1, t)$ of the neighbors are first read by cell $r$ (gather operation) and then combined through $F$ to update $s(r)$ at the next time step. This formulation is standard in the CA community.

However, in the community of lattice gas automata (LGA) and lattice Boltzmann (LB) [3] a different formulation is preferred. In LGA or LB, the state $f(r, t)$ of a cell is multi-valued, associating one value with each neighbor. For instance, in 1D, $f(r, t)$ is a three-value column vector

$$f(r, t) = (f_0(r, t), f_1(r, t), f_2(r, t))^T \tag{2}$$

The quantity $f_0$ is an information only known to the cell itself whereas $f_1$ and $f_2$ are data intended to the left and right neighbors.

The update scheme for this formulation is called the *collision-propagation* paradigm. During the propagation phase, the state $f_i$ at location $r$ is moved to

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 144–151, 2008.

the state $f_i$ at location $r + v_i$, i.e. $f_i(r) \to f_i(r + v_i)$. In 1D, with $v_0 = 0$, $v_1 = 1$, $v_2 = -1$ the effect of propagation is to shift the states $i = 1$ to the right and those with $i = 2$ to the left. In general, this streaming of data can be expressed as

$$f^{in}(t+1) = Pf^{out}(t) \tag{3}$$

where $P$ is the propagation operator and it contains the information about the neighborhood topology. Here we consider $f(t)$ as the whole set of states at time $t$. Thus $P$ takes a full configuration and creates a new one in which each internal state is shifted to the appropriate direction. Note that we have introduced the upperscript *in* and *out* to better distinguish incoming and outgoing information.

After propagation, $f$ can be updated according to the chosen evolution rule. We call this phase the *collision* and we denote by $C$ the operator which transform $f^{in}(r, t+1)$ into $f^{out}(r, t+1)$ for all $r \in D$

$$f_i^{out}(r,t) = C_i\left(f^{in}(r,t)\right) \qquad \text{or, in short} \qquad f^{out}(t) = Cf^{in}(t) \tag{4}$$

By combining eqs. (3) and (4) we get

$$f^{in}(t+1) = PCf^{in}(t) \qquad \text{or} \qquad f^{out}(t+1) = CPf^{out}(t) \tag{5}$$

which we call the collision-propagation (PC) formulation. In what follows, $f$ will denote either $f^{in}$ or $f^{out}$, depending on the context.

Note that at the boundary of the domain $D$, some of the $f_i$ may not be properly defined after the propagation step. In our 1D example, $f_1$ at the left-most $r$ and $f_2$ at the right-most $r$ will be unknown. Boundary conditions must be then supplied before collision can be applied. We call $B$ the operator acting upon the configuration $f$ and providing the required information, which is obviously problem dependent. Then, eq. (5) becomes $f^{in}(t+1) = PCBf^{in}(t)$.

The above structure is at the heart of the Complex Automata (CxA) approach recently proposed by us [5] for coupling several CA's together. The description of all the CA's in terms of the $P$, $B$ and $C$ operator provide a generic way to define coupling between different models (see also [1,6]).

## 2 Cellular Automata on a Graph Topology

CA are usually defined on a regular lattice of cells. This is quite natural if the CA represents a spatial domain. But some problems are more efficiently described by a complex network [2]. This is the case of many applications in economy, social science, epidemiology or system biology.

There is no well established way to define a dynamical evolution of a system whose structure is a complex network or a graph. Clearly, CA have been quite successful to model complex dynamical systems on a regular topology and it is natural to extend the definition of a CA to any set of interconnected cells. We call such an extension a CAG (Cellular Automata on a Graph). Here we use the collision-propagation (CP) paradigm to express it.

Informally a CAG is defined as a triple $(V, E, C)$ where $(V, E)$ is a *directed* graph with $V$ the set of nodes and $E$ the set of edges. We assume that each node $r$ contains internal state variables $f_i^{in}(r)$ and $f_i^{out}(r)$. The number of $f_i^{in}(r)$ is equal to the in-degree $k_{in}$ of node $r$ and the number of $f_j^{out}(r)$ to its out-degree $k_{out}$. In addition we may have two extra states $f_0^{in}(r)$ and $f_0^{out}(r)$.

The quantity $C$ is the collision operator which acts synchronously and locally at each node $r \in V$ and computes the outgoing values from the incoming ones

$$f_j^{out}(r) = C_j(r) f^{in}(r) \qquad 1 \leq j \leq k_{out}$$

Note that now the action of $C$ may depend on the node $r$ for the simple reason that different nodes may have a different number of neighbors.

The propagation $P$ as well as the neighborhood are naturally defined from the graph edges in $E$: assume there is an edge in $E$ from node $r_0$ to node $r_1$. In node $r_0$ we suppose this outgoing edge is labeled with index $i_k$. In $r_1$, let us say this incoming edge is labeled $i_\ell$. Then, the propagation $P$ will move $f_{i_k}^{out}(r_0, t)$ to $f_{i_\ell}^{in}(r_1, t+1)$.

As before, the $C$ and $P$ operators may be supplemented by an operator $B$ to define boundary conditions. Note however that in a graph, all existing entering edges are expected to come from an existing node (a bit like in a periodic system) and $P$ does not create any missing information. In a CAG, interaction with the external world is then naturally implemented with special nodes having given boundary values and only outgoing edges.

From the above discussion, it is clear that the CP formalism easly describes a CAG with the same compact relation as before, that is $f^{in}(t+1) = P f^{out}(t)$ and $f^{out}(t+1) = C f^{in}(t+1)$.

A software environment has been recently developed to implement a CAG [8]. Its output is illustrated in Fig. 1. The application we have considered is a simple economical model on a complex network. The links of the network represents the possible interactions between idealized agents trading goods against money. Here we assumed that interactions between persons obey a scale-free topology.

In this application, the operator $P$ implements the exchanges of good and money between connected pairs of agents. Based on a local and self-adapting price $p(r, t)$, the operator $C$ computes how an agent's fortune is split among the sellers in its neighborhood in order to buy their goods. In a second phase $C$ also computes how much goods each seller gives to each of its buyer in exchange of its money.

This simple market model and the simulation results are discussed in details in [8]. Here we only want to stress the behavior observed in Fig. 1. Depending on the initial condition and the graph topology, we can see the emergence of sub-market, i.e. subgraphs that result from the deletion of the transaction between some pairs of agents. It is indeed observed that two agents $r_1$ and $r_2$ that are not in the same *strongly connected component* of the graph will gradually reduce their interaction until the flux of money or goods that traverse the links $r_1 \to r_2$ or $r_2 \to r_1$ drops to zero. As time goes on, the dynamics reaches a stationnary state. An interesting result of this model is that the local prices in each emergent submarket converge to a unique value, but a different one in each submarket [8].

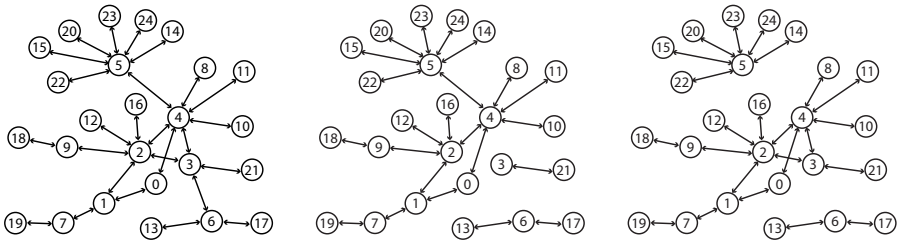



**Fig. 1.** Left panel: the initial graph topology. Middle and right panels: the resulting emergent submarkets with different initial money distributions.

## 3 Equivalence between the Gather-Update and Collision-Propagation Formulation

In this section we show that the PC and GU formulations, despite their conceptual differences, are mathematically equivalent.

In GU, all the neighbors of cell $r_0$ will gather the *same* information from $r_0$. When the neighbors of $r_0$ need different data, the CP formulation is more natural. This is the case in the above economical model in which a cell $r_0$ gives a different amount of money or goods to each of its neighbors. The same happens in LGA/LB models because a different amount of particle flows to the different neighbors. In the PC formulation, the flux of data between neighbors is made explict whereas it is not in the GU model.

Another difference between the two formulations is that the GU combines in one operation the reading of the neighbors state and the update of the site. On the other hand the CP approach clearly disentangles the *local* part of the rule ($C$) and its *non-local* part ($P$), thus giving a formulation which is closer to the computer implementation, at least in the case of a parallel code.

The translation from one formulation to the other is straightforward. Let us first assume we have a rule in the CP approach where the state $f(r)$ has, say, $q+1$ components. We write $f(r) = (f_0(r), f_1(r), \ldots, f_q(r))^T$ with $f_i(r)$ the quantity that will be send to neighbor $r + v_i$ ($v_0 = 0$). From $f^{out}(t+1) = CPf^{out}(t)$, we have

$$f_i^{out}(r) = C_i(f_0^{out}(r), f_1^{out}(r - v_1) \ldots f_q^{out}(r - v_q))^T \qquad \forall i \quad 0 \leq i \leq q$$

In the GU approach, a multi-valued state can be easily defined as $s(r,t) = f^{out}(r,t)$. Since $s$ has $q+1$ components, the evolution rule $F$ must be a set of $q+1$ rules $F_0, \ldots, F_q$ such that

$$s_i(r, t+1) = F_i(s(r,t), \ldots, s(r - v_q, t))$$

By definition of the GU formalism, $F_i$ is a function of the entire state of the neighbor cells. Therefore, in order for $F$ to match the CP formulation, one has to restrict $F$ to one component only of the neighbor's state. By introducing a

*selection* operator $S_i$ such that $S_i s = s_i$, we see that the equivalence between a CP rule and a GU rule is obtained by choosing $s(r,t) = f^{out}(r,t)$ and

$$\begin{aligned} F_i(s(r,t),\ldots,s(r-v_q,t)) &= F_i(S_0 s(r), S_1 s(r-v_1), ..., S_q s(r-v_q)) \\ &= C_i(S_0 s(r), S_1 s(r-v_1), ..., S_q s(r-v_q)) \end{aligned} \tag{6}$$

As an example, let us consider the simple 1D CA in which $C$ is the identity. By applying $CP$, the states $f_1$ move east and the states $f_2$ move west. In accordance to eq. (6) this rule can also be written as

$$s_1(r,t+1) = s_1(r-1,t) \qquad s_2(r,t+1) = s_2(r+1,t) \tag{7}$$

Thus the identity rule in CP implements simultaneously two rules in the GU formulation, namely the so-called *east* and *west* rules.

We can now consider the inverse problem, i.e. how to write a GU rule in the CP form. Let us assume we have

$$s(r,t+1) = F(s(r,t), s(r-v_1,t)\ldots, s(r-v_q,t))$$

where $s$ and $F$ are possibly multi-component quantities. To obtain a CP version, the first step is to replicate $s$ for all $q+1$ neighbor directions. We thus introduce an *expansion* operator $E$ such that $f^{out}(r) = Es(r)$ means $f_i^{out}(r) = s(r)$, for $0 \le i \le q$. We can also define the inverse of $E$, the projection operator $\Pi$ such that $\Pi f^{out}(r) = f_0^{out}(r) = s(r)$. The propagation can then be applied to $f^{out} = Es$. To match the behavior of $F$ in the GU formulation, one has to choose $C_i = F$, for all $0 \le i \le q$. It is indeed easy to check that

$$Es(t+1) = f(t+1) \qquad \text{or} \qquad \Pi CPEs(t) = s(t+1)$$

To illustrate this construction we take the *east* CA rule $s(r,t+1) = s(r-1,t)$ which simply moves information to east. The rule is $F(s(r), s(r-1), s(r+1)) = s(r-1)$. On a spatial configuration $\ldots abc \ldots$ it acts as follows

$$\ldots abc \ldots \xrightarrow{F} \ldots zab \ldots$$

where $z$ is the state of the cell on the left. In the CP case, with the collision $C_0(f_0,f_1,f_2) = C_1(f_0,f_1,f_2) = C_2(f_0,f_1,f_2) = f_1$ we have for the vector $f = Es$

$$\ldots \begin{pmatrix} a \\ a \\ a \end{pmatrix} \begin{pmatrix} b \\ b \\ b \end{pmatrix} \begin{pmatrix} c \\ c \\ c \end{pmatrix} \xrightarrow{P} \ldots \begin{pmatrix} a \\ z \\ b \end{pmatrix} \begin{pmatrix} b \\ a \\ c \end{pmatrix} \begin{pmatrix} c \\ b \\ d \end{pmatrix} \ldots \xrightarrow{C} \ldots \begin{pmatrix} z \\ z \\ z \end{pmatrix} \begin{pmatrix} a \\ a \\ a \end{pmatrix} \begin{pmatrix} b \\ b \\ b \end{pmatrix} \ldots$$

which is, as expected, $Es(t+1)$.

## 4 Invertible CA

The question of finding the inverse of a CA rule has been discussed in several papers [4], but mostly in the framework of a GU formulation. In the CP model,

there is an easy way to build an invertible rule, using the physical interpretation of a collision process followed by particle motion [3,7]: assume we have $f(n) = (PC)^n f(0)$ for some given $C$ operator. After $n$ iterations let us apply a still to be specified local transformation $A$ to the configuration $f(n)$ and apply, for $n$ steps, another collision operator $C'$. A rule is said *invertible* if we then obtain back the initial configuration $f(0)$, up to the transformation $A$. This condition reads $(PC')^n A(PC)^n f(0) = Af(0)$ or, simply, $(PC')^n A(PC)^n = A$ since it is true for any $f(0)$. With

$$(PC')^n A(PC)^n = (PC')^{n-1} PC' APC(PC)^{n-1}$$

we see, by induction over $n$ that condition $(PC')^n A(PC)^n = A$ is equivalent to

$$A = PC' APC \tag{8}$$

In other words, if we can find $A$ and $C'$ such that the above condition holds, the CA is invertible.

It is easy to check that a way to solve (8) is to impose

$$A = R^{-1}C \qquad PRP = R^{-1} \qquad C'R^{-1}C = R \tag{9}$$

for some *reverse* operator $R$ having an inverse $R^{-1}$. This solution is inspired from LGA models of particles [3,9]: to reverse the particles motion, one needs to perform an extra collision, reverse their velocity ($R$ does it) and then run the LGA $n$ times to finally obtain the initial state with reversed velocities.

It is interesting to note that eq. (9) includes Fredkins method [9] to produce a reversible rule in the GU approach

$$s(r, t+1) = F(\{s(r - v_i, t)\}_{i=0}^{q}) \oplus s(r, t-1) \tag{10}$$

where $\oplus$ denotes the logical xor and $s$ is a Boolean state. The rule is reversible (self-inverse) for any $F$ since it can be written as $s(r, t-1) = F(\{s(r-v_i, t)\}_{i=0}^{q}) \oplus s(r, t+1)$ due to the property of the xor.

Let us now transform (10) in a CP form. To keep the notation simple we consider a 1D case. Two states per cell must be stored: the current and previous time step. Let us call them $s$ and $\bar{s}$. According to the procedure above, we construct

$$f(r, t) = E(s, \bar{s})^T = (s(r, t), \bar{s}(r, t), s(r, t), \bar{s}(r, t), s(r, t), \bar{s}(r, t))^T$$

where, by choice, components 3,4 are propagated to the right, components 5,6 to the left and components 1,2 are the rest states. Let us define the collision $C$ as follows

$$\begin{aligned} C_i f &= F(f_1, f_3, f_5) \oplus f_{i+1} \qquad i = 1, 3, 5 \\ C_2 f &= f_1 \qquad C_4 f = f_5 \qquad C_6 f = f_3 \end{aligned} \tag{11}$$

Let us define $R$ as the operator which, first, swaps the values which propagate in one direction with those propagating in the other direction and, second, swaps

$f_i$ and $f_{i+1}$ for $i = 1, 3, 5$. $R$ can be expressed by a $6 \times 6$ matrix

$$R = \begin{pmatrix} B & 0 & 0 \\ 0 & 0 & B \\ 0 & B & 0 \end{pmatrix} \qquad \text{with} \qquad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

It is easily verified that $B^2 = 1$ and $R = R^{-1}$. Let us now check whether condition (9) holds. We have

$$\begin{aligned} PRPf(r) &= PR(f_1(r), f_2(r), f_3(r-1), f_4(r-1), f_5(r+1), f_6(r+1))^T \\ &= P(f_2(r), f_1(r), f_6(r+1), f_5(r+1), f_4(r-1), f_3(r-1))^T \\ &= (f_2(r), f_1(r), f_6(r+1-1), f_5(r+1-1), f_4(r-1+1), f_3(r-1+1))^T \\ &= Rf(r) \end{aligned} \tag{12}$$

Thus $PRP = R$. For the collision condition $C'R^{-1}C = R$ we choose $C' = C$ since the rule is expected to be its own inverse. By dropping the cell location $r$ and writing $F$ for $F(f_1, f_3, f_5)$ we have

$$\begin{aligned} CRCf &= CR(F \oplus f_2, f_1, F \oplus f_4, f_5, F \oplus f_6, f_3)^T \\ &= C(f_1, F \oplus f_2, f_3, F \oplus f_6, f_5, F \oplus f_4)^T \\ &= (F \oplus F \oplus f_2, f_1, F \oplus F \oplus f_6, f_5, F \oplus F \oplus f_4, f_3)^T \\ &= (f_2, f_1, f_6, f_5, f_4, f_3)^T = Rf \end{aligned} \tag{13}$$

Thus $CRC = R$ and Fredkins method is a special case of eq. (9).

It should be noted that another way to solve eq. (8) is

$$A = PR \qquad PRP = R^{-1} \qquad C'R^{-1}C = R \tag{14}$$

If we choose $R = P^{-1}$ ($P^{-1}$ always exists since it corresponds to moving the information backwards) then (14) requires $C'PCP = 1$. Consider a 1D rule $s(r, t+1) = F(s(r-1, t), s(r, t), s(r+1, t))$ and its potential inverse $s(r, t+1) = G(s(r-1, t), s(r, t), s(r+1, t))$. We set $f(r) = (f_1, f_0, f_2) = Es(r) = (s(r), s(r), s(r))$ and we associate $C$ to $F$ and $C'$ to $G$. Thus

$$C'PCPf = C'P \begin{pmatrix} F(s(r-1), s(r), s(r+1)) \\ F(s(r-1), s(r), s(r+1)) \\ F(s(r-1), s(r), s(r+1)) \end{pmatrix} = C' \begin{pmatrix} F(s(r-2), s(r-1), s(r)) \\ F(s(r-1), s(r), s(r+1)) \\ F(s(r), s(r+1), s(r+2)) \end{pmatrix}$$

Then $C'PCP = 1$ if

$$G[F(s(r-2), s(r-1), s(r)), F(s(r-1), s(r), s(r+1)), F(s(r), s(r+1), s(r+2))] = s(r) \tag{15}$$

which is the expected *non-local* relation expressing that rules $F$ and $G$ are inverse of each other. It is clear that rule *east* $F(s(r-1), s(r), s(r+1)) = s(r-1)$ and rule *west* $G(s(r-1), s(r), s(r+1)) = s(r+1)$ obey this inverse relation.

In general, the validity of eq. (15) requires to check all $2^5$ values of $s(r-2)$, $s(r-1)$, $s(r)$, $s(r+1)$, $s(r+2)$. In 2D and a von Neumann neighborhood $2^{13} = 8192$ tests would be necessary to check whether $C'PCP = 1$. In contrast, the verification of the *local* condition $C'R^{-1}C = R$ requires only $3 \times 2^3$ operations in 1D and $5 \times 2^5 = 160$ in 2D.

## 5 Conclusions

We have formally discussed the relation between the collison-propagation (CP) and the gather-update (GU) formulation of a CA rule. The CP formulation is naturally adapted to situations where the flow of information depends on the neighbors. It is also well suited to couple several CA [5].

Furthermore we have introduced the concept of CAG (cellular automata on a graph) by applying the CP approach to irregular topologies and asymmetrical neighborhoods.

We have finally explored the conditions of finding the inverse of a CA rule using both formulation. We found that two classes of invertible CA can be identified: the information to inverse the rule is local (as in the Fredkins construction or in discrete fluid models); or the information to inverse the rule is non-local (shared by the neighbors) and the problem is numerically more intensive.

This work is supported by the European Commission (COAST project EU-FP6-IST-FET Contract 033664) and the Swiss National Science Foundation.

## References

1. Caiazzo, A., Falcone, J.-L., Chopard, B., Hoekstra, A.: Error investigations in complex automata models for reaction-diffusion systems. In: Umeo, H., et al. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 260–267. Springer, Heidelberg (2008)
2. Caldarelli, G.: Scale-Free Networks. Oxford University Press, Oxford (2007)
3. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press, Cambridge (1998)
4. Durand-Lose, J.: Representing reversible cellular automata with reversible block cellular automata. In: Discrete Mathematics and Theoretical Computer Sciences Proceedings AA (DM-CCG), pp. 145–154 (2001)
5. Hoekstra, A., Lorenz, E., Falcone, J.-L., Chopard, B.: Towards a complex automata formalism for multuscale modeling. Int. J. Multiscale Computational Engineering 5(6), 491–502 (2008)
6. Hoekstra, A.G., Falcone, J.-L., Caiazzo, A., Chopard, B.: Multi-scale modeling with cellular automata: The complex automata approach. In: Umeo, H., et al. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 192–199. Springer, Heidelberg (2008)
7. Marconi, S., Chopard, B.: Discrete physics, cellular automata and cryptography. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 617–626. Springer, Heidelberg (2006)
8. Razakanirina, R.M.: Cellular automata on graphs applied to financial flow simulation. Master's thesis, University of Geneva, Computer Science department, Master thesis (2007)
9. Toffoli, T., Margolus, N.: Cellular Automata Machines: a New Environment for Modeling. MIT Press, Cambridge (1987)

# Exploring $CA$ State Space to Synthesize Cellular Automata with Specified Attractor Set*

Sukanta Das[1], Chandrama Shaw[1], and Biplab K. Sikdar[2]

[1] Dept. of Information Technology
{sukanta,chandrama}@it.becs.ac.in
[2] Dept. of Computer Science & Technology
biplab@cs.becs.ac.in
Bengal Engineering & Science University, Shibpur, West Bengal, India, 711103

**Abstract.** The cellular automaton ($CA$) with multiple attractors in its state space creates immense interest to devise solutions for pattern classification, pattern recognition, design of associative memory, query processing, etc. This work characterizes the $CA$ state space to explore the essential properties of 1-dimensional nonlinear cellular automata with single cycle attractors. The characterization of pseudo-exhaustive bits ($PE$ bits) is done to uniquely identify the attractor set of such a $CA$. Theoretical framework thus evolved provides means to synthesize a $CA$ for a given attractor set with specified $PE$ bits.

**Keywords:** Nonlinear cellular automata, attractor, MACA, $PE$ bit, classifier.

## 1 Introduction

The concept of Cellular Automata ($CA$) was initiated in 1950s by von Neumann and Ulam [9]. Neumann's $CA$ involved 5-neighborhood interactions among the cells with 29 states per cell. Researchers had tried to view rather simplified structure of $CA$ with the target to characterize its behavior, essentially keeping the flavour as that of Neumann's model [1,3,8]. In early 1980s, Stephen Wolfram [10] studied a family of simple 3-neighborhood 1-dimensional cellular automata [7] with two states per cell. This structure attracted a large section of researchers working in diverse fields [2].

While characterizing the $CA$ state space, the researchers identified a set of $CA$ states towards which the neighboring states asymptotically approach in the course of dynamic evolution [11]. They referred this set of states as the attractor of $CA$ state space forming a basin of attractions. The $CA$ with multiple attractors in its state space were of primary interest [2,6]. The single cycle attractor is one where the number of states of an attractor is one [2]. Characterization of single cycle attractors of the linear/ additive $CA$ has been reported in [2,5,6].

* This research work is supported by the Sponsored CA Research Projects, Bengal Engineering and Science University, Shibpur, West Bengal, India-711103.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 152–159, 2008.

The applications of linear/ additive $CA$ having multiple single cycle attractors are also investigated. However, the characterization of single cycle attractors in nonlinear $CA$ state space is yet to be explored.

The above scenario motivates us to concentrate on the characterization of single cycle attractors in 1-dimensional cellular automata [10]. A theoretical framework has been developed to explore all the attractors of such a $CA$. This further enables identification of pseudo-exhaustive ($PE$) bits [2] to address an attractor and then synthesis of a $CA$ for the given set of attractors & $PE$ bits. The algorithms/ schemes are proposed for efficient synthesis of such desired $CA$.

The preliminaries of $CA$, relevant for this work, have been reported in the twin paper (*Characterization of Non-reachable States in Irreversible CA State Space*). In the next section, we report the proposed characterization.

## 2 Characterization of Single Cycle Attractors

This section detailed out the theoretical framework developed to explore the single cycle attractors of a given $CA$ and the $PE$ bits specifying the attractors. It is based on the analysis of $RMT$s [4] of each cell rule of the $CA$.

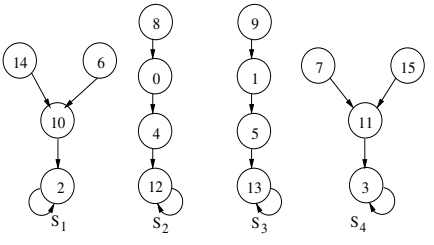



**Fig. 1.** State transitions of a $CA$ with rule vector $< 10, 69, 204, 68 >$

### 2.1 Identification of Attractors of a $CA$

Since the next state (NS) of a single cycle attractor is the attractor itself (*Fig.1*), there should be at least one $RMT$ (*Table 1*) of each cell rule ($\mathcal{R}_i$) for which the $CA$ cell ($i$) does not change its state. For example, $RMT$ $x0x$ ($x = 0/1$) is considered to find the NS of cell $i$ when the current states of its left neighbor ($(i-1)^{th}$ cell), self and right neighbor ($((i+1)^{th}$) cell) are $x$, 0 and $x$ respectively. It implies, if such $RMT$ is '0', the state change of cell ($i$) is $0 \rightarrow 0$. That is, for the rule $\mathcal{R}_i$, if $RMT$ 0 (000), 1 (001), 4 (100) or 5 (101) is 0, the $CA$ cell $i$ configured with $\mathcal{R}_i$ does not change its state. Similarly, if the $RMT$ 2 (010), 3 (011), 6 (110) or 7 (111) is 1, it ensures a cell configured with $\mathcal{R}_i$ can stick to its current state. For example, all $RMT$s of rule 204 help formation of attractors (*Fig.2*).

*Property 1*: A rule $\mathcal{R}_i$ can contribute to the formation of single cycle attractor(s) if at least one of the $RMT$s 0, 1, 4 or 5 is 0, or the $RMT$ 2, 3, 6 or 7 is 1.

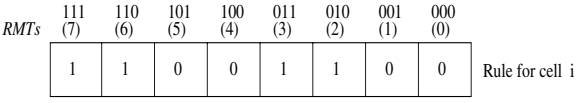



**Fig. 2.** *RMT*s of rule 204

If any rule of a $CA$ does not maintain *Property 1*, the $CA$ can not have single cycle attractors. The following recursive algorithm scans the rule vector $\mathcal{R} =< \mathcal{R}_1, \cdots, \mathcal{R}_i, \cdots, \mathcal{R}_n >$ of a $CA$ and explores all of its attractors.

Algorithm 1: **FindAttractors (*i*, *State*)**
**Input**: $n$ ($CA$ size), $CA$ rule vector.
**Output**: set of attractors.
A). if $i = 1$, *i.e*, for the first rule {
if $RMT$ 0 is 0 then
Set first two bits of *State* as 00 and call *FindAttractors (i+1, State)*
if $RMT$ 1 is 0 then
Set first two bits of *State* as 01 and call *FindAttractors (i+1, State)*
if $RMT$ 2 is 1 then
Set first two bits of *State* as 10 and call *FindAttractors (i+1, State)*
if $RMT$ 3 is 1 then
Set first two bits of *State* as 11 and call *FindAttractors (i+1, State)* }
else if $(1 < i < n)$, *i.e*, for an intermediate rule {
B). Set $(i+1)^{th}$ bit of *State* as 0, and
Compute $k$ – decimal equivalent of $(i-1)$, $i$, and $(i+1)^{th}$ bit sequence of *State*
Check whether the $RMT$ $k$ of the $i^{th}$ rule can stick to the $i^{th}$ bit of *State*.
C). If yes, call *FindAttractors (i+1, State)*
D). Set $(i+1)^{th}$ bit of *State* as 1
Compute $k$ and check whether $RMT$ $k$ of $i^{th}$ rule can stick to $i^{th}$ bit of *State*.
If yes, call *FindAttractors (i+1, State)* }
else
E). Compute $k$= decimal equivalent of the $(n-1)^{th}$ bit, $n^{th}$ bit of *State* and 0
If $RMT$ $k$ of last rule is the last bit of *State*, output the *state* as an attractor.

The argument *State* contains partially constructed attractor states. To find the exact form of an attractor, we need to run $FindAttractors(1, State)$.

*Example 1.* Consider a 4-cell $CA$ with rule vector $< 10, 69, 204, 68 >$ (*Table 1*). To find attractors, we call *FindAttractors(1,State)*, where *State* (- - - -) is empty. Initially, the algorithm finds that $i = 1$ and $RMT$ 0 for the first rule is 0. Hence if a state starts with 00, the first bit will stick to 0. That is, the state may be an attractor. Therefore, the first two bits of 4-bit *State* is filled up with 00, and the recursive algorithm *FindAttractors* ($i = 2$ and *State* (0 0 - -)) is called.

*First recursive call (i = 2 and State = 00- -)*: According to the algorithm, $(i+1)^{th}$ bit -that is, the third bit of *State* is set to 0 (*Algorithm 1 Step B*). Hence, $k = 0$ (000), the $(i-1)^{th}$ bit of the *State* $= 0$, $i^{th}$ bit $= 0$, and $(i+1)^{th}$ bit $= 0$.

**Table 1.** $RMT$s of the $CA < 10, 69, 204, 68 >$ cell rules

| RMT | 111 (7) | 110 (6) | 101 (5) | 100 (4) | 011 (3) | 010 (2) | 001 (1) | 000 (0) | Rule |
|---|---|---|---|---|---|---|---|---|---|
| *First cell* | $d$ | $d$ | $d$ | $d$ | 1 | 0 | 1 | 0 | 10 |
| *Second cell* | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 69 |
| *Third cell* | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 204 |
| *Fourth cell* | $d$ | 1 | $d$ | 0 | $d$ | 1 | $d$ | 0 | 68 |

$RMT$ stands for rule min term. $d$ denotes don't care bit. It can either be 0/1.

However, the $i^{th}$ (second) bit of the *State* -that is, 0 and the $RMT$ $x$ ($RMT$ 0) of $i^{th}$ rule (69, *Table 1*) is (1) not the same. Then, execute Step D, and the second bit of *State* is set to 1. Now the *State* is 001-, and $k = 1$. The $RMT$k (1) of the second rule (69) is 0 -that is, the second bit value of *State*, and so $FindAttractors$ is called with $i = 3$ and *State* (001-).

*Second recursive call ($i$ = 3 and State = 001-)*: Since $i < 4$, as like the $1^{st}$ recursive call, the next bit (fourth bit) of *State* is set to 0. Therefore, the final *State* is 0010, and $k = 2$ (010). The $RMT$ 2 of third rule, however, is 1 -that is, the third bit value of the *State*. So, $FindAttractors$ is called with $i = 4$ and $State = 0010$.

*Third recursive call ($i$ = 4 and State = 0010)*: Here it is checked whether the last bit of *State* can be justified by the last cell rule to recognize the *State* as an attractor. Now, $k = 4$ (100) and $RMT$ 4 of the last rule can generate the fourth bit of the *State*. Therefore, the *State* (0010) is a single cycle attractor (2).

Now $i = 3$. The fourth bit of *State* is set to 1. Hence the *State* is 0011, and $k = 3$ (011). The $RMT$ 3 of third rule is again 1 -that is, the third bit of the *State*. So, the algorithm is further called with $i = 4$ and $State = 0011$.

*Fourth recursive call ($i$ = 4 and State = 0011)*: As like the third recursive call, it checks whether the new last bit of *State* can be justified by the last cell rule. Here $k = 6$ (110) and the $RMT$ 6 of last rule is 1. Hence, the *State* 0011 (3) is also an attractor.

If the execution of algorithm is continued, we can identify another two attractors - 1100 (12) and 1101 (13) (*Fig.1*).

To identify all the attractors of a $CA$, we may not repeat $FindAttractors$. An attractor $A$ can be derived once $B$ and $C$ are identified as the attractors.

**Theorem 1.** *If* $B = b_1b_2 \cdots b_{i-1}b_ib_{i+1} \cdots b_n$ *and* $C = c_1c_2 \cdots c_{i-1}c_ic_{i+1} \cdots c_n$ *are two attractors of an* $n-$*cell* $CA$ *and* $b_{i-1} = c_{i-1}$, $b_i = c_i$ & $b_{i+1} \bigoplus c_{i+1} = 1$, *then* $B' = b_1b_2 \cdots b_{i-1}b_ic_{i+1} \cdots c_n$ *and* $C' = c_1c_2 \cdots c_{i-1}c_ib_{i+1} \cdots b_n$ *are also the attractors of the* $CA$ *[4].*

*Example 2.* Let $B = 010101$ and $C = 000111$ be the two attractors of a 6-cell $CA$. For the attractors $B$ and $C$, the third and fourth bits are the same ($b_{i-1} = c_{i-1}$ and $b_i = c_i$). Whereas the fifth bits ($b_{i+1}$ and $c_{i+1}$) are different. Therefore, the derived attractors are $B' = 010111$ and $C' = 000101$.

**Corollary 1.** *The derived attractors ($B'$ and $C'$) are same as the original ($B$ and $C$) if $b_k = c_k$, $\forall k$, where $1 \leq k < i-1$.*

*Proof.* If $b_k = c_k$, $\forall k$ where $1 \leq k < i-1$, then we can write $B' = b_1 b_2 \cdots b_{i-1} b_i c_{i+1} \cdots c_n = c_1 c_2 \cdots c_{i-2} b_{i-1} b_i c_{i+1} \cdots c_n = c_1 c_2 \cdots c_{i-2} c_{i-1} c_i c_{i+1} \cdots c_n = C$. Similarly, it can be shown that $C' = B$. Hence the proof.

**Corollary 2.** *An $n-$cell $CA$ synthesized from 2 arbitrary attractors, can have maximum $2^{m+1}$ attractors, where $m = \lfloor \frac{n-1}{3} \rfloor$.*

*Proof.* Let us consider a $CA$ is synthesized from the attractors $B$ and $C$. For one set of $(i-1,\ i,\ i+1)$, maintaining *Theorem 1*, the number of attractors is doubled. That is, two pairs $(2^2)$ of attractors are there. If there is another such set of $(i-1,\ i,\ i+1)$, each pair of attractors derives another pair of attractors. Hence, for two such sets of $(i-1,\ i,\ i+1)$, number of attractors is $2^{2+1}$.

However, it is obvious that if $i = 2$ (*Theorem 1*), then $B = B'$ and $C = C'$. That is, no new attractor is derived. Therefore, excluding the left most bit, maximum number of possible $(i-1,\ i,\ i+1)$ set is $\lfloor \frac{n-1}{3} \rfloor$. Therefore, maximum number of possible attractors is $2^{m+1}$, where $m = \lfloor \frac{n-1}{3} \rfloor$. Hence the proof.

### 2.2 Extraction of *PE*-Bits

A number of works [2,5] have been reported, where $MACA$ (multiple attractor cellular automata) is considered to classify the set of data. The $CA$ of *Fig.1* can be employed to classify the patterns of two classes. Class I is represented by $S_1$ & $S_2$ with attractors 2 (0010) & 12 (1100). Whereas class II ($S_3$ & $S_4$) is represented by the attractors 13 (1101) & 3 (0011). To find the class of a pattern $p$ (1010), the $CA$ is to be run for same time steps considering $p$ as the seed. Finally, the $CA$ settles to an attractor $A$ (2). Hence $p$ (1010) belongs to the class of $A$ - that is, class I.

In *Fig.1*, there are 4 attractors. We need 4 places to store the class information. The places can be identified by the least significant 2 bits of the attractors. That is, for an input pattern, if the CA settles to the attractor 12 (1100), then we have to search the place 00. These two least significant bit positions are the pseudo-exhaustive ($PE$) bits of the CA.

The scheme to identify the $PE$ bits of a linear $MACA$ has been reported in [2]. However, extraction of $PE$-bit positions in nonlinear $CA$ is yet to be addressed. We next report characterization of $MACA$ that guides identification of pseudo-exhaustive bits of the attractors of an $MACA$.

**Theorem 2.** *$2^k$ attractors, derived from 2 attractors, can uniquely be identified by $k$ bit positions.*

*Proof.* Consider, a $CA$ is synthesized from two given attractors $B$ and $C$. According to *Theorem 1*, for one set of $(i-1,\ i,\ i+1)$, the number of attractors is doubled. For these four attractors, $(i+1)^{th}$ and $(i-2)^{th}$ bits (or any bit from 1 to $(n-2)$) are unique. Therefore, the four attractors can be identified by

these two bits. If there is another such set $(i-1,\ i,\ i+1)$, each of the attractor pairs derives another pair of attractors. An additional bit -that is, last bit of the set, is required to identify the attractors. Hence, for two such sets of $(i-1,\ i,\ i+1)$, total number of attractors are $2^{2+1}$ and 3 bits are sufficient to identify the attractors. Therefore, if there are $(k-1)$ sets of $(i-1,\ i,\ i+1)$, then we can find $2^k$ number of attractors that can be identified by $k$ bits. Hence the proof.

*Example 3.* The 4 attractors (010101, 000111, 010111 & 000101) of *Example 2*, derived from 010101 and 000111, can uniquely be identified by the second and fifth bits (from left) of the attractors.

*Theorem 2* states that if we can construct two attractors that can derive in total $2^k$ attractors (*Theorem 1*), and then if a $CA$ is synthesized considering those 2 attractors, the $CA$ will must have $2^k$ attractors with $k$ $PE$-bits.

## 3 Synthesis of $CA$ with Single Cycle Attractors

The theoretical framework reported in the earlier section enables synthesis of a $CA$ $(\mathcal{R})$ for a given set of attractors. The following algorithm *SynMACA* reports the synthesis of such a $CA$.

Algorithm 2: **SynMACA**
**Input:** set of attractors.
**Output:** $CA$ (rule vector).
For each of the attractors {
for $i^{th}$ $CA$ cell $(1 \leq i \leq n)$
**S1:** Set $RMT$ $k$ as the $i^{th}$ bit of the attractor, where $k$ is the decimal equivalent of the sequence of $(i-1)$, $i$, and $(i+1)$ bits of the attractor, assuming $0^{th}$ and $(n+1)^{th}$ bits are 0. }
**S2:** Set the unfilled $RMT$s such that those $RMT$s as a whole can not contribute to generate single cycle attractors. Output the $CA$ rule vector.

The avoidance of $RMT$s that can generate single cycle attractors (S2, *Algorithm 2*), can be realized following the designed next algorithm. It synthesizes a $CA$ that does not have any single cycle attractor.

Algorithm 3: **CAWithoutSingleCycle**
**Input:** $n$ ($CA$ size)
**Output:** $CA$
*Step 1:* Randomly synthesize an $n$-cell $CA$.
*Step 2:* If the $CA$ is having no single cycle, goto *Step 6*.
*Step 3:* Select a $CA$ cell arbitrarily as the victim.
*Step 4:* Identify the $RMT$s of victim cell rule that generates attractors.
*Step 5:* Replace the value of each such $RMT$ by its complement.
*Step 6:* Report the final $CA$.

*Example 4.* This example illustrates the execution steps of *Algorithm 2*. Consider the three 4-bit single cycle attractors - 0010 (2), 0101 (5) and 1111 (15). The

algorithm scans each attractor from left to right and sets the $RMT$s accordingly. Since the $CA$ is a 3-neighborhood $CA$, a 3-bit window can be considered that slides from left to right, assuming the left of the leftmost bit and right the rightmost bit are 0. Hence while scanning attractor 2, it sets $RMT$ 0 (000) of first rule as 0, $RMT$ 1 (001) of $2^{nd}$ rule as 0, $RMT$ 2 (010) of third rule as 1 and 0 to $RMT$ 4 (100) of the $4^{th}$ rule. Similarly, the attractors 3 and 15 can be considered to fill up the $RMT$s of cell rules. However, a number of $RMT$s remain unfilled. The unfilled $RMT$s are set in such a way that those can not contribute to produce attractors. Here for simplicity, we set each unfilled $RMT$ $abc$ by $b'$, where $b'$ is the complement of $b$ (underlined $RMT$s of *Table 2*). The $CA$ synthesized from *Algorithm 3*, for a given set of attractors S, may have additional attractors called the spurious attractors that are not belong to S. For example, *Algorithm 3* outputs a $CA < 14, 145, 191, 65 >$ for the attractor set - {0010 (2), 1011 (11) and 1111 (15)}. The attractor set of the CA contains two spurious attractors– 1010 (10) and 0011 (3) (*Theorem 1*).

**Table 2.** Formation of $CA < 8, 181, 151, 69 >$

| RMT | 111 (7) | 110 (6) | 101 (5) | 100 (4) | 011 (3) | 010 (2) | 001 (1) | 000 (0) | Rule |
|---|---|---|---|---|---|---|---|---|---|
| *First cell* | $d$ | $d$ | $d$ | $d$ | $\underline{1}$ | 0 | $\underline{0}$ | $\underline{0}$ | 8 |
| *Second cell* | $\underline{1}$ | 0 | 1 | 1 | 0 | $\underline{1}$ | $\underline{0}$ | 1 | 181 |
| *Third cell* | $\underline{1}$ | 0 | $\underline{0}$ | 1 | 0 | $\underline{1}$ | 1 | 1 | 151 |
| *Fourth cell* | $d$ | $\underline{1}$ | $d$ | $\underline{0}$ | $d$ | $\underline{1}$ | $d$ | 1 | 69 |

The synthesis of an $n$-cell $CA$ with $k$ ($k \leq 1 + \lfloor \frac{n-1}{3} \rfloor$, *Corollary 2*) pseudo-exhaustive bits is described in the following algorithm. It exploits *Theorem 1* and constructs an n-bit attractor (A) randomly. Then a new one (say B), based on the A, is formed such that while synthesizing an MACA with the attractors A & B, a number ($2^k$-2) of spurious attractors are generated.

Algorithm 4: **SynMACAwithPE**
**Input:** n (length of $MACA$), k (number of $PE$-bits)
**Output:** non-linear $MACA$ (rule vector) with $k$ pseudo-exhaustive bit positions
*Step 1:* Randomly synthesize an $n$-bit attractor $A$.
*Step 2:* Arbitrarily identify $k$ bit positions on $A$ such that an identified bit can have at least one identified bit either of its left or right at a distance not less than 3 bit.
*Step 3:* Synthesize an $n$-bit attractor $B$ following the rules -

($i$) Identified bits of $A$ & $B$ should be complement to each other.

($ii$) If $i^{th}$ and $j^{th}$ bits are two consecutive identified bits and $|i-j| \geq 3$, $\forall i, j$, the bits starting from $(j+1)^{th}$ to $(i-1)^{th}$ positions of $A$ & $B$ are the same.

($iii$) Randomly fill up the other bits of $B$ such that the non-identified bits can not behave like an identified bit - that is, ($i$) and ($ii$) are denied.
*Step 4:* Synthesize $CA$ that includes the attractors $A$ and $B$ (*Algorithm 3*).

*Example 5.* Let us consider synthesis of an $n = 4$ cell $MACA$ with $k = 2$ $PE$ bits. Assume $A = 0101$ is randomly selected as an attractor. The $1^{st}$ and

$4^{th}$ bits are identified as $PE$. The attractor $B = 1100$ is synthesized from $A$ following *Step 3* of *Algorithm 4*. If we run *Algorithm 3*, a $CA < 9, 119, 3, 20 >$ is synthesized. It is having 4 single cycle attractors (0100, 0101, 1100, and 1101) that can be identified by the $1^{st}$ and $4^{th}$ bits (00, 01, 10, and 11) only.

The earlier discussion points to the fact that for a given attractor set [the PE-bits], we can synthesize an MACA following *Algorithm 3* [*Algorithm 4*]. However, the performance of *Algorithm 4* is limited by the number of PE-bits ($k$) expected.

## 4 Conclusion

This paper reports a detail characterization of single cycle attractors in the $CA$ state space. Pseudo-exhaustive ($PE$) bits to identify the single cycle attractors are identified. A theoretical framework is proposed to synthesize a $CA$ with specified $PE$ bits and the attractor set.

## References

1. Banks, E.R.: Information Processing and Transmission in Cellular Automata. PhD thesis, MIT (1971)
2. Pal Chaudhuri, P., Roy Chowdhury, D., Nandi, S., Chatterjee, S.: Additive Cellular Automata - Theory and Applications, vol. 1. IEEE Computer Society Press, California (1997)
3. Codd, E.F.: Cellular Automata. Academic Press Inc., London (1968)
4. Das, S., Das(Bit), S., Sikdar, B.K.: Non-linear Cellular Automata Based Design of Query Processor for Mobile Network. In: IEEE SMC conference, Hawaii, vol. 3, pp. 2751–2756 (October 2005)
5. Ganguly, N.: Cellular Automata Evolution: Theory and Applications in Pattern Recognition and Classification. PhD thesis, Bengal Engineering College (a Deemed University), India (2004)
6. Maji, P., Shaw, C., Ganguly, N., Sikdar, B.K., Pal Chaudhuri, P.: Theory and Application of Cellular Automata For Pattern Classification. Special issue of Fundamenta Informaticae on Cellular Automata 58, 321–354 (2003)
7. Martin, O., Odlyzko, A.M., Wolfram, S.: Algebraic Properties of Cellular Automata. Comm. Math. Phys. 93, 219–258 (1984)
8. Moore, E.F.: Machine models of self reproduction. In: Burks, A.W. (ed.) Essays on Cellular Automata, University of Illinois Press, Urbana (1970)
9. von Neumann, J.: The theory of self-reproducing Automata. In: Burks, A.W. (ed.), Univ. of Illinois Press, Urbana (1966)
10. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55(3), 601–644 (1983)
11. Wuensche, A., Lesser, M.J.: The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata. Santa Fe Institute Studies in the Science of Complexity. Addison Wesley, Reading (1992)

# Characterization of Non-reachable States in Irreversible $CA$ State Space*

Sukanta Das[1] and Biplab K. Sikdar[2]

[1] Department of Information Technology
sukanta@it.becs.ac.in
[2] Department of Computer Science & Technology
biplab@cs.becs.ac.in
Bengal Engineering & Science University, Shibpur, West Bengal, India, 711103

**Abstract.** This paper targets characterization of the non-reachable states of 1-dimensional irreversible cellular automata ($CA$). A theoretical framework has been developed to design algorithms for computing the number of non-reachable states as well as the number of single cycle attractors of such a $CA$.

**Keywords:** Irreversible $CA$, non-reachable states, reachability tree, attractor.

## 1 Introduction

In the early 1950s, von Neumann and Stan Ulam [6] initiated the concept of cellular automata ($CA$). Stephen Wolfram first studied a family of simple 3-neighborhood 1-dimensional cellular automata that could simulate complex behaviors [7]. This structure attracted a large section of researchers working in the diverse fields and a specialized class of 1-dimensional $CA$, called linear/additive $CA$, had gained the primary attention [1]. The matrix algebraic tool provided the framework for characterization of linear/ additive $CA$. However, characterization of 3-neighborhood nonlinear $CA$ is yet to be explored. This motivates us to concentrate on the non-linear $CA$ - its characterization and analysis of its state space. In this work, we target the special class of $CA$ called irreversible $CA$. The *non-reachable* states and the *attractors* of irreversible $CA$ are characterized. The theoretical framework thus developed leads to the design of algorithms for computing the number of non-reachable states as well as the number of single cycle attractors in an irreversible $CA$.

## 2 Cellular Automata Basics

A Cellular Automaton ($CA$) consists of a number of cells organized in the form of a lattice. It evolves in discrete space and time, and can be viewed as an

* This research work is supported by the Sponsored CA Research Projects, Bengal Engineering and Science University, Shibpur, West Bengal, India-711103.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 160–167, 2008.

autonomous finite state machine ($FSM$). Each cell stores a discrete variable at time $t$ that refers to the present state of the cell. The next state of the cell at $(t+1)$ is affected by its state and the states of its *neighbors* at time $t$. In this work, we concentrate on such 3-neighborhood $CA$ (self, left and right neighbors), where a $CA$ cell is having two states - 0 or 1. Therefore, the next state $S_i^{t+1}$ of the $i^{th}$ $CA$ cell is specified by the *next state function* $f_i$ as

$$S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t) \tag{1}$$

where $S_{i-1}^t$, $S_i^t$ and $S_{i+1}^t$ are the present states of the neighbors at time $t$.

The $\mathcal{S}^t = (S_1^t, S_2^t, \cdots, S_n^t)$ is the present state of an $n-$cell $CA$ and

$$\mathcal{S}^{t+1} = (f_1(S_0^t, S_1^t, S_2^t), f_2(S_1^t, S_2^t, S_3^t), \cdots, f_n(S_{n-1}^t, S_n^t, S_{n+1}^t)) \tag{2}$$

If $S_0^t = S_n^t$ and $S_{n+1}^t = S_1^t$ (that is, left neighbor of the left most cell is the right most cell and vice versa), then the $CA$ is referred to as *periodic boundary* $CA$. On the other hand, if $S_0^t = S_{n+1}^t = 0$, the $CA$ is *null boundary*.

If the next state function of the $i^{th}$ cell is expressed in the form of a truth table, then the decimal equivalent of its output is conventionally referred to as the 'Rule' $\mathcal{R}_i$ [7]. In a two-state 3-neighborhood $CA$, there can be a total of $2^8$ (256) rules. Three such rules 90, 150, and 75 are illustrated in *Table 1*. The first

**Table 1.** Truth table for rule 90, 150 and 75

| Present state : | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 | *Rule* |
|---|---|---|---|---|---|---|---|---|---|
| (*RMT*) | (7) | (6) | (5) | (4) | (3) | (2) | (1) | (0) | |
| (i) Next State : | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 90 |
| (ii) Next State : | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 150 |
| (iii) Next State : | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 75 |

Note: *RMT* stands for *Rule Min Term*. The value 0/1 noted in $3^{rd}/4^{th}/5^{th}$ row shows the output of the three variable switching function.

row of the table lists the possible $2^3$ (8) combinations of the present states of $(i-1)^{th}$, $i^{th}$ and $(i+1)^{th}$ cells at time $t$. The last three rows indicate the next states of the $i^{th}$ cell at $(t+1)$ for the rules, 90, 150 and 75 respectively.

**Definition 1.** *A rule is* **Balanced** *if it contains equal number of 1s and 0s in its* $8-$*bit binary representation; otherwise it is an* **Unbalanced** *rule.*

**Definition 2.** *The set of rules* $\mathcal{R} =< \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_i, \cdots, \mathcal{R}_n >$ *that configures the cells of a* $CA$ *is called the* **rule vector**.

The sequence of states generated (state transitions) during its evolution with time directs the $CA$ behavior. The state transition diagram of an irreversible $CA$ may contain *cyclic* (lies in a cycle) and *non-cyclic* states (*Fig.1*). Further, in an *irreversible* $CA$ there are some states that are not reachable (*non-reachable*

states) from the other state and some states are having more than one predecessor [4,5]. For example, the states marked as 5 and 13 of *Fig.1* are the non-reachable states. Whereas 15 and 7 have more than one predecessor.

An irreversible $CA$ contains one or more cycles, called attractors. *Fig.1* contains two cycles – one of length 3 ($7 \rightarrow 3 \rightarrow 11 \rightarrow 7$) and other is of length 1 (15). This paper concentrates on the characterization of non-reachable states and the attractors of length 1 (single cycle attractors) in an irreversible $CA$. Here we refer such single cycle attractors as simply attractors. The next section introduces the concept of Reachability tree to formalize the characterization.

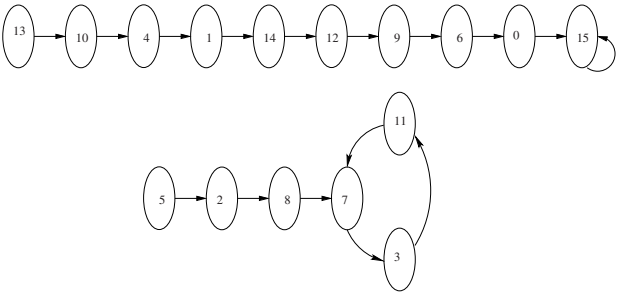



**Fig. 1.** State transitions of an irreversible $CA < 105, 177, 171, 75 >$

## 3 Reachability Tree

Reachability Tree, proposed in [2,3], is a binary tree that represents the reachable states of a $CA$. Each node of the tree is constructed with $RMT$(s) of a rule (*Section 2*). The left edge of a node of the tree is considered as the 0-edge and the right edge is as the 1-edge (*Fig.2*). The number of levels of the reachability tree for an $n-$cell $CA$ is $(n+1)$. Root node is at Level 0 and the leaf nodes are at Level $n$. The nodes of Level $i$ are constructed following the selected $RMT$s of $(i+1)^{th}$ $CA$ cell rule $\mathcal{R}_{i+1}$, while computing the next state.

The number of leaf nodes in the reachability tree denotes the number of reachable states of a $CA$ and a sequence of edges from the root to a leaf node, representing an $n-$bit binary string, is the reachable state. The binary string is formed assuming that the 0-edge and 1-edge represent 0 and 1 respectively.

During next state computation of a CA cell, the $RMT$s of the rule configuring the CA cell take the leading role. However, the $RMT$s of two consecutive cell rules $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$ are related while the $CA$ changes its state. Since the $CA$ is in 3-neighborhood, the $RMT$s are of 3-bit. So, a three bit window can be considered that slides over the present state, from left to right, to get the next state [2]. If the $RMT$ window for $i^{th}$ cell is $(b_{i-1}b_ib_{i+1})$, $b_i = 0/1$, then the $RMT$ window for $(i+1)^{th}$ cell will be either $(b_ib_{i+1}0)$ or $(b_ib_{i+1}1)$. In other words, if the $i^{th}$ $CA$ cell changes its state following the $RMT$ $k$ (decimal equivalent of $b_{i-1}b_ib_{i+1}$) of rule $\mathcal{R}_i$, then the $(i+1)^{th}$ cell will generate its next state following the $RMT$

**Table 2.** Relationship between $RMT$s of cell $i$ and cell $(i+1)$ for next state computation

| $RMT$ at $i^{th}$ rule | $RMT$s at $(i+1)^{th}$ rule |
|---|---|
| 0 | 0, 1 |
| 1 | 2, 3 |
| 2 | 4, 5 |
| 3 | 6, 7 |
| 4 | 0, 1 |
| 5 | 2, 3 |
| 6 | 4, 5 |
| 7 | 6, 7 |

**Table 3.** $RMT$s of the $CA < 8, 112, 44, 68 >$

| RMT | 111 (7) | 110 (6) | 101 (5) | 100 (4) | 011 (3) | 010 (2) | 001 (1) | 000 (0) | Rule |
|---|---|---|---|---|---|---|---|---|---|
| *First cell* | *d* | *d* | *d* | *d* | 1 | 0 | 0 | 0 | 8 |
| *Second cell* | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 112 |
| *Third cell* | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 44 |
| *Fourth cell* | *d* | 1 | *d* | 0 | *d* | 1 | *d* | 0 | 68 |

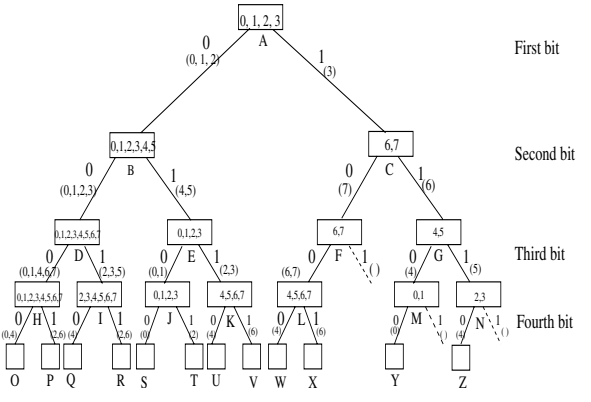



**Fig. 2.** Reachability Tree for the $CA < 8, 112, 44, 68 >$

$2k \ mod \ 8$ $(b_i b_{i+1} 0)$ or $(2k+1) \ mod \ 8$ $(b_i b_{i+1} 1)$ of rule $\mathcal{R}_{i+1}$. This relationship between the $RMT$s of $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$, while computing the next state of a $CA$, is shown in *Table 2*. The relation, noted in the table, plays an important role for characterizing the $CA$ behavior configured with different cell rules.

*Fig.2* is the reachability tree for a $CA < 8, 112, 44, 68 >$. The $RMT$s of the $CA$ rules are noted in *Table 3*. The decimal numbers within a node at level $i$ represent the $RMT$s of the $CA$ cell rule $\mathcal{R}_{i+1}$ following which the cell $(i+1)$ may change its state. The $RMT$s of a rule for which we follow 0-edge or 1-edge are noted in the bracket. For example, the root node (level 0) of *Fig.2* is constructed with $RMT$s 0, 1, 2 and 3 as cell 1 (rule 00001000) can change its state following any one of the $RMT$s 0, 1, 2, and 3. As the state of its left neighbor is always 0, the $RMT$s 4, 5, 6 & 7 are the *don't care*s for cell 1. It is obvious from *Fig.2* that there are 12 possible sequences of edges in the tree. That is, 12, out of 16, $CA$ states are reachable and the rests are non-reachable. Based on the theory of

Reachability tree, we next report the proposed characterization of non-reachable states and the attractors of a $CA$.

## 4 Characterization of $CA$ Targeting Non-reachable States and the Attractors

This section presents a scheme to characterize the irreversible $CA$ states. It identifies the non-reachable states and also computes the number of non-reachable states of an irreversible $CA$ in linear time. It also finds the number of single cycle attractors of a $CA$. The theoretical aspects of such characterization are formulated in the following theorems.

**Theorem 1.** *An $n-$cell irreversible $CA$ contains at least $2^{n-3}$ non-reachable states.*

*Proof.* In 3-neighborhood, $\frac{1}{8}$ of the total $CA$ states are to be determined by each of the 8 $RMT$s of $i^{th}$ $CA$ cell rule $\mathcal{R}_i$. Since the $CA$ is irreversible, there is at least one $RMT$ of $\mathcal{R}_i$ that causes an unbalanced reachability tree for the $CA$. Therefore, $\frac{1}{8}$ of total states are obviously non-reachable. Hence, the number of non-reachable states is at least $\frac{2^n}{8} = 2^{n-3}$.

**Theorem 2.** *An $n-$cell irreversible $CA$ constructed only with the balanced rules contains at least $2^{n-2}$ non-reachable states.*

*Proof.* Let us consider the reachability tree for an $n-$cell irreversible $CA$, configured only with balanced rules, is balanced up to the $i^{th}$ level and rule $\mathcal{R}_i$ is responsible for that. Since $\mathcal{R}_i$ is balanced, therefore, there exist at least 2 $RMT$s that cause the tree as unbalanced. As $\frac{1}{8}$ of total states are determined by an $RMT$, total number of non-reachable states for such $CA$ is $(\frac{1}{8} + \frac{1}{8} = \frac{1}{4})$ of the total states. Hence an $n-$cell irreversible $CA$, configured with balanced rules, contains at least $2^{n-2}$ non-reachable states. Hence the proof.

**Corollary 1.** *An $n-$cell linear/additive irreversible $CA$ contains at least $2^{n-2}$ non-reachable states.*

*Proof.* Since a linear/additive rule is balanced [2,3], the result is directly followed from *Theorem 2*.

We next propose an algorithm that calculates the number of non-reachable states of an irreversible $CA$ utilizing the concept of reachability tree.

### 4.1 Computing the Number of Non-reachable States

The algorithm (*CalNonReachableStates*) assumes the variables $S$, an array of sets, and *nos* (the number of sets in $S$). The arrays $oldWeight$ and $newWeight$ are used to store the number of states that may be reachable and to store

the number of states that are non-reachable respectively. The number of non-reachable states are finally stored in the variable *NS*.

Algorithm 1: **CalNonReachableStates**
**Input**: $n$ ($CA$ size), $Rule[n][8]$ ($CA$).
**Output**: number of non-reachable states.
*Step 1*: Find $S[1] = \{j\}$, where $Rule[1][j] = 0$ and $1 \leq j \leq 3$,
and $S[2] = \{j\}$, where $Rule[1][j] = 1$ and $1 \leq j \leq 3$.
If $S[i] = \phi$ ($i$=1/2), set NS := $2^{n-1}$, oldWeight[1] := $2^{n-1}$ and $nos := 1$.
Otherwise, set oldWeight[1] := $2^{n-1}$, oldWeight[2] := $2^{n-1}$ and $nos := 2$.
*Step 2*: For $i = 2$ to $n - 1$ do 2.1 to 2.4
2.1 For $j = 1$ to $nos$
Determine $RMT$s for the next level nodes from $S[j]$ following *Table 2*.
Distribute these $RMT$s of $i^{th}$ rule with value 0 into $S'[2j - 1]$ and 1 into $S'[2j]$.
Set newWeight[2j-1] := oldWeight[j]/2 and newWeight[2j] := oldWeight[j]/2.
If $S'[k] = \phi$, set NS := NS + newWeight[k], where $k = 2j - 1, 2j$.
2.2 Replace $RMT$s 4, 5, 6 and 7 by equivalent $RMT$s 0, 1, 2 and 3 respectively for each $S'[k]$.
2.3If $S'[k] = S'[k']$ for any $k'$, set oldWeight[k] := newWeight[k] + newWeight[$k'$]; otherwise, set oldWeight[k] := newWeight[k].
2.4 Assign unique sets of $S'$ to $S$, and $nos$ := number of sets in $S$.
*Step 3*: For $j = 1$ to $nos$
Determine next $RMT$s of $S[j]$, of which 2 are invalid since it is the last rule.
Distribute these $RMT$s of last rule with value 0 into $S'[2j - 1]$ and 1 into $S'[2j]$.
If $S'[k] = \phi$, then set NS := NS +oldWeight[k]/2, where $k = 2j - 1, 2j$.
*Step 4*: Report the value of NS as the number of non-reachable states of the $CA$.

**Complexity:** Since *Algorithm 1* uses a loop in *Step 2* that depends on $n$, and the maximum value of $nos$. $nos$ is constant. Therefore, the time complexity of the algorithm is $O(n)$.

### 4.2 Computing the Number of Attractors

Since the next state of a single cycle attractor is the attractor itself (attractor 15 of *Fig.1*), there should be at least one $RMT$ (*Table 1*) of each cell rule ($\mathcal{R}_i$) for which the $CA$ $\mathcal{R}$ cell ($i$) does not change its state. For example, the $RMT$ $x0x$ ($x = 0/1$) of a rule is considered to find the next state of cell $i$ when the current states of its left neighbor ($(i-1)^{th}$ cell), self ($i^{th}$) and right neighbor ($((i+1)^{th}$ cell) are $x$, 0 and $x$ respectively. It implies, if the $RMT$ is '0', the state change of the cell ($i$) is $0 \rightarrow 0$. That is, for the rule $\mathcal{R}_i$, if the $RMT$ 0 (000), 1 (001), 4 (100) or 5 (101) is 0, the $CA$ cell $i$ configured with $\mathcal{R}_i$ does not change its state. Similarly, if the $RMT$s 2 (010), 3 (011), 6 (110) or 7 (111) is 1 in $\mathcal{R}_i$, the cell configured with $\mathcal{R}_i$ can stick to its current state in the next time step. For

| RMTs | 111 (7) | 110 (6) | 101 (5) | 100 (4) | 011 (3) | 010 (2) | 001 (1) | 000 (0) | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | Rule for cell i |

**Fig. 3.** $RMT$s of rule 204

example, if a $CA$ cell is configured with the rule 204 (*Fig.3*), all $RMT$s of the rule help formation of attractors.

*Property 1*: A rule $\mathcal{R}_i$ can contribute to the formation of single cycle attractor(s) if at least one of the $RMT$s 0, 1, 4 or 5 is 0, or the $RMT$ 2, 3, 6 or 7 is 1.
If any rule does not maintain *Property 1*, the $CA$ can not have single cycle attractors. The following algorithm *CalNoOfAttractors* scans a $CA$ rule vector $\mathcal{R}$ from left to right and explores all the attractors of the $CA$.

Algorithm 2: **CalNoOfAttractors**
**Input**: $n$ ($CA$ size), $Rule[n][8]$ ($CA$).
**Output**: NoA (number of attractors).
*Step 1*: If any rule does not maintain *Property 1*, return $NoA = 0$.
*Step 2*: If $RMT$ $j$ ($j$ =0, 1, 2, 3) is capable of generating the attractors, assign $S[1] = \{j\}$, where $Rule[1][j] = 0$, $S[2] = \{j\}$, and $Rule[1][j] = 1$.
If $S[i] = \phi$, then set oldWeight[1] := $2^{n-1}$ and $nos := 1$, where $i = 1$ or 2.
Otherwise, set oldWeight[1] := $2^{n-1}$, oldWeight[2] := $2^{n-1}$ and $nos := 2$.
*Step 3*: For $i = 2$ to $n - 1$ do 3.1 to 3.4
3.1 For $j = 1$ to $nos$
Determine $RMT$s for the next level nodes from $S[j]$ following *Table 2*.
Remove the $RMT$s that are not capable of generating attractors.
Distribute these $RMT$s of $i^{th}$ rule with value 0 into $S'[2j - 1]$ and 1 into $S'[2j]$.
Set newWeight[2j-1] := oldWeight[j]/2 and newWeight[2j] := oldWeight[j]/2.
3.2 Replace $RMT$s 4, 5, 6 and 7 by equivalent $RMT$s 0, 1, 2 and 3 respectively for each $S'[k]$.
3.3If $S'[k] = S'[k']$ for any $k'$, set oldWeight[k] := newWeight[k] + newWeight[$k'$];
Otherwise, set oldWeight[k] := newWeight[k].
3.4 Assign unique sets of $S'$ to $S$, and $nos$ := number of sets in $S$.
*Step 4*: For $j = 1$ to $nos$
Determine next $RMT$s of $S[j]$, of which 2 are invalid since it is the last rule.
Remove the $RMT$s that are not capable of generating attractors.
Distribute these $RMT$s of last rule with value 0 into $S'[2j - 1]$ and 1 into $S'[2j]$.
If $S'[k] = \phi$, then set NoA := NoA +oldWeight[k]/2, where $k = 2j - 1, 2j$.
*Step 5*: Report NoA as the number of attractors.

**Complexity:** The complexity of *Algorithm 2* is also $O(n)$ as it is for *Algorithm 1*.

## 5 Conclusion

This paper presents an efficient scheme to calculate the number of non-reachable states of an irreversible $CA$ in linear time. An algorithm is also proposed that computes the number of single cycle attractors of such a $CA$. A theoretical framework has been reported to characterize the non-reachable states as well as the attractors of the $CA$.

## References

1. Pal Chaudhuri, P., Roy Chowdhury, D., Nandi, S., Chatterjee, S.: Additive Cellular Automata - Theory and Applications, vol. 1. IEEE Computer Society Press, California (1997)
2. Das, S., Sikdar, B.K.: Classification of CA Rules Targeting Synthesis of Reversible Cellular Automata. In: Proceedings of International Conference on Cellular Automata for Research and Industry, ACRI, France, pp. 68–77 (September 2006)
3. Das, S., Sikdar, B.K., Chaudhuri, P.P.: Characterization of Reachable/Nonreachable Cellular Automata States. In: Proceedings of Sixth International Conference on Cellular Automata for Research and Industry, ACRI, The Netherlands, pp. 813–822 (October 2004)
4. Moore, E.F.: Machine models of self reproduction. In: Burks, A.W. (ed.) Essays on Cellular Automata, University of Illinois Press, Urbana (1970)
5. Myhill, J.: The converse of moore's garden of eden theorem. In: Proceedings of American Mathematical Society, vol. 14, pp. 685–686 (1963)
6. von Neumann, J.: The theory of self-reproducing Automata. In: Burks, A.W. (ed.), Univ. of Illinois Press, Urbana (1966)
7. Wolfram, S.: Cellular Automata and Complexity - Collected Papers. Addison-Wesley, Reading (1994)

# An Efficient $n \times n$ Boolean Mapping Using Additive Cellular Automata

Sourav Das and Dipanwita RoyChowdhury

Dept. of CSE, Indian Institute of Technology, Kharagpur, India
{souravd,drc}@cse.iitkgp.ernet.in

**Abstract.** The cellular automata (CA) have found its application in diverse fields. But in order to utilize the whole state space of a maximum length CA one has to run $2^n - 1$ clock cycles. Using the closure property of a group CA this paper shows how to reach a state which is far from the initial state using very few cycles. It also develops two algorithms to utilize the whole state space of CA using very few cycles. Using these algorithms it creates $n \times n$ pseudo-random boolean mappings having very good statistical properties.

**Keywords:** Additive Cellular Automata, Pseudo-Random Number Generator.

## 1 Introduction

In order to utilize the whole state space of a CA, a very high number of cycles is required. For an $n$ cell maximum length CA the number of cycles required to utilize the whole state space is as high as $2^n$. So for a large length of CA this exponential growth of the number of cycles to utilize the whole state space is not very practical. This might limit the utilization of CA having very large length, even if the properties of the CA may be useful for different types of application.

In this work we present two algorithms for exploring the whole state space of CA by running very few cycles. Using these algorithms an efficient $n \times n$ boolean mapping can be generated. There can be a total of $2^n - 1$ different $n \times n$ mappings for $2^n - 1$ (except all zeros as a seed) seeds for each of the algorithms. We evaluate some of these mappings with NIST statistical test suit [2] to evaluate the randomness of these mappings. The evaluation shows promising results that can be used to qualify these mappings as random.

We start this paper with a brief introduction of cellular automata followed by how to reach a very high state using a very few cycles in section 2. In section 3 we describe two heuristic algorithms to run through the whole state space of the CA. Using these algorithms, we mention how to generate highly random $n \times n$ mappings with hardware implementation and statistical results in section 4.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 168–173, 2008.

## 2 Reaching States Far from Initial State of CA Using Very Few Cycles

The CA structure can be viewed as a lattice of cells that evolve in each time step depending on some combinational logic on itself and its neighbors. The characteristic matrix of a CA operating over GF(2) is a matrix that describes the behavior of the CA [1]. A characteristic matrix is constructed as:

$T[i,j]$ =1, if the next state of the ith cell depends on jth cell
=0, otherwise

If $S(t)$ represents the state of the CA at the $i$th instant of time then the state at the next time instant can be represented as:
$S(t+1) = [T]S(t)$ and $S(t+2) = [T]^2S(t)$ and so on. So we can write:
$S(t+p) = [T]^pS(t)$.

The CA can be used to generate a very good set of pseudorandom generators [3], [4], [5], [6]. However, when the length of the CA is high, the number of cycles required to utilize the whole state space is very high. For example, $2^n - 1$ clock cycles are required to utilize the whole state space of an n-cell maximum length CA. This makes the CA based applications slow for larger length of CA.

Here we present a theoretical scheme to achieve a high degree of $T$ matrix by running a very few cycles of the CA or, equivalently, multiplying the $T$ matrices for a very few times.

$$T^i \oplus T^j = T^k \tag{1}$$

This can also be expressed in terms of states of the CA as,

$$S(t+i) \oplus S(t+j) = S(t+k) \tag{2}$$

where, $k \gg i, j$. The existence of equation 1 and equation 2 comes from the closure property of the group CA [1].

*Example 1.* In a four cell maximum length CA having rule vector $\langle 90, 150, 90, 150\rangle$, consider, $T^1$ and $T^3$, XORing the two we get $T^9$. So here we can reach $T^9$ by running only three cycles.

*Example 2.* In a sixteen cell maximum length group CA, consider, $T^1$ and $T^2$, XORing the two we get $= T^{60630}$. So here we can reach $T^{60630}$ by running only two cycles.

## 3 Exploring Full State Space

After finding an easy way to find a very high power of $T$ matrix by running very few cycles, the next challenge is to utilize the whole state space of the CA. It is obvious that if we use lesser number of bits for $i$ and $j$ in the equation $T^i \oplus T^j = T^k$, we cannot utilize the whole state space of the CA. Also the mapping is not one to one as two different pairs of $(i, j)$ can give rise to the same value of $k$.

However, we have found two linear-time algorithms that give unique value for $T^k$ and also utilizes the whole state space. The first algorithm (algorithm 1) needs $2n$ cycles whereas the second algorithm(algorithm 2) needs only $n$ cycles to run through $2^n$ state space. Even if the second algorithm is an improvement of the first one, both run in linear time and can be used to generate different $n \times n$ boolean mappings. Hence, we mention both algorithms here with more emphasis on the algorithm 2.

**Algorithm 1.** Utilization of Whole State Space Using $2n$ Cycles

**Require:** Choose any seed. Take $n$ bit maximum length CA.
1: Take an $n$ bit input.
2: $a = [0]$
3: **for** $i = 1$ to $n$ **do**
4: **if** $i$th bit of input= 1 **then**
5: $a = a \oplus T^{2*i-1}$
6: **else**
7: $a = a \oplus T^{2*(i-1)}$
8: **end if**
9: **end for**
10: Output $a$

Normally, running through the whole state space of cellular automata requires exponential time but the algorithm 1 runs in linear time with $2n$ cycles. The example 3 shows how this mapping works.

*Example 3.* Let us take n=8 and show how the proposed method runs through the whole state space for all different inputs by running only 8*2=16 cycles. Consider a seed 10101010.

For an input 00000000

The output of the above algorithmic function is:

$(T^0 \oplus T^2 \oplus T^4 \oplus T^6 \oplus T^8 \oplus T^{10} \oplus T^{12} \oplus T^{14}) * seed = 00011110$

Let us denote this function as $f$.

Then,

$f(10101010) = (T^0 \oplus T^3 \oplus T^4 \oplus T^7 \oplus T^8 \oplus T^{11} \oplus T^{12} \oplus T^{15}) * seed = 00101110$

Again, $f(11111111) = (T^1 \oplus T^3 \oplus T^5 \oplus T^7 \oplus T^9 \oplus T^{11} \oplus T^{13} \oplus T^{15}) * seed = 00100111$

and so on.

The algorithm 2 requires only $n$ cycles to create $n \times n$ boolean mapping. The example 4 shows how this mapping works in algorithm 2.

*Example 4.* Let us take n=8 and show how the proposed method runs through the whole state space for all different inputs by running only 8 cycles using algorithm 2. Consider a seed 10101010.

For an input 00000000

The output of the above algorithmic function is:

$(T^{2^8-1} = T^{255} = I) * seed = 10101010$

**Algorithm 2.** Utilization of Whole State Space Using $n$ cycles

**Require:** Choose any seed. Take $n$ bit maximum length CA.

```
1: Take an n bit input.
2: a = [I]
3: for i = 1 to n do
4:    if ith bit of input= 1 then
5:       a = a ⊕ T^i
6:    else
7:       {Do Nothing}
8:    end if
9: end for
10: Output a
```

Let us denote this function as $f$.

Then, $f(00010100) = (I \oplus T^3 \oplus T^5) * seed = 01000110$

Again, $f(10101010) = (I \oplus T^2 \oplus T^4 \oplus T^6 \oplus T^8) * seed = 10100001$

Again, $f(11111111) = (I \oplus T^1 \oplus T^2 \oplus T^3 \oplus T^4 \oplus T^5 \oplus T^6 \oplus T^7 \oplus T^8) * seed = 00001101$

and so on.

Note that, different boolean functions will be generated for different seeds in all the examples. We have verified in software that both algorithms 1 and 2 do run through the whole state space of a Cellular Automata of length 4, 8 and 16 with different seeds and it can be easily verified computationally for other lengths of cellular automata.

Both algorithms 1 and 2 have been characterized using the $T$ matrices. This is equivalent to running a number of cycles by choosing a seed and the corresponding rule vector. Since $T^i$ is equivalent to running a CA for $i$ times, this algorithm is equivalent to choosing a CA and running suitable number of cycles.

## 4 An $n \times n$ Boolean Mapping

Using the theory developed above to use the whole state space of a Cellular Automata, we develop an $n \times n$ boolean mapping. This mapping takes an input of $n$ bits that determines the number of cycles that the CA needs to run that will be XORed to give the $n$ bit output. The mapping also takes an $n$ bit input for the seed of the CA which will be constant. For different seeds a different mapping will be generated.

We have used different seeds using algorithms 1 and 2 to get different $8 \times 8$ mappings. In fact it is possible to generate $2^8 - 1 = 255$ (excluding all zeros input) for eight cell CA for each of the algorithm. Table 1 shows an example of such mappings generated with an input seed 10101010 using algorithm 2.

### 4.1 Statistical Tests

To evaluate the mappings for statistical randomness, we have used NIST statistical test suit [2]. This test tool contains sixteen statistical tests to evaluate

**Table 1.** Generated $8 \times 8$ Boolean Mapping with Initial Seed 10101010

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| aa | 81 | 69 | 42 | 8c | a7 | 4f | 64 | f1 | da | 32 | 19 | d7 | fc | 14 | 3f |
| 60 | 4b | a3 | 88 | 46 | 6d | 85 | ae | 3b | 10 | f8 | d3 | 1d | 36 | de | f5 |
| 9b | b0 | 58 | 73 | bd | 96 | 7e | 55 | c0 | eb | 03 | 28 | e6 | cd | 25 | 0e |
| 51 | 7a | 92 | b9 | 77 | 5c | b4 | 9f | 0a | 21 | c9 | e2 | 2c | 07 | ef | c4 |
| c1 | ea | 02 | 29 | e7 | cc | 24 | 0f | 9a | b1 | 59 | 72 | bc | 97 | 7f | 54 |
| 0b | 20 | c8 | e3 | 2d | 06 | ee | c5 | 50 | 7b | 93 | b8 | 76 | 5d | b5 | 9e |
| f0 | db | 33 | 18 | d6 | fd | 15 | 3e | ab | 80 | 68 | 43 | 8d | a6 | 4e | 65 |
| 3a | 11 | f9 | d2 | 1c | 37 | df | f4 | 61 | 4a | a2 | 89 | 47 | 6c | 84 | af |
| 08 | 23 | cb | e0 | 2e | 05 | ed | c6 | 53 | 78 | 90 | bb | 75 | 5e | b6 | 9d |
| c2 | e9 | 01 | 2a | e4 | cf | 27 | 0c | 99 | b2 | 5a | 71 | bf | 94 | 7c | 57 |
| 39 | 12 | fa | d1 | 1f | 34 | dc | f7 | 62 | 49 | a1 | 8a | 44 | 6f | 87 | ac |
| f3 | d8 | 30 | 1b | d5 | fe | 16 | 3d | a8 | 83 | 6b | 40 | 8e | a5 | 4d | 66 |
| 63 | 48 | a0 | 8b | 45 | 6e | 86 | ad | 38 | 13 | fb | d0 | 1e | 35 | dd | f6 |
| a9 | 82 | 6a | 41 | 8f | a4 | 4c | 67 | f2 | d9 | 31 | 1a | d4 | ff | 17 | 3c |
| 52 | 79 | 91 | ba | 74 | 5f | b7 | 9c | 09 | 22 | ca | e1 | 2f | 04 | ec | c7 |
| 98 | b3 | 5b | 70 | be | 95 | 7d | 56 | c3 | e8 | 00 | 2b | e5 | ce | 26 | 0d |

a random number generator for cryptographic purposes. We have used a CA of length 20 with a particular seed and the output for all the $2^{20}$ inputs were fed to the tool. Out of sixteen tests, twelve tests passed, two tests failed and two tests passed for some runs of the test and failed for other runs. Table 2 shows the result of the statistical tests.

**Table 2.** Statistical Tests Summary

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Frequency MonoBit | Pass | Runs | Pass | Lempel Ziv | Pass | Longest Run of ones in a block | Pass |
| Linear Complexity | Pass | Rank | Fail | Random Excursion | Some Runs Pass | overlapping template matching | Pass |
| Block Frequency | Pass | DFT | Pass | Approximate Entropy | Pass | Non-overlapping template matching | Pass |
| Serial Test | Pass | CUSUM | Pass | Random Excursion Variant | Some Runs Pass | Universal Statistical Test | Fail |

### 4.2 Implementation

The biggest advantage of this mapping is that its hardware implementation is very easy and extremely efficient apart from providing high scalability and modularity. There can be three different approaches of hardware implementation: Time Optimized (running using only one clock cycle), Space Optimized (using a much smaller number of gates, even if at the expense or requiring a higher number of machine cycles) and finally Both Time and Space Optimized (where both the above approaches are used). Both algorithms can be implemented easily in hardware using any of the approaches above.

The figure 1 shows the schematic of hardware implementation of algorithm 2 using just one cycle. Note that running a number of cycles of a CA can be equivalently realized using the powers of $T$, its characteristic matrix.

In this implementation, we realize $n$ number of powers of $T$ matrices from $T^1, T^1, T^2, .., T^n$ in parallel. Depending on each bit in the n bit input, the corresponding powers of $T$ transformation will be selected by simple AND operation for XORing to give the output. In addition, the seed is also fed to the XOR directly to realize $I$. Each $T$ transformation can be realized by $n$ input (worst case) XOR gates for each output bits and a total of $n$ such XOR gates for $n$ bit

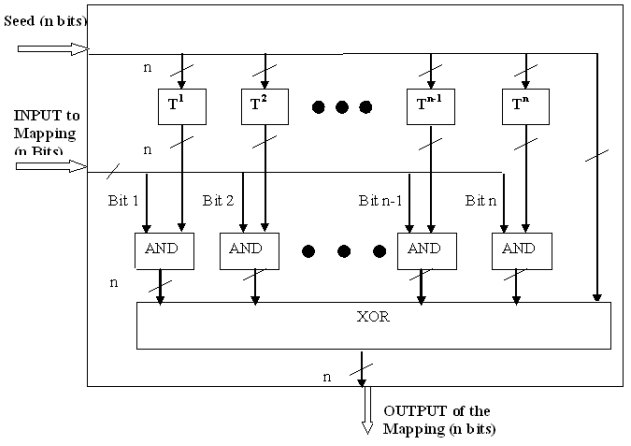



**Fig. 1.** Time Optimized Hardware Implementation of Algorithm 2

output. There are $n$ such transformations, giving rise to a total of $n^2$ such XOR gates required to realize all the $T$ transformations. The final XORing of selected $T$ matrix outputs needs another $n$ input XOR gates for each of the $n$ bit output.

## 5 Conclusion

In this paper we have utilized the closure property of group CA to reach a very high state in the additive cellular automata state space using very few cycles. We have devised two linear time algorithms to explore the full state space of CA that can be used in an $n \times n$ boolean mapping generation. As a future work this theory can be applied various applications including cryptography, coding theory and VLSI test pattern generation.

## References

1. Chaudhury, P.P., Chowdhury, D.R., Nandi, S., Chattopadhay, S.: Additive Cellular Automata Theory and Application, vol. 1. IEEE Computer Society Press, Los Alamitos
2. NIST Statistical Test Suit, http://csrc.nist.gov/rng/
3. Serra, M., Slater, T., Muzio, J.C., Miller, D.M.: Analysis of One Dimensional Cellular Automata and their Aliasing Probabilities. IEEE Trans. on CAD 9(7), 767–778 (1990)
4. Tomassini, M., Sipper, M., Perrenoud, M.: On the Generation of High-Quality Random Numbers by Two-dimensional Cellular Automata. IEEE Trans. on Computers 49(10), 1146–1151 (2000)
5. Nandi, S., et al.: Analysis of Periodic and Intermediate Boundary 90/150 Cellular Automata. IEEE Trans. on Computers 45(1), 1–12 (1996)
6. Hortencius, P.D., McLeod, R.D., Pries, W., Miller, D.M., Card, H.C.: Cellular Automata based Pseudo-random Number Generators for Built-in Self-Test. IEEE Trans. on CAD 8(8), 842–859 (1989)

# Controlling the Dynamics of the Fuzzy Cellular Automaton Rule 90, I.

Samira El Yacoubi[1] and Angelo Mingarelli[2]

[1] Laboratory of Mathematics, Physics and Systems, University of Perpignan
52, Paul Alduy Avenue, 66860 Perpignan, Cedex, France
yacoubi@univ-perp.fr
[2] School of Mathematics and Statistics, Carleton University,
Ottawa, Canada, K1S 5B6
amingare@math.carleton.ca

**Abstract.** Control problems on Cellular Automata (CA) models have been introduced in a rigorous mathematical framework [10]. In this paper, we attempt to apply the control theory concept to the special class of fuzzy CA for which more freedom is gained using a continuum state space. Focusing on the case of fuzzy rule 90, we investigate the possibility of finding a control $u = (u_0, u_1, \cdots, u_{T-1})$ which forces the system at a localized cell, to achieve a given desired state at time $T$. The problem is studied starting from an initial configuration consisting of a single seed on a zero background.

**Keywords:** Fuzzy Cellular automata, long-term evolution, Control.

## 1 Introduction

Cellular automata constitute a very interesting modelling approach which has been explored from mathematical and computational points of view for theoretical as well as practical aspects [17,4,3,13,14,19]. However, from the mainstream literature, CA in their classical form are treated as closed systems, as they do not take into account the interaction between the system and its environment. Considering control problems on systems using CA approaches should be beneficial for this field of research and makes connections with the field of systems theory.

The basic idea of control theory states that systems behavior is caused by a response to an outside stimulus and may be influenced so as to achieve a desired goal [18]. In order to implement this influence, engineers build devices that incorporate various mathematical techniques.

An appropriate way of introducing controls in CA models in order to make them more useful in systems theory has been given in [10,6]. Some concepts related to the control theory (regional controllability, identification, spreadability) has been studied mainly in the case of additive CA [7,8,9]. However, the problem of obtaining analytical results is still posed.

We consider in this paper the fuzzy version of CA which constitute a direct generalization of the classical binary CA models. These elementary cellular

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 174–183, 2008.

automata (ECA) which have been studied by Wolfram and others, are good examples of systems with simple rules that may produce unusually complex behavior. We investigate the *Fuzzy CA* (FCA) as a real-valued version of ECA which seem to provide the best results regarding the control problems.

Considering a FCA evolution on a time interval $[0, T]$, we will address the question whether some particular target state is reachable starting from a specific initial condition. The control value to be found is given by the vector $u = (u_0, u_1, \cdots, u_{T-1})$. We enlarge the state space from $[0, 1]$ to $\mathbb{R}$ in order to obtain more flexibility regarding the control values and then a necessary and sufficient condition is found. The same result is obtained when working on $[0, 1]$ but only for specific desired states or small values of $T$.

## 2 Basic Definitions

### 2.1 Cellular Automata

A cellular automaton (CA) may be thought of as a linear collection of cells where all cells share the same local space (i.e., the set of values for the cells) the same neighborhood structure (i.e., the cells on each side of a cell), and the same local function or rule (i.e., the function defining the effect of the neighbors on each cell, also called the transition function or rule function). The global evolution of the CA is defined by the synchronous update of all cell values according to repeated applications of the local function to the neighborhood of each cell. A configuration of the automaton is a state of all lattice cells [22].

Cellular automata were one of the first abstract models for parallel computing. Conceived by John von Neumann [17] in the early 1950's to investigate self-reproduction, CA have been used mainly for studying parallel computing methods and the formal properties of model systems.

Given a bi-infinite lattice of cells on a line, the local space $\{0, 1\}$, the usual neighborhood structure $\langle$ left neighbor, itself, right neighbor $\rangle$, and a rule function $g : \{0, 1\}^3 \longrightarrow \{0, 1\}$, the global dynamics of an *elementary CA* are defined by:

$$f : \{0, 1\}^{\mathbb{Z}} \longrightarrow \{0, 1\}^{\mathbb{Z}}$$

$$\forall i \in \mathbb{Z}, f(x)_i = g(x_{i-1}, x_i, x_{i+1}).$$

The rule function or *local rule* is then defined by the 8 possible local configurations a cell detects in its direct neighborhood:

$$(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (r_0, \cdots, r_7),$$

where each triplet above represents a local configuration of the left neighbor, the cell itself, and the right neighbor. In general, the value of the sum $\sum_{i=0}^{7} 2^i r_i$ is used as the name of the rule. It is well-known that the local rule of any boolean CA can be expressed canonically as a *disjunctive normal form (DNF)*, that is,

$$g(x_1, x_2, x_3) = \vee_{i|r_i=1} \wedge_{j=1:3} x_j^{d_{ij}}$$

where $d_{ij}$ is the $j$-th digit, read from left to right, of the binary expression of $i$, and $x^0$ (resp. $x^1$) stands for $\neg x$ (resp. $x$).

### 2.2 Fuzzy Cellular Automata

The initial string now consists of a set of *fuzzy* states, that is, a collection of arbitrary but fixed *real numbers* in the closed interval $[0,1]$ (as opposed to the two-point set $\{0,1\}$). The process of *fuzzification* described below entails redefining the local rule $g$ above so that it can now act on triples of real numbers (as opposed to triples of boolean numbers) and map the unit box $[0,1]^3 \in \mathbb{R}^3$ into the unit interval $[0,1]$.

Inherent in this procedure is the fact that fuzzification will allow one to move from the discrete (boolean CA) to the continuous (fuzzy CA, or FCA) by extending the domain of definition of the rule in such a way that the new "rule" agrees with the original rule when we restrict its domain to the boolean set $\{0,1\}$. We describe a natural method of fuzzifying a given boolean rule herewith, the source of which is in [2]. We adopt the now standard terminology from Flocchini *et al*, [11].

**Definition 1.** *A "fuzzy" cellular automaton or fuzzy CA or FCA for brevity, is obtained by* fuzzifying *the local function of a given boolean CA in the following way: For real numbers $a, b \in [0,1]$ we redefine the quantities $(a \vee b)$ to be $(a+b)$, $(a \wedge b)$ to be $(ab)$, and $(\neg a)$ to be $(1-a)$ in the DNF. Thus $a \vee b = a+b$, $a \wedge b = a \cdot b$, and $\neg a = 1-a$, where $+$ and "$\cdot$" are ordinary addition and multiplication of real numbers.*

The case under consideration is rule 90. Since $90 = 2^1 + 2^3 + 2^4 + 2^6$ we see that its rule number, $90 = \sum_{i=0}^{7} r_i\, 2^i$, forces $r_i = 1$ for $i = 1, 3, 4, 6$. Using the DNF above gives us the rule function of FCA 90 in the form

$$g(x, y, z) = x + z - 2xz, \tag{1}$$

for $(x, y, z) \in [0,1]^3$. We emphasize that in "fuzzifying" the DNF (**??**), we replaced $\neg x$ by $1 - x$, $x \vee y$ by $x + y$, and $x \wedge y$ in (**??**) by their product, $x \cdot y$.

In this case, the local fuzzy rule maps the triples of zeros and ones as follows:

$$000, 001, 010, 011, 100, 101, 110, 111 \to 0, 0, 1, 1, 1, 0, 1, 0.$$

## 3 Control of Fuzzy Rule 90

### 3.1 Problem Statement

CA have been used extensively as a modelling tool to approximate nonlinear discrete and continuous dynamical systems in a variety of applications. However the inverse problem of determining a/the CA that satisfies some specified constraints has received very little attention. A possible formulation for an inverse problem involves the search of an appropriate CA rule capable of carrying a given system from an initial state to a desired final configuration during a time horizon $T$.

If the rule has the form $s_{t+1} = F_u(s_t, u_t)$, the problem, usually referred to as the *controllability problem*, consists in finding a control $u = (u_0, u_1, \cdots, u_{T-1})$ in an appropriate control space such that, for some $T \geq 0$,

$$s_T = S_d$$

where $S_d$ is the desired state, given in a suitable space of so-called *reachable states* and $s_T$ is the CA configuration at time $T$.

### 3.2 The Case of an Excited Cell with a Single Initial Seed in a Background of Zeros

The notion of controllability is geared to the possibility of forcing a system into a particular state by using one or more appropriate control signals. In this work we consider the case where the signals are applied at $t = 0, 1, \ldots, T-1$ so as to influence the space-time diagram of the rule in order to achieve a desired state $A^T_{-T}, \cdots, A^T_{-1}, A^T_0$, at time $T$ (cf., Table 1 below).

Hence the work presented here is related to the most general problem of control theory using Cellular Automata models. We solve it in the case where the initial string consists of two cells, one of which is a given fuzzy state $x^0_0 = a$, which may or may not be in $[0, 1]$, and the other of which is a control, $x^0_1 = u_0$, under which, in the cells $x^i_1$, $i = 1, 2, \ldots, T-1$, there is a string of $T-1$ other controls, all immersed in a background of zeros (the so-called *homogeneous background* case). The space-time diagram under consideration is of the form of Table 1

**Table 1.** The space time diagram showing the evolution of a rule function starting from a single seed **a** with a column of values assigned to the controlled cell

| | $-T$ | $\cdots$ | $-3$ | $-2$ | $-1$ | $0$ | $1$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $\cdots$ | 0 | 0 | 0 | $a$ | $u_0$ |
| 1 | 0 | $\cdots$ | 0 | 0 | $a$ | $u_0$ | $u_1$ |
| 2 | 0 | $\cdots$ | 0 | $a$ | $u_0$ | . | $u_2$ |
| $\vdots$ | | | $\cdots$ | | | $\cdots$ | $\vdots$ |
| $i$ | $\cdots$ | 0 | $a$ | $u_0$ | $\ldots$ | $\cdots$ | $u_i$ |
| $\vdots$ | | | | | | | $\vdots$ |
| $T-1$ | 0 | $a$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $u_{T-1}$ |
| $T$ | $x^T_{-T}$ | $\cdots$ | $x^T_{-3}$ | $x^T_{-2}$ | $x^T_{-1}$ | $x^T_0$ | |

and the problem is to reach a state $(x^T_{-T}, \cdots, x^T_{-1}, x^T_0)$ which coincides with a desired one $A = (A^T_{-T}, \cdots, A^T_{-1}, A^T_0) \in \mathbb{R}^{T+1}$ at time $T$. In other words, the so-called *input-to-final-state reachability map*

$$\begin{aligned} K : \quad \mathbb{R}^T \quad &\longrightarrow \mathbb{R}^{T+1} \\ u = (u_0, u_1, \ldots, u_{T-1}) \quad &\longrightarrow x_T = (x^T_{-T}, \cdots, x^T_{-1}, x^T_0) \end{aligned}$$

is surjective on some appropriate subset of the range. Prior to formulating our results we need a few lemmas.

**Lemma 1.** *The rule function $g(x, y, z) \neq 1/2$ if and only if $x \neq 1/2$ and $z \neq 1/2$.*

*Proof.* Sufficiency: Assume, on the contrary, that $g(x, y, z) = 1/2$. Then $x + z - 2xz = 1/2$; and this implies that $x(1 - 2z) = 1/2 - z = (1 - 2z)/2$ or since $z \neq 1/2$ then $x = 1/2$ which is impossible. On the other hand, if either $x = 1/2$ or $z = 1/2$, the rule function $g(x, y, z) = 1/2$ which contradicts the assumption.

Our first result deals with the case where the domain of rule 90 is enlarged to all of $\mathbb{R}^3$. By doing so, we can obtain a necessary and sufficient condition for the controllability of the system.

**Theorem 1.** *Let $T > 0$ be a given (final) time, $A = (A^T_{-T}, \cdots, A^T_{-1}, A^T_0) \in \mathbb{R}^{T+1}$ be a given desired state. Consider the controllability problem associated with Table 1 with $a \in \mathbb{R}$, $g(x, y, z) = x + z - 2xz$ defined on all of $\mathbb{R}^3$.*

1. *If $A^T_{-T} \neq a$ then the control problem has no solution.*
2. *If $A^T_{-T} = a$, then the control problem has a (unique) solution in the admissible set $\mathcal{U}_{ad} = \mathbb{R}^T \setminus \{(1/2, 1/2, \ldots, 1/2)\}$ if and only if all the coordinates of $A$ are different from $1/2$,* i.e., *for every $N$, $0 \leq N \leq T$ we have $A^T_{-T+N} \neq 1/2$.*

*Proof.* From Table 1 it is clear that $A^T_{-T} = a$, this proves the first claim. Hence this last equality is in force throughout. We proceed on a case-by-case basis for $i = 1, 2, 3$ and then appeal to an induction argument for the general case. The case $i = 0$ being trivial we proceed immediately to the next case.

In the case where $i = 1$ we observe that $x^i_{-i+1} = x^1_0 = g(0, a, u_0) = u_0$, for every $i$, $0 \leq i \leq T$.

When $i = 2$ note that $x^1_1 = u_1$, and since $x^2_0 = g(x^1_{-1}, x^1_0, x^1_1)$ by definition, we get

$$x^2_0 = x^1_{-1} + (1 - 2x^1_{-1})u_1. \tag{2}$$

Similarly,

$$\begin{aligned} x^3_{-1} &= g(x^2_{-2}, x^2_{-1}, x^2_0), \\ &= x^2_{-2} + (1 - 2x^2_{-2})x^2_0, \\ &= x^2_{-2} + (1 - 2x^2_{-2})(x^1_{-1} + (1 - 2x^1_{-1})u_1), \quad \text{(by (2))} \\ &= x^2_{-2} + (1 - 2x^2_{-2})x^1_{-1} + (1 - 2x^2_{-2})(1 - 2x^1_{-1})u_1. \end{aligned} \tag{3}$$

The form of the next term, $x^4_{-2}$ necessary for our purposes is found similarly. Thus, substituting (3)) for $x^3_{-1}$ we find,

$$\begin{aligned} x^4_{-2} &= g(x^3_{-3}, x^3_{-2}, x^3_{-1}), \\ &= x^3_{-3} + (1 - 2x^3_{-3})x^3_{-1}, \\ &= x^3_{-3} + (1 - 2x^3_{-3})(x^2_{-2} + (1 - 2x^2_{-2})x^1_{-1} + (1 - 2x^2_{-2})(1 - 2x^1_{-1})u_1), \\ &= x^3_{-3} + (1 - 2x^3_{-3})x^2_{-2} + (1 - 2x^3_{-3})(1 - 2x^2_{-2})x^1_{-1} + \\ &\qquad + (1 - 2x^3_{-3})(1 - 2x^2_{-2})(1 - 2x^1_{-1})u_1. \end{aligned}$$

Continuing in this way we define the terms $x^n_{-m}$ when $n-m=2$ recursively and find that the cell values along this diagonal are given by

$$x^n_{-n+2} = x^{n-1}_{-n+1} + \sum_{k=1}^{n-2} \prod_{j=k}^{n-2} (1 - 2x^{j+1}_{-j-1})\, x^k_{-k} + u_1 \prod_{j=1}^{n-1} (1 - 2x^j_{-j}). \tag{4}$$

We calculate one more case, $i=3$, prior to stating the general form of the $x^T_{-T+i}$ for any $i$ and for any diagonal. Note that $x^2_1 = u_2$. Next,

$$x^3_0 = x^2_{-1} + (1 - 2x^2_{-1})u_2, \tag{5}$$

and

$$\begin{aligned} x^4_{-1} &= g(x^3_{-2}, x^3_{-1}, x^3_0), \\ &= x^3_{-2} + (1 - 2x^3_{-2})x^3_0, \\ &= x^3_{-2} + (1 - 2x^3_{-2})(x^2_{-1} + (1 - 2x^2_{-1})u_2), \quad \text{(by (5))} \\ &= x^3_{-2} + (1 - 2x^3_{-2})x^2_{-1} + (1 - 2x^3_{-2})(1 - 2x^2_{-1})u_2. \end{aligned}$$

The term $x^5_{-2}$ is found as above. This gives,

$$\begin{aligned} x^5_{-2} &= g(x^4_{-3}, x^4_{-2}, x^4_{-1}), \\ &= x^4_{-3} + (1 - 2x^4_{-3})x^4_{-1} \\ &= x^4_{-3} + (1 - 2x^4_{-3})(x^3_{-2} + (1 - 2x^3_{-2})x^2_{-1} + (1 - 2x^3_{-2})(1 - 2x^2_{-1})u_2) \\ &= x^4_{-3} + (1 - 2x^4_{-3})x^3_{-2} + (1 - 2x^4_{-3})(1 - 2x^3_{-2})x^2_{-1} + \\ &\quad + (1 - 2x^4_{-3})(1 - 2x^3_{-2})(1 - 2x^2_{-1})u_2. \end{aligned}$$

As before, we proceed by induction to find that, for this third left-diagonal (here $n-m=3$)

$$x^n_{-m} = x^{n-1}_{-m-1} + \sum_{k=1}^{m} x^{k+1}_{-k} \prod_{j=k}^{m} (1 - 2x^{j+2}_{-j-1}) + u_2 \prod_{j=1}^{m+1} (1 - 2x^{j+1}_{-j}),$$

or, upon setting $m-3$, we obtain

$$x^n_{-n+3} = x^{n-1}_{-n+2} + \sum_{k=1}^{n-3} x^{k+1}_{-k} \prod_{j=k}^{n-3} (1 - 2x^{j+2}_{-j-1}) + u_2 \prod_{j=1}^{n-2} (1 - 2x^{j+1}_{-j}). \tag{6}$$

This argument extends to the general case of the $i$-th left diagonal, where $i \leq T-1$. In this case every cell value of this diagonal is given by terms of the form

$$\begin{aligned} x^n_{-n+i} = x^{n-1}_{-n+i-1} + \sum_{k=1}^{n-i} x^{i+k-2}_{-k} \prod_{j=k}^{n-i} (1 - 2x^{i+j-1}_{-j-1}) \; + \\ + u_{i-1} \prod_{j=1}^{n-i+1} (1 - 2x^{i+j-2}_{-j}). \end{aligned} \tag{7}$$

where $n = 0, 1, 2, \ldots, T$. So, for the desired final row (see Table 1), $n = T$ and $i = N$ therefore

$$x^T_{-T+N} = x^{T-1}_{-T+N-1} + \sum_{k=1}^{T-N} x^{N+k-2}_{-k} \prod_{j=k}^{T-N} (1 - 2x^{N+j-1}_{-j-1}) + \\ + u_{N-1} \prod_{j=1}^{T-N+1} (1 - 2x^{N+j-2}_{-j}), \tag{8}$$

where $N = 0, 1, \ldots, T$. Let $A^T_{-T+N} = x^T_{-T+N}$ be the given final state at time $T$, where $0 \le N \le T$.

For $N = 1$ the coefficient of $u_0$ in (8) is 1 by Table 1. Hence a control $u_0$ exists with the property that the space-time diagram of Table 1 will reach $A^T_{-T+1}$ at time $T$. This unique control $u_0$ is now fixed.

The case $N = 2$ gives that the coefficient of $u_1$ in (8) is given by

$$\prod_{j=1}^{T-1} (1 - 2x^j_{-j}) = \prod_{j=1}^{T-1} (1 - 2a) = (1 - 2a)^{T-1}.$$

Since $A^T_{-T} = a \neq 1/2$ by hypothesis, it follows that $u_1$ exists and is unique. This now fixes the control $u_1$.

When $N = 3$, since $x^{j+1}_{-j} = u_0$ for all $j$, the coefficient of $u_2$ in (8) is

$$\prod_{j=1}^{T-2} (1 - 2x^{j+1}_{-j}) = \prod_{j=1}^{T-2} (1 - 2u_0) = (1 - 2u_0)^{T-2}.$$

It follows that the control $u_2$ exists and is unique. The three controls $u_0, u_1, u_2$ now found will bring the system in Table 1 to the given values $A^T_{-T}, A^T_{-T+1}, A^T_{-T+2}$ at time $T$.

For general $N$ the coefficient of $u_{N-1}$ is

$$\prod_{j=1}^{T-N+1} (1 - 2x^{N+j-2}_{-j})$$

where we require that the $x^{N+j-2}_{-j} \neq 1/2$ for all $j$, $j = 1, 2, \ldots, T - N + 1$. We show next that this is always the case.

If possible, let $j$ be a subscript with $x^{N+j-2}_{-j} = 1/2$. Then

$$\begin{aligned} x^{N+j-1}_{-j-1} &= g(x^{N+j-2}_{-j-2}, x^{N+j-2}_{-j-1}, x^{N+j-2}_{-j}) \\ &= x^{N+j-2}_{-j-2} + 1/2 - 2x^{N+j-2}_{-j-2}/2 \\ &= 1/2. \end{aligned}$$

Similarly, we get that $x^{N+j}_{-j-2} = 1/2$, and so on for all the cells down this diagonal (cells whose terms are necessarily of the form $x^{N+j-k}_{-j+k-2}$, $k = 0, 1, 2, \ldots$). Since this

diagonal must intersect the row of $A$'s in Table 1, we see that there must be some $M$ such that $A^T_{-T+M} = 1/2$ and this contradicts the hypothesis. It now follows that a unique control $u_{N-1}$ exists satisfying the requirement of controllability stated at the outset. Incidentally, this same argument shows that the $u_{N-1} \neq 1/2$. Since $N$ is arbitrary, we get that all $u_N \neq 1/2$, for $N = 0, 1, \ldots, T-1$.

The necessity is straightforward. For if such controls $u_N$, $N = 0, 1, \ldots, T-1$ exist satisfying the controllability hypothesis then we claim that $A^T_{-T+N} \neq 1/2$ for all $N$. Otherwise using Lemma 1 we can "work our way up and to the right" of this cell and deduce that some $u_N = 1/2$. Since this is not an admissible control value by hypothesis this contradiction then proves the necessity.

We give the general idea on how to proceed: if for some $N$ we have $A^T_{-T+N} = 1/2$, then by Lemma 1 either the cell directly above it and to the left or the cell directly above it and to the right must have value equal to $1/2$. The worst case scenario is if we keep going up and left, away from the line of controls on the extreme right using repeated applications of Lemma 1 (cf., Table 1). Then at some point in this procedure we must cross the first non-zero diagonal $\mathcal{L}_0^-$ at the extreme left which consists of the quantity "a" only. Since $a \neq 1/2$ by hypothesis, this cannot occur. Hence as we work our way up the space-time diagram using said Lemma, eventually the process must bring us to some control $u_N$ with value necessarily equal to $1/2$; and this is a contradiction.

*Example 1.* Let $a = 1/3$, and $T = 5$ in Table 1. In addition let $A^5_{-5} = 1/3$, $A^5_{-4} = 1/4$, $A^5_{-3} = 1/5$, $A^5_{-2} = 1/6$, $A^5_{-1} = 1/7$, $A^5_0 = 1/8$ be the set of reachable states. Recall that $u_0 = 1/4$. Then a straightforward calculation using the rule function $g(x, y, z) = x + z - 2xz$ gives the following table of values;

**Table 2.**

| $t$ | $-5$ | $-4$ | $-3$ | $-2$ | $-1$ | $0$ | $1$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1/3 | $1/4 = u_0$ |
| 1 | 0 | 0 | 0 | 0 | 1/3 | 1/4 | $-119/5 = u_1$ |
| 2 | 0 | 0 | 0 | 1/3 | 1/4 | $-38/5$ | $-13/6 = u_2$ |
| 3 | 0 | 0 | 1/3 | 1/4 | $-11/5$ | $-5/6$ | $788/1702 = u_3$ |
| 4 | 0 | 1/3 | 1/4 | $-2/5$ | $-1/6$ | 19/63 | $7/32 = u_4$ |
| 5 | 1/3 | 1/4 | 1/5 | 1/6 | 1/7 | 1/8 | . |

When we speak of *controllability on* $J$ where $J$ is either $[0, 1]$ or $\mathbb{R}$ we mean that the admissible controls set $\mathcal{U}_{ad}$ defined in Theorem 1 above is of the form $\mathcal{U}_{ad} = J^T \setminus \{(1/2, 1/2, \ldots, 1/2)\}$ and the $A^T_{-T+N} \in J$ for all $N = 0, 1, \ldots, T-1$. The next theorem gives some results for the more difficult problem of controllability on $[0, 1]$.

*Remark 1.* Note that in the preceding example $x^5_{-3} = 1/5$ by hypothesis. In this case the resulting equation admits the solution $u_1 = -119/5 < 0$; indeed many such values are negative. Thus the controllability problem on $[0, 1]$ has no solution, although the same problem on $\mathbb{R}$ has a unique solution (as per

Theorem 1) and it is exhibited in Table 2. This example shows us that for controllability we must expect a delicate generally nonlinear interplay between the values of the $A$'s, $a$ and the controls $u_i$.

*Remark 2.* We have presented in this paper only the case of $J = \mathbb{R}$. The general controllability problem on $[0, 1]$ for fuzzy rule 90 have also been solved but only for $T \leq 5$. We gather that, in general, we can expect a solution to the controllability problem for $[0, 1]$ provided the $A^T_{-T+N}$, $N \geq 3$, are very close to but not equal to $1/2$ and the time $T$ is not too large. This is because the exponentially fast asymptotes derived in [11] guarantee that $A^T_{-T+N} \to 1/2$ as $T \to \infty$ for every $N$. As one can gather from the obtained results the rule of thumb is, the larger the time, the closer the initial conditions are to be to $1/2$, in which case there is some hope for solvability.

## References

1. Bunimovich, L.A.: Coupled Map Lattices: one Step Forward and two Steps Back. Physica D 86, 248–255 (1995)
2. Cattaneo, G., Flocchini, P., Mauri, G., Santoro, N.: Cellular automata in fuzzy backgrounds. Physica D 105, 105–120 (1997)
3. Chopard, B., Droz, M.: Cellular automata modelling of physical systems. In: Collection Alea-Sacley. Cambridge University Press, Cambridge (1998)
4. Conway, J. H.: Game of life (1976)
5. Culik II, K., Yu, S.: Undecidability of CA classification schemes. Complex Systems 2, 177–190 (1988)
6. El Yacoubi, S., El Jai, A.: Notes on control and observation in Cellular automata models. WSEAS Transaction on Computers 4(2), 1086–1092 (2003)
7. El Yacoubi, S., El Jai, A., Ammor, N.: Regional controllability with cellular automata models. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493, pp. 357–367. Springer, Heidelberg (2002)
8. El Yacoubi, S., El Jai, A.: Cellular Automata and Spreadability. Journal of Mathematical and Computer Modelling 36, 1059–1074 (2002)
9. El Yacoubi, S., Jacewicz, P.: A genetic programming approach to structural identification of cellular automata models. Journal Of Cellular Automata
10. El Yacoubi, S.: A Mathematical method for control problems on Cellular Automata models. International Journal of Systems Sciences 39(5), 529–538 (2008)
11. Flocchini, P., Geurts, F., Mingarelli, A., Santoro, N.: Convergence and aperiodicity in fuzzy cellular automata: revisiting rule 90. Physica D 42, 20–28 (2000)
12. Flocchini, P., Santoro, N.: The chaotic evolution of information in the interaction between knowledge and uncertainty. In: Stonier, R.J., Yu, X.H. (eds.) Complex Systems: Mechanism of Adaptation, pp. 337–343. IOS Press, Amsterdam (1994)
13. Garzon, M.: Models of massive parallelism. Analysis of cellular automata and neural networks. Springer, Heidelberg (1995)
14. Hedlund, G.A.: Endomorphisms and automorphisms of the shift dynamical systems. Mathematical Systems Theory 3, 320–375 (1969)
15. Mingarelli, A.B.: The global evolution of general fuzzy cellular automata. J. Cellular Automata 1(2), 141–164 (2006)

16. Mingarelli, A.B., El Yacoubi, S.: On the decidability of the evolution of the fuzzy cellular automaton, FCA 184. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 360–366. Springer, Heidelberg (2006)
17. Von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Urbana (1966)
18. Sontag, E.D.: Mathematical Control Theory: Deterministic Finite Dimensional Systems, 2nd edn. Springer, Heidelberg (1998)
19. Sipper, M.: Non-Uniform Cellular Automata: Evolution in Rule Space and Formation of Complex Structures. In: Brooks, R.A., Maes, P. (eds.) Artificial Life IV, pp. 394–399. MIT Press, Cambridge (1994)
20. Toffoli, T.: Cellular automata as an alternative to differential equation in modeling physics. Physica D 10, 117–127 (1984)
21. Vichniac, G.Y.: Simulating physics with cellular automata. Physica D 10, 96–115 (1984)
22. Wolfram, S.: Cellular Automata and Complexity. Collected Papers. World Scientific, Singapore (1994)

# Examples of Fast and Slow Convergence of 2D Asynchronous Cellular Systems

Nazim Fatès and Lucas Gerin

[1] INRIA Nancy Grand-Est – LORIA – MaIA team
nazim.fates@loria.fr
[2] IECN - Univ. Nancy I
lucas.gerin@iecn.u-nancy.fr

**Abstract.** We study the convergence properties of cellular automata under fully asynchronous updating, i.e., when a single cell is selected at random at each time unit. We tackle this question for the two-dimensional totalistic cellular automata. As a first step for studying this class, we focus on a few examples that are, in our view, representative of the diversity of the behaviours found in dimension two. The richness of the evolutions we consider underlines that the updating scheme plays a central role in the evolution of a cellular automaton.

## 1 Introduction

In a recent series of studies, cellular automata (CA) have been examined in the light of their robustness to asynchronous updating: the question is to determine to which extent the behaviour of a cellular automaton is due to the method used to update the cells. In particular, what happens if this method is probabilistic, for example when each cell has a probability $\alpha$ to be fired at each time step (the $\alpha$-asynchronous dynamics), or when a single cell is updated at random at each time step (the fully asynchronous dynamics)?

For the sake of conciseness, we refer to [3,1] for a review of works related to asynchronism and for a discussion on updating schemes. One of the first analytical results on asynchronous CA led to classify a small set of one-dimensional rules (the double-quiescent Elementary CA) according to their convergence properties [3]. The present paper is intended as a first step to broaden the scope of our research to the two-dimensional case. We follow the pioneering work by D. Regnault *et al.* which concerned the Minority rule [2]. We select some totalistic two-dimensional rules that are, in our view, representative of the different classes of behaviour in two dimensional CA under fully asynchronous updating. Among these rules, we distinguish three examples for which we apply techniques already used in one dimension. Furthermore, we exhibit two rules for which a finer geometrical analysis is needed. This provides us with a new type of asymptotic convergence time, specific to the two-dimensional case.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 184–191, 2008.

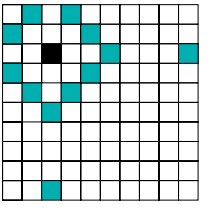
## 2 Definitions

**Topology of the environment.** Let $\Lambda$ be the two-dimensional square grid $\{1,\dots,L\}\times\{1,\dots,L\}$, with toric boundary conditions (*i.e.*, we identify $\{1,\dots,L\}$ with $\mathbb{Z}/L.\mathbb{Z}$). We denote by $n=L^2$ the total number of cells.

For a cell $c$ and an integer $k$, we define the *sphere* $\partial B(c,k)$ as:

$$\partial B(c,k)=\{c'\in\Lambda\,,d(c,c')=k\},$$

where $d$ is the graph distance on the torus. Below is a representation of the sphere $\partial B(c,3)$, with $c$ in black and $L=10$:



To each cell $c$ we associate a state $\sigma_c$ in $\{0,1\}$, and we call $\sigma=\{\sigma_c\}_{c\in\Lambda}$ a *configuration*. To express the locality of the interactions, we introduce the function $N:\Lambda\to\Lambda^5$ that associates to each cell $c\in\Lambda$, its *von Neumann* neighbourhood:

$$N(c)=\{c,c+\boldsymbol{n},c-\boldsymbol{n},c+\boldsymbol{e},c-\boldsymbol{e}\},$$

where $\boldsymbol{n},\boldsymbol{e}$ denote the vectors $(0,1)$ and $(1,0)$.

**Totalistic 2D Cellular Automata.** A *local transition rule* is a function $\phi:\{0,1\}^5\to\{0,1\}$. In this paper, we consider only the *totalistic functions*, *i.e.*, functions $\phi$ such that:

$$\phi(q_1,\dots,q_5)=f(q_1+\cdots+q_5),$$

where $f:\{0,\dots,5\}\to\{0,1\}$. There are 64 such totalistic rules ; we denote the totalistic rule $\phi$ associated to a function $f$ by the code $\mathbb{T}i$ where $i=f(0)\cdot 2^0+f(1)\cdot 2^1+\cdots+f(5)\cdot 2^5$.

We restrict our study to the *fully asynchronous dynamics*: only one cell is updated at each time step. To define this type of asynchronism, we associate to each local rule $\phi$ a *global rule* $\Phi:\{0,1\}^\Lambda\times\Lambda\to\{0,1\}^\Lambda$ where $\Phi(\sigma,\tilde{c})$ is the configuration obtained by updating cell $\tilde{c}$ in configuration $\sigma$, according to the local rule $\phi$. More formally, if we write $\sigma'=\Phi(\sigma,\tilde{c})$:

$$\forall c\in\Lambda,\ \ \sigma'_c=\begin{cases}f(\sigma_c+\sigma_{c+\boldsymbol{n}}+\sigma_{c-\boldsymbol{n}}+\sigma_{c+\boldsymbol{e}}+\sigma_{c-\boldsymbol{e}}) & \text{for } c=\tilde{c};\\ \sigma_c & \text{otherwise.}\end{cases}$$

Let us denote by $U_t$ the cell updated at time $t$. A local rule $\phi$ and a sequence $(U_t)_{t\in\mathbb{N}}$ defines the sequence of configurations $(\sigma^t)_{t\in\mathbb{N}}$:

$$\begin{aligned}&\sigma^0\in\{0,1\}^\Lambda,\\ &\sigma^{t+1}=\Phi(\sigma^t,U_t),\ \text{ for } t\in\mathbb{N}.\end{aligned}$$

As we are interested in random updates of the system, we assume that $(U_t)_{t\in\mathbb{N}}$ is a sequence of independent random variables, uniformly sampled from $\Lambda$. This random sequence makes $(\sigma^t)$ a stochastic dynamical system in $\{0,1\}^\Lambda$. Our objective is to determine the convergence properties of this system.

For a local rule $\phi$, we denote by $\mathfrak{F}(\phi)$ the set of the fixed points of the global rule $\Phi$ associated to $\phi$. It consists of the configurations that remain unchanged, whatever the cell selected for updating:

$$\mathfrak{F}(\phi) = \{\sigma \in \{0,1\}^\Lambda, \text{for any cell } c \in \Lambda, \Phi(\sigma, c) = \sigma\}.$$

Note that for a given $\phi$, the sets of fixed points under synchronous and asynchronous updating are identical.

**Definition 1.** *For a given function $\phi$ and an initial configuration $\sigma^0$, let $T_\phi(\sigma^0)$ be the time of convergence of the sequence $(\sigma^t)_{t\in\mathbb{N}}$, that is, the random variable:*

$$T_\phi(\sigma^0) = \min\{t \in \mathbb{N}, \sigma^t \in \mathfrak{F}(\phi)\},$$

*with $\min\emptyset = +\infty$. The* Worst Expected Convergence Time *(WECT) of rule $\phi$ is given by:*

$$\mathrm{WECT}_\phi(n) = \max_{\sigma^0} \mathbb{E}[T_\phi(\sigma^0)].$$

Intuitively, if we think of cellular automata as models of physical or artificial systems, studying the WECT provides us with an estimation of the maximum time needed to go back to equilibrium when a perturbation is applied.

Fatès *et al.* showed that the asymptotic behaviour of $\mathrm{WECT}_f(n)$ provides a relevant classification of the Elementary Cellular Automata with two quiescent states [3]. More precisely, they have shown that these rules may be classified into 5 families, according to whether $\mathrm{WECT}_f(n)$ is $\Theta(n\log n)$, $\Theta(n^2)$, $\Theta(n^3)$, $\Theta(n2^n)$ or infinite[1].

## 3 Convergence Times

The examples of 2D totalistic rules that we exhibit show that we recover some of the classes mentioned above, as well as a new one: $\Theta(n^{3/2})$. The five different behaviours found in 2D are summarised in table below:

| Name | $\phi$ | $\mathrm{WECT}_\phi(n)$ | Observed behaviour |
|---|---|---|---|
| Coupon collector | $\mathbb{T}63$ | $\Theta(n\log n)$ | Fast convergence to $1^\Lambda$ |
| Epidemic | $\mathbb{T}62$ | $\approx n^{3/2}$ | Fast convergence to $1^\Lambda$ |
| Majority | $\mathbb{T}56$ | $\Theta(n^2)$ | Fast convergence to a local equilibrium |
| Erratic | $\mathbb{T}10$ | $> \lambda^n$ (conjectured) | Slow convergence (metastability) |
| Parity Counter | $\mathbb{T}21$ | infinite | Noise-like evolution |

[1] We write $f_n = \Theta(g_n)$ when there exist positive numbers $C^-, C^+$ such that, for $n$ large enough, $C^- g_n \leq f_n \leq C^+ g_n$.

### 3.1 A *Coupon Collector* Automaton

Our examination begins with the rule $\mathbb{T}63$:

| $s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $f(s)$ | 1 | 1 | 1 | 1 | 1 | 1 |

An update sets the cell into state 1 whatever the initial state. Under the synchronous dynamics $\mathbb{T}63$ reaches the fixed point $1^\Lambda$ in one step. In the asynchronous case, this fixed point is attained once each cell has been updated at least once (see Fig. 1). This kind of process often arises in the analysis of algorithms ; it is usually called a *Coupon collector* process.

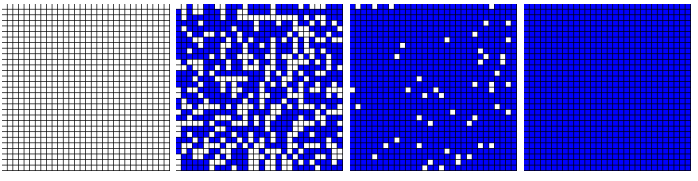



**Fig. 1.** Example of a simulation for the *coupon collector* rule $\mathbb{T}63$. Cells in state 1 (resp. 0) are coloured in black (resp. white), a convention which is kept throughout the paper. Simulations were obtained with the *FiatLux* CA simulator [4].

**Theorem 1**

$$\mathrm{WECT}_{63}(n) = \Theta(n \log n).$$

**Sketch of proof.** Remark that when $k$ cells among the $n$ cells are still in state 0, it takes in average $n/k$ time steps to update one of them. Thus, the slowest convergence time holds for $\sigma^0 = 0^\Lambda$. It is equal to:

$$\frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1} = \Theta(n \log n).$$

### 3.2 The *Majority* Rule

We now turn to the *Majority* rule $\mathbb{T}56$:

| $s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $f(s)$ | 0 | 0 | 0 | 1 | 1 | 1 |

An update sets cell $c$ to the state that is most present in $N(c)$. The global effect of the rule is to converge quickly to equilibrium (see Fig. 2).

**Theorem 2**

$$\mathrm{WECT}_{56}(n) = \Theta(n^2).$$

**Sketch of proof.** We begin by introducing some notations. The *similarity function* $\mathcal{I}$ counts the number of neighbouring cells in the same state:

$$\mathcal{I}(\sigma) = \mathrm{card}\{(c, c') \in \Lambda\,, d(c, c') = 1 \text{ and } \sigma_c = \sigma_{c'}\}.$$

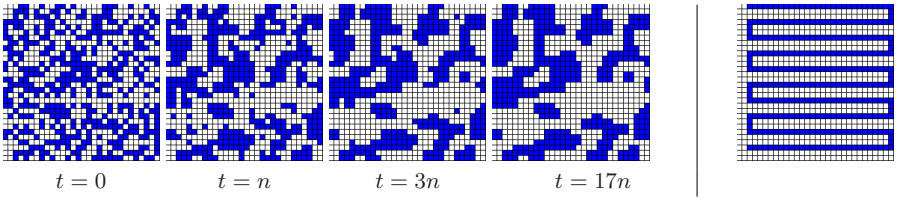



**Fig. 2.** Left : Example of a simulation for rule $\mathbb{T}56$, when $\sigma^0$ is a random uniform configuration. Right : A snake-like configuration gives a lower bound on the WECT.

For a configuration $\sigma$, we define $M_{q,r}(\sigma)$ as the numbers of cells in state $q \in Q$, with $r$ neighbours in state 1.

**Upper bound.** We obtain an upper bound on the convergence time by showing that the sequence $\mathcal{I}(\sigma^t)$ is increasing. To see why this holds, let us list all the cases where a cell may change its state:

| Pattern | $M_{0,4}$ | $M_{0,3}$ | $M_{1,0}$ | $M_{1,1}$ |
|---|---|---|---|---|
| |  |  |  |  |
| $\Delta\mathcal{I}$ | $+8$ | $+4$ | $+8$ | $+4$ |

With the notation $\Delta\mathcal{I}_t = \mathcal{I}(\sigma^{t+1}) - \mathcal{I}(\sigma^t)$, we have that:

$$\mathbb{E}[\Delta\mathcal{I}_t \mid \sigma^t] = \frac{1}{L^2}\big(\,8M_{0,4} + 4M_{0,3} + 8M_{1,0} + 4M_{1,1}\,\big)(\sigma^t).$$

Remark that if $\sigma^t$ is not a fixed point, then this quantity is greater than $4/L^2$. Using arguments on Lyapunov functions, we obtain that, for any initial configuration $\sigma^0$:

$$\mathbb{E}[T_{56}(\sigma^0)] \leq \frac{L^2}{4}\mathcal{I}(\sigma^0).$$

As $\mathcal{I}$ is bounded by $4L^2$, it follows that:

$$\mathbb{E}[T_{56}(\sigma^0)] \leq L^4 = n^2.$$

**Lower bound.** To obtain a lower bound on the convergence time, we consider the *snake-like* initial configuration drawn in Fig.2. Intuitively, this initial configuration is chosen in order to ensure that the only way of reaching equilibrium is to "shrink" the snake by updating its two extremities. The snake is made of about $n/3$ cells in state 1 ; as the probability to update one of the two extremities is $2/n$, we obtain:

$$\mathbb{E}[T_{56}(\sigma^0)] \geq C^- n^2.$$

As the lower bound and the upper bound scale in $n^2$, the theorem is proved. Remark that the evolution of the *Majority* rule is different in 1D and 2D since the WECT of the *Majority* rule in 1D is of order $n \log n$ [3].

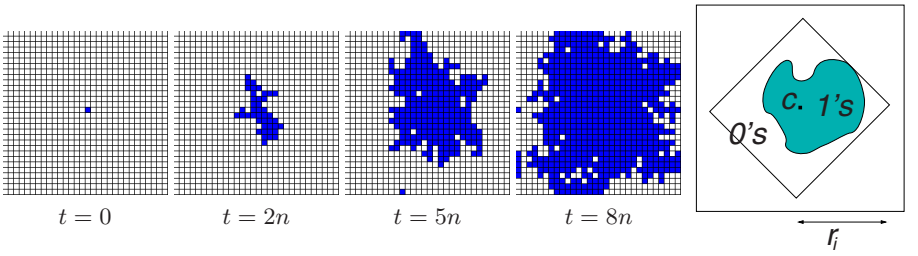



**Fig. 3.** Left : Example of a simulation for rule $\mathbb{T}62$. Right : Schematic view of a configuration where a region of 1's reaches the sphere $\partial(c, r_i)$.

### 3.3 The *Epidemic* Automaton

Let us consider the *epidemic* rule $\mathbb{T}62$:

| $s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $f(s)$ | 0 | 1 | 1 | 1 | 1 | 1 |

A cell in state 0 (healthy) turns to state 1 (infected) if one of its neighbours is in state 1; it then remains in this state.

Up to now, the classes of convergence that we have met were already known as they were identified for 1D cellular automata [3]. This rule has a new type of convergence, specific to the two-dimensional case.

**Theorem 3.** *There exist two constants $C^-, C^+$ such that, for $n$ large enough,*

$$\frac{C^-}{\log n} n^{3/2} \leq \mathrm{WECT}_{62}(n) \leq C^+ (\log n) n^{3/2}.$$

**Remark.** We conjecture that one may tighten these bounds and show that $\mathrm{WECT}_{62}(n)$ is actually of order $n^{3/2}$.

**Sketch of proof.** The first step is to prove that the largest convergence time is obtained for the configurations which contain only one cell in state 1. For such an initial configuration, the region of 1's spreads all over the grid (see Fig. 3). The second step consists in estimating the time needed for this region to cover the whole grid.

To prove the first step, let us introduce a partial order $\prec$ on $\{0,1\}^\Lambda$ by:

$$\sigma \prec \eta \quad \text{if and only if} \quad \forall c \in \Lambda, \sigma_c \leq \eta_c.$$

It is then enough to note that $\Phi$ conserves the order $\prec$ in the sense that:

$$\forall U_t \in \Lambda,\ \sigma \prec \eta \Rightarrow \Phi(\sigma, U_t) \prec \Phi(\eta, U_t).$$

We therefore obtain: $\sigma^t = 1^\Lambda \Rightarrow \eta^t = 1^\Lambda$ ; which proves the first step.

To obtain the bounds around $n^{3/2}$, we analyse how the region of 1's spreads over $\Lambda$ as the system evolves. The key point is to cut $\Lambda$ in concentric spheres $\partial B(c, r_i)$ for some suitably chosen integers $(r_i)$ and to estimate the time needed for the region of 1's to reach every $\partial B(c, r_i)$ (see Fig. 3 - Right). The choice of the sequence $r_i$ is different for the two bounds of the theorem.

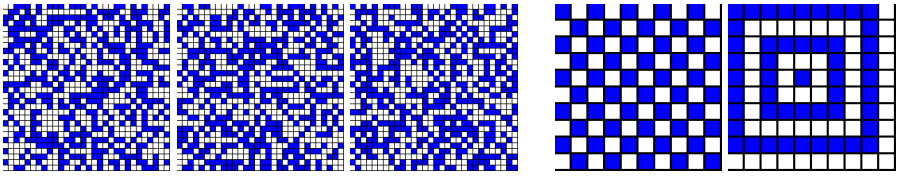


**Fig. 4.** Left: Three typical noise-like configurations observed in the evolution of the *erratic* rule $\mathbb{T}10$. Right: Two fixed points of this rule.

### 3.4 An *Erratic* Automaton

When exploring systematically the 64 totalistic rules, we observed that some of them, *e.g.* $\mathbb{T}10$, evolved in a metastable regime: the observation of the configurations attained gives the impression that the system evolves "erratically" and will never converge.

However, a detailed analysis of $\mathbb{T}10$:

| $s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $f(s)$ | 0 | 1 | 0 | 1 | 0 | 0 |

shows that this rule do possesses fixed points (see Fig.4) and that some of them may be attained with a convenient sequence of updates.

**Conjecture.** *There exists a constant $\lambda > 1$ such that, for $n$ large enough,*

$$\lambda^n \leq \mathrm{WECT}_{10}(n) < +\infty.$$

In [3], it is proved that there exist 1D CA for which the worst convergence time is exponential in $n$. It seems however that the methods used in dimension one do not apply here. While we can describe precisely the set $\mathfrak{F}(\mathbb{T}10)$ of fixed points, we are not able to quantify the proximity between a given configuration and $\mathfrak{F}(\mathbb{T}10)$, as opposed to the previous rules. The difficulty comes from the complexity of the trajectories in dimension two, and from the large quantity of fixed points.

A way of tackling this problem would be to change $\phi = \mathbb{T}10$ into $\phi^\varepsilon$, a *probabilistic* local rule defined by the following table:

| $s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $f(s)$ | 0 | 1 | 0 | 1 | $\varepsilon$ | 0 |

The effect of $\phi^\varepsilon$ is identical to that of $\mathbb{T}10$, except if the updated cell $c$ has four neighbours in state 1 ; in this case, $c$ takes the state 1 with probability $\varepsilon$. The perturbation with probability $\varepsilon$ makes the process $(\sigma^t)$ a *reversible* Markov chain, for which quantitative and efficient tools are available (see *e.g.* [5]). Consequently, it may be easier to estimate some quantities concerning this new system, and then to derive the bounds needed on the convergence time of the original system by taking $\varepsilon$ infinitely small.

### 3.5 The Parity Counter

We finish our exploration with rule $\mathbb{T}42$:

| $s$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $f(s)$ | 0 | 1 | 0 | 1 | 0 | 1 |

It is sometimes referred to as the *Parity Counter*: an update on cell $c$ turns it into `1` if and only if the number of `1`'s in $N(c)$ is odd.

**Theorem 4.** *If $\sigma^0$ is not a fixed point, then $T_{42}(\sigma^0) = +\infty$. Consequently,*

$$\mathrm{WECT}_{42}(n) = +\infty.$$

**Sketch of proof.** To show that $T_{42}(\sigma^0)$ is infinite when $\sigma^0$ is not a fixed point, we need to prove that, unlike the *erratic* automaton, there is no sequence of updates leading from $\sigma^0$ to a fixed point. The argument is that all the transitions are reversible: remark that updating a cell $c$ changes both its state and the "parity" of $N(c)$, and therefore $c$ remains unstable under the application of the rule (although other cells in $N(c)$ may become stable).

### 3.6 Concluding Remark

According to our simulations, the spectrum of convergence times that we studied here covers most of the totalistic rules. We ask whether there are other convergence types, for example in $\theta(n^3)$ or in $\theta(n^4)$.

**Acknowledgements.** Authors are grateful to P.-Y. Louis for fruitful discussions, and to M. Krikun for his many comments on an earlier version of this paper.

## References

1. Grilo, C., Correia, L.: Asynchronous Stochastic Dynamics and the Spatial Prisoner's Dilemma Game. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNAI), vol. 4874. Springer, Heidelberg (2007)
2. Regnault, D., Schabanel, N., Thierry, É.: Progresses in the Analysis of Stochastic 2D Cellular Automata: a Study of Asynchronous 2D Minority. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708. Springer, Heidelberg (2007)
3. Fatès, N., Morvan, M., Schabanel, N., Thierry, É.: Fully asynchronous behavior of double-quiescent elementary cellular automata. Theoretical Computer Science 362, 1–16 (2006)
4. Fatès, N.: FiatLux CA Simulator in JAVA. Sources and software, `http://nazim.fates.free.fr/Logiciel.htm`
5. Norris, J.: Markov Chains. Cambridge University Press, Cambridge (1997)

# Multi-scale Modeling with Cellular Automata: The Complex Automata Approach

Alfons G. Hoekstra[1], Jean-Luc Falcone[2], Alfonso Caiazzo[1], and Bastien Chopard[2]

[1] Section Computational Science, Faculty of Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{alfons,acaiazzo}@science.uva.nl
http://www.science.uva.nl/research/scs

[2] Department of Computer Science, University of Geneva, 24, Rue du Général Dufour, 1211 Geneva 4, Switzerland
{Bastien.Chopard,Jean-Luc.Falcone}@cui.unige.ch
http://spc.unige.ch/

**Abstract.** Cellular Automata are commonly used to describe complex natural phenomena. In many cases it is required to capture the multi-scale nature of these phenomena. A single Cellular Automata model may not be able to efficiently simulate a wide range of spatial and temporal scales. It is our goal to establish a Cellular Automata modeling paradigm for multi-scale processes. Here we will demonstrate that Complex Automata, a paradigm that we recently introduced, are capable to facilitate such modeling.

**Keywords:** Multi-Scale Modeling, Complex Automata.

## 1 Introduction

Cellular Automata (CA) are generally acknowledged to be a powerful way to describe and model natural phenomena. [1,2,3] There are even tempting claims that Nature itself is one big (quantum) information processing system, e.g. [4], and that CA may actually be nature's way to do this processing [5,6,7]. We will not embark on this philosophical road, but ask ourselves a more mundane question. Can we use CA to model the inherently multi-scale processes in nature *and* use these models for efficient simulations on digital computers?

The ever increasing availability of experimental data on every scale, from 'atom to material' or from 'gene to health', in combination with the likewise ever increasing computational power [8,9], facilitates modeling and simulation of natural phenomena taking into account all required spatial and temporal scales (see e.g. [10]). Multi-scale modeling and simulation, as a paradigm in Computational Science, is becoming more and more important, as witnessed by e.g. dedicated special issues [11] and thematic journals [12,13].

When using CA to model a natural process, the underlying lattice and one iteration of the CA have a clear physical meaning in terms of lattice spacing

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 192–199, 2008.

and time step. We denote by $A(\Delta x, \Delta t, L, T)$ the spatio-temporal domain of a CA, whose spatial domain is made of cells of size $\Delta x$ and it spans a region of size $L$, while the quantity $\Delta t$ is the time step and $T$ is the end of the simulated time interval. Therefore, processes with time scales between $\Delta t$ and $T$ can be represented and spatial scales ranging from $\Delta x$ to $L$ can be resolved. When executing such CA on a digital computer we note that the execution time $T_{ex}$ scales as

$$T_{ex} \sim \frac{T}{\Delta t} \left( \frac{L}{\Delta x} \right)^D , \tag{1}$$

where $D$ is the spatial dimension of the simulated domain. Trying to model a multi-scale system with a single CA would require to choose $\Delta x$ and $\Delta t$ in such a way that the smallest microscopic details and fastest dynamical response of the system are captured, yet the overall system size ($L$) and slowest dynamical time scale ($T$) need to be covered. For instance, in modeling human physiology the relevant range of spatial scales is from nanometer to meter (i.e. a factor $10^9$) whereas temporal scale is from microseconds to human lifetime (i.e a factor $10^{15}$). Such numbers, in combination with Eq. 1 immediately show that one will never be able to simulate multi-scale systems with a single CA spanning a wide range of scales.

In this paper we propose Complex Automata (CxA), a set of single scale CA representing processes operating on different spatio-temporal scales, supplied with adequate coupling templates between the scales, as a CA-based methodology to model and simulate multi-scale phenomena. These CA are typically based on Lattice Boltzmann Models (LBM) [1,3] or other generalized CA. Besides generalized CA to represent single scale models, the CxA approach also includes Agent Based Models (ABM).

## 2 Multi-scale Cellular Automata

The literature on using Cellular Automata to model multi-scale phenomena is relatively small, maybe with the exception of using CA to model land usage and geographical systems (e.g. [14]). Furthermore, many papers exist that use CA in multi-scale modeling, but there CA is typically coupled to other types of models (e.g. [15]).

The bulk of CA multi-scale attempts are grid refinement methods, also termed multi-blocks. The idea is to adapt the local grid size to the local process scale, i.e. using a fine grid in regions where small scale processes occur and a coarse grid where larger scales are sufficient. A common approach is to couple grids of different scales with an overlap region [16]. To exchange information from the coarse grid to the fine grid, an interpolation scheme must be used. However, as both space and time scale are different, interpolation must be spatial and temporal. To allow a smooth coupling, a re-scaling of the variables is is also required (e.g. [17]). Another interesting way to couple a coarse and fine grid for a CA was proposed by Weimar [18]. When the two grids are not overlapping, each grid must provide boundary conditions for the other one. The information

provided by the fine grid to the coarse grid is simply the averaging of the micro cell adjacent to a macro cell. However, to reconstruct the lacking information at microscopic boundaries, the author proposes an iterative statistical scheme to ensure the information flowing in each boundary micro cell exhibit the same correlations than the information flowing between micro cells.

Other ways of coupling multi-scale CA come from two theoretical frameworks. The first one is based on higher-order CA [19]. In this framework, the CA rules are not only able to change the cell state, but also the rules themselves, the neighborhood and the topology. Moreover, these models are also able to take in account hierarchical CA where higher level cells are connected to one or more lower level cells. The second one results from the work of Israeli and Goldenfeld [20] who have shown that it is possible to coarse-grain 1D nearest-neighbor CA, by defining a macroscopic CA whose behavior is similar to a microscopic CA. That is an important result because the authors have achieved the coarse-graining of CA known to be irreducible.

## 3 Complex Automata

Complex Automata (CxA) were recently introduced as a modeling paradigm for multi-scale systems using CA, LBM and Agent Based Models (ABM) as the building blocks [21,22]. The key idea behind CxA is that a multi-scale system can be decomposed into N single-scale Cellular Automata that mutually interact across the scales. The decomposition is achieved by building a Scale Separation Map (SSM) on which each system can be represented as an area according to its spatial and temporal scales. Processes having well separated scales are easily identified as the components of the multi-scale model.

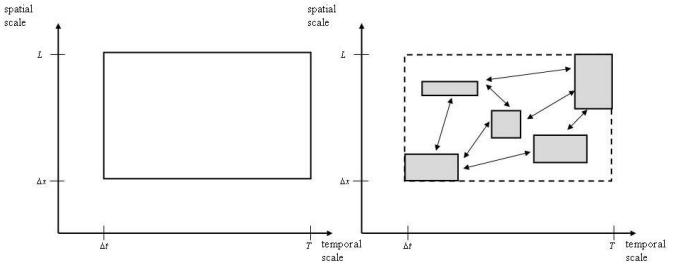



**Fig. 1.** The Scale Separation Map with left a single CA and right a hypothetical CxA modeling the same process

Fig. 1 shows a SSM, where the horizontal axis represents the temporal scales and the vertical axis the spatial scales. On the left a CA with spatio-temporal domain $A(\Delta x, \Delta t, L, T)$ is represented on the SSM. Assuming that the process to be simulated is really multi-scale in the sense that it contains relevant sub-processes on a wide range of scales, simulations based on the finest discretizations

are not really feasible (Eq. 1), the approach we propose in CxA modeling is to try to split the original CA into a number of single-scale CA and let these CA exchange information in such a way that the dynamical behavior of the multi-scale process is mimicked as accurate as possible. This is shown schematically in right part in Fig. 1. In [22] we explain in much more detail how this can be achieved and provide a large number of examples. Here we will continue to introduce a more formal definition of a CxA, providing argument why we believe that the CxA approach *does* represent a possible way to construct CA-based models of multi-scale systems.

### 3.1 Preliminary Definitions

Formally, a Cellular Automaton can be defined as

$$\mathrm{CA} = \{(\Delta x, L, \Delta t, T), \mathbb{F}, \Phi, f_{init} \in \mathbb{F}, \mathbf{u}\} \tag{2}$$

specifying the spatial-temporal domain with discretization parameters $(\Delta x, \Delta t)$, the space of states $(\mathbb{F})$, the initial condition $(f_{init})$ and the update rule $(\Phi)$. In Eq. (2) we include in the definition a field $\mathbf{u}$, collecting the external data the CA depends on. The state of the system is described by a $\hat{f}^{t_n} \in \mathbb{F}$, denoting the numerical solution at the $n$-th time step:

$$\begin{aligned} &\hat{f}^0 = f_{init}[u_0], \text{ initial condition} \\ &\hat{f}^{t_n+\Delta t} = \Phi[\mathbf{u}; \hat{f}^n], \quad n > 0 \end{aligned} \tag{3}$$

with $u_0$ an external field connected to the initial condition.

Additionally, restricting to CA, LBM or ABM, we constrain the update rule $\Phi$ to the form

$$\Phi[\mathbf{u}; f] = (B[u_B] \circ P \circ C[u_C])\,[f], \tag{4}$$

i.e. written as a composition of three operators: *collision* $C[u_C]$, depending on external parameters $u_C$, *propagation* $P$, depending on the topology of the domain, and *boundary condition* $B[u_B]$, depending on external parameters denoted by $u_B$. This form of a CA update rule is inspired by Lattice Gas Automata (see e.g. [1,3]), and applied in defining coupling templates for CxA [22]. Note that Chopard et al. discuss this in more detail in another manuscript submitted to ACRI08.

In what follows, we let the definition of CA (2) depend on a (small) parameter $h$, related to spatial and temporal discretizations (for example $\Delta x_h = h$, $\Delta t_h = \alpha h$). Accordingly, the evolution space and the update rule depend on $h$ as well: $\Phi_h : \mathbb{F}_h \to \mathbb{F}_h$ . Shortly, we will call $\hat{f}_h$ the numerical outcome of the $\mathrm{CA}_h$.

### 3.2 Complex Automata Formalism

To begin with, as in the left diagram in Fig. 1, we consider a multi-scale system represented as a single $\mathrm{CA}_h$ defined as in (2). Building a CxA, instead of

describing the system with a single $\hat{f}_h$, we lower the dimension of the problem and the computational complexity, introducing a set of discretization parameters $H = (h_1, \ldots, h_M)$ and building a corresponding Complex Automaton $\mathrm{CxA}_H = (\mathrm{CA}_{h_1}, \ldots, \mathrm{CA}_{h_M})$, where each $\mathrm{CA}_{h_m}$ is an object as in (2). Similarly as before, we denote with $\hat{f}_H$ the numerical outcome of the complex automata simulation and $\hat{f}_{h_m}$ the state variable of the single CAs. Note that for the evolution space of a CxA it holds $\mathbb{F}_H^{CxA} \subset \mathbb{F}_{h_1} \times \ldots \times \mathbb{F}_{h_M}$. In fact, part of the single scale evolution spaces could be shared by several Automata, in case of space overlap and single domain coupling [22].

### 3.3 Coupled Evolution of a CxA

For the sake of simplicity, we describe the formalism restricting to the evolution of two coupled single scale models. From equations (3)-(4), we have the following general representation

$$\begin{aligned}
&\hat{f}_{1,h_1}^{t_{1,0}} = f_{init,1}[\hat{f}_2] \\
&\hat{f}_{1,h_1}^{t_{1,n_1}+\Delta t_{h_1}} = \left(B_{h_1}[\hat{f}_2] \circ P_{h_1} \circ C_{h_1}[\hat{f}_2]\right)[\hat{f}_{h_1}^{t_{1,n_1}}], \quad n_1 > 0 \\
&\hat{f}_{2,h_2}^{t_{2,0}} = f_{init,2}[\hat{f}_1] \\
&\hat{f}_{2,h_2}^{t_{2,n_2}+\Delta t_{h_2}} = \left(B_{h_2}[\hat{f}_1] \circ P_{h_2} \circ C_{h_2}[\hat{f}_1]\right)[\hat{f}_{h_2}^{t_{2,n_2}}], \quad n_2 > 0
\end{aligned} \tag{5}$$

where two CAs are fully coupled in all the components. In detail,

- $f_{init,1}[\hat{f}_2]$ denotes a coupling through initial conditions (i.e. the initial condition of 1 depends on the results of 2)
- $B_{H_1}[\hat{f}_2]$ expresses coupling through boundary conditions,
- $C_{H_1}[\hat{f}_2]$ expresses the coupling through collision operator.

In general, for different situations (multidomain/singledomain, time/space separation/overlap) we can restrict the set of possible couplings to a well-specified coupling template. [22] Consider the example of a microscopic fast process coupled to a macroscopic slow process (micro-macro coupling). The macroscopic process takes input from explicit simulations of microscopic processes at each time step and on each lattice site of the macroscopic process. The microscopic processes run to completion, assuming that they are much faster than the macroscopic process and therefore are in quasi-equilibrium on the macroscopic time scales (this approach is known in the literature as the Heterogeneous Multi-scale Method, see [23]). The macroscopic process could e.g. be a fluid flow with takes its viscosity from an underlying microscopic process (e.g. explicit suspension model).

In Fig. 2 we show for this example of micro-macro coupling the SMM (left) and the coupling template (right). The later is defined in [22] and shows how the operators as defined in (4) are coupled to each other. A close inspection of this coupling template shows indeed that upon each iteration of the macroscopic process the microscopic process executes a complete simulation, taking input

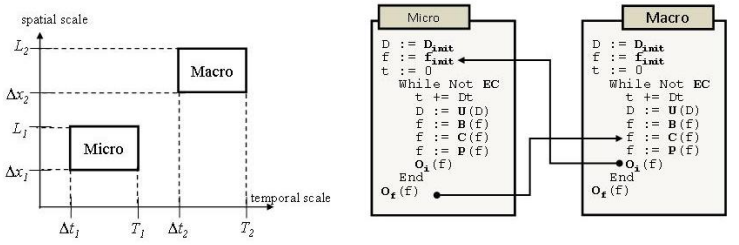



**Fig. 2.** Micro-macro coupling example. Left: SSM. Right: coupling template.

from the macroscopic process. In turn, the output from the microscopic process is fed into the Collision operator of the macroscopic process.

We can formulate the CxA dynamics as follows (based on Eq. (5))

$$\begin{aligned} \hat{f}_{1,h_1}^{t_{1,0}} &= f_{init,1}[\hat{f}_2] \\ \hat{f}_{1,h_1}^{t_{1,n_1}+\Delta t_{h_1}} &= (B_{h_1} \circ P_{h_1} \circ C_{h_1})\,[\hat{f}_{1,h_1}^{t_{1,n_1}}], \quad n_1 > 0 \end{aligned} \tag{6}$$

$$\begin{aligned} \hat{f}_{2,h_2}^{t_{2,0}} &= f_{init,2} \\ \hat{f}_{2,h_2}^{t_{2,n_2}+\Delta t_{h_2}} &= \left(B_{h_2} \circ P_{h_2} \circ C_{h_2}[\hat{f}_1]\right)[\hat{f}_{h_2}^{t_{2,n_2}}], \quad n_2 > 0 \end{aligned}$$

where 1 refers to the micro-scale and 2 to the macro-scale. As drawn in Fig. 1, the micro-scale model 1 is run until completion (i.e. until the final time $T_1$), then a single time step $\Delta t_{h_2}$ is performed for the macro-scale model.

We can now compare an estimation of the execution time of the CxA model of Fig. 2 with that of using a single CA for the same system, as in the left part of Fig. 1. For the single CA the execution time would be $T_{CA} = k_{CA} \frac{T_2}{\Delta t_1} \left(\frac{L_2}{\Delta x_1}\right)^D$, which is Eq. (1) using the subscripts as introduced in Fig. 2. For the CxA the execution time becomes

$$T_{CxA} = k_2 \frac{T_2}{\Delta t_2} \left(\frac{L_2}{\Delta x_2}\right)^D \left(1 + k_1 \frac{T_1}{\Delta t_1} \left(\frac{L_1}{\Delta x_1}\right)^D\right). \tag{7}$$

Note that $k_{CA}$, $k_1$ and $k_2$ are constant numbers. The execution time is determined by the total number of iterations of the macro scale process, and the time per iteration is in turn determined by on the one hand the macro scale operators and on the other hand the time needed for the micro scale CA simulations. Next one can compute a speedup, comparing the single scale CA formulation and the CxA formulation as $S = T_{CA}/T_{CxA}$. After some algebra we find

$$S = \left(\frac{k_{CA}}{k_2} \frac{\Delta t_2}{\Delta t_1} \left(\frac{\Delta x_2}{\Delta x_1}\right)^D\right) \Big/ \left(1 + k_1 \frac{T_1}{\Delta t_1} \left(\frac{L_1}{\Delta x_1}\right)^D\right). \tag{8}$$

Under the reasonable assumption that the execution time for a full micro scale simulation needs much more time than a single iteration of the macro scale

model, i.e. when $k_1 \frac{T_1}{\Delta t_1} \left(\frac{L_1}{\Delta x_1}\right)^D >> 1$, Eq. (8) reduces to $S = \frac{k_{CA}}{k_1 k_2} \frac{\Delta t_2}{T_1} \left(\frac{\Delta x_2}{L_1}\right)^D$. Note that $\frac{\Delta t_2}{T_1} > 1$ and $\frac{\Delta x_2}{L_1} > 1$, and can be interpreted as the distance on the SSM (Fig. 2). So, if the scale separation is large enough, the obtained speedups can be huge, principally rendering a CxA simulation feasible.

### 3.4 Error Analysis

The previous arguments demonstrate the improvements in computational efficiency offered by the CxA formulation. On the other hand, replacing the original multi-scale model with many coupled single-scale algorithms, we face a partial loss of precision.

A possible measure of this lowering in accuracy can be obtained considering the difference in the numerical results of he original $CA_h$ and the Complex Automaton $CxA_H$, which we call *scale-splitting error* [24].

In general, a detailed and rigorous investigation of the scale-splitting error requires a good base knowledge of the single scale Automaton and of the full multiscale algorithm. An example of error investigation using the formalism introduced in section 3.2 for a simple CxA model can be found in [24] and another contribution submitted to ACRI 2008. [25]

## 4 Conclusions

In this manuscript we have introduced a methodology for CA based multi-scale modeling and simulation. Our approach, Complex Automata, relies on the possibility to split a multi-scale model into a set of mutually couple single scale CAs. This manuscript introduces a formalism for CxA modeling and presents a few partial results for the case of micro-macro coupling. Development of CxA methodology is an ongoing project, and in the near future we intend to publish a more concise formalism, containing a larger set of examples.

**Acknowledgments.** This research is supported by the European Commission, through the COAST project (EU-FP6-IST-FET Contract 033664, www.complex-automata.org).

## References

1. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press, Cambridge (1998)
2. Deutsch, A., Dormann, S.: Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis. Birkhäuser, Basel (2005)
3. Sloot, P.M.A., Hoekstra, A.G.: Modeling Dynamic Systems with Cellular Automata. In: Fishwick, P. (ed.) Handbook of Dynamic System Modeling, ch. 21, Chapman & Hall/CRC (2007)
4. Lloyd, S.: Computational Capacity of the Universe. Phys. Rev. Lett. 23, 237901 (2002)

5. Zuse, K.: The Computing Universe. Int. J. Theor. Phys. 21, 580–600 (1982)
6. Zuse, K.: Rechnender Raum, http://www.idsia.ch/~juergen/digitalphysics.html
7. Wolfram, S.: A New Kind of Science. Wolfram Media, Inc. (2002)
8. Bader, D.A. (ed.): Petascale Computing: Algorithms and Applications. Chapman & Hall/CRC (2008)
9. Hoekstra, A.G., Portegies Zwart, S., Bubak, M., Sloot, P.M.A.: Towards Distributed Petascale Computing. In: Bader, D.A. (ed.) Petascale Computing: Algorithms and Applications, ch. 8, Chapman & Hall/CRC (2008)
10. Sloot, P.M.A., Frenkel, D., van der Vorst, H., et al.: White paper on computational e-science, studying complex systems in silico, a national research invitiative (2007), http://www.science.uva.nl/research/pscs/papers/archive/Sloot2007a.pdf
11. Special issue on Multiphysics modeling. IEEE Comput. Sci Eng. 7, 14–53 (2005)
12. SIAM Multiscale Modeling and Simulation, http://epubs.siam.org/sam-bin/dbq/toclist/MMS
13. International Journal for Multiscale Computational Engineering, http://www.edata-center.com/journals/61fd1b191cf7e96f.html
14. White, R.: Modeling Multi-scale Processes in a Cellular Automata Framework. In: Portugali, J. (ed.) Complex Artificial Environments, Simulation, Cognition and VR in the Study and Planning of Cities, pp. 165–177. Springer, Heidelberg (2006)
15. Ribba, B., Alarcón, T., Marron, K., Maini, P.K., Agur, Z.: The Use of Hybrid Cellular Automata Models for Improving Cancer Therapy. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 444–453. Springer, Heidelberg (2004)
16. Lin, C., Lai, Y.G.: Lattice Boltzmann method on composite grids. Phys. Rev. E 62, 2219–2225 (2000)
17. Dupuis, A., Chopard, B.: Theory and Applications of an Alternative Lattice Boltzmann Grid Refinement Algorithm. Phys. Rev. E 67, 066707 (2003)
18. Weimar, J.R.: Coupling Microscopic and Macroscopic Cellular Automata. Int. J. of Parallel Computing 27, 601–611 (2001)
19. Baas, N.A., Helvik, T.: Higher Order Cellular Automata. Advances in Complex Systems 8, 169–192 (2005)
20. Israeli, N., Goldenfeld, N.: Computational Irreducibility and the Predictability of Complex Physical Systems. Phys. Rev. Let. 92, 74105 (2004)
21. Hoekstra, A.G., Lorenz, E., Falcone, J.-L., Chopard, B.: Towards a Complex Automata Framework for Multi-scale Modeling: Formalism and the Scale Separation Map. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4487, pp. 922–930. Springer, Heidelberg (2007)
22. Hoekstra, A.G., Lorenz, E., Falcone, J.-L., Chopard, B.: Toward a Complex Automata Formalism for MultiScale Modeling. Int. J. Multiscale Comput. Eng. 5, 491–502 (2007)
23. W., E., Engquist, B., Li, X., Ren, W., Vanden-Eijnden, E.: Heterogeneous Multiscale Methods: A Review. Commun. Comput. Phys. 2, 367–450 (2007)
24. Caiazzo, A., Falcone, J.L., Chopard, B., Hoekstra, A.G.: Scale-splitting error in Complex Automata models for Reaction-Diffusion systems. In: Bubak, M., et al. (eds.) ICCS 2008, part II. LNCS, vol. 5102, pp. 291–300. Springer, Heidelberg (2008)
25. Caiazzo, A., Falcone, J.L., Chopard, B., Hoekstra, A.G.: Error Investigations in Complex Automata models for Reaction-Diffusion systems. In: Umeo, H., et al. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 260–267. Springer, Heidelberg (2008)

# Reconfiguring Circuits Around Defects in Self-Timed Cellular Automata

Tadashi Kunieda[1], Teijiro Isokawa[1], Ferdinand Peper[2,1], Ayumu Saitoh[1], Naotake Kamiura[1], and Nobuyuki Matsui[1]

[1] Division of Computer Engineering, Graduate School of Engineering, University of Hyogo, 2167 Shosha, Himeji, 671-2280, Japan
isokawa@eng.u-hyogo.ac.jp
[2] Nano ICT Group, National Institute of Information and Communications Technology, Japan

**Abstract.** For the realization of nanocomputers it will be important to have built-in defect-tolerance, which is the ability to overcome the unreliability caused by defective components. This paper explores defect-tolerance for nanocomputers based on Self-Timed Cellular Automata—an asynchronously timed CA of which the functionality can be expressed through a small number of transition rules. The proposed method assumes that defects are coped with in an initial phase by detecting and isolating them in cellular space from non-defective cells. The phase after this—the main topic of this paper—includes a scheme to efficiently lay out circuits on the cellular space in areas that are not affected by defects. The scheme is self-contained, i.e., it is carried out through the transition rules defined for the CA and does not require external circuitry.

## 1 Introduction

Cellular Automata(CA) have much potential as architectures for computers built with nanometer-scale feature sizes (nano-computers), because their regular structures and local connectivity are very suitable for manufacturing by molecular self-assembly [1,2]. Two important issues for realizing nanocomputers are (1) the reduction of power consumption and heat dissipation and (2) reliability. The first issue has resulted in models governed by an asynchronous scheme of timing, which offers an energy-efficient alternative to the clock signals commonly used in synchronous schemes. Models combining the key advantage of CA's regularity with the advantages of asynchronous timing thus offer a promising road to the realization of nanocomputers. One such a model is the Self-Timed Cellular Automaton (STCA), which requires only a small number of transition rules to express the functionality required for computation. Computations on this model are typically carried out by simulating so-called Delay-Insensitive circuits on the cell space [3,4], which are a kind of circuit that is robust to delays in the signals processed by it.

The second issue concerns the reduced reliability of nanometer-scale devices as compared to their VLSI counterparts, due to defects arising during manufacturing. In a previous study on defects in STCA models [5], defective cells in

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 200–209, 2008.

the cellular space are detected and isolated in an initial phase, followed by the placement of circuit modules in free areas of the cell space that lie between isolated defects. This scheme is, however, rather inefficient in its use of cell space, since the placement of modules is rudimentary, allowing modules to be placed only along a narrow zone that can be extended in one dimension only, but not in a second dimension. This would leave some non-defective areas unused for the placement of modules. The scheme in [5] also lacks completeness, in that the placement of circuit modules is not self-contained. A more complete mechanism would require, for example, a so-called universal constructor [6] in cellular space to direct the detection and measurement of defect-free areas and to direct the placement of modules.

In this paper, we present a novel scheme for defect tolerance on STCA that is an improved version of the scheme in [5]. A constructor for placing circuit modules is implemented that has the ability to adjust the placement location of each module by measuring the sizes of the free areas in the cellular space. All two dimensions of the cell space are efficiently used for the placement of the modules into which the circuit can be subdivided. In the proposed scheme, the circuit modules communicate with each other through a combination of strategies. First, signals have the ability to go around defective areas autonomously, in a similar way as in [5]. This ability is implemented through the model's transition rules. Second, for the case modules' placements causes their input and output lines to be misaligned with each other, there is a bus configuration implemented on the cell space that carries signals from source to the appropriate destination in a flexible way. Lacking in the model in [5], this bus lies at the basis for the proposed model's ability to efficiently use the two dimensions of the cell space for laying out circuit modules.

## 2 Preliminaries

### 2.1 Self-Timed Cellular Automaton

A self-timed cellular automaton (STCA)[3] is a two-dimensional asynchronous CA of identical cells, each of which has a state that is partitioned into four parts in one-to-one correspondence with its neighboring cells. Each part of a cell has a state. Figure 1 shows a cell in STCA where the states of the partitions are denoted as $p_n$, $p_e$, $p_s$, and $p_w$. For simplicity the diagonal lines indicating partitions in cells are left out in the remainder of the paper. Each cell undergoes transitions in accordance with a transition function $f$ that operates on the four parts of the cell $p_n$, $p_e$, $p_s$, $p_w$ and the nearest part of each of its four neighbors $q_n$, $q_e$, $q_s$, $q_w$. The transition function $f$ is defined by

$$f(p_n, p_e, p_s, p_w, q_n, q_e, q_s, q_w) \rightarrow (p'_n, p'_e, p'_s, p'_w, q'_n, q'_e, q'_s, q'_w), \tag{1}$$

where a state symbol to which a prime is attached denotes the new state of a partition after update (see Fig.2). Function $f$ can be described by a finite set of

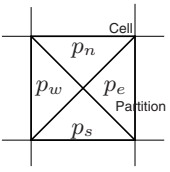



**Fig. 1.** A cell in Self-Timed Cellular Automaton

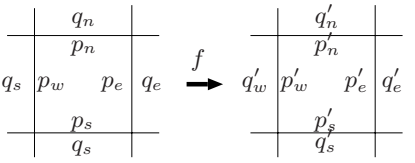
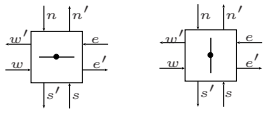
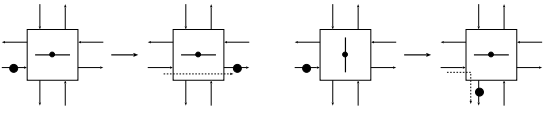



**Fig. 2.** Transition rule in accordance with the function $f$




(a) H-state (b) V-state

**Fig. 3.** A Rotary Element



(a) Parallel case (b) Orthogonal case

**Fig. 4.** The operation of an RE on a signal

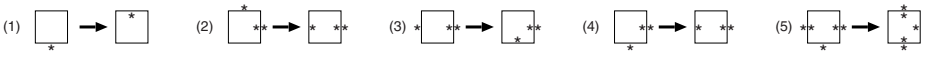



**Fig. 5.** Transition rules implementing the RE. A '$*$' symbol in a partition denotes the corresponding state '$*$', and a blank denotes the state '  '.

transition rules on the STCA. Dummy transitions, which are transitions without any changes of the partition states, are not included in a set of the transition rules, so we assume that the left-hand side of Fig. 2 always differs from the right-hand side. Furthermore, we assume that transition rules on an STCA are rotation-symmetric, thus each of the rules has four rotated analogues.

In an STCA, transitions of the cells occur at random times, independent of each other. Furthermore, it is assumed that neighboring cells never undergo transitions simultaneously to prevent a situation in which such cells write different values in shared partitions at the same time.

There are several approaches to perform computation on STCAs, such as simulating synchronous CA on them, or embedding delay-insensitive circuits on them[7,4]; we will use the latter approach, because of its lower overhead in terms of the number of required cell states and transition rules. We use Morita's Rotary Element [8] for conducting computation. This element has been proven to be computational universal.

**Rotary Element(RE) and Its Implementation on STCA.** A *Rotary Element (RE)* [8] is a logic element with four input lines $\{n, e, s, w\}$, four output lines $\{n', e', s', w'\}$, and two states—the H-state and the V-state—which are displayed as horizontal and vertical rotation bars respectively (Fig. 3). A signal coming from a direction parallel to the rotation bar passes straight through to the opposite output line, without changing the direction of the bar (Fig.4(a));

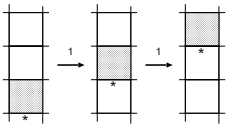


**Fig. 6.** Signal transmission northwards

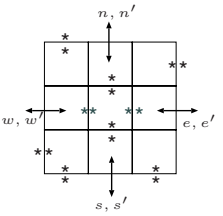



**Fig. 7.** RE realized on STCA

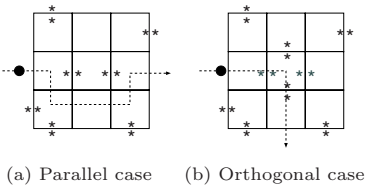



**Fig. 8.** Signal passing through RE

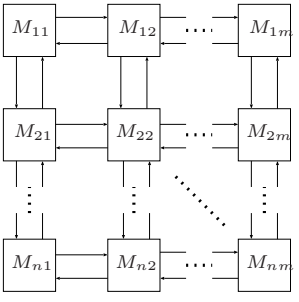



**Fig. 9.** Modules are arranged in two-dimensional space according to a mesh-connected scheme

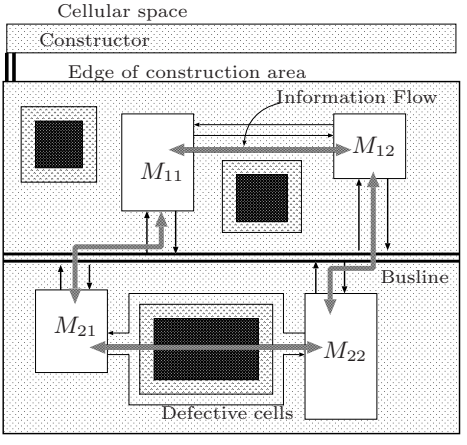



**Fig. 10.** Cellular space with four circuit modules ($M_{ij}$) and three isolated areas of defective cells (black surrounded by gray layers)

a signal coming from a direction orthogonal to the bar turns right and rotates the bar by 90 degrees (Fig. 4(b)).

Embedding an arbitrary RE-circuit on the cellular space of STCA can be achieved by the two states of a partition and the five transition rules shown in Fig. 5 (see [6] for details). Rule #1 defines a signal used for ordinary computation. Applying it to the successive shaded cells in Fig. 6 results in a signal proceeding northwards on a straight continuous path of cells, all bits of which are 0. Transmitting a signal towards the east, south, or west is done similarly, because of the rotation-symmetry of transition rules. Rules #2–4 are responsible for routing a signal according to the direction from which the signal comes from. The operation of the rotation bar in an RE is conducted by rule #5.

A rudimentary RE is shown in Fig. 7. The input/output paths at the sides of this RE are bi-directional. Connecting so-called Input/Output Multiplexers [6] to these paths results in a regular RE with four input and four output paths, all unidirectional. Figure 8 illustrates the traces of a signal input to a rudimentary RE from a line (a) parallel with or (b) orthogonal to the rotation bar. Since all the

elements of RE-circuits can be realized by stable partition pairs, any circuit on this STCA can be represented by a certain configuration of stable partition-pairs.

## 3 Laying Out Circuit Modules on STCA

In our model circuit modules are connected through a mesh-connected scheme, with the nodes in the mesh represented by modules (Fig. 9). The input (resp. output) ports of a module are matched to the output (resp. input) ports of its neighboring modules in a one-to-one fashion, i.e., the signal lines between modules are not merged or bifurcated.

In the underlying STCA model, each cell partition can be in one of 18 states, which are denoted by a set of symbol $\{0, \cdots, 9, A, \cdots, H\}$. The partition state '$*$' and '  ' in section 2 are mapped to the partition state 2 and 3 in this section, respectively.

Defects in the CA are of the so-called stuck-at-fault type [5], i.e., a cell's state cannot be updated by the cell itself or by an outside source, but it can still be read by neighboring cells. Defects are detected and isolated through a procedure outlined in [5]. This results in a cell space (Fig. 10) in which defective areas (colored black) are isolated by a layer of cells (the gray cells surrounding the black areas) all of which are in the same state.

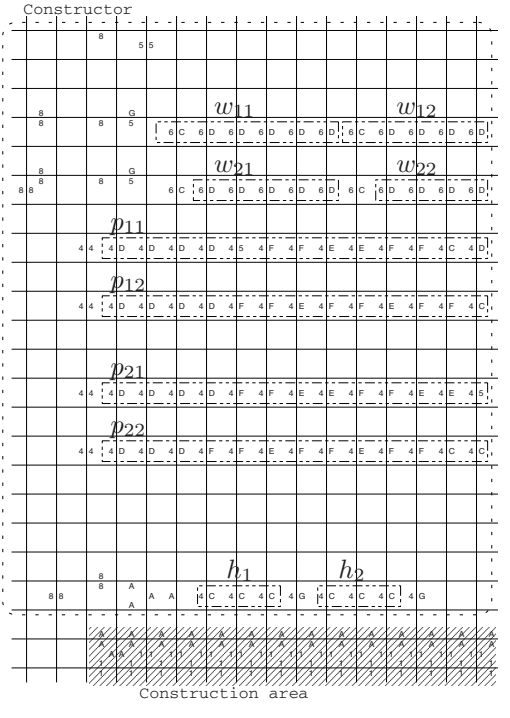



**Fig. 11.** Signals in constructor. For reasons of space only the left hand side of the constructor configuration is shown.

The areas between defects are used for the placement of circuit modules. Fig. 10 shows four modules placed in a cell space with three defective (and isolated) areas. There is also a constructor at the top, which directs the construction of the circuit on the cell space. Modules exchange computational signals between each other via signal lines. Module $M_{11}$ and $M_{12}$ are directly connected to each other, without any obstacles in between. Modules $M_{21}$ and $M_{22}$, on the other hand, have a large area of defective cells in between, which obstructs signals between them. Communication

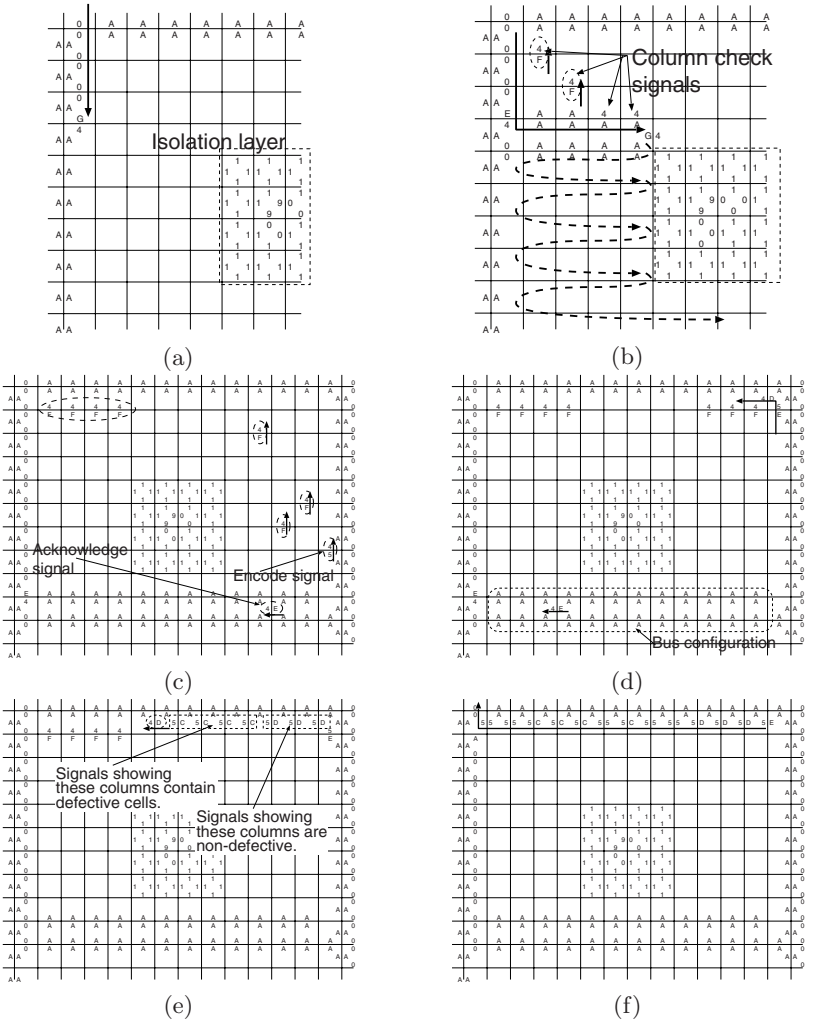



**Fig. 12.** Scanning free areas, followed by placing a bus. (a) *h*-signal injected from the top-left proceeds to the south. (b) When the encoded height is reached, the head starts to scan horizontally. Each step right generates a *column-check signal*, which, moving upwards, will be collected at the top of the area, but only at those columns that do not contain defects. Upon running into the isolation around defects, the head will move down one row, and starts a new scan. (c) When a whole row is free of defects it will be used for the bus line, which is created as part of the scanning process. At the right side of the area a so-called *Encode-signal* moves up. An *Acknowledge-signal* moves to the left to open up the area below the bus line for a similar process if a new *h*-signal is available. (d) The Encode-signal transforms the column-check signals at the top of the area, (e) into a form suitable to move out of the construction area and (f) move back into the constructor.

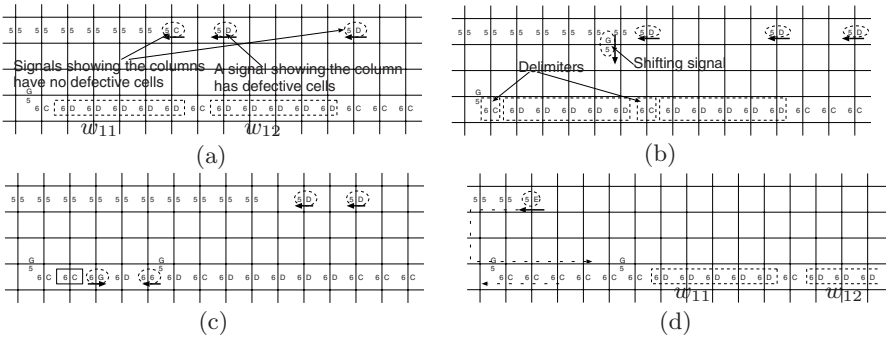



**Fig. 13.** Preparation for placement of circuit modules. (a) The train of signals encoding the positions of defect-free columns enters the constructor. Each of these signals encodes the absence resp. presence of defects in its corresponding column. (b) A signal encoding defects in a column generates a *Shift-signal*, which is dropped onto the $w$-signal below. (c) The Shift-signals cause the $w$-signal to be internally shifted, such as to reflect the proper positioning of modules in defect-free areas. (d) The train of signals and the dropped Shift-signals are cleaned up after use. Also, the $w$-signal starts to move toward the construction area to place the writing head in the appropriate position to begin processing of the $p$-signal.

is made possible in this case by signals having the ability to autonomously find their way around defects, as in [5]. Yet another case occurs between modules $M_{11}$ and $M_{21}$ (as well as between $M_{12}$ and $M_{22}$). Even though there are no defective areas between these modules, they are misaligned due to their forced placement between defective areas that are not aligned properly. Communications between these modules take place through a *Bus line* configuration, which is described in Section 3.2.

### 3.1 Scanning Non-defective Areas and Placing Modules

To place modules in the non-defective areas, we need to investigate the sizes of these areas and adjust the constructions signals such as to make the modules fit. Directly below the area in which a row of modules is to be placed we place a bus line, as part of the construction process. The constructor doing all this is attached at the top of the construction area (Fig. 10), in which the construction signals are injected in an appropriate order.

The constructor and the signals in it are shown in Fig. 11. In the initial stage, the signals are still static: they do not move until they receive an appropriate start signal. There are three types of signals: $h$-signals, denoted by $h_i$, express the maximum height in unary coding of the row $i$ of modules to be placed; $w$-signals, denoted by $w_{ij}$, express the width of module $M_{ij}$ in unary coding; and $p$-signals, denoted by $p_{ij}$, express the circuitry in module $M_{ij}$ (see Fig. 11).

These signals are processed at the writing head: they move the head in the construction area and direct the head to write appropriate information in the cells of the CA. The written information usually has the form of a stable pair of states in adjacent cells.

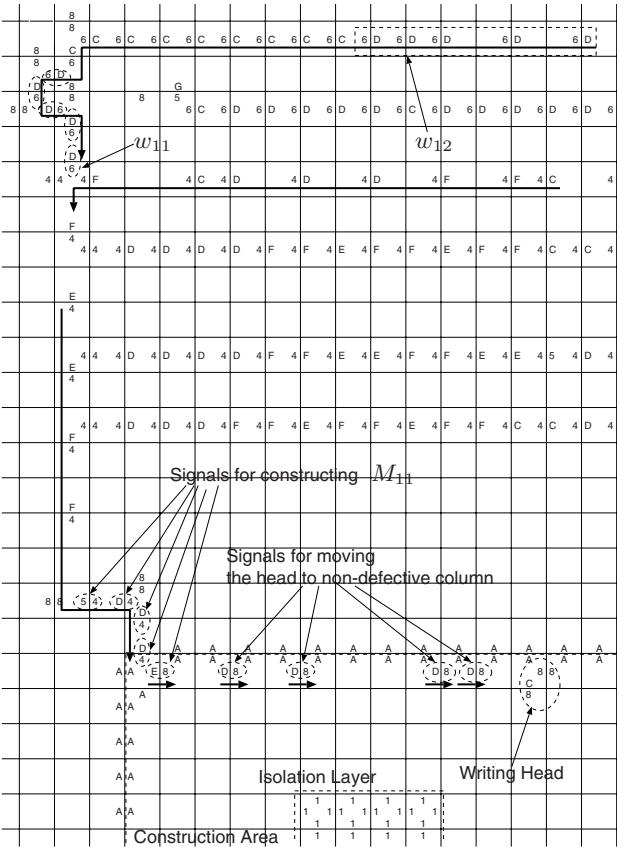



**Fig. 14.** Positioning of head by $w$-signals, followed by the placement of modules by $p$-signals

To place bus lines in the construction area, the constructor will scan the area for defects (Fig. 12).

Moving downward, this scanning process places a bus line at the first completely empty row of cells that it finds (for details see caption of Fig. 12).

An important product of the scanning process is a train of signals that encodes for each column of cells in the area whether there is a defect. This train of signals moves back into the constructor, and is placed in a position above the corresponding construction signal $w$, with the purpose to encode the positions of the defect-free areas in $w$ (Fig. 13).

The actual construction of the modules on the cell space starts with a $w$-signal moving into the construction area to position the writing head. The $p$-signal following directly after this will then result in the modules being written at the appropriate position, i.e. in a defect-free area (Fig. 14). Details of the construction process can be examined on the web page in [9].

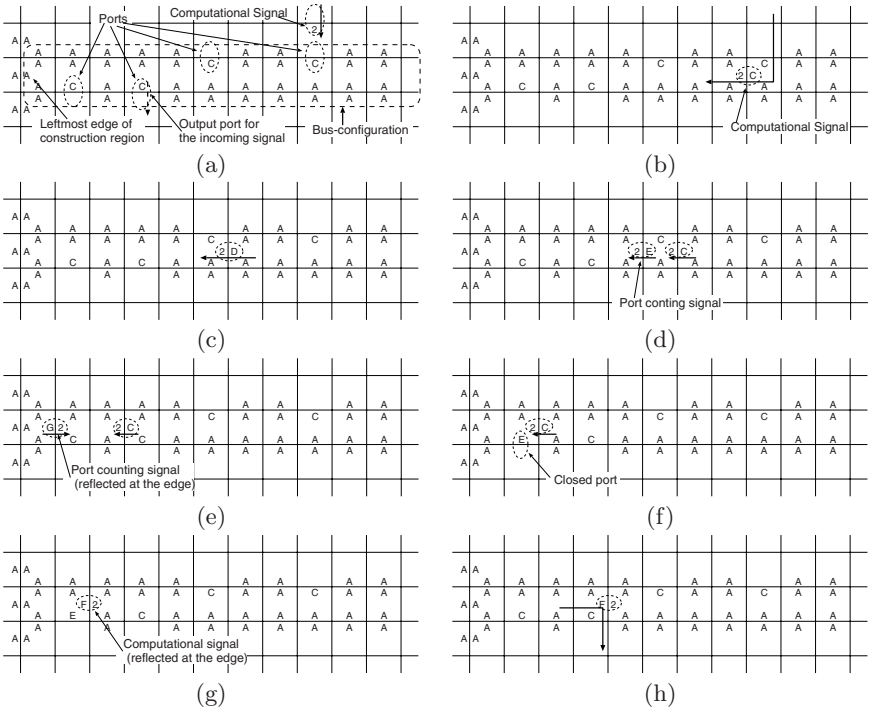



**Fig. 15.** A computational signal going through a bus. (a) Basic structure of a bus line. In this case the bus line contains two ports at its top and two ports at its bottom. (b) Signal entering right-most port at the top. This signal is supposed to exit from the right-most port at the bottom. (c) Signal moves along the bus line. (d) When encountering a port at its right, the signal emits a counting signal at its head. (e) Counting signal travels left and is reflected by the left side of bus. (f) The reflected counting signal closes the next port at its right and is annihilated. (g) The computational signal is also reflected, and it passes closed ports at its right, in the process opening them for the next signal. (h) Upon encountering an open port at its right, the computational signal exits via this port. By using counting signals to keep track of the number of ports to be passed, the computational signal can always find its appropriate output port.

### 3.2 Computation by the Modules

Once constructed, the modules can receive a computational signal, which will start the computation. Since this process is similar to that described in [5], we will not go into the details here, except for describing how communication via the bus line takes place, due to its novelty. The bus line contains ports that line up with the ports of the constructed modules. These ports on the bus line have been placed there by the construction process. A pair of ports on the bus line—one at the top of the bus line and one at the bottom—then connects two terminals, thus forming a connection between the corresponding modules. Transition rules are designed such that signals always move between matching pairs. The details of this process are outlined in Fig. 15. Again, details are on the web page in [9].

## 4 Conclusion

We have presented a novel scheme for defect tolerance on STCA that is able to flexibly place circuitry in cell space. The scheme is able to cope with defects in the construction area, but not with defects in cells used by the constructor itself or cells over which signals pass to and from the constructor, at the edges of the construction area. The cells that should be defect-free may be realized by reliable technology in implementations.

A novel element in our construction is a bus line configuration, which is used for communication between modules that are vertically unaligned. Due to these improvements, the circuit can be subdivided into modules in two-dimensional space and can be placed in non-defective areas more efficiently than in the previous scheme in [5].

The number of states per partition is 18 and the number of transition rules is 342. Both numbers are much larger than those in [5]. One reason for this is due to the use of an increased number of signal types and the need to appropriately control all these different signals. We expect that this number can be reduced if the bus line configuration can be applied not only for vertical connections between modules, but also for horizontal connections, since this would make redundant the previously proposed mechanism in [5] to guide signals around defective areas in an autonomous way.

## References

1. Biafore, M.: Cellular automata for nanometer-scale computation. Physica D 70, 415–433 (1994)
2. Durbeck, L.J.K., Macias, N.J.: The cell matrix: an architecture for nanocomputing. Nanotechnology 12, 217–230 (2001)
3. Peper, F., Lee, J., Adachi, S., Mashiko, S.: Laying out circuits on asynchronous cellular arrays: a step towards feasible nanocomputers? Nanotechnology 14, 469–485 (2003)
4. Lee, J., Peper, F., Adachi, S., Morita, K., Mashiko, S.: Reversible computation in asynchronous cellular automata. In: 3rd Int. Conf. on Unconventional Models of Computation 2002, pp. 220–229. Springer, Heidelberg (2002)
5. Isokawa, T., Kowada, S., Takada, Y., Peper, F., Kamiura, N., Matsui, N.: Defect-Tolerance in Cellular Nanocomputers. New Generation Computing (2007)
6. Takada, Y., Isokawa, T., Peper, F., Matsui, N.: Universal construction and self-reproduction on self-timed cellular automata. Int. J. of Modern Physics C 17(7), 985–1007 (2006)
7. Peper, F., Lee, J., Abo, F., Isokawa, T., Adachi, S., Matsui, N., Mashiko, S.: Fault-Tolerance in Nanocomputers: A Cellular Array Approach. IEEE Trans. on Nanotechnology 3(1), 187–201 (2004)
8. Morita, K.: A simple universal logic element and cellular automata for reversible computing. In: Margenstern, M., Rogozhin, Y. (eds.) MCU 2001. LNCS, vol. 2055, pp. 102–113. Springer, Heidelberg (2001)
9. http://www.eng.u-hyogo.ac.jp/eecs/isokawa/acri08

# Theory of Composing Non-linear Machines with Predictable Cyclic Structures

Debdeep Mukhopadhyay[1], Dipanwita RoyChowdhury[2], and Chester Rebeiro[3]

[1] Assistant Professor, Department of Computer Science and Engg., Indian Institute of Technology Kharagpur, India
debdeep@cse.iitkgp.ernet.in
[2] Professor, Department of Computer Science and Engg., Indian Institute of Technology, Kharagpur, India
drc@cse.iitkgp.ernet.in
[3] MS Student, Department of Computer Sc and Engg, Indian Institute of Technology Madras, India
chetrebeiro@gmail.com

**Abstract.** The paper proposes construction techniques for group non-linear Cellular Automata (CA) composing smaller non-linear invertible CA with linear group CA. We prove that such a scheme generates machines with state transitions having predictable cyclic properties. We show that with appropriate choice of the rules of the linear CA we may obtain invertible, balanced Boolean mappings with strong non-linearity. Extensive experimental results are provided to support the claims made.

**Keywords:** Non-linearity, Group Cellular Automata, cyclic properties, Boolean functions.

## 1 Introduction

Cellular Automata (CA) is an important mathematical abstraction for modeling complex behavior ([1]). It may be conceived as a discrete lattice of cells, with local inter-connects and an array of combinational gates, deriving their inputs from local neighborhood. A linear CA employs XOR gates, whereas use of XNOR gates realizes what is known as complementary CA. Both the above type of machines form the additive Cellular Automata. However if we use gates like AND and OR gates, then we obtain a non-additive or non-linear CA. These kind of CA are hard to predict, mainly because the super-position principle of states does not work ([2]). There are very few methods known to analyze these class of machines better than an exhaustive simulation which takes $O(t)$ time to predict the state of machine at the $t^{th}$ step. The results presented in ([2]) do not show construction methods for non-linear group CA with explicit group properties, analogous to linear CA. Indeed these group properties are one of the reasons why linear CA has become an important field of study, both from the research and application point of view. The absence of such methods is one of the reasons why non-linear CA has not found much applications. However they

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 210–219, 2008.

would be of much significance in applications like the design of strong block and stream ciphers, which require non-linearity as one of the prime features for their security ([3]). It has been shown in [3], that the affine nature of linear CA is a major bottleneck to its application in cryptography [4]. However, the cyclic properties of the linear CA structures is an important feature as they help in performing decryption using the same encryption algorithm, thus leading to efficient cryptographic schemes [4]. Thus it would be ideal for the application of CA to cryptography to develop a solution which combines the best of both the types of CA: non-linear CA with cyclic group properties. In order to achieve this, we apply a result from [5,6].

Design of cryptographically robust Boolean functions is a challenging task [7]. They are employed for constructing various primitives for cryptography. For hardware based stream ciphers, they are employed as the non-linear combiner functions of the states generated by the array of Linear Feedback Shift Registers (LFSRs). The combiner outputs a key stream $(K_i, i \geq 0)$ which is XORed with the message stream $(M_i, i \geq 0)$, to result in the cipher bit stream $(C_i, i \geq 0)$. The operation may be represented as follows: $C_i = M_i \oplus K_i$.

In order to improve the throughput of stream ciphers, it is often advised in the cryptographic literature to combine the output bits of $n$ stage LFSR by an $n \times k$ mapping, to obtain $k$ bits of key streams [8]. The key bits are then XORed with $k$ bits of the message to obtain $k$ bits of the cipher bits. The throughput of the stream cipher is thus $k$ times more than when using a single Boolean function as the combiner. However, as reported attempts show the construction of the non-linear Boolean functions are not trivial and comes at the cost of a large overhead on hardware. The area required to implement the Boolean functions grow exponentially. The problem becomes even more challenging when more than one Boolean function is desired. The non-linear mappings, often referred to as Substitution Boxes (*S-Boxes*) are also used in block ciphers, like DES and AES. For block ciphers like AES they are further required to be invertible, that is the mappings should form a permutation. With the pressing need of providing security in resource constrained environments, like hand-held devices, smart cards, vehicular systems, we require more elegant solutions to realize non-linear mappings with lesser hardware requirements. This paper investigates whether the *simple rules* of Cellular Automata can be used to generate invertible, non-linear mappings with predictable cyclic structures.

In this paper, we propose a method to compose small non-linear invertible Cellular Automata (CA) with additive group Cellular Automata to construct large non-linear machines with predictable cyclic properties. Our claim is that the state transitions of the composed machine posses the cyclic structures of the additive machine and hence are easy to analyze. At the same time, the state transitions define a mapping with the much needed non-linearity. We provide experimental evidence to show that governing the rules of the linear CA we are able to control the state transitions of the composed non-linear machine.

The outline of the paper is as follows: *section 2* presents the preliminaries required for the work. *Section 3* shows the composition of group non-linear

machines. *Section 4* describes the construction of small non-linear invertible CA and *section 5* shows the construction of a 6 cell non-linear CA with predictable cyclic structures. *Section 6* concludes the work.

## 2 Preliminaries on Cellular Automata and Non-linearity of Boolean Functions

Cellular Automaton (CA) was investigated as a mathematical model for self-organizing statistical systems [9]. The CA structure can be viewed as a discrete lattice of sites (cells) where each cell can assume either value 0 or 1. The next state of a cell is assumed to depend on itself and on its two neighbours (three-neighbourhood dependency). The cells evolve in discrete time steps according to some deterministic rule that depends only on local neighbours. In effect, each cell consists of a storage element (D-flipflop/DFF) and a combinational logic implementing the next state function. The next state function (transition) for a three neighbourhood CA cell can be expressed as follows: $q_i(t+1) = f[q_{i-1}(t), q_i(t), q_{i+1}(t)]$

Here, $i$ denotes the position of an individual cell in the one-dimensional array of cells and $t$ denotes the time step in the evolution of the CA. The output state of the $i^{th}$ cell at the $t^{th}$ and $(t+1)^{th}$ time step are respectively denoted by $q_i(t)$ and $q_i(t+1)$. The transition function is indicated by $f$ and is realized through combinational logic. The combination logic is described by a term called *rule* of the CA, which is the decimal equivalent of the truth table of the next state function, $f$. For example, rule 150 denotes the combinational logic: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$.

The state transition of a CA which uses only XORs and XNORs (additive CA) can be expressed as $Y = T(X)$. In the above equation $T$ is an $n \times n$ matrix to represent the characteristic matrix of an $n$ cell CA. The input state of the CA is denoted by the $n$-bit vector $X$ while $Y$ denotes the output bit vector of the CA. An important class of CA is the group CA. A CA is a group CA iff the determinant det $T$=$|T|$=1, where $T$ is the characteristic matrix of the CA. The CA under such a transformation forms a cyclic group. An $n$-cell maximum length CA is a sub-class of group CA and is characterized by the presence of a cycle of length $(2^n - 1)$ with all non-zero states.

If the state transition of the CA employs operations like *AND/OR*, then the CA is a non-additive or non-linear Cellular Automata. Each output of such a CA represents a non-linear Boolean function of the inputs. Next we present some definitions on the non-linearity of Boolean functions.

**Definition 1.** *Affine Function: We call* $h(x)$=$a_1x_1 \oplus \ldots \oplus a_nx_n \oplus c$ *an affine function where* $x = (x_1, \ldots, x_n)$ *and* $a_j$,$c \in GF(2)$. *In particular,* $h$ *will be called a* linear function *if* $c = 0$.

**Definition 2.** *Non-linear Function: Any Boolean function can be represented in the Algebraic Normal Form (ANF) as* $h(x) = a_0 \oplus \bigoplus_{i=1}^{n} a_i x_i \oplus \bigoplus_{i=1,j=2}^{n-1,n} a_{ij} x_i x_j \oplus$

$\ldots \oplus a_{1,2,\ldots,n} x_1 x_2 \ldots x_n$. *All the Boolean functions except the Affine Functions are* Non-linear Functions. *Here,* $x_i$, $a_i$, $a_{ij}$, $\ldots$, $a_{1,2,\ldots,n} \in GF(2)$.

We can quantify the non-linearity of Boolean functions as follows:

**Definition 3.** *Hamming Distance: Let* $f$ *and* $g$ *be functions on* $V_n$*, which is a vector of* $n$ *bits. Then* $d(f,g)=\sum_{f(x)\neq g(x)} 1$*, where the addition is over the reals, is called the Hamming distance between* $f$ *and* $g$.

**Definition 4.** *Non-linearity: Let* $\psi_0,\ldots$ $\psi_{2^{n+1}-1}$ *be the affine functions on* $V_n$. *Then* $N_f = min_{i=0,\ldots,2^{n+1}-1}\ d(f,\psi_i)$ *is called the non-linearity of* $f$*, where* $d$ *denotes the Hamming Distance between the two functions* $f$ *and* $\psi_i$. *It is well-known that the non-linearity of* $f$ *on* $V_n$ *satisfies* $N_f \leq 2^{n-1}\text{-}2^{n/2-1}$*, when* $n$ *is even.*

**Definition 5.** *Balancedness: The vector space of* $n$ *tuples of elements from GF(2) is denoted by* $V_n$. *Let* $f$ *be a (Boolean) function from* $V_n$ *to* $GF(2)$. *The truth table of* $f$ *is defined as* $(f(\alpha_0), f(\alpha_1),\ldots,f(\alpha_{2^n-1}))$*, where* $\alpha_i$, $i = 0, 1, \ldots, 2^n - 1$*, denote vectors in* $V_n$. $f$ *is said to be balanced if its truth table has an equal number of zeroes and ones.*

A non-linear reversible Cellular Automata essentially generates non-linear Boolean functions at each output bit. It may be easily verified that if the CA is also reversible, then the mapping from the input to the output of the CA is one-one, and thus each bit is also a balanced Boolean function.

## 3 Proposed Composition of the Non-linear Cellular Automata with Group Properties

The construction of the group Non-linear machine has essentially two components: a small null neighbourhood invertible non-linear Cellular Automata and a large linear group CA. The invertible non-linear CA is denoted by the notation $f$ and its inverse by $f^{-1}$. The linear group CA is denoted by the symbol, $T$. The overall composed machine is denoted by the symbol, $T_{NCA}$.

**Theorem 1.** *If the linear Cellular Automata with* $T$ *as the characteristic matrix is a group Cellular Automata, so is the composed CA,* $T_{NCA} = f \circ T \circ f^{-1}$.

*Proof.* Since, $T$ represents a group CA, there exists an integer $l$ such that $T^l(X) = X$, for all $X$ [9]. Thus we have:
$Y = T^l_{NCA}(X) = (f \circ T \circ f^{-1})(f \circ T \circ f^{-1}) \ldots (\text{l times})(f \circ T \circ f^{-1})(X) = X$.

Thus for all $X$, $T^l_{NCA}(X) = X$. Thus the composed CA, denoted by $T_{NCA}$ is also a group CA.

The above result shows that the final CA represented by the transformation $T_{NCA}$ is a group CA.

Thus we conjecture: *We can construct group non-linear CA from invertible non-linear CA and group linear CA. We can predict that if the group CA generates a cyclic structure among the states, so does the composed non-linear machine. It is convenient to analyze the properties of the transitions of the composed*

*machines, as it can be done by analyzing the linear CA, for which we have efficient mathematical methods. For example, we can state that the maximum cycle lengths for both $T$ and $T_{NCA}$ will be identical.*

As we see the construction has two important parts: the construction of the non-linear layers with inverse and the application of linear group CA. In our construction we shall use small non-linear CA to realize the invertible non-linear mappings. In the next section we propose a method to realize small non-linear invertible CA.

## 4 Construction of Small Non-linear Invertible CA

We propose the construction of non-linear mappings of 3 bits to 3 bits. It may be noted that the same design principle may be generalised to other bits sizes as well. Consider a LFSR of size 3 with feedback polynomial $p(x) = x^3 + x + 1$, which is a primitive polynomial for $GF(2^3)$. The LFSR is loaded with the pattern $(1, 0, 0)$ and we obtain $(2^3 - 1) = 7$ patterns (**Fig. 1**). When the feedback polynomial is primitive, each of these sequences are called $m$-sequences or maximal length sequences. The truth table of each of the 3 bits with an extra zero added at the beginning is converted to an equivalent AND-XOR form using Reed-Muller expansion [10,11]. This gives a non-linear permutation, and hence leads to a non-linear invertible CA.

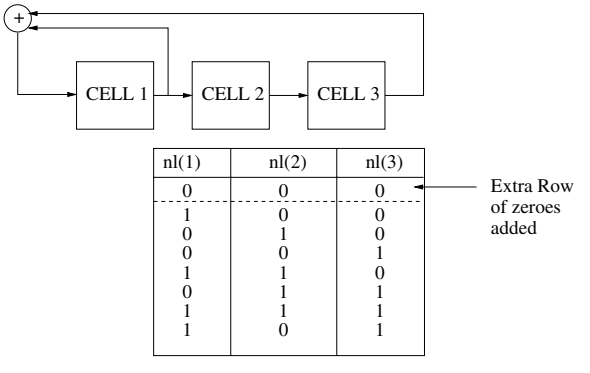



**Fig. 1.** The LFSR based construction of non-linear permutation in $V_3$

Minimizing the truth-table shown in **Fig. 1** the final Boolean equations are:

$$y_1 = x_1 \oplus x_3 \oplus x_2x_3$$
$$y_2 = x_1 \oplus x_2 \oplus x_1x_2 \oplus x_2x_3$$
$$y_3 = x_1x_2 \oplus x_2x_3 \oplus x_1x_3$$

Thus, the non-linear permutation is $y = \{y_1, y_2, y_3\}$. It may be verified that the final function is a non-linear permutation. The non-linearity of any non-zero

linear combination is 2, which is the maximum non-linearity for a Boolean function in 3 bits. Also the resultant Boolean function is balanced.

The inverse Boolean functions are as follows:

$$x_1 = y_1 \oplus y_3 \oplus y_1y_2 \oplus y_1y_3 \oplus y_2y_3$$
$$x_2 = y_1 \oplus y_2 \oplus y_1y_2 \oplus y_2y_3$$
$$x_3 = y_1 \oplus y_1y_3 \oplus y_2y_3$$

In the following section, we present the final construction of a 6 cell non-linear Group Cellular Automata using the above composition principle. It may be noted that although we have shown for a 6 cell structure, the results may be generalised to other sizes by appropriately choosing the dimensions of the small invertible non-linear CA. The modularity of the composition and CA helps in achieving this feature.

## 5 Construction of 6 Cell Non-linear Group Cellular Automata

**Fig. 2** depicts the construction of a 6 cell Non-linear Cellular Automata using 3 bit invertible non-linear CA and a 6 bit linear group CA.

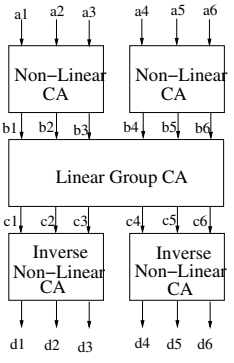



**Fig. 2.** The Composed 6 bit Non-linear Cellular Automata

The 3 bit non-linear CA and their inverse derive their logic from the equations described in *section 4*. For the linear group CA, we have chosen null boundary group structures with each cell having a radius of 1. The characteristic matrix of the linear group CA is represented as:

$$T = \begin{pmatrix} b_1 & b_2 & 0 & 0 & 0 & 0 \\ b_3 & b_4 & b_5 & 0 & 0 & 0 \\ 0 & b_6 & b_7 & b_8 & 0 & 0 \\ 0 & 0 & b_9 & b_{10} & b_{11} & 0 \\ 0 & 0 & 0 & b_{12} & b_{13} & b_{14} \\ 0 & 0 & 0 & 0 & b_{15} & b_{16} \end{pmatrix},$$

where $b_i$ $(1 \leq i \leq 16) \in \{0,1\}$. We varied the values of the $b_i's$ and noted down the cases when the characteristic matrix was that of a group Cellular Automata. We found that there are 3840 group Cellular Automata with the three neighbourhood structure mentioned above.

For the various values of the characteristic matrices we observed the cycle structure that the composed non-linear machine generates. We also noted the non-linearity of the bits obtained from the composed function for the choices for $T$'s. We explain some cases in the form of examples.

**Example 1.** *For a choice of the group linear CA with rules:* $< 90, 150, 90, 150, 90, 150 >$*, the composed machine is a non-linear maximum length Cellular Automata. The linear CA is a maximum cycle length with all the* 63 *non-zero states lying in one cycle. Thus, the maximum cycle length is* 63*. Using our theorem, we predict that the composed machine is also a maximum length CA with all the* 63 *non-zero states lying on one cycle. The diagram shown in* **Fig. 3** *depicts the state transitions of the composed non-linear machine and shows the correctness of our prediction.*

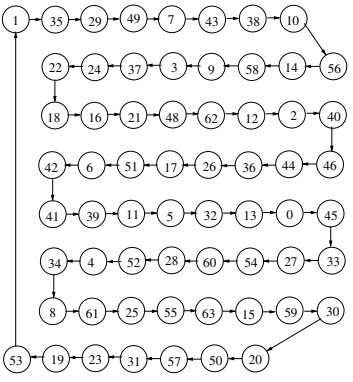



**Fig. 3.** State Transitions of a 6 bit non-linear maximum length CA

*The generated* $6 \times 6$ *mapping is a one-one mapping and is hence invertible. Each of the bits have a high non-linearity and are also balanced. The non-linearity of the* 6 *bits obtained are found to be quite high* $\{24, 24, 16, 16, 20, 22\}$*. It may be noted that the highest non-linearity possible for a* 6 *bit balanced Boolean function is* 24*.*

**Example 2.** *For a choice of the group linear CA with rules:* $< 153, 153, 153, 153, 153, 153 >$*, the composed machine is a non-linear Cellular Automata with cycles of equal length.*

*We know from [12] that the state transitions of such a complemented linear group CA form cycles of equal length. The length of each cycle is equal to*

$$l = 2^{\lfloor log 6 \rfloor + 1} = 8\,. \tag{1}$$

*Hence, the state transitions of the composed non-linear Cellular Automata also forms* 8 *cycles of length* 8 *each. The diagram shown in* **Fig. 4** *depicts the state*

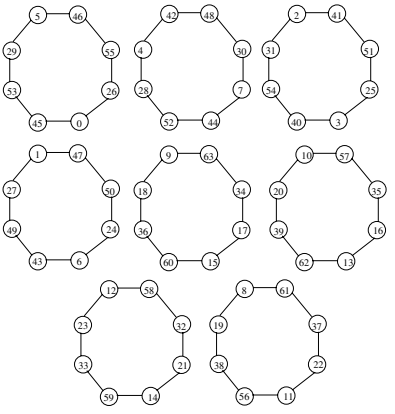



**Fig. 4.** State Transitions of a 6 bit non-linear complemented CA

**Table 1.** Cycle Structure and non-linearity of the composed $6 \times 6$ Cellular Automata based Mapping

| | Group CA Rule | Maximum Cycle Length of Linear CA | Maximum Cycle Length of Non-linear CA | Non-linearity |
|---|---|---|---|---|
| Weak Mappings | 102,240,204,170,60,204 | 3 | 3 | 16,16,18,16,16,0 |
| | 204,204,204,170,60,204 | 3 | 3 | 16,16,18,16,0,0 |
| | 102,204,204,170,60,204 | 6 | 6 | 16,16,18,0,0,16 |
| | 102,90,204,170,60,204 | 3 | 3 | 16,16,18,16,0,16 |
| | 204,90,240,170,240,204 | 4 | 4 | 16,18,18,16,0,16 |
| | 204,102,240,170,240,204 | 6 | 6 | 16,18,18,0,0,16 |
| | 102,102,240,170,240,204 | 6 | 6 | 16,18,18,0,16,0 |
| | 102,90,204,170,60,204 | 3 | 3 | 16,16,18,16,0,16 |
| | 102,240,204,170,240,204 | 6 | 6 | 16,18,18,16,16,0 |
| Moderate Mappings | 170,240,204,170,60,204 | 6 | 6 | 16,16,18,16,16,16 |
| | 240,170,240,170,240,204 | 2 | 2 | 16,18,18,16,16,16 |
| | 102,170,240,170,240,240 | 4 | 4 | 16,18,18,16,16,16 |
| | 102,90,240,170,240,204 | 14 | 14 | 16,18,18,16,16,16 |
| | 204,60,204,170,60,204 | 6 | 6 | 16,16,18,16,16,16 |
| | 170,60,204,170,60,204 | 3 | 3 | 16,16,18,16,16,16 |
| | 170,90,204,170,60,204 | 12 | 12 | 16,16,18,16,16,16 |
| | 204,102,204,170,60,204 | 6 | 6 | 16,16,18,16,16,16 |
| | 102,102,204,170,60,204 | 12 | 12 | 16,16,18,16,16,16 |
| Strong Mappings | 90,240,170,60,90,60 | 6 | 6 | 24,24,24,16,16,16 |
| | 102,240,170,60,90,60 | 6 | 6 | 24,24,24,16,16,24 |
| | 204,204,170,60,90,60 | 6 | 6 | 24,24,24,16,16,24 |
| | 102,204,170,60,90,60 | 6 | 6 | 24,24,24,16,16,16 |
| | 204,60,170,60,90,60 | 6 | 6 | 24,24,24,16,16,16 |
| | 170,60,170,60,90,60 | 6 | 6 | 24,24,24,16,16,24 |
| | 170,90,170,60,90,60 | 6 | 6 | 24,24,24,24,16,16 |
| | 102,90,170,60,90,60 | 6 | 6 | 24,24,24,24,24,16 |
| | 204,102,170,60,90,60 | 6 | 6 | 24,24,24,16,24,16 |
| | 102,102,170,60,90,60 | 6 | 6 | 24,24,24,16,24,16 |

*transitions of the complemented non-linear CA and shows that indeed they have exactly the same transition pattern as their linear counterpart (the complemented linear CA).*

*The generated* $6 \times 6$ *mapping is hence a one-one mapping and is hence invertible. The non-linearity of the* 6 *bits obtained are found to be quite high* $\{24, 24, 16, 18, 18, 20\}$.

The choice of the group linear CA has a strong influence on the final non-linearity of the bits generated by the composed machine. We performed an extensive experimentation on all the group rules, and found that they can be broadly categorized into three regions:

1. Region of *weak mappings*: At least one bit with non-linearity 0.
2. Region of *moderate mappings*: All the bits have more than 16 as non-linearity but there is no bit with non-linearity, 24, which is the highest non-linearity for a 6 bit balanced Boolean function.
3. Region of *strong mappings*: There is no bit with non-linearity less than 16, but there is at least one bit with non-linearity 24.

In **Table 1** we show a snap-shot of the results from each of the three regions mentioned above.

## 6 Conclusion

We have presented in the paper a composition technique for small invertible non-linear CA with linear group CA to realize non-linear CA with group properties. We show that the maximum cycle lengths of such composed machines may be predicted by analyzing the linear components. We also show with extensive experimental results that choosing the group rules for the linear machine we can generate strong non-linear mappings which are also invertible.

## References

1. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55(3), 601–644 (1983)
2. Moore, C.: Predicting non-linear cellular automata quickly by decomposing them into linear ones. Physica (D) 111, 27–41 (1998)
3. Merphy, S., Blackburn, S., Paterson, K.: Comments on theory and applications of cellular automata in cryptography. IEEE Transactions on Computers 46(5), 637–638 (1997)
4. Kar, B., Nandi, S., Pal Chaudhury, P.: Theory and applications of cellular automata in cryptography. IEEE Transactions on Computers 43(12), 1346–1357 (1994)
5. Mukhopadhyay, D., RoyChowdhury, D.: Cellular automata: An ideal candidate for a block cipher. In: Ghosh, R.K., Mohanty, H. (eds.) ICDCIT 2004. LNCS, vol. 3347. Springer, Heidelberg (2004)

6. Joshi, P., Mukhopadhyay, D., Roy Chowdhury, D.: Design and Analysis of a Robust and Efficient Block Cipher Using Cellular Automata. In: Proceedings of $20^{th}$ International Conference on Advanced Information Networking and Applications, AINA, vol. 2, pp. 67–71 (2006)
7. Sarkar, P., Mitra, S.: Efficient Implementation of Cryptographically Useful "Large" Boolean Functions. IEEE Transactions on Computers 52(4), 410–416 (2003)
8. Khoo, K., Gong, G., Lee, H.-K.: The Rainbow Attack on Stream Ciphers based on Maiorana McFarland Functions (Revised Version), Cryptology ePrint Archive, Report 2007/242 (2007)
9. Pal Chaudhuri, P., Roy Chowdhury, D., Nandi, S., Chattopadhyay, S.: Additive Cellular Automata Theory and its Application, ch.4, vol. 1. IEEE Computer Socuiety Press, Los Alamitos (1997)
10. RayChaudhuri, D.: Digital Circuits, ch. 6, vol. 1, pp. 254–258 (2001)
11. Seberry, J., Zhang, X.M., Zheng, Y.: Systematic Generation of Cryptographically Robust S-boxes. In: $1^{st}$ Conference Computer and Communication Security, VA, USA, pp. 171–182 (1993)
12. Mukhopadhyay, D., RoyChowdhury, D.: Characterization of a class of complemented group cellular automata. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305. Springer, Heidelberg (2004)

# Combined Effect of Topology and Synchronism Perturbation on Cellular Automata: Preliminary Results*

Jean-Baptiste Rouquier[1] and Michel Morvan[1,2]

[1] Université de Lyon, IXXI and ENS Lyon, France
[2] EHESS and Santa Fe Institute

**Abstract.** The aim of this paper is to experimentally study the combined effect of the introduction of two kinds of structural perturbations to the behavior of cellular automata. We present the results obtained by simultaneously perturbing synchronism and topology of elementary cellular automata. We show that very interesting and different behaviors appear, including phase transitions and non monotonicity (i.e. introduction of both perturbations is less effective than the introduction of only one of them). These results lead us to think that this study is worth to be now developed more accurately.

## 1 Introduction

The last decade has seen the development of an interest to the effect of structural perturbation applied to discrete dynamical model [1,2,3,4,5]. Almost all such studies were focused on the introduction of one kind of perturbation. However, in [6], it is shown that combining two perturbations, namely the introduction of asynchronism and a topology perturbation, could lead to interesting new behaviors of the Game of Life. In particular, it has been shown that while the Game of Life was very sensitive to asynchronism, this sensitivity was strongly decreased when coupled with the breaking of a small percentage of links in the grid.

The aim of this paper is to develop an exhaustive study of the coupling of the same kind of perturbations for elementary cellular automata. We will show that very different behaviors appear. On the one hand, part of these results were expected: some ECA are robust to both perturbations, even when applied simultaneously, other are sensitive to only one, the other having no effect, and some are sensitive to both of them when they are applied simultaneously. On the other hand, some interesting and more unexpected behavior appear in this last case. For example, the combination of both perturbation can in some cases lead either to the vanishing of a phase transition induced by one perturbation alone or to the appearance of a new phase transition. Moreover, rule 45 exhibits the same behavior as the Game of Life: one perturbation makes the system robust

* This work has been partially supported by French ANR project NT05-2_4913 CARPVIRTUEL and European NEST Pathfinder project 043322 MORPHEX.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 220–227, 2008.

to the introduction to the other, and so the effects of the perturbations are not combined in a monotonous way.

The main conclusion of this exploratory paper is that the robustness to the combination of different perturbations is a subject which deserves to be studied in more details, even in very simple systems as ECA.

The paper is organized as follows. In Section 2, we describe the definitions and in particular the definition of the perturbations we will apply to ECA. We also present the experimental protocol. Section 3 contains the result and brief analysis of the experimental results. We classify the different observed behavior and highlight those that are, in our opinion, the most interesting for future studies. We end by a discussion in which we propose promising research directions.

## 2 Definitions and Experimental Protocol

### 2.1 Perturbed Cellular Automata

**Definition 1.** *A Cellular Automata (CA) with periodic boundary condition is a tuple consisting in*

- $Q$ *a set of* states,
- $\mathcal{U} := (\mathbb{Z}/n\mathbb{Z})^d$ *the finite set of* cells *($d$ is the dimension, $n$ is a parameter),*
- $V := \{\vec{v}_1, \ldots, \vec{v}_{|V|}\}$ *is the* neighborhood*, a finite set of vectors of* $\mathbb{Z}^d$*,*
- $\delta : Q^{|V|} \to Q$ *is the* update rule*.*

A *configuration* specifies the state of each cell, and is thus a function $\mathcal{U} \to Q$. The usual dynamics on a CA is the following: considering a configuration $c_t$, the next configuration $c_{t+1}$ is defined by $\forall z, c_{t+1}(z) := \delta(c_t(z+\vec{v}_1), \ldots, c_t(z+\vec{v}_{|V|}))$.

This early works focuses on a particular class of CA, namely the Elementary Cellular Automata (ECA), the simplest ones: $Q = \{0,1\},\ d = 1,\ V = \{-1,0,1\}$. We use the classical notation introduced by Wolfram: an ECA is denoted by the code $\delta(0,0,0) \times 2^0 + \delta(0,0,1) \times 2^2 + \delta(0,1,0) \times 2^2 + \cdots + \delta(1,1,1) \times 2^7$.

The fact that it is one dimensional implies that $n$ is also the number of cells.

We perturb this dynamics in two ways. First, not all cells are updated at each time step. Instead, at each time step, each cell has a fixed probability $p$ to get updated, independently of the other cells. $p$ is a parameter. The case $p = 1$ is the unperturbed (synchronous) case. Updating only one cell at a time would also be a classical model [7].

Second, a cell might not always "see" all its neighbors. Instead, at each time step, when the cell $z$ requests the states of its neighbors to compute its next state, it gets no answer with probability $r$. It then assumes the neighbor is in state $q$. $q$ and $r$ are parameters. The case $r = 0$ (no link cut) is the unperturbed case.

The final dynamics is thus:

$$c_{t+1} := \begin{cases} \delta\big(c'_t(z-1),\ c_t(z),\ c'_t(z+1)\big) & \text{with probability } p \\ c_t(z) & \text{with probability } 1-p \end{cases} \tag{1}$$

$$\text{where } c'_t(y) := \begin{cases} q & \text{with probability } r \\ c_t(y) & \text{with probability } 1-r \end{cases}$$

*Motivation of the Model* Not updating a cell can be seen as a delay before updating. It can also be seen as a defective cell, where we chose that a defective cell keeps its state. When a cell does not sees one of its neighbor, it corresponds to a defective (oriented) link between the cells.

All defects are temporary. This is necessary for ECA since otherwise a defect would prevent all communication between its left and right sides.

### 2.2 Experimental Protocol

We call *run* the temporal evolution of a CA when all parameters are chosen. The macroscopic measure we use on a run is the density of cells in state 1. This measure has been shown relevant in [8]. For a configuration $c$, we note the density

$$\rho(c) := \#\{z \in \mathcal{U} \mid c(z) = 1\} \ / \ |\mathcal{U}|$$

**Sampled Parameters.** We do one run for each possible combination of the following parameters.

For a given choice of the rule (including the parameter $q$), we sample many values of $p$ and $r$, that is, we combine many strengths of synchronism and topology perturbation. For the former, we sample the whole space of synchrony rates, including the synchronous case, by steps of 0.01: $p \in \{0.01, 0.02, \ldots 1.00\}$. For the latter, cutting 10% of the links is already a strong perturbation, and we do not perturb the model more. That is, $r \in \{0.00, 0.01, 0.02, \ldots 0.10\}$.

We sample both values (0 and 1) for the parameter $q$. For a given rule, e.g. 110, we note the parameter $q$ as a subscript, e.g. $110_0$ and $110_1$. For autoconjugate rules (i.e. rules that are unchanged under the exchange of state 0 and 1), both choices of the parameter $q$ leads to the same dynamics (actually to symmetric ones). We therefore study only one, which we note $105_a$.

When working on ECA, one usually considers only 88 rules among the 256 possible ones, thanks to symmetry considerations. The left/right symmetric of a rule $\delta$ (i.e. the rule $\delta'(a, b, c) := \delta(c, b, a)$) leads to configurations symmetric to the ones obtained with . Therefore, we will only consider one rule among each pair $(\delta, \delta')$. However, the other usual symmetry, namely exchanging the states 0 and 1, is no longer possible. Indeed, the parameter $q$ has introduced an asymmetry between both states.

**Fixed Parameters.** The following parameters are identical for all runs.

- The ring size is n=10 000 cells. [4] estimated that 2 000 was already sufficient in a similar problem.
- The initial configuration is random (each cell is in state 0 with probability 0.5, independently from the other cells) and distinct for each run.
- To speed up computation, we do not measure  at each time step, instead we do the same number of measurements (roughly 200) in each decade.
- To detect when the density has settled to a constant value, we do a linear regression on the last 200 measure points and wait for the fitted line to be almost horizontal (we actually do three independent fits on three portions of the history). We then average the density over the history.

– If the density has not stabilized after $10^6$ steps, we stop the simulation and ignore this point.

**Limits of the Protocol.** The density might be seen as a quite rough measure, and so we will certainly miss some phenomenons occuring in this model. However, the variety of results shows that this measure is able to detect one kind of sensibility to the perturbations applied here.

Another limit is that this protocol does not measure the true asymptotic density, but instead a long enough stage where the density is stable. This is the only stage observable experimentally if the asymptotic regime occur only after exponential time. Nevertheless, decision to stop a run might occur early if the density is very slowly decreasing.

Last, for each value of the parameters, we compute only one data point. The smoothness of most surfaces shows that variance among runs is low, and that we do not need to average over several data points.

## 3 Exhaustive Study of the ECA Space

We now apply this protocol to each ECA and plot $\rho$ against $p$ and $r$ (Fig. 1). Reviewing the sampling surfaces suggests the classification in four classes of Table 1, based on the sensitivity to each perturbation. The subclasses a/, b/ and c/ are explained in Section 3.4. We now review each class.

**Table 1.** Classification of the ECA from their robustness against each perturbation

| Synchronism perturbation | Topology perturbation: insensitive | Topology perturbation: sensitive |
|---|---|---|
| insensitive | $0_0$ $0_1$ $2_0$ $8_0$ $8_1$ $10_0$ $23_a$ $24_0$ $28_0$ $32_0$ $34_0$ $40_0$ $42_0$ $51_a$ $56_0$ $60_0$ $60_1$ $74_0$ $105_a$ $128_0$ $128_1$ $130_0$ $136_0$ $136_1$ $138_0$ $152_0$ $154_1$ $160_0$ $162_0$ $168_0$ $200_1$ $204_a$ | $4_1$ $5_0$ $12_1$ $13_0$ $15_a$ $29_0$ $29_1$ $32_1$ $36_1$ $40_1$ $44_1$ $72_0$ $72_1$ $76_1$ $78_1$ $104_0$ $104_1$ $132_1$ $140_1$ $160_1$ $164_1$ $168_1$ $170_a$ $172_1$ $184_0$ $184_1$ $200_0$ $232_a$ |
| sensitive | $1_0$ $4_0$ $5_1$ $6_0$ $9_0$ $11_0$ $11_1$ $12_0$ $18_0$ $19_0$ $19_1$ $22_0$ $26_0$ $33_0$ $35_0$ $35_1$ $36_0$ $38_0$ $41_0$ $41_1$ $44_0$ $46_0$ $50_0$ $54_0$ $58_0$ $76_0$ $90_1$ $108_0$ $110_0$ $110_1$ $132_0$ $134_0$ $146_0$ $164_0$ | a/ $2_1$ $6_1$ $10_1$ $24_1$ $34_1$ $38_1$ $42_1$ $45_0$ $45_1$ $46_1$ $56_1$ $73_0$ $73_1$ $74_1$ $77_a$ $94_0$ $94_1$ $130_1$ $134_1$ $138_1$ $152_1$ $154_0$ $162_1$<br>b/ $18_1$ $26_1$ $50_1$ $58_1$ $106_0$ $106_1$ $146_1$ $178_a$<br>c/ $1_1$ $3_0$ $3_1$ $7_0$ $7_1$ $9_1$ $13_1$ $14_0$ $14_1$ $22_1$ $25_0$ $25_1$ $27_0$ $27_1$ $28_1$ $30_0$ $30_1$ $33_1$ $37_0$ $37_1$ $43_a$ $54_1$ $57_0$ $57_1$ $62_0$ $62_1$ $78_0$ $90_0$ $108_1$ $122_0$ $122_1$ $126_0$ $126_1$ $140_0$ $142_a$ $150_a$ $156_0$ $156_1$ $172_0$ |

### 3.1 Rules Insensitive to Both Perturbations

Most of the rules in this class have a surface that is just 0 everywhere (Fig 1.a), sometimes with an exception in the unperturbed case ($p = 1$, $r = 0$). A few

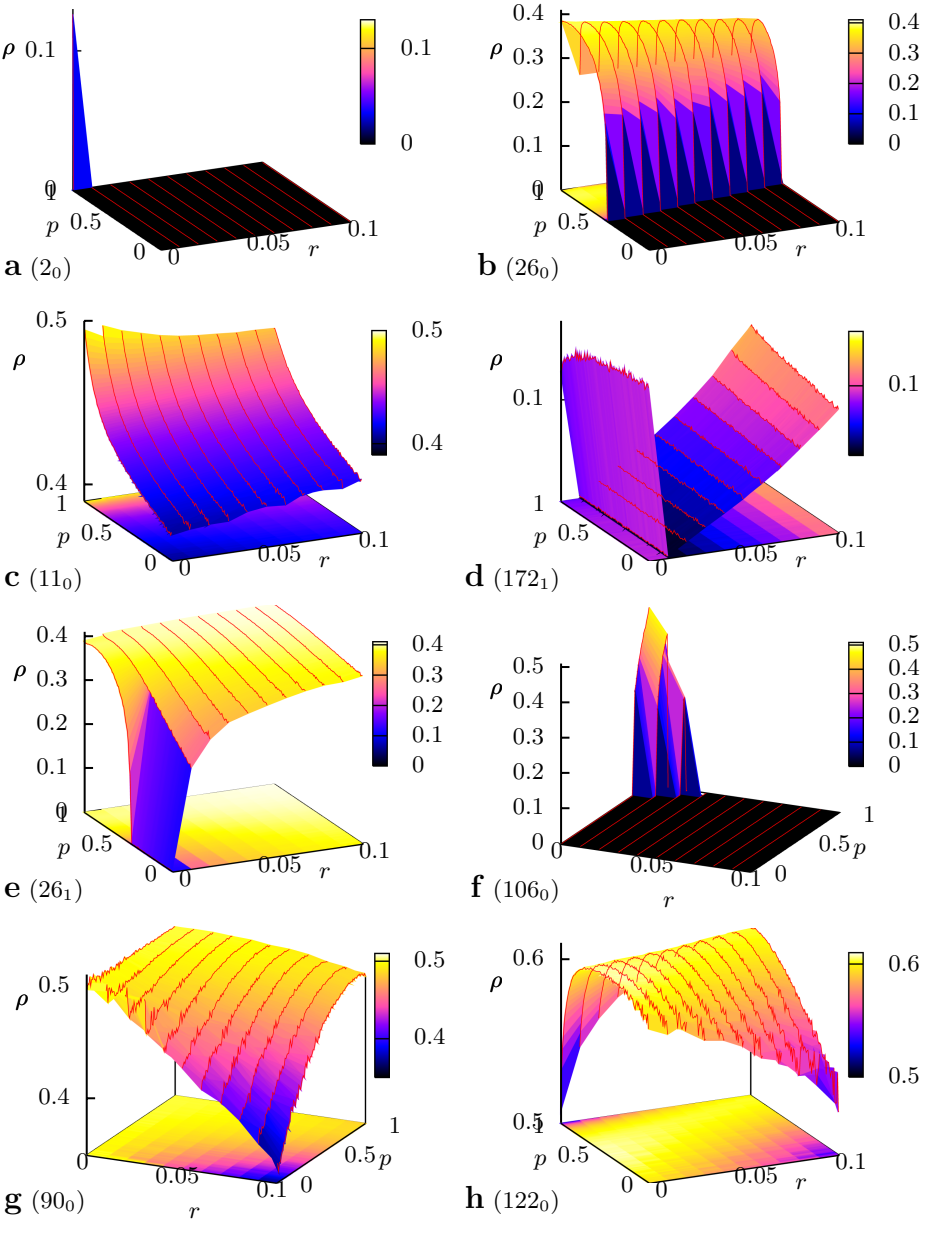



**Fig. 1.** Some sampling surfaces (color online). Axis $x$ (width) represents topology perturbation, $y$ (depth) synchronism perturbation, $z$ (height) asymptotic density. The difference between Figures 1.a and 1.f is that in the former only the unperturbed case ($p = 1, r = 0$) is non zero, whereas in the latter there is a continuous transition from non-zero to zero, with respect to both $p$ and $r$.

rules (9%) outside this class also have a noticeably different behavior in the unperturbed case, but this is left for further study.

Since they are by definition insensitive to the perturbations, the other rules of this class also have a horizontal surface, (the altitude is often around $= 0.5$).

### 3.2 Rules Sensitive Only to Synchronism Perturbation

Some rules of this class undergo a phase transition when $p$ changes (Fig 1.b). The conditions of a conjecture by [9] are fulfilled, so this transition should belong to the universality class of directed percolation. (For an introduction to directed percolation in CA see [3], and for a detailed review of phase transitions see [10].)

It is interesting to note that the topology perturbation slightly shifts the threshold (the position of the transition) of most of those rules.

The remaining surfaces of this class are smooth, for instance like Fig 1.c. Since these rules are almost insensitive to topology perturbation, all slices along a red line (grey for a B&W version) are nearly identical.

### 3.3 Rules Sensitive Only to Topology Perturbation

Some rules of this class ($4_1$, $5_0$, $12_1$, $13_0$, $36_1$, $44_1$, $72_0$, $72_1$, $76_1$, $78_1$, $104_0$, $104_1$, $132_1$, $140_1$, $164_1$, $170_a$, $172_1$, $184_0$, $184_1$, $200_0$, $232_a$) are sensitive to the introduction of topology perturbation, i.e., they have a brutal change of behavior between $r = 0$ and $r > 0$ (Fig 1.d). A few rules actually have a density that depends on $p$, but only when $r = 0$.

The other surfaces of this class are smooth. The only phase transitions of this class are thus first order phase transitions (i.e. discontinuity of $\rho$),and they are all located at the introduction of topology perturbation.

### 3.4 Rules Sensitive to Both Perturbations

This is the most interesting class, as we will see many different behaviors. We examine specifically how both perturbations interact.

**a/ Sensitive to Introduction of Perturbation.** As in the class 3.3, the surfaces of this subclass exhibit a gap between $r = 0$ and $r > 0$. This is a first order phase transition, and it is not the only phase transition of class 3.4.

**b/ Created or Destroyed Phase Transition.** We have seen in section 3.2 that one of the two ways for the density to vary when $p$ changes is a phase transition. For the ECA of Section 3.2, this phase transition was robust against topology perturbation. Here it is not the case since the phase transition is destroyed when $r > 0$ (Fig 1.e). We looked at the configurations (not reproduced here due to lack of space) and it seems that cutting a link in a region of cells in state 0 (i.e. in the absorbing state) corresponds to reseeding an active site at this point. The configuration is thus constantly reseeded with active sites and $0^n$ is no more an absorbing state. This is the case for $18_1$, $26_1$, $50_1$, $58_1$, $106_1$, $146_1$.

The case of $178_a$ is the opposite: when the topology is not perturbed, this rule is autoconjugate and can't have a preferred state. In [11], without topology perturbation, it is suggested that this rule could have a phase transition of the $\mathbb{Z}_2$ symmetric percolation universality class when $p$ varies. The slightest topology perturbation introduces a bias towards $0^n$, which becomes the only absorbing state. The rule then satisfies the conditions of the conjecture by [9], and should thus belong to the universality class of directed percolation. Therefore, we conclude that topology perturbation has created a new phase transition.

**c/ Smooth.** The last subclass contains the remaining surfaces, which are smooth.

**Some Remarkable Rules.** We now would like to point out some remarkable rules of the present class (Section 3.4).

Rule $106_0$ (Fig 1.f) exhibits two phase transitions, one when $p$ varies and the other when $r$ varies.

Rule $90_0$ (Fig 1.g) is almost insensitive to a single perturbation. However, both perturbations combined induce a noticeable decrease of the density. (This study focuses on small $r$, but it is not hard to see that for high enough $r$, rule $90_0$ has a vanishing density.)

Rule $122_0$ (Fig 1.h) has a density close to 0.5 in the unperturbed case. When applying any one of the perturbations, the density tends to 0.6. But when applying both, the density tends to back 0.5. (However, the configurations are quite different from the unperturbed case, only the density has been restored.) One perturbation can be seen as a way to make the CA robust against the other perturbation. This is one of the rare cases where perturbations are not combining in a monotonous way.

Rule 73 (which has the same qualitative behavior for both values of $q$) has a constant asymptotic density when topology is not perturbed, for all values of $p$ (except $p = 1$). With a small perturbation of the topology, spans the range 0.3 to 0.5. The interesting effect is that introducing topology perturbation has made the rule sensitive to synchronism perturbation. Same remark applies to rules $14_0$, $14_1$, $57_0$, $90_0$, $150_a$ $156_0$ $156_1$ and to almost all rules of subclass a/.

The opposite behavior also exists: 45, with no topology perturbation, has a density that depends on $p$. With $r > 0$, the density becomes constant for all $p$: perturbing the topology makes the rule robust against synchronism perturbation.

## 4 Discussion and Perspectives

By means of *dynamic* perturbation, we studied topology perturbation on ECA and its interplay with synchronism perturbation. Topology perturbation is a perturbation on the neighboring cells, while synchronism perturbation is a perturbation on the central cell. The macroscopic parameter chosen, namely the density, already allows one to see a rich set of behaviors, even if there may be more than what this measure reveals. This is both a confirmation of the relevancy of the density and a confirmation that this model could be studied further, possibly with analytical results.

There does not seem to be any relation between this classification and classical classifications about synchronous deterministic CA.

The topology perturbation introduced in this paper should be studied for itself, possibly with the following generalization of the parameter $q$. When a cell does not know the state of its neighbor, it currently assumes it is in state $q$. It could instead decide its state according to a Bernoulli law. Future work will study the combined effect of $r$ and this new parameter $q$.

A few surfaces contains the characteristic curve of second order phase transition. Future work should check if they belong to the universality class of directed percolation, according to the aforementioned conjecture.

## References


1. Fatès, N., Morvan, M.: An experimental study of robustness to asynchronism for elementary cellular automata. Complex Systems 16, 1–27 (2005)
2. Ingerson, T.E., Buvel, R.L.: Structure in asynchronous cellular automata. Physica D Nonlinear Phenomena 10, 59–68 (1984)
3. Fatès, N.: Asynchronism induces second order phase transitions in elementary cellular automata. Journal of Cellular Automata (March 2007)
4. Rouquier, J.B., Morvan, M.: Coalescing cellular automata: Synchronization by common random source for asynchronous updating. Journal of Cellular Automata (accepted, 2008)
5. Schönfisch, B., de Roos, A.: Synchronous and asynchronous updating in cellular automata. Biosystems 51(3), 123–143 (1999)
6. Fatès, N., Morvan, M.: Perturbing the topology of the game of life increases its robustness to asynchrony. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 111–120. Springer, Heidelberg (2004)
7. Fatès, N., Morvan, M., Schabanel, N., Thierry, E.: Fully asynchronous behavior of double-quiescent elementary cellular automata. Theoretical Computer Science 362, 1–16 (2006)
8. Fatès, N.: Experimental study of elementary cellular automata dynamics using the density parameter. In: Discrete models for complex systems, DMCS 2003 (Lyon). Discrete Mathematics Theoretical Computer Science Proceedings, AB, Nancy. Assoc. Discrete Math. Theor. Comput. Sci, pp. 155–165 (2003)
9. Grassberger, P.: Are damage spreading transitions generically in the universality class of directed percolation? J. Stat. Phys. 79, 13–23 (1995)
10. Hinrichsen, H.: Nonequilibrium critical phenomena and phase transitions into absorbing states. Advances in Physics 7, 815–958 (2000)
11. Fatès, N.: Robustesse de la dynamique des systèmes discrets: le cas de l'asynchronisme dans les automates cellulaires. PhD thesis, ENS Lyon (December 2004)

# Finite Size Stability Analysis for Stochastic Cellular Automata

Yukio Sakisaka[1], Yukio Iwamura[1], Nariyuki Nakagiri[2], Jin Yoshimura[3], and Kei-ichi Tainaka[3]

[1] Ibaraki Prefectural University of Health Sciences, Ami 300-0394, Japan
[2] University of Hyogo, Himeji 670-0092, Japan
[3] Shizuoka University, Hamamatsu 432-8561, Japan

**Abstract.** Real simulations are performed on a finite size of lattice. It is therefore very difficult to predict a phase diagram on an infinitely large lattice. Here, we present a Finite Size Stability Analysis (FSSA) to know whether the phase is sustainable or not. Although this analysis is a hypothesis, it enables us to determine the boundary of phase diagram. We apply FSSA to multi-state system. For example we study ten-species system in ecology. From computer simulations on various sizes of lattices, we obtain the waiting time $\tau$ to extinction. The system is found to have two phases: the coexistence of all species is either unstable or marginally (neutrally) stable. In the latter case, $\tau$ diverges on a power law with the increase of lattice size.

## 1 Introduction

In recent decades, stochastic cellular automata have been extensively studied in various fields [1,2]. The present paper focuses on the problem "how to determine the phase diagram?" When each site takes one of many states, then there are many phases in stationary state (equilibrium). However, it is not easy to know what kinds of states coexist in each phase. When a lattice size is small, annihilation of almost all states always occurs. To determine the phase diagram is one of important problems. In this paper, we present a "Finite Size Stability Analysis" (FSSA). From the result of real simulations, FSSA predicts whether the coexistence of several states is sustainable or not on an infinitely large lattice.

For example, we consider two-state model: each lattice site takes one of two states A or B. There are three possible phases in the final equilibrium; namely, the survival phase of A (phase A), the survival phase of B (phase B) and the coexistence of both states (phase A & B). In the case of contact process, the phase boundaries can be determined by mathematics, and the dynamics is known for various sizes of lattices [3,4].

The contact process has been introduced by Harris [5] as a simple model for the spread of infection. It has been extensively studied in various fields, such as mathematics [3,6,7,8,9], physics [10,11,12] and ecology [13,14]. Consider a single species on a $d$-dimensional regular lattice $\mathbf{Z}^d$; each lattice site takes one of two

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 228–235, 2008.

states: either empty ($E$) or occupied ($X$) by an individual. Birth and death processes are respectively defined by

$$X + E \longrightarrow 2X \quad (\text{rate: } r) \tag{1a}$$

and

$$X \longrightarrow E \quad (\text{rate: } m), \tag{1b}$$

where the parameters $r$ and $m$ represent the reproduction and mortality rates of an individual, respectively ($r > 0$, $m > 0$). So far, many researchers have mathematically studied the contact process, and reported the following properties [6,7,9,10,12,15,16]:

1) When the lattice size is infinite, the system dynamics goes into one of two equilibrium phases; namely the species X survives (phase $X$ & $E$) or goes extinct (phase $E$).
2) When the lattice size is finite, the annihilation always occurs. The waiting time $\tau$ to annihilation increases with the increase of the lattice size ($L$) as shown in Fig. 1. In the case of phase $X$ & $E$, the value $\tau$ diverges on an exponential scale with the increase of $L$. In the case of phase $E$, the value $\tau$ diverges on an log scale with the increase of $L$.
3) If the system locates on a phase boundary, a power law ($\tau \propto L^{\alpha}$) holds, where the exponent $\alpha$ is a positive constant.

We apply this idea to multiple-state model. Here the waiting time $\tau$ is defined by the period until the first annihilation of a state occurs.

## 2 Model

We study multiple contact process on a finite size ($L$) of a 2-dimensional lattice [17]. Each lattice site is empty ($E$) or occupied by species $i$ ($X_i$), where $i = 1, \ldots, N$. Overall reactions are defined as follows:

$$X_i + E \longrightarrow 2X_i \quad (\text{rate: } r) \tag{2a}$$

and

$$X_i \longrightarrow E \quad (\text{rate: } m). \tag{2b}$$

Here the parameters $r$ and $m$ are assumed to take the same value for any species, respectively. The model of (2a) and (2b) was first presented by Miyazaki et al [18] to explain the paradox of plankton. Many plankton species can coexist in a pond or a lake, although they compete with each other. This contradiction is the orgin of "paradox of plankton" [19]. The simulation is carried out in two different ways: global and local interactions. We first describe the simulation method for the local interaction version of lattice model:

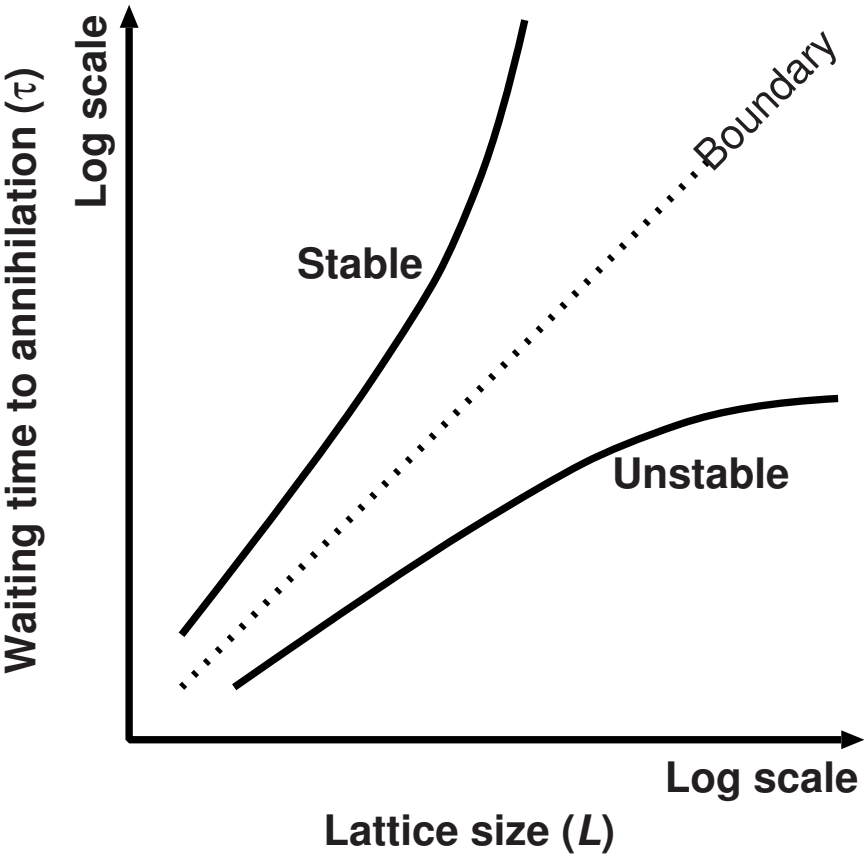



**Fig. 1.** A schematic diagram of finite size stability analysis (FSSA). We obtain the waiting time $\tau$ until a state annihilate for various values of lattice size $L$. 1) In the unstable phase of coexistence, $\tau$ diverges on a log-log scale with the increase of $L$. 2) In the stable phase, $\tau$ exponentially diverges with $L$. 3) The boundary case between unstable and stable is represented by straight dotted line on a log-log scale (i.e. power law).

1) Initially, we distribute individuals on a lattice.
2) Reactions (2a) and (2b) are performed in the following two steps:
   (i) We perform two-body reaction (2a). Choose one lattice site randomly, and then randomly specify one of $2d$ neighboring sites. If both chosen sites are $X_i$ and $E$, then $E$ is changed to $X_i$ with probability $r$.
   (ii) We perform one-body reaction (2b). Choose one lattice point randomly; if the site is occupied by $X_i$, the site will become $E$ with probability $m$.
3) Repeat step 2) $L^2$ times, where $L^2$ is the total number of a square lattice site. This total number is called the Monte Carlo step (MCS) [20].
4) Repeat step 3) until the first species goes extinct.

Next, we describe the method of global interaction in which the interaction occurs between any pair of lattice sites. The global simulation is very similar to local case, but step (i) in 2) is replaced as follows:

(i′) A pair of lattice sites is chosen not only randomly but also independently.

Under global interaction, the population dynamics of the system of (2a) and (2b) is given by the mean-field theory. Let $x_i$ be the overall density of species $i$, then we have the following dynamics:

$$\frac{dx_i}{dt} = -mx_i + rx_ie, \tag{3}$$

where $e$ is the density of empty site ($e = 1 - \sum_i x_i$). The first and second terms on the right hand side of (3) comes from death and birth processes, respectively. By summing up (3) with respect to $i$, we have the following logistic equation:

$$\frac{dS}{dt} = RS(1 - S/K), \tag{4}$$

where $S$ is the total density ($\sum_i x_i$) and $K = 1 - m/r$. Thus, when $m/r < 1$, then $S$ is stable; namely, $S \to K$ for $t \to \infty$. When $m/r \geq 1$, all species go extinct ($S \to 0$ for $t \to \infty$).

## 3 Simulation Results

We assume that our system contains ten kinds of species ($N = 10$) on a finite size of square lattice. To obtain the waiting time to extinction, we set the simulation condition as follows.

1) The values of reproduction rates for both global and local interactions are such chosen that the stationary densities take the same value.
2) Initially, all species have the identical density which is just the same as the steady-state density.

For various sizes of lattices, we obtain a waiting time ($\tau$) until the first species goes extinct. A typical result is illustrated in Fig. 2, where the vertical axis denotes the waiting time ($\tau$), and the horizontal axis means the lattice size ($L$). In Fig. 3, we can easily know the approximation lines of both global and local simulations on a log-log scale. All statistical analyses for Figs. 2 and 3 were conducted in R Development Core Team [21]. From Figs. 2 and 3, we find the following power law:

$$\tau \propto L^{\alpha}. \tag{5}$$

The value of exponent takes $\alpha = 2.1$ for global and $\alpha = 2.6$ for local interactions. Hence, simulations for both methods exhibit the similar critical behavior. Equation (5) always holds for any values of $r$ or $m$, so long as the condition $m/r < c$ is satisfied. Here $c$ is the critical value; $c = 1$ for global and $c \approx 0.625$

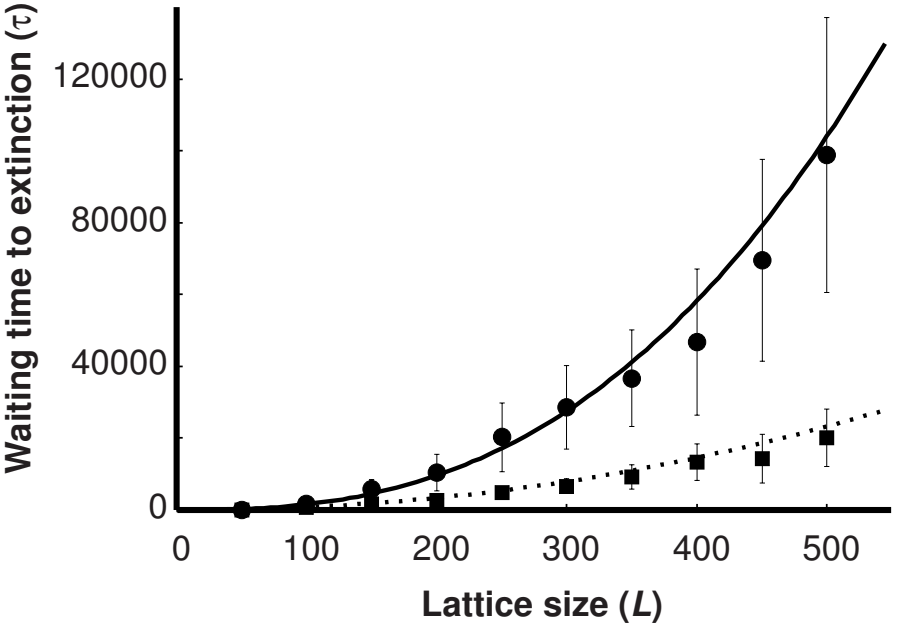



**Fig. 2.** Results of multiple contact process simulations ($m = 0.3$). All reproduction rates ($r$) and mortality rates ($m$) of ten species take the same value, respectively ($N = 10$). The waiting time ($\tau$) until the first species becomes extinct is plotted against the lattice size ($L$). The plots of filled circles represent the results of local interaction, where $r$ takes values of 0.9. The plots of filled squares represent the results of global interaction, where $r$ takes values of 0.786. Bars indicate standard deviation (SD) of each 20 simulation runs. Almost all plots are well approximated by $\tau = cL^{\alpha}$, where we have $(c, \alpha) = (0.010, 2.6)$ for local interaction (solid curve) and (0.050, 2.1) for global interaction (broken curve).

for local interactions [12]. In contrast, when $m/r \geq c$, all species go extinct. Note that all species coexist in neutrally stable phase, but there is no stable phase.

Previous works [4] for multiple contact process reported the following outcomes: when the lattice size is infinitely large ($L \to \infty$), the equilibrium phases were well known. In the case of global interaction, coexistence of all species is possible. However, in local interaction, coexistence of plural number of species is impossible for any condition. Only one species can survive at most. After the extinctions, the system is eventually equivalent to the contact process (single-species or two-state system). However, Fig. 2 and Fig. 3 indicates peculiar results never predicted from the conclusion for equilibrium analyses:

(i) Although extinction is unavoidable for local interaction, the critical behavior is observed.
(ii) The exponent $\alpha$ for local interaction takes a lager value than that for global interaction.

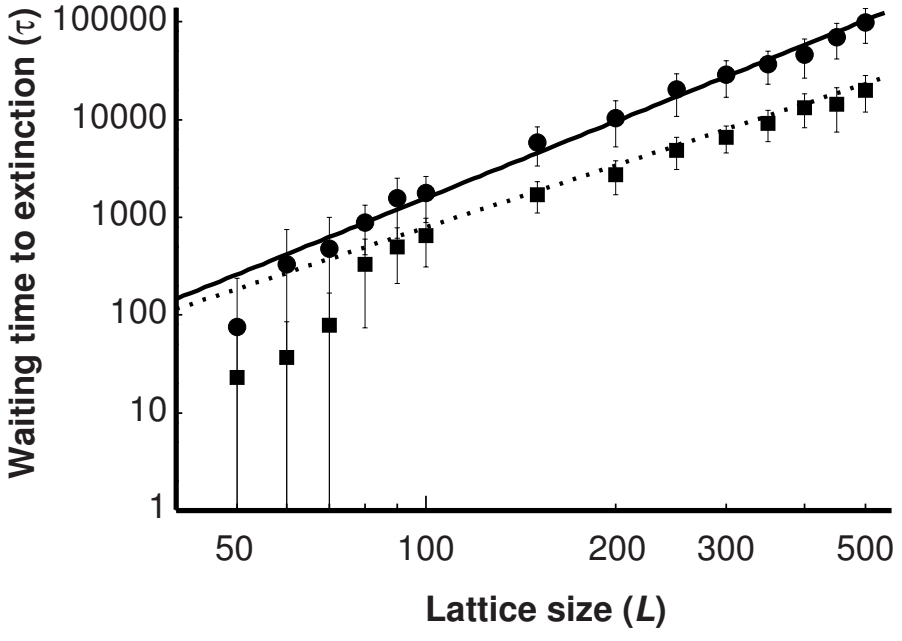



**Fig. 3.** The same data of Fig. 2 plotted on a log-log sacale. Both vertical and horizontal axises are correspond to Fig. 1's. The function $\tau = cL^{\alpha}$ forms a line on this scale.

It is therefore found that species survive for a very long time, so long as the local interaction is applied.

## 4 Conclusions and Discussions

We present a new method called finite size stability analysis (FSSA). By the use of FSSA, we can determine whether multiple states is stable or not in each phase. We obtain the waiting time ($\tau$) until the first state annihilates against the lattice size ($L$) as shown in Fig. 1. If the power law (5) holds, then the coexistence is neutrally stable (boundary between different phases).

An example is an ecological model of (2a) and (2b) to explain the coexistence of plankton species. In this system, there are eleven states: each lattice site takes empty ($E$) or individual ($X_i$) of species $i$. This system evolves into one of two phases: extinct or survival phases. When $m/r \geq c$ (extinct phase), all lattice sites become empty ($E$). Here $c = 1$ for global and $c \approx 0.625$ for local interactions. In contrast, when $m/r < c$, the coexistence of all species becomes neutrally stable: (5) always holds for any values of $r$ or $m$. From the mean-field theory, readers may have the following question: why the coexistence is neutrally stable in spite of logistic equation (4). The answer is simple. Equation (4) predicts the stability of the total density $S$, but the density of each species is neutrally stable [17].

We here suggest the origin of the power law (5) for local interaction. This law can be observed for other values of parameters, as long as all species have the same reproduction rates and same mortality rates, respectively. Moreover, our results never change for different number of species. We carry out simulations for 9-, 10- and 11-species systems. When species number increases, the waiting time ($\tau$) decreases. However, the value of exponent never changes.

In the two-species contact process, Neuhauser [4] indicated that the problem of coexistence or competitive exclusion can be replaced by that of collision of random walking particles. Random walking and extinction processes of many particles on a lattice space are studied extensively in the field of diffusion-limited reactions. According to these studies, the number of particles decreases proportionally to $t^{-\beta}$ in most cases ($\beta$ is a positive constant) [22,23,24]. By assuming that extinction occurs at about $1/L$ particles, the extinction time $\tau$ satisfies the relation $1/L \sim \tau^{-\beta}$. Therefore, we can expect the power law $\tau \sim L^{\alpha}$. Note that it holds for large values of $L$.

## References


1. Kawai, T., Tadokoro, Y., Tainaka, K., Hayashi, T., Yoshimura, J.: A lattice model of fashion propagation with correlation analysis. Int. J. Systems Science (in press, 2008)
2. Wolfram, S.: Computation theory of cellular automata. Commun. Math. Phys. 96, 15–57 (1984)
3. Liggett, T.M.: Interacting particle systems. Springer, Berlin (1985)
4. Neuhauser, C.: Ergodic theorems for the multitype contact process. Probab. Theor. Relat. Field. 91, 467–506 (1992)
5. Harris, T.E.: Contact interactions on a lattice. Ann. probab. 2, 969–988 (1974)
6. Durrett, R., Liu, X.-F.: The contact process on a finite set. Ann. Probab. 16, 1158–1173 (1988)
7. Durrett, R., Schonmann, R.H.: The contact process on a finite set II. Ann. Probab. 16, 1570–1583 (1988)
8. Liggett, T.M.: Stochastic interacting systems: contact, voter and exclusion processes. Springer, Berlin (1999)
9. Sudbury, A.W.: Rigorous lower bounds for the critical birth-rate in the diffusive contact process. J. Appl. Probab. 38, 1074–1078 (2001)
10. Jensen, I., Dickman, R.: Time-dependent perturbation theory for diffusive nonequilibrium lattice models. J. Phys. A: Math. Gen. 26, L151–L157 (1993)
11. Katori, M., Konno, N.: Upper bounds for survival probability of the contact process. J. Stat. Phys. 63, 115–130 (1991)
12. Marro, J., Dickman, R.: Nonequilibrium phase transitions in lattice models. Cambridge University Press, Cambridge (1999)
13. Kubo, T., Iwasa, Y., Furumoto, N.: Forest spatial dynamics with gap expansions: total gap area and gap size distribution. J. Theor. Biol. 180, 229–246 (1996)
14. Yoshimura, J., Tainaka, K., Suzuki, T., Sakisaka, Y., Nakagiri, N., Togashi, T., Miyazaki, T.: The role of rare species in the community stability of a model ecosystem. Evol. Ecol. Res. 8, 629–642 (2006)
15. Durrett, R., Schonmann, R.H., Tanaka, N.: The contact process on a finite set III. The critical case. Ann. Probab. 17, 1303–1321 (1989)

16. Liggett, T.M.: Improved upper bounds for the contact process critical value. Ann. Probab. 23, 697–723 (1995)
17. Sakisaka, Y., Tainaka, K., Sugimine, N., Yoshimura, J., Hayashi, T., Aihara, K., Togashi, T., Miyazaki, T.: Power law for extinction process in multiple contact process. J. Phys. Soc. Jpn. 76, 023101–1–4 (2007)
18. Miyazaki, T., Tainaka, K., Togashi, T., Suzuki, T., Yoshimura, J.: Spatial coexistence of phytoplankton species in ecological timescale. Popul. Ecol. 48, 107–112 (2006)
19. Hutchinson, G.E.: The Paradox of the Plankton. Am. Nat. 95, 137–145 (1961)
20. Tainaka, K.: Lattice model for the Lotka-Volterra system. J. Phys. Soc. Jpn. 57, 2588–2590 (1988)
21. R Development Core Team: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2005)
22. Kanno, S., Tainaka, K.: Power-Law spectra in diffusion-limited reaction A+B→0 with source. J. Phys. Soc. Jpn. 62, 2275–2278 (1993)
23. Lythe, G.: Diffusion-limited reaction in one dimension. Physica D 222, 159–163 (2006)
24. Toussaint, D., Wilczek, F.: Particle-antiparticle annihilation in diffusive motion. J. Chem. Phys. 78, 2642–2647 (1983)

# On the Addition of Recurrent Configurations of the Sandpile-Model

Matthias Schulz

University of Karlsruhe, Department for Computer Sciences
Am Fasanengarten 5, 76128 Karlsruhe, Germany
schulz@ira.uka.de

**Abstract.** The sandpile model, introduced by Bak, Tang and Wiesenfeld in 1987, is the standard example for a dynamic model showing Self-Organized Criticality (SOC). Also, it has many nice algebraic properties; for example, there is a set of configurations which is a group with a certain naturally defined addition.

We look at elements $c, d$ of this group and try to find out how long it takes to naively compute the sum $c \oplus d$. While we can easily give an upper bound, it is harder to find a lower bound. We prove some facts about the number of topplings (elementary operations) that have to be performed during the addition of two elements of the group and give a heuristic for quickly finding local minima.

## 1 Introduction

The sandpile model was introduced by Bak, Tang and Wiesenfeld in 1987 [1] as a model to explain $\frac{1}{f}$ noise. Grains of sand fall onto a grid, and if four or more grains are lying upon a site in the grid, one grain of sand falls to the left, right, top and bottom respectively. It has been found that the model displays Self-Organized Criticality (SOC), which means that from some time on a critical state is maintained if grains keep on falling randomly onto the grid.

Dhar and others [3] found many interesting algebraic properties of the sandpile model, especially the set of recurrent configurations (configurations which can occur infinitely often in the process described above). One of the most interesting findings was the fact that the set of recurrent configurations, together with an addition, is an Abelian group.

In this paper, we consider the question how many topplings there will be at least if we add two recurrent configurations. While it is hard to find the global optimum, we introduce a probabilistic algorithm which gives us at least local minima whose quality depends on the strategy used at one point.

## 2 Basics

Consider the grid $Z = \{0, \ldots n-1\} \times \{0, \ldots n-1\}$. For each $z \in Z$, let $N(z)$ be the set of sites in the von-Neumann-neighborhood of $z$.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 236–243, 2008.

From a configuration $c : Z \to \mathbb{N}_0$ and a site $z \in Z$ which satisfies $c(z) > 3$, we get the successive configuration $c_z$ defined as

$c_z = c - 4e_z + \sum_{z' \in N(z)} e_{z'}$,

where $\forall z \in Z : e_z(z') = \begin{cases} 1 & \text{if } z' = z \\ 0 & \text{otherwise} \end{cases}$.

Figuratively, if a site contains 4 or more grains of sand, 4 grains fall off this site and onto the sites in the neighborhood or off the grid, when the site was at the border of the grid. We say that the site *toppled* and a *toppling* occurred.

If all sites containing at least 4 grains topple at the same time and we bound the number of grains initially in $c$, we get the rule for a cellular automaton.

It has been shown that the process of letting sites with at least 4 grains topple eventually ends and leads to a configuration where each site contains at most 3 grains of sand (cf. for example [2]). Also, the order in which the sites topple is irrelevant, as all possible sequences lead to the same configuration, and the number of times a site $z$ topples during the process is also independent of the order of the topplings. This process is called a *relaxation* and the resulting configuration when starting from configuration $c$ will be denoted by $c_{rel}$.

Let $c$ be a configuration on $Z$. For all $z \in Z$, let $f_c(z)$ be the number of topplings of $z$ during the relaxation of $c$. Further, let $B \in \mathbb{Z}^{Z \times Z}$ be the matrix with

$$B(i,j) = \begin{cases} 4 & \text{if } i = j \\ -1 & \text{if } i \in N(j) \\ 0 & \text{otherwise.} \end{cases}$$

Then $c_{rel} = c - B \cdot f_c$.

Further, $B$ is invertible; both is shown in [2].

Let $\mathcal{C}$ be the set of configurations $c : \mathbb{N}_0 \to Z$ with $c_{rel} = c$. (This means: $\forall z \in Z : 0 \leq c(z) \leq 3$.) The elements of $\mathcal{C}$ are called *stable configurations*.

We define the operation $\oplus$ on $\mathcal{C}$ by $c \oplus d = (c + d)_{rel}$. This operation is associative and commutative, as shown in [3].

We define the maximal stable configuration $m$ satisfying $\forall z \in Z : m(z) = 3$ and the set of configurations $\mathcal{R} = m \oplus \mathcal{C}$. The elements of $\mathcal{R}$ are called *recurrent configurations*.

Note that for each configuration $c \in \mathcal{C}$ and for each configuration $d \in \mathcal{R}$, there exists a configuration $e \in \mathcal{C}$ such that $c \oplus e = d$:

From the definition of $\mathcal{R}$, we know that there is a configuration $e' \in \mathcal{C}$ such that $m \oplus e' = d$, and therefore $c \oplus ((m - c) \oplus e') = (c \oplus (m - c)) \oplus e' = m \oplus e' = d$.

Also, if $c$ is a recurrent configuration, then for all $d \in \mathcal{C}$ $c \oplus d \in \mathcal{R}$, too.

Therefore, if we have a Markov chain $(\mathcal{C}, P)$ with the states being stable configurations and transition probability matrix satisfying $P(c,d) = 0$ if there exists no $z \in Z$ such that $d = c \oplus e_z$, $\mathcal{R}$ is the set of recurrent configurations of this Markov chain.

A very interesting fact about $\mathcal{R}$ is that $(\mathcal{R}, \oplus)$ is an Abelian group, which is proven in [3].

By $id \in \mathcal{R}$ we denote the identity element of the group $(\mathcal{R}, \oplus)$.

Let $b \in \mathcal{C}$ be the configuration which satisfies

$\forall z \in Z : b(z) = 4 - |N(z)|$.

(So, $b(z) = 2$ if $z$ is a site in corner of $Z$, $b(z) = 1$ if $z$ is a site on the border of $Z$, and $b(z) = 0$ otherwise.)

It has been shown in [4] that

$c \in \mathcal{R} \iff c \oplus b = c$ and

$c \in \mathcal{R} \iff f_{c+b} = \mathbf{1}$,

where $\forall z \in Z : \mathbf{1}(z) = 1$.

## 3 The Problem

Consider two recurrent configurations $c$ and $d$. A naive way to compute $c \oplus d$ would be to compute the sum $c + d$ and to relax this configuration. This means $\mathbf{1}^\top f_{c+d}$ topplings have to be done.

It is quite easy to see that the worst case, i.e. the case for which $|f_{c+d}| = \mathbf{1}^\top f_{c+d}$ is maximal, is when $c = d = m$; it can be shown that $|f_{m+m}| \in O(n^4)$ holds.

On the other hand, it is much harder to find a tight lower bound for $|f_{c+d}|$ when $c, d \in \mathcal{R}$, or to find configurations $c, d \in \mathcal{R}$ for which $|f_{c+d}|$ is minimal. While we can find a configuration $c$ such that $m - c \in \mathcal{R}$ for $n \leq 4$, there is a very obvious reason why there cannot be such a configuration $c$ for $n > 4$: In this case the configuration $c + d$ contains more than $3n^2$ grains of sand and the surplus grains have to fall off the grid, which means that sites on the border of $Z$ must topple.

In the following sections, we will show that each recurrent configuration contains at least $2n^2 - 2n$ grains of sand, which means that during the relaxation of the sum $c + d$ of two recurrent configurations $c, d$ at least $n^2 - 4n$ grains of sand must fall off the grid.

For $n = 4$ we will give recurrent configurations $c, d \in \mathcal{R}$, such that $c + d = m$.

We will show how to find for $c \in \mathcal{R}$ a configuration $\bar{c} \in \mathcal{R}$ such that $|f_{c+\bar{c}}|$ is minimal.

We will show how we can get from a recurrent configuration $c$ to a recurrent configuration $c'$ such that $f_{c'+(\bar{c'})} \leq f_{c+\bar{c}}$ holds. ($\leq$ here means component-wise less or equal.) By repeating this process, we reach a local minimum.

## 4 Number of Grains in Recurrent Configurations

Let $c$ be recurrent configuration. All grains that remain on a site $z$ after it toppled during the relaxation of $c + b$ can be taken away without the resulting configuration $c'$ becoming non-recurrent, since the same sequence of topplings during the relaxation of $c + b$ is a possible toppling sequence for $c' + b$: If a site $z$ contained more than four grains of sand at the moment it toppled, then the "surplus" grains are the ones that were taken away to get $c'$. Therefore, during the relaxation of $c' + b$ each site contains exactly four grains of sand at the moment it topples if we use the same sequence of topplings as for $c + b$.

We call a recurrent configuration $c$ *minimal recurrent*, if $\forall z \in Z : c - e_z \notin \mathcal{R}$. This means that no grain of sand can be taken from $c$ without getting a non-recurrent configuration; it follows that each site contains exactly four grains of sand at the moment it fires.

By counting the grains of sand each site contains just before it fires during the relaxation of $c+b$, we obviously get $4n^2$ grains. Since we counted each grain of sand that still is in $(c+b)_{rel}$ twice and each grain of sand that got lost at the edge once, we get

$4n^2 = 2|c| + |b|$

and therefore, since $|b| = 4n$,

$|c| = 2n^2 - 2n$ .

This means that each recurrent configuration contains at least $2n^2 - 2n$ grains of sand and that the sum of two recurrent configurations $c$ and $d$ contains at least $4n^2 - 4n$ grains of sand. From this we get $|f_{c+d}| \geq \frac{n^2-4n}{2}$, since a toppling can lead to the loss of no more than 2 grains of sand (except for $n = 1$, in which case $\frac{n^2-4n}{2} < 0$ anyways).

## 5 Examples

It is easy to see that the highest $n$ for which we can hope to find $c, d \in \mathcal{R}$ such that $|f_{c+d}| = 0$ (and therefore obviously minimal) is 4. We here give two configurations $c, d \in \mathcal{R}$ which satisfy $c + d \leq m$.

$$c = \begin{pmatrix} 2\,1\,3\,1 \\ 1\,2\,2\,1 \\ 2\,1\,1\,2 \\ 2\,0\,2\,1 \end{pmatrix}, d = \begin{pmatrix} 1\,2\,0\,2 \\ 2\,1\,1\,2 \\ 1\,2\,2\,1 \\ 1\,3\,1\,2 \end{pmatrix}$$

Note that $c + d = m$ and we get $d$ by the point reflection in the center of the grid. (Therefore $c \in \mathcal{R} \iff d \in \mathcal{R}$). It is easy to verify that $c \oplus b = c$ and $d \oplus b = d$.

However, for $n = 5$, we don't find configurations $c, d$ with $f_{c+d} = 3$, which is the lower bound we get from the inequality $|f_{c+d}| \geq \frac{n^2-4n}{2}$, although the proof is rather inelegant.

## 6 Searching for Local Minima

Now, we take a look at larger values for $n$ and describe a search strategy for recurrent configurations $c, d$ such that $|f_{c+d}|$ becomes small.

First, we will show how to construct $\bar{c} \in \mathcal{R}$ such that for all $d \in \mathcal{R}$ $f_{c+\bar{c}} \leq f_{c+d}$ holds.

Then we will describe how to optimize $c$.

### 6.1 Minimizing $f_{c+d}$ for Fixed $c$

Let $c$ be a recurrent configuration. Then for all $d \in \mathcal{R}$ the following inequality holds:

$f_{c+d} \geq f_{c+((m-c)\oplus id)}$:

There are no topplings during the relaxation of $(c \oplus d) + (m - (c \oplus d))$, and therefore

$c + d - B \cdot f_{c+d} + (m - (c \oplus d)) =$
$c + d + (m - (c \oplus d)) - B \cdot f_{c+d} =$
$c \oplus (d \oplus (m - (c \oplus d))) =$
$c \oplus (d + (m - (c \oplus d)) - B \cdot f_{d+(m-(c\oplus d))})$.

We define $\bar{c} = d \oplus (m - (c \oplus d))$ and get

$c + d + (m - (c \oplus d)) - B \cdot f_{c+d} = c + \bar{c} - B \cdot f_{c\oplus\bar{c}}$
$\Rightarrow c+d+(m-(c\oplus d))-B\cdot f_{c+d} = c+d+(m-(c\oplus d))-B\cdot f_{d+(m-(c\oplus d))}-B\cdot f_{c+\bar{c}}$
$\Rightarrow f_{c+d} = f_{c+\bar{c}} + f_{d+(m-(c\oplus d))}$,

since $B$ is invertible.

We see that $f_{c+\bar{c}} \leq f_{c+d}$.

We know that $d \oplus (m - (c \oplus d)) \in \mathcal{R}$ since $d \in \mathcal{R}$, and we know that $c \oplus (d \oplus (m - (c \oplus d))) = m$.

On the other hand, $id \oplus (m - c) \in \mathcal{R}$ and $c \oplus (id \oplus (m - c)) = m$, and it follows that for all $d \in \mathcal{R}$ the equation $id \oplus (m - c) = d \oplus (m - (c \oplus d))$ holds.

Therefore for all $d \in \mathcal{R}$ the inequality $f_{c+((m-c)\oplus id)} \leq f_{c+d}$ holds.

From now on, we call the configuration $(m - c) \oplus id$ the *minimizing configuration* of $c$, denoted as $\bar{c}$.

The task to minimize the topplings during the relaxation of the sum of two recurrent configurations is now reduced to finding a recurrent configuration $c$ such that $|f_{c+\bar{c}}|$ becomes minimal.

### 6.2 The Cutting Algorithm

Let $c$ be a recurrent configuration, $\bar{c}$ the minimizing configuration of $c$ and $e$ a configuration in $\mathcal{C}$ which is component-wise less or equal to $c$.

It can be easily shown that $f_{c+\bar{c}} = f_{(c-e)+(\bar{c}\oplus e)} + f_{\bar{c}+e}$.

This means that there are fewer topplings during the relaxation of $(c - e) + (\bar{c} \oplus e)$ if there were topplings during the relaxation of $\bar{c} + e$. If we make sure that $c - e$ is still a recurrent configuration, we find a pair of recurrent configurations which induces fewer topplings during the relaxation of their sum.

(Also, if $c - e$ is recurrent, it is easy to see that $\bar{c} \oplus e$ is the minimizing configuration of $c - e$.)

The Cutting Algorithm gives for the recurrent configuration $c$ (and its minimizing configuration $\bar{c}$) a recurrent configuration $d$ (and its minimizing configuration $\bar{d}$) such that $d \oplus \bar{d} = c \oplus \bar{c}$ and $f_{d+\bar{d}} \leq f_{c+\bar{c}}$ hold.

We start with $c + b$, work along a possible toppling sequence for $c + b$ (i.e. in each time step, we choose a site that can topple in the changed configuration) and transfer grains of sand that are left on a site $z$ after $z$ has toppled from $c$ to $\bar{c}$.

As in the beginning of section 4, the chosen toppling sequence for $c+b$ is also a toppling sequence for the configuration we get after transferring grains from $c$ to $\bar{c}$.

After each grain transfer we let the changed configuration $\bar{c}$ relax; in the end we get the configurations $d = c - e \in \mathcal{R}$ and $\bar{c} \oplus e$, which is the minimizing

configuration $\bar{d}$ of $d$. (The configuration $e$ here is the configuration which assigns each site the number of grains that were transferred on this site from $c$ to $\bar{c}$.)

Now, $(c-e) \oplus (\bar{c} \oplus e) = m$ and $f_{(c-e)+(\bar{c}\oplus e)} = f_{c+\bar{c}} - f_{\bar{c}+e} \leq f_{c+\bar{c}}$.

(Note that $e$ depends on the toppling sequence and the number of grains which are taken transferred from each site.)

We will denote the results $(c-e, \bar{c}\oplus e)$ of the Cutting Algorithm for arguments $c, \bar{c}$ as $cut(c, \bar{c})$.

The nice thing about the Cutting Algorithm is the fact that you can repeat it and thereby get better results:

After computing $(d, \bar{d}) = cut(c, \bar{c})$, it is often possible to find a configuration $e \in \mathcal{C}, e \leq d$ such that $d - e \in \mathcal{R}$ and $f_{\bar{d}+e} \neq 0$.

In pseudo code this process could be described as follows:

1: DO
2: $\quad (c, \bar{c}) \leftarrow cut(c, \bar{c})$
3: $\quad (\bar{c}, c) \leftarrow cut(\bar{c}, c)$
4: WHILE(!*exit condition*)

A local minimum is reached in $(c, \bar{c})$ if neither $cut(c, \bar{c})$ nor $cut(\bar{c}, c)$ can be a "better" pair than $(c, \bar{c})$.

The exit condition in the pseudo code program should be chosen in a way that makes it very likely that a local minimum has been reached. (A simple possibility would be to count how many times in a row the vector $f_{c+\bar{c}}$ has not changed and exit if this number is higher than a chosen threshold.)

### 6.3 The Local Minimum Condition

In this subsection we will show how to determine whether a pair $(c, \bar{c})$ is a local minimum. We will also be able to find a "better" pair $(d, \bar{d})$.

A pair $(c, \bar{c})$ is a local minimum if there exists no $e \in \mathcal{C}$ such that either $c - e \in \mathcal{R} \wedge f_{\bar{c}+e} \neq 0$ or $\bar{c} - e \in \mathcal{R} \wedge f_{c+e} \neq 0$ holds, since we could get a "better" pair $(c - e, \bar{c} \oplus e)$ respectively $(c \oplus e, \bar{c} - e)$ otherwise.

If $(c, \bar{c})$ is not a local minimum, then there exists a configuration $e \in \mathcal{C}$ such that $c - e \in \mathcal{R} \wedge f_{\bar{c}+e} \neq 0$ or $\bar{c} - e \in \mathcal{R} \wedge f_{c+e} \neq 0$ holds. Without loss of generality, we assume the former. Then there exists a site $z \in Z$ and a number $k \in \mathbb{N}$ such that $(\bar{c} + ke_z)(z) \geq 4$ and $c - ke_z \in \mathcal{R}$; surely $k \geq 4 - \bar{c}(z)$ holds.

It follows that $f_{\bar{c}+(4-\bar{c}(z))e_z} \neq 0$ and $c - (4 - \bar{c}(z))e_z \in \mathcal{R}$ hold, and we can formulate a local minimum condition as follows:

If $\exists e \in \mathcal{C} : \bar{c} - e \in \mathcal{R} \wedge f_{c+e} \neq 0$ holds, we can find a site $z \in Z$ such that $f_{c+(4-c(z))e_z} \neq 0$ and $\bar{c} - (4 - c(z))e_z \in \mathcal{R}$ hold.

So we can determine whether a given pair $(c, \bar{c})$ is a local minimum by looking at each site $z \in Z$ and checking for $z$ if any of the two configurations $c - (4 - \bar{c}(z))e_z$ and $\bar{c} - (4 - c(z))e_z$ is recurrent; this has a time complexity in $O(n^4)$ and is therefore rather slow.

(One case where it is easy to verify that a local minimum has been reached: If both $c$ and $\bar{c}$ are minimal recurrent configurations, we know that we have reached a local minimum.)

We define the function $check(c, \bar{c})$ which checks for all sites $z \in Z$ if $c - (4 - \bar{c}(z))e_z \in \mathcal{R}$ holds. In this case, $c$ is set to $c - (4 - \bar{c}(z))e_z$ and $\bar{c}$ to $(\bar{c} + (4 - \bar{c}(z))e_z)_{rel}$, and at the end of the procedure the value 0 is returned. If no site $z$ satisfies this condition, the value 1 is returned. (Note that $|f_{c+\bar{c}}|$ decreases if 0 is returned.)

### 6.4 Outline of the Algorithm

For a fixed number $k$, we set $(c, \bar{c})$ by turns to $cut(c, \bar{c})$ and $cut(\bar{c}, c)$, each time checking whether $f_{c+\bar{c}}$ changes.

After $f_{c+\bar{c}}$ has not changed for $k$ runs of the loop, we use $check(c, \bar{c})$ and $check(\bar{c}, c)$ to see whether a local minimum has been reached or setting $(c, \bar{c})$ to a pair $(c', \bar{c'})$ with $f_{c'+\bar{c'}} \leq f_{c+\bar{c}}$ and $f_{c'+\bar{c'}} \neq f_{c+\bar{c}}$.

If a local minimum has not been reached, we again use the Cutting Algorithm until $f_{c+\bar{c}}$ has not changed for $k$ times and use the *check* algorithm again.

This is repeated until a local minimum is reached, which eventually must happen, since $|f_{c+\bar{c}}|$ always decreases when the *check* algorithm finds that no local minimum has been reached.

## 7 Analysis

There are two functions used to decrease $|f_{c+\bar{c}}|$: $cut(c, \bar{c})$ and $check(c, \bar{c})$.

While $cut(c, \bar{c})$ is generally much faster to compute than $check(c, \bar{c})$, $cut(c, \bar{c})$ is a probabilistic function and may not reduce $|f_{c+\bar{c}}|$ although it would be possible; on the other hand, $check(c, \bar{c})$ reduces $|f_{c+\bar{c}}|$ if $(c, \bar{c})$ is not a local minimum, but generally needs much more time to do so than a call of $cut(c, \bar{c})$ does.

Therefore, we recommend using $cut(c, \bar{c})$ most of the time and $check(c, \bar{c})$ only after no improvements have been made for some time.

### 7.1 Strategies for Choosing the Toppling Sequence

During $cut(c, \bar{c})$, we have to choose a sequence for the sites of $Z$ to topple during the relaxation of $c + b$, and for each site that toppled we have to choose how many grains we transfer to $\bar{c}$.

Several strategies have been tried, in every strategy we used rectangular distribution:

1. *Random/Biased*: If we choose a *random* strategy, we randomly choose a site $z$ which contains at least 4 grains at each step and let it topple.

   If we choose a *biased* strategy, we choose a number $k$ and pick up to $k$ times a site $z$ with at least 4 grains on it and choose the first one for which the sum of grains in both configurations is at least 8 and let it topple. If no such site is picked, we choose the last of the $k$ sites and let it topple.
2. *Generous/Sparing*: A *generous* strategy always transfers all grains on $z$ after it toppled from one configuration to the other.

A *sparse* strategy only transfers grains on $z$ from one configuration to the other if this leads to topplings, and in this case only as many as necessary for a toppling to occur.

Experiments suggest that a *generous* strategy yields the better results than a *sparing* strategy, considering running time as well as $|f_{c+\bar{c}}|$.

For example, on a $200 \times 200$ grid with $k = 10$, the *random, generous* strategy yielded a result after 188 iterations of the inner loop and 2 *check* calls (which confirmed the local optimum) a configuration $c$ for which $|f_{c+\bar{c}}| = 56865720$.

On the same grid, the *biased, generous* strategy with three tries needed 183 iterations and just two *check* calls to verify the local minimum and yielded a configuration $c$ for which $|f_{c+\bar{c}}| = 56847147$.

The *random, sparing* strategy needed 839 iterations of the inner loop and 12 *check* calls to yield a a configuration $c$ for which $|f_{c+\bar{c}}| = 57027080$.

## 8 Conclusion

We have introduced the problem of finding two recurrent configurations such that the relaxation of their sum needs as few topplings as possible.

While it is easy to find a second recurrent configuration which minimizes the number of topplings if the first configuration is given, it is non-trivial to find the global minimum among all pairs of recurrent configurations.

We have given a probabilistic algorithm which finds local minima. The performance of the algorithm depends on the strategy to choose the sequence of sites $z \in Z$ during one routine; the best strategy found is to choose the next site randomly, although further experiments concerning the quality of the various strategies should be done.

We conjecture that for each local minimum $(c, \bar{c})$ the vector $f_{c+\bar{c}}$ is minimal, meaning there are no recurrent configurations $d, \bar{d}$ such that $f_{d+\bar{d}} \leq f_{c+\bar{c}}$ and $f_{d+\bar{d}} \neq f_{c+\bar{c}}$ hold.

Other future work includes the searching for better strategies for the choosing of a toppling sequence for $c + b$ in the Cutting Algorithm and the question how hard it is to decide whether a given pair $(c, \bar{c})$ is the global optimum.

## References

1. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: An explanation of the 1/f noise. Phys. Rev. Lett. 59, 381–384 (1987)
2. Chung, F., Ellis, R.: A chip-firing game and dirichlet eigenvalues. Discrete Mathematics 257, 341–355 (2002)
3. Dhar, D., Ruelle, P., Sen, S., Verma, D.N.: Algebraic aspects of abelian sandpile models. J.PHYS.A 28, 805 (1995)
4. Majumdar, S.N., Dhar, D.: Equivalence between the abelian sandpile model and the $q \longrightarrow 0$ limit of the potts model. Physica A: Statistical and Theoretical Physics 185, 129–145 (1992)

# A Construction Method of Moore Neighborhood Number-Conserving Cellular Automata

Naonori Tanimoto and Katsunobu Imai

Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima, Japan
{naonori,imai}@iec.hiroshima-u.ac.jp

**Abstract.** A number-conserving cellular automaton (NCCA) is a cellular automaton such that all states of cells are represented by integers and the total of the numbers (states) of all cells of a global configuration is conserved throughout its computing process. It can be thought to be a kind of modelization of the physical conservation law of mass or energy. In this paper, we show a sufficient condition for a Moore neighborhood CA to be number-conserving. According to this condition, the local function of rotation-symmetric NCCA is expressed by a summation of quaternary functions. On this framework, we construct a 6-state logically universal NCCA.

**Keywords:** cellular automata, number-conservation, logical universality.

## 1 Introduction

A number-conserving cellular automaton (NCCA) is a cellular automaton such that all states of cells are represented by integers and the total of the numbers (states) of all cells of a global configuration is conserved throughout its computing process. It can be thought to be a kind of modelization of the physical phenomena, for example, fluid dynamics and highway traffic flow[6].

Boccara et al[2]. studied number conservation of one-dimensional CAs on circular configurations. Durand et al[3]. studied the two-dimensional case and the relation between several boundary conditions. Although their theorems are useful for deciding if a given CA is number-conserving or not, it is quite difficult to design NCCAs with complex transition rules.

As for von Neumann neighborhood NCCAs with and without rotation-symmetry, necessary and sufficient conditions to be number-conserved are shown respectively[7]. According to these conditions, the local function of a rotation-symmetric NCCA is expressed by a summation of two binary functions. Designing the two binary functions, we can construct a rotation-symmetric NCCA easily. On this framework, we constructed 14-state logically universal NCCA with rotation-symmetry. (A logically universal NCCA can simulate any boolean circuits. If it has a cyclic configuration, it can simulate a universal Turing machine.)

It seems that there is a trade-off between the size of state-number of a universal NCCA and the size of its neighborhood, and there is no rotation-symmetric von Neumann neighborhood NCCA of which the element number of the state

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 244–251, 2008.

set (state set size) is less than five[8]. To find a smaller state universal NCCA, we consider Moore neighborhood which is also used widely. A necessary and sufficient condition for Moore neighborhood CA to be number-conserving is shown by Durand et al[3]. But it is still difficult to construct Moore neighborhood NCCAs by using the condition directly.

Consequently, we show another sufficient condition for a CA to be number-conserving. According to this condition, the local function of rotation-symmetric NCCA is expressed by a summation of quaternary functions $\phi$. $\phi$ indicates the behavior of values within a block composed of four cells. Assigning a function $\phi$ is enough to design a rotation-symmetric Moore neighborhood NCCA. It can be thought that the transition of an NCCA can be expressed based on local flows of value such as $\phi$, whatever the neighborhood is. Although the shape of partition is not unique for a neighborhood, $\phi$ is the most natural one in case of the Moore neighborhood. Employing this method, we construct a 6-state logically universal NCCA.

## 2 Definitions

**Definition 1.** *A deterministic two-dimensional Moore neighborhood cellular automaton is a system defined by* $A = (\mathbf{Z}^2, Q, f, q)$*, where* $\mathbf{Z}$ *is the set of all integers,* $Q$ *is a non-empty finite set of internal states of each cell,* $f : Q^9 \to Q$ *is a mapping called a local function and* $q \in Q$ *is a quiescent state that satisfies* $f(q, q, q, q, q, q, q, q, q) = q$*.*

A *configuration* over $Q$ is a mapping $\alpha : \mathbf{Z}^2 \to Q$ and the set of all configurations over $Q$ is denoted by $\mathrm{Conf}(Q)$, i.e., $\mathrm{Conf}(Q) = \{\alpha | \alpha : \mathbf{Z}^2 \to Q\}$. The function $F : \mathrm{Conf}(Q) \to \mathrm{Conf}(Q)$ is defined as follows and called the *global function* of A.

$$\forall (x, y) \in Z^2,$$
$$F(\alpha)(x, y) = f(\alpha(x-1, y-1), \alpha(x, y-1), \alpha(x+1, y-1), \alpha(x-1, y), \alpha(x, y),$$
$$\alpha(x+1, y), \alpha(x-1, y+1), \alpha(x, y+1), \alpha(x+1, y+1)).$$

In the case of von Neumann neighborhood, the above definition is the same except that its local function $f$ is $f : Q^5 \to Q$ and its global function $F$ is defined as follow.

$$\forall (x, y) \in Z^2, F(\alpha)(x, y) = f(\alpha(x, y), \alpha(x, y+1), \alpha(x+1, y), \alpha(x, y-1), \alpha(x-1, y)).$$

We assume a quiescent state $q$ such that $f(q, q, q, q, q) = q$.

In this paper, we only consider CAs with *finite configurations*, i.e., the number of cells which states are not quiescent is finite. $A$ is said to be *number-conserving* when it satisfies $\sum_{(x,y)\in\mathbf{Z}^2}\{F(\alpha)(x, y) - \alpha(x, y)\} = 0$ for all finite configurations $\alpha$.

Next we define some symmetry conditions.

**Definition 2.** *CA A is said to be rotation-symmetric if its local function f satisfies the following condition.*

$$\forall c, u, r, d, l, ul, ur, dl, dr \in Q, f(ul, u, ur, l, c, r, dl, d, dr) = f(dl, l, ul, d, c, u, dr, r, ur).$$

*In the case of von Neumann neighborhood, these symmetries is defined as above.*

## 3 Von Neumann Neighborhood Number-Conserving Cellular Automata

We showed a necessary and sufficient condition for a von Neumann neighborhood CA to be number-conserving in [7].

**Theorem 1.** *[7] A deterministic two-dimensional von Neumann neighborhood CA* $A = (\mathbf{Z}^2, Q, f, q)$ *is number-conserving iff* $f$ *satisfies*

$$\begin{aligned}
&\exists g_U, g_R, g_D, g_L, h_{UR}, h_{RD}, h_{DL}, h_{LU} : Q^2 \to \mathbf{Z}, \forall c, u, r, d, l \in Q,\\
&f(c,u,r,d,l) = c + g_U(c,u) + g_R(c,r) + g_D(c,d) + g_L(c,l)\\
&\qquad\qquad + h_{UR}(u,r) + h_{RD}(r,d) + h_{DL}(d,l) + h_{LU}(l,u),\\
&g_U(c,u) = -g_D(u,c), g_R(c,r) = -g_L(r,c),\\
&h_{UR}(u,r) = -h_{DL}(r,u), h_{RD}(r,d) = -h_{LU}(d,r).
\end{aligned}$$

Next, we derived a necessary and sufficient condition for a rotation-symmetric CA to be number-conserving from the condition.

**Corollary 1.** *[7] A deterministic two-dimensional rotation-symmetric von Neumann neighborhood CA* $A = (\mathbf{Z}^2, Q, f, q)$ *is number-conserving iff* $f$ *satisfies*

$$\begin{aligned}
&\exists g, h : Q^2 \to \mathbf{Z}, \forall c, u, r, d, l \in Q,\\
&f(c,u,r,d,l) = c + g(c,u) + g(c,r) + g(c,d) + g(c,l)\\
&\qquad\qquad + h(u,r) + h(r,d) + h(d,l) + h(l,u),\\
&g(c,u) = -g(u,c), h(u,r) = -h(r,u).
\end{aligned}$$

According to this condition, a local function of a rotation-symmetric NCCA is expressed by a summation of two binary function $g$ and $h$. The binary function $g$ indicates the number flow between two cells in a vertical or horizontal direction, and $h$ does in a diagonal direction. In a vertical or horizontal flow, a value moves on two cells of which states are arguments of $g$. But in the diagonal flow case, cells on which a value moves don't correspond to arguments of $h$. This causes divergence of the state set size.

In order to design a rotation-symmetric NCCA, we only have to define $g$ and $h$. Although it may not be a CA for divergence of the state set size, there exists a procedure to modify these functions for construction of an NCCA after designing $g$ and $h$.

## 4 A Sufficient Condition Based on Quaternary Functions

Durand et al[3]. showed a necessary and sufficient condition for an NCCA in the case of $n \times m$ neighborhood. We show the condition which is restricted to Moore neighborhood.

**Theorem 2.** *[3] A deterministic two-dimensional Moore neighborhood CA* $A = (\mathbf{Z}^2, Q, f, q)$ *is number-conserving iff* $f$ *satisfies*

$$
\begin{aligned}
&\forall c, u, r, d, l, ul, ur, dl, dr, \in Q,\\
&f(ul, u, ur, l, c, r, dl, d, dr) = ul + f(q, u, ur, q, c, r, q, d, dr) + f(q, q, u, q, q, c, q, q, d)\\
&\quad + f(q, q, q, q, u, ur, q, c, r) + f(q, q, q, q, q, u, q, q, c) + f(q, q, q, q, q, q, q, u, ur)\\
&\quad + f(q, q, q, q, q, q, q, q, u) + f(q, q, q, l, c, r, dl, d, dr) + f(q, q, q, q, l, c, q, dl, d)\\
&\quad + f(q, q, q, q, q, l, q, q, dl) + f(q, q, q, q, q, q, l, c, r) + f(q, q, q, q, q, q, q, l, c)\\
&\quad + f(q, q, q, q, q, q, q, q, l)\\
&\quad - f(q, ul, u, q, l, c, q, dl, d) - f(q, q, ul, q, q, l, q, q, dl) - f(q, q, q, ul, u, ur, l, c, r)\\
&\quad - f(q, q, q, q, ul, u, q, l, c) - f(q, q, q, q, q, ul, q, q, l) - f(q, q, q, q, q, q, ul, u, ur)\\
&\quad - f(q, q, q, q, q, q, q, ul, u) - f(q, q, q, q, q, q, q, q, ul) - f(q, q, q, q, c, r, q, d, dr)\\
&\quad - f(q, q, q, q, q, c, q, q, d) - f(q, q, q, q, q, q, q, c, r) - f(q, q, q, q, q, q, q, q, c). \quad (1)
\end{aligned}
$$

It is difficult to design $f$ which satisfies the condition. Because when we allocate one value of $f$, we have to consider other 24 values at most. In addition, each 24 values must satisfies the condition. Therefore we introduce another representation of sufficient condition to be number-conserving and show the condition is usable to design Moore neighborhood NCCAs.

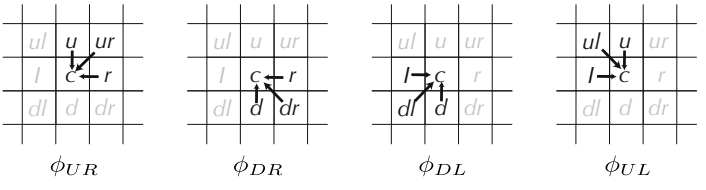



**Fig. 1.** An interpretation of $\phi_{XX}$

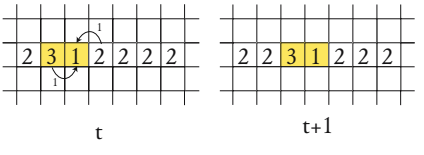



**Fig. 2.** A configuration of a wire and a signal in $A_{ex}$

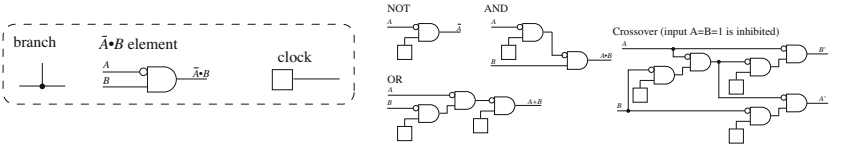



**Fig. 3.** A construction of basic logic elements [4]

**Corollary 2.** *A deterministic two-dimensional Moore neighborhood CA* $A = (\mathbf{Z}^2, Q, f, q)$ *is number-conserving if* $f$ *satisfies*

$$\exists \phi_{UL}, \phi_{UR}, \phi_{DL}, \phi_{DR} : Q^4 \to \mathbf{Z}, \forall c, u, r, d, l, ul, ur, dl, dr \in Q,$$
$$f(ul, u, ur, l, c, r, dl, d, dr) = c + \phi_{UR}(c, u, ur, r) + \phi_{DR}(c, r, dr, d) + \phi_{DL}(c, d, dl, l) + \phi_{UL}(c, l, ul, u), \quad (2)$$
$$\phi_{UR}(c, u, ur, r) + \phi_{DR}(u, ur, r, c) + \phi_{DL}(ur, r, c, u) + \phi_{UL}(r, c, u, ru) = 0. \quad (3)$$

*Proof.* We assume a CA $A$ of which $f$ satisfies the equations (2) and (3). Then this $f$ suffuses (1). Therefore the CA $A$ is number-conserving.

The local function $f$ is expressed by a summation of four quaternary functions $\phi_{UR}, \phi_{DR}, \phi_{DL}$ and $\phi_{UL}$, see Equation (2). Each function represents the received value by the central cell from the other three cells, see Fig. 2.

When we design the function $\phi$s, we have only to take into account the interactions of four cells appeared in each $\phi$ at a time. The interaction should satisfy the equation (3). Especially if $\phi \equiv \phi_{UR} = \phi_{DR} = \phi_{DL} = \phi_{UL}$, the NCCA is rotation-symmetric. To obtain the actual state-number of the NCCA by this construction method, we need to follow the procedure as below.

1. Choose a *partial* state set $\tilde{Q}$ of size $k(> 0)$ and design $\phi(i, j, k, l)$ for $(i, j, k, l) \in \tilde{Q}^4$.
2. Determine the state set $Q$ by extending $\tilde{Q}$ to $Q$. This is performed by calculating the set $Q'$ which contains all next values of local function $f(ul, u, ur, l, c, r, dl, d, dr)$ for all $(ul, u, ur, l, c, r, dl, d, dr) \in \tilde{Q}^9$ and $Q := Q' \cup \tilde{Q}$. The domain of $\phi$ is extended from $\tilde{Q}$ to $Q$. (Note that $\phi(x_0, x_1, x_2, x_3) \equiv 0$ for any element of $\{(x_0, x_1, x_2, x_3) \in Q | \exists i, x_i \in (Q - \tilde{Q})\}$)

*Example 1.* Let's consider an NCCA $A_{ex}$ depicted in Fig. 2. It realizes structures which can be regarded as wires and signals. A wire is a connected set of cells of width 1 of which the states are 2. A signal is encoded by two cells whose states are 1 and 3, and it flows along a wire at speed 1, and at the end of the wire(i.e., when it faces to three quiescent cells), the signal is terminated. The state set $\tilde{Q}_{ex} = \{0, 1, 2, 3\}$ is 4-state and the function $\phi_{ex}(i, j, k, l), (i, j, k, l) \in \tilde{Q}_{ex}$ has non-zero values only at $(i, j, k, l) = (3, 1, 0, 0), (1, 0, 0, 3), (1, 2, 0, 0), (2, 0, 0, 1)$, thus $\phi_{ex}(3, 1, 0, 0) = -1, \phi_{ex}(1, 0, 0, 3) = 1, \phi_{ex}(1, 2, 0, 0) = 1$, and $\phi_{ex}(2, 0, 0, 1) = -1$ realize the movement of signals. By calculating its local function $f_{ex}$ for all $(ul, u, ur, l, c, r, dl, d, dr) \in \tilde{Q}_{ex}$, its state set $Q_{ex} = \tilde{Q}_{ex}$. Thus the obtained NCCA is 4-state Moore neighborhood NCCA $A_{ex} = (\mathbf{Z}^2, Q_{ex}, f_{ex}, 0)$.

**Table 1.** Defined non-zero values of $g_{14}$ and $h_{14}$

| |
|---|
| $g_{14}(1,2) = 1$, $g_{14}(1,3) = 1$, $g_{14}(1,4) = 1$, $g_{14}(1,6) = 1$, $g_{14}(7,5) = 1$, |
| $g_{14}(4,3) = 1$, $g_{14}(5,4) = 1$, $g_{14}(4,8) = 1$, $g_{14}(1,5) = 1$, $g_{14}(5,11) = 1$, |
| $g_{14}(-2,8) = 1$, $h_{14}(8,5) = 1$ |

**Table 2.** Defined non-zero values of $\phi$

| | | | |
|---|---|---|---|
| $\phi(3,1,0,0)=-1$, | $\phi(1,0,0,3)=1$, | $\phi(1,2,0,0)=1$, | $\phi(2,0,0,1)=-1$, |
| $\phi(4,1,0,0)=-1$, | $\phi(1,0,0,4)=1$, | $\phi(1,2,0,2)=1$, | $\phi(2,0,2,1)=-1$, |
| $\phi(1,2,2,0)=1$, | $\phi(2,2,0,1)=-1$, | $\phi(1,2,3,3)=1$, | $\phi(3,3,1,2)=-1$, |
| $\phi(2,2,1,3)=-1$, | $\phi(2,1,3,2)=1$, | $\phi(3,1,2,0)=-1$, | $\phi(1,2,0,3)=1$, |
| $\phi(4,1,0,1)=-1$, | $\phi(1,0,1,4)=1$ | | |

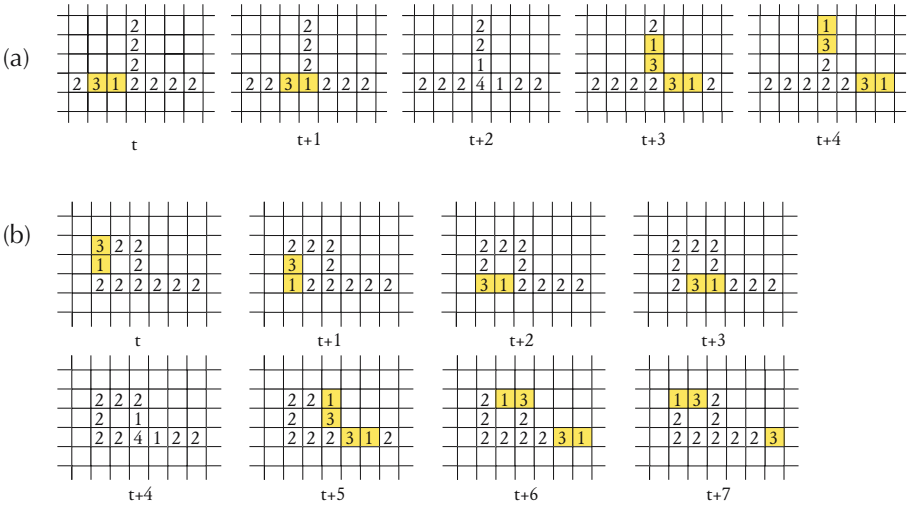



**Fig. 4.** Configurations of a junction(a) and a clock(b)[5]

## 5 A Logically Universal NCCA

Banks[1] shows a two-state logically universal von Neumann neighborhood CA, i.e., it is possible to embed any boolean circuit into its cellular space as a configuration and its transition process can simulate the circuit. His model has configurations of a signal and a wire. Wires are allowed to have branching points and there are two types of branching points, a junction and a $\bar{A}\cdot B$ logic element. His model also has a clock (periodic signal generator). He shows that above elements can be embedded into his two-state von Neumann neighborhood CA and combining these elements, he also shows that it is possible to realize AND, OR, NOT, and signal crossover elements depicted in Fig. 3.

In NCCA case, we showed a 14-state logically universal CA $A_{14}$ with von Neumann neighborhood and rotation-symmetry. $A_{14}=(\mathbf{Z}^2, N_{[-2,11]}, f_{14}, 0)$[7]. The flow functions $g_{14}, h_{14}$ of $f_{14}$ have the values in Table 1 and satisfy $g_{14}(x,y)=-g_{14}(y,x)$ and $h_{14}(x,y)=-h_{14}(y,x)$ for all $x,y\in N_{[-2,11]}$. Values not defined by Table 1 are 0. Fig. 4 shows configurations of a junction and a period 9 clock.

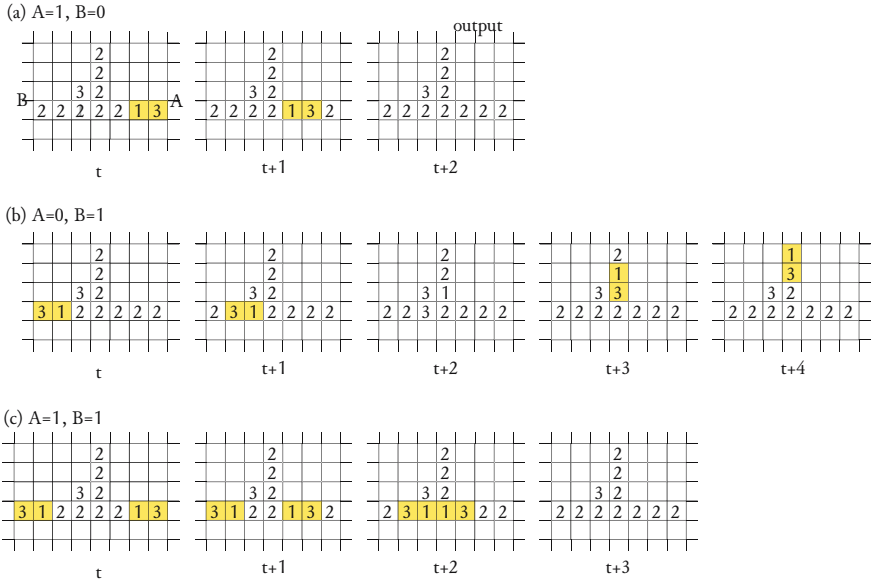



**Fig. 5.** Configurations of $\bar{A} \cdot B$ logic element for each inputs

In Moore neighborhood NCCA, we construct logically universal CA $A_6$ with rotation-symmetry. $A_6 = (\mathbf{Z}^2, \{0, 1, 2, 3, 4, 5\}, f_6, 0)$. $\phi$ which defines $f_6$ has the values in Table 2. Values not defined by Table 2 are 0. A junction and a clock are same as $A_{14}$ (Fig. 4). Fig. 5 shows configurations of $\bar{A} \cdot B$ logic elements for each inputs.

## 6 Conclusion

In this paper, we gave a sufficient condition for a Moore neighborhood CA to be number-conserving and showed that the condition is usable to construct a Moore neighborhood NCCA. According to this condition, the local function of rotation-symmetric NCCA is expressed by a summation of quaternary functions $\phi$. $\phi$ indicates the behavior of values within a block composed of four cells. Assigning a function $\phi$ is enough to design a rotation-symmetric Moore neighborhood NCCA. Employing this method, we constructed a 6-state logically universal NCCA.

An NCCA which is constructed by $\phi$ has a stronger symmetry than the rotation-symmetry. But the symmetry is not clear so far. In case of a larger radius neighborhood, a construction method of an NCCA may be established by the same approach which separates a neighborhood into several blocks.

**Acknowledgments.** The authors thank Kenichi Morita and Chuzo Iwamoto for their daily advices. The authors also thank anonymous reviewers for their attentive comments.

## References

1. Banks, E.R.: Universality in cellular automata. In: Proc. Eleventh Annual Symposium on Switching and Automata Theory, pp. 194–215 (1970)
2. Boccara, N., Fukś, H.: Number-conserving cellular automaton rules. Fundamenta Informaticae 52, 1–13 (2003)
3. Durand, B., Formenti, E., Grange, A., Róka, Z.: Number conserving cellular automata: new results on decidability and dynamics. In: Morvan, M., Rémila, É. (eds.) Discrete Models for Complex Systems, DMCS 2003. Discrete Mathematics and Theoretical Computer Science Proceedings, vol. AB, pp. 129–140 (2003)
4. Imai, K., Ikazaki, A., Iwamoto, C., Morita, K.: A logically universal number-conserving reversible cellular automaton with a unary table-lookup function. Trans. IEICE E87-D(3), 694–699 (2004)
5. Imai, K., Furuta, K., Iwamoto, C., Morita, K.: A universal number conserving cellular automaton whose number of inter-cell moving particle is restricted. In: 12th International Workshop on Cellular Automata (2006)
6. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal of Physics I France 2, 2221–2229 (1992)
7. Tanimoto, N., Imai, K.: A characterization of von Neumann neighbor number-conserving cellular automata. In: Proceedings of 2nd International Workshop on Natural Computing, 10 pages (2007)
8. Tanimoto, N., Imai, K., Iwamoto, C., Morita, K.: On constraints of rotation-symmetry of number-conserving cellular automata on their state numbers. In: LA symposium 2007 winter (in Japanese), pp. 4.1–4.5 (2008)

# Changing Neighborhoods of CA: Reduced Local Structures and Embeddings for Universality

Thomas Worsch[1,⋆] and Hidenosuke Nishio[2]

[1] University of Karlsruhe, Germany
worsch@ira.uka.de
[2] ex. Kyoto University, Japan
yra05762@nifty.com

**Abstract.** This paper consists of two parts. In the first we pick up again the question under which circumstances different pairs of a local function and a neighborhood give rise to the same global behavior of CA and disprove a conjecture made in an earlier paper. In the second part we reconsider a construction showing that one can achieve universality by only changing the (positions in the) neighborhood of a CA, while not providing any information about the CA to be simulated in the initial configuration. The construction uses an embedding which in some sense is "non-local". We show that under mild conditions this is necessary.

## 1 Introduction

Usually investigations of cellular automata without further discussion assume some standard neighborhood because it is "without loss of generality". In general this is correct, except, of course, when one is interested in questions specifically concerning neighborhoods. This is the guiding line of the current paper. It is a continuation e. g. of [3] and [5] (an extended version will appear in [4]).

The rest of the paper is organized as follows: In Section 2 we introduce the notions and notations used throughout the paper. Sections 3 and 4 concerned with a "normal form" of pairs $(f, \nu)$ of a local function and a neighborhood and the question under which circumstances different such pairs can give rise to the same global behavior of CA. Finally, in Section 5 we take a second look at a construction showing some kind of universality [5] and prove that the embedding used for the simulation of CA necessarily has to have a certain propoerty.

## 2 Basics

We assume that readers are familiar with the basic concepts of cellular automata (CA). In the first part of this paper we will consider $d$-dimensional Euclidean CA for any $d \in \mathbb{N}_+$. We will write $R$ for the set $\mathbb{Z}^d$ of all cells. In the second part for the sake of simplicity we will assume that $d = 1$. The set of states of a

⋆ Corresponding author.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 252–259, 2008.

single cell will usually be denoted by $Q$ (or $Q_A, \ldots$). A local transition function is of the form $f : Q^n \to Q$. Of course, one is interested in the case $n \geq 1$. But for technical reasons (a simpler proof of Lemma 3) we also allow $n = 0$; in this case $f : Q^0 \to Q$ simply is a constant.

A *neighborhood* is a mapping $\nu : \mathbb{P}_n \to R$, where $\mathbb{P}_n$ denotes the set $\{1, 2, \ldots, n\}$ of positive integers. As a special case we use $\mathbb{P}_0$ to denote the empty set. This can equivalently be seen as a list $\nu$ with $n$ components, written as $\nu(1), \ldots, \nu(n)$. Note that for the use with a local function the order may be important. The set of *all* neighborhoods of size $n$ will be denoted as $\mathcal{N}_n$.

A pair $(f, \nu)$ of a local function $f : Q^n \to Q$ and a neighborhood $\nu \in \mathcal{N}_n$ is called a *local structure*. We call $n$ the *arity* of the local structure.

Since $Q$ and $R$ are already implicit in $f$ and $\nu$ we will simply speak of a CA $\mathcal{A} = (f, \nu)$. As usual, each local structure $\mathcal{A} = (f, \nu)$ induces a global transition function $Q^R \to Q^R$ (which we also denote by $\mathcal{A}$) of a CA by

$$\forall\, x \in R : \mathcal{A}(c)(x) = f(c(x + \nu(1)), c(x + \nu(2)), \ldots, c(x + \nu(n))) \,.$$

The general question we are interested in is:

> What can and what cannot happen when changing the neighborhood of a local function, i. e. when going from a local structure $(f, \nu)$ to a local structure $(f, \nu')$?

A restricted version of this question is: What can and what cannot happen when permuting the neighbors of a local function, i. e. when going from a local structure $(f, \nu)$ to a local structure $(f, \nu')$, where $\nu'$ is a permutation of $\nu$?

## 3 Reduced Local Structures

**Definition 1.** *A local structure is called* reduced*, if and only if the following conditions are fulfilled:*

- *$f$ depends on each argument, i. e. for each $i \in \mathbb{P}_n$ there are $q_1, \ldots, q_{i-1}, q_i, q_i', q_{i+1}, \ldots, q_n \in Q$ such that*

$$f(q_1, \ldots, q_{i-1}, q_i, q_{i+1}, \ldots, q_n) \neq f(q_1, \ldots, q_{i-1}, q_i', q_{i+1}, \ldots, q_n) \,.$$

- *$\nu$ is injective, i. e. no $x \in R$ appears twice in the list. Such neighborhoods are called* non-degenerate *in [3].*

A somewhat special case is a *constant* local function $f$ and the empty neighborhood. In this case $f$ is of the form $f : Q^0 \to Q$, i. e. the only "argument list" for $f$ is the empty list, and $\nu : \mathbb{P}_0 \to R$ is the empty list as well. This local structure is reduced.

Obviously, for $n \geq 1$ the transition function of each reduced local structure is non-constant.

**Definition 2.** *Two local structures* $(f,\nu)$ *and* $(f',\nu')$ *are called* equivalent *if and only if they induce the same global function. In that case we sometimes write* $(f,\nu) \approx (f',\nu')$.

**Lemma 3.** *For each local structure* $(f,\nu)$ *there is an equivalent reduced local structure* $(f',\nu')$.

*Proof.* Let $n$ denote the arity of $(f,\nu)$. Assume that $(f,\nu)$ is not reduced. We will see that $n \geq 1$ and show how to construct an equivalent local structure $(f',\nu')$ with arity $n-1$.

**Case 1:** $\nu$ is not injective. Then clearly $n \geq 2$. Let $i$ and $j$ be indices such that $i < j$ and $\nu_i = \nu_j$. Define $\nu' \in \mathcal{N}_{n-1}$ as

- $\nu'_k = \begin{cases} \nu_k & \text{iff } k < j \\ \nu_{k+1} & \text{iff } k \geq j \end{cases}$, i. e. drop the $j$-th component of $\nu$, and
- $f' : Q^{n-1} \to Q$ by $f'(q_1,\ldots,q_{n-1}) = f(q_1,\ldots,q_{j-1},q_i,q_j,\ldots,q_{n-1})$

For any configuration $c \in Q^R$ holds:

$$\begin{aligned} F'(c)(0) &= f'(c(\nu'_1),\ldots,c(\nu'_{n-1})) \\ &= f(c(\nu'_1),\ldots,c(\nu'_{j-1}),c(\nu'_i),c(\nu'_j),\ldots,c(\nu'_{n-1})) \\ &= f(c(\nu_1),\ldots,c(\nu_{j-1}),c(\nu_i),c(\nu_{j+1}),\ldots,c(\nu_n)) \\ &= f(c(\nu_1),\ldots,c(\nu_{j-1}),c(\nu_j),c(\nu_{j+1}),\ldots,c(\nu_n)) \\ &= F(c)(0) \end{aligned}$$

Since application of local functions commutes with shifts, it follows $F'(c)(x) = F(c)(x)$ for all $x \in R$.

**Case 2:** $f$ does not depend on all arguments. Then clearly $n \geq 1$. Assume that it does not depend on argument $i$, $1 \leq i \leq n$. Define $\nu' \in \mathcal{N}_{n-1}$ as

- $\nu'_k = \begin{cases} \nu_k & \text{iff } k < i \\ \nu_{k+1} & \text{iff } k \geq i \end{cases}$, and
- $f'(q_1,\ldots,q_{n-1}) = f(q_1,\ldots,q_{i-1},q,q_{i+1},\ldots,q_{n-1})$ for any $q \in Q$. Since $f$ does not depend on the $i$-th argument, $f'$ is well defined.

For any configuration $c \in Q^R$ holds:

$$\begin{aligned} F'(c)(0) &= f'(c(\nu'_1),\ldots,c(\nu'_{i-1}),c(\nu'_i),c(\nu'_{i+1}),\ldots,c(\nu'_{n-1})) \\ &= f(c(\nu'_1),\ldots,c(\nu'_{i-1}),q,c(\nu'_i),c(\nu'_{i+1}),\ldots,c(\nu'_{n-1})) \\ &= f(c(\nu_1),\ldots,c(\nu_{i-1}),q,c(\nu_{i+1}),c(\nu_{i+2}),\ldots,c(\nu_n)) \\ &= f(c(\nu_1),\ldots,c(\nu_{i-1}),c(\nu_i),c(\nu_{i+1}),c(\nu_{i+2}),\ldots,c(\nu_n)) \\ &= F(c)(0) \end{aligned}$$

As in case 1 it follows that $F'(c)(x) = F(c)(x)$ for all $x \in R$. □

The construction above does *not* imply that the equivalent reduced local structure itself is unique. In fact in general it is not: As a simple example consider the local function $f : \{0,1\}^2 \to \{0,1\} : (x_1, x_2) \mapsto x_1 \wedge x_2$. Since the order of the arguments $x_i$ does not matter for the value $f(x_1, x_2)$ the local structures $(f, (0,1))$ and $(f, (1,0))$ are equivalent. At the same time both are obviously reduced.

**Open problem 4.** Given any non-reduced local structure $(f, \nu)$ and an equivalent reduced local structure $(f', \nu')$, is there always a sequence of operations as in the proof of Lemma 3 that transforms $(f, \nu)$ into $(f', \nu')$?

## 4 Equivalence of Local Structures

**Definition 5.** *For $n \geq 1$ let $\pi \in S_n$ denote a permutation of the numbers in $\mathbb{P}_n$.*

- *For a neighborhood $\nu$ denote by $\nu^\pi$ the neighborhood defined by $\nu^\pi_{\pi(i)} = \nu_i$.*
- *For an $n$-tuple $\ell \in Q^n$ denote by $\ell^\pi$ the permutation of $\ell$ such that $\ell^\pi(i) = \ell(\pi(i))$ for $1 \leq i \leq n$.*
- *For a local function $f : Q^n \to Q$ denote by $f^\pi$ the local function $f^\pi : Q^n \to Q$ such that for all $\ell$: $f^\pi(\ell) = f(\ell^\pi)$.*

In the first part of the definition we have preferred the given specification over the equally possible $\nu^\pi_i = \nu_{\pi(i)}$, because the former leads (in our opinion) to a slightly nicer formulation of the following lemma.

**Lemma 6.** *$(f, \nu)$ and $(f^\pi, \nu^\pi)$ are equivalent for any permutation $\pi$.*

*Proof.* For any configuration $c$:

$$\begin{aligned} F^\pi(c)(0) &= f^\pi(c(\nu^\pi_1), \ldots, c(\nu^\pi_n)) \\ &= f(c(\nu^\pi_{\pi(1)}), \ldots, c(\nu^\pi_{\pi(n)})) \\ &= f(c(\nu_1), \ldots, c(\nu_n)) \\ &= F(c)(0) \end{aligned}$$

□

We are now going to show that for *reduced (!)* local structures the relationship via a permutation $\pi$ is the *only* possibility to get equivalence.

**Lemma 7.** *If $(f, \nu)$ and $(f', \nu')$ are two reduced local structures which are equivalent, then there is a permutation $\pi$ such that $\nu^\pi = \nu'$.*

It should be noted that e. g. for local structures with a local transition function $f$ which does not depend on an input the claim of the lemma is wrong. Changing the position of the neighbor which "provides" the irrelevant input for $f$ of course does not change the global behavior.

*Proof.* Assume that there is an $x$ which does not appear in $\nu'$ but does appear in $\nu$, say at position $i$. Since $(f, \nu)$ is reduced, $f$ does depend on its $i$-th argument

and there are two configurations $c$ and $\bar{c}$, which do *only* differ at cell $x$, such that $F(c)(0) \neq F(\bar{c})(0)$.

Since $\nu'$ does not contain $x$, it is clear that $F'(c)(0) = F'(\bar{c})(0)$. It is therefore impossible that $F(c)(0) = F'(c)(0)$ and simultaneously $F(\bar{c})(0) = F'(\bar{c})(0)$. Hence $F(c) \neq F'(c)$ and $F \neq F'$. □

By choosing different neighborhoods which are not permutations of each other one immediately gets the following; compare Theorem 1 of [3]:

**Corollary 8.** *For each reduced non-constant local function there are infinitely many compatible neighborhoods inducing pairwise different global CA functions.*

**Lemma 9.** *If $(f, \nu)$ and $(f', \nu')$ are two reduced local structures which are equivalent, then there is a permutation $\pi$ such that $(f^\pi, \nu^\pi) = (f', \nu')$.*

*Proof.* By Lemma 7 we already know that $\nu$ and $\nu'$ are permutations of each other: $\nu' = \nu^\pi$ for some $\pi$; and $(f, \nu) \approx (f', \nu^\pi)$. But it is clear that different local functions induce different global functions, if they use the same neighborhood. Hence if one assumes $f' \neq f^\pi$, then $(f', \nu^\pi) \not\approx (f^\pi, \nu^\pi)$ which together with $(f^\pi, \nu^\pi) \approx (f, \nu)$ (Lemma 6) contradicts $(f, \nu) \approx (f', \nu^\pi)$. □

## 5 Universality Revisited

In order to keep notation simple, we will only consider one-dimensional CA in this section.

### 5.1 A Short Review of the Construction

In a previous paper [5] we have sketched a construction which allows to do the following:

Fix an arbitrary set of states $Q_A$ of cardinality $a \geq 2$ and consider the set of *all* CA $\mathcal{A}_i$ with local structures $(f_i, \nu_i)$ of arbitrary local functions $f_i : Q_A^{n_i} \to Q_A$ of any arity $n_i$ and compatible arbitrary neighborhoods $\nu_i \in \mathcal{N}_{n_i}$.

Then there is

- one set of states $Q_B$ of cardinality $b$,
- one embedding $E : Q_A^R \to Q_B^R$ of configurations and
- *one* local function $g : Q_B^k \to Q_B$ ($k = 5$ in [5])

such that

- for each $(f_i, \nu_i)$ one can effectively construct a neighborhood $\nu_i' \in \mathcal{N}_k$ such that
- for each configuration $c : R \to Q_A$
- using $(g, \nu_i')$ on initial configuration $E(c)$ simulates each step of each cell of the CA given by $(f_i, \nu_i)$.

In the construction we use $Q_B = Q_A \times Q'$ where $Q'$ contains the symbols $\circ$ and $\_$ (among others) and the embedding is defined as

$$E(c)(x) = \begin{cases} (c(x), \circ) & \text{if } x = 0 \\ (c(x), \_) & \text{otherwise} \end{cases}$$

### 5.2 Discussion of the Embedding

In this subsection we will discuss the embedding $E$ in more detail.

The idea of setting a special marker in one cell has appeared in the literature before. Durand-Lose [1] has used it in his construction to simulate irreversible one-dimensional CA on reversible one-dimensional CA for infinite configurations. This result has to be contrasted with a theorem by Hertling [2] stating that such a simulation is impossible. Both authors are right, simply because they use different notions of simulation. In particular, Durand-Lose uses an embedding of configurations which does set a special marker in cell 0, while Hertling only consider embeddings of a certain type called weak morphisms. $E$ is a weak morphims if there is an integer $m$ such that

$$E(\sigma(c)) = \sigma^m(E(c))$$

holds for all configurations $c$; here $\sigma$ is the shift $(\forall\, x :\ \sigma(c)(x) = c(x{+}1))$. Clearly, an embedding setting a special marker in cell 0 violates this requirement.

**Open problem 10.** We note in passing that in addition Hertling's notion of simulation is not compatible with that used by Durand-Lose, and that it is an open problem whether all of the differences are really necessary in order to be able to realize a simulation of irreversible CA on reversible CA for infinite configurations.

We now turn to the universality construction mentioned above and have a closer look at spatially period configurations.

**Definition 11.** *A configuration $c$ is spatially periodic or simply periodic*[1] *if there is a positive integer $p \geq 1$ such that $\sigma^p(c) = c$. If $\ell$ is the smallest such integer we also say that $c$ is $\ell$-periodic.*

It should be clear that the application of a global CA function $F$ maps configurations with period $p$ to configurations with period $p$ again. The smallest positive period may decrease from $\ell$ to a divisor of $\ell$.

**Observation 12.** If $\sigma^p(c) = c$ and $E(\sigma(c)) = \sigma^m(E(c))$, then $E(c) = E(\sigma^p(c)) = \sigma^{pm}(E(c))$. I. e., under weak morphisms periodic configurations have to be embedded into periodic ones and *the period can increase by at most a constant factor* $(m)$.

For the moment fix some arity $n$ and consider all $\mathcal{A}_i$ with local structures $(f_i, (1, 2, \ldots, n))$ of this arity. There are $a^{a^n}$ different such CA.

We will use a periodic configuration $c_n$ which as its building block uses a de Bruijn sequence of length $a^n$. That is, the $a^n$ subwords of length $n$ starting at positions 1, 2, ..., $a^n$ in $c_n$ are pairwise different (and their union is $Q_A^n$). Configuration $c_n$ has smallest positive period $a^n$. Hence, according to Observation 12 $c'_n = E(c_n)$ has a smallest positive period of at most $ma^n$.

We are interested in running some CA $\mathcal{B}_i$ with $(g, \nu'_i)$ on the periodic configuration $c'_n$. In general the following holds:

[1] We will not consider periodicity in time in this paper.

**Lemma 13.** *Let $c$ denote an $\ell$-periodic configuration for some CA $\mathcal{B} = (g, \mu)$ and let $\mathcal{B}' = (g, \mu')$ where $\mu'$ denotes the neighborhood with $\mu'(i) = \mu(i) \bmod \ell$ for all $i$. Then for all $t \geq 0$ holds:*

$$\mathcal{B}^t(c) = \mathcal{B}'^t(c) \ .$$

This is so since $c$ and all of its successor configurations have period $\ell$ and hence it does not matter if the position of a neighbor is shifted by a multiple of $\ell$.

**Theorem 14.** *If the embedding is a weak morphism, it is impossible to achieve universality as described in Subsection 5.1.*

*Proof.* Configuration $c_n$ has been chosen such that all $a^{a^n}$ different CA produce $a^{a^n}$ different successor configurations. Assume that $c'_n$ is $\ell'$-periodic. According to Lemma 13 there are at most $\ell'^k$ different global behaviors possible for the $\mathcal{B}_i$ when starting from $c'_n$. Thus one needs $\ell'^k \geq a^{a^n}$. Writing $a'$ for $a^{1/k}$ and denoting the minimum positive period of $c_n$ by $\ell$ this means $\ell' \geq (a')^\ell$. In other words the periods of configurations have to grow exponentially in $\ell$. Since we want to be able to simulate all CA $\mathcal{A}_i$, one can choose $n$ sufficiently large and increase the gap between $\ell$ and $\ell'$ arbitrarily. Comparing this with Observation 12 immediately gives the claim. □

**Open problem 15.** The embedding described in Subsection 5.1 does more than "significantly" increasing the period length of periodic configurations. It completely breaks spatial periodicity. It is an open problem whether that necessarily has to be the case.

For the sake of a simple proof of Theorem 14 we have assumed that one wants to be able to simulate *all* CA. Looking at the proof one can see that we only used the fact for each of the neighborhoods $(1, 2, \ldots, n)$ we used all local functions of arity $n$. But in fact somewhat weaker conditions suffice: For example the number $m_n$ of local functions using such a neighborhoods does not have to be $a^{a^n}$. One only needs that the condition $\ell'^k \geq m_n$ implies that $\ell'$ grows more than linearly in $\ell$. To this end $\sqrt[k]{m_n} \in \omega(a^n)$ is sufficient, e. g. $m_n \geq (a^n)^{2k}$.

## 6 Summary and Outlook

We have shown that "factoring out" relations by permutation different reduced local structures are exactly those which give rise to different global CA functions.

The universality contruction shows what can be achieved by changing the neighborhood if the local function is chosen carefully. We have proven that in order to achieve universality, the embedding cannot ever be a weak morphism.

It remains the main open problem for future research to find ways of characterizing at least partially how much can possibly happen for local functions if they are not designed for a specific task, but coming from a non-trivial set of local functions.

## References

1. Durand-Lose, J.O.: Reversible space-time simulation of cellular automata. Theoretical Computer Science 246, 117–129 (2000)
2. Hertling, P.: Embedding cellular automata into reversible ones. In: Calude, C.S., Casti, J., Dinneen, M.J. (eds.) Unconventional Models of Computation, pp. 243–256. Springer, Heidelberg (1998)
3. Nishio, H.: Changing the neighborhood of cellular automata. In: Durand-Lose, J., Margenstern, M. (eds.) MCU 2007. LNCS, vol. 4664, pp. 255–266. Springer, Heidelberg (2007)
4. Worsch, T., Nishio, H.: Achieving universality of CA by changing the neighborhood. Journal of Cellular Automata (accepted for publication)
5. Worsch, T., Nishio, H.: Variations on neighborhoods. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 581–588. Springer, Heidelberg (2007)

# Error Investigations in Complex Automata Models for Reaction-Diffusion Systems

Alfonso Caiazzo[1], Jean Luc Falcone[2],
Bastien Chopard[2], and Alfons G. Hoekstra[1]

[1] Section Computational Science, University of Amsterdam, The Netherlands
{acaiazzo,alfons}@science.uva.nl
[2] Computer Science Department, University of Geneva, Switzerland
{Jean-Luc.Falcone,Bastien.Chopard}@cui.unige.ch

**Abstract.** Complex Automata (CxA) have been recently introduced as a paradigm to simulate multiscale systems as a collection of generalized Cellular Automata on different scales. We present a basic mathematical framework to investigate the behavior of a CxA model depending on scale separation and modeling choices. In particular, a simple CxA model for a reaction-diffusion system, based on the lattice Boltzmann method, will be formally analyzed. Theoretical error estimates will be derived and numerically validated.

**Keywords:** Complex Automata modeling, reaction-diffusion, lattice Boltzmann method, asymptotic expansion.

## 1 Introduction

Complex Automata (CxA) are a recently proposed paradigm for simulation of multiscale systems [5,6,7,10]. A CxA is based on the idea that an algorithm designed to simulate a process spanning a wide range of temporal and spatial scales (a *multiscale algorithm*), can be replaced by a collection of *single scale models*, simulating relevant sub-processes of the original system. Addionally, these single scale algorithms are constrained to Cellular Automata (CA), lattice Boltzmann methods (LBM), or Agent-Based models (ABM). These algorithms (which can be regarded as *generalized CA*), have in common a specific structure of the update rule which allows computationally efficient implementations (see [3,7] for more details and other properties of the CxA approach). This procedure improves the performance of the multiscale simulations, at the expense of loss of accuracy.

Aim of this paper is to introduce a formalism for the CxA modeling and investigate the *scale-splitting error*, i.e. the difference between the numerical solution obtained using a single algorithm based on fine time and space discretizations and the numerical solution obtained using a CxA model.

We focus on a simple benchmark, based on a reaction-diffusion system solved with the lattice Boltzmann method, which is introduced in section 2. A CxA model based on an operator splitting approach will be constructed. In section 3 we discuss error definitions and estimates, while numerical validations are shown in section 4.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 260–267, 2008.

## 2 Complex Automaton for a Reaction-Diffusion System

We consider a reaction-diffusion process for a concentration field $\rho(t,x)$

$$\begin{aligned} &\partial_t \rho = d\partial_{xx}\rho + \kappa(\rho_\lambda - \rho),\ t \in (0, T_{end}],\ x \in (0, L] \\ &\rho(0,x) = \rho_0(x) \end{aligned} \tag{1}$$

with periodic boundary conditions in $x$, initial condition $\rho_0$, $\rho_\lambda(x)$ being a given function. With $\rho_\lambda(x) = \sin(\lambda x)$, for $\frac{\lambda}{2\pi} \in \mathbb{Z}$, (1) has the analytical solution:

$$\rho^*_\lambda(t,x) = \exp(-(4d\pi^2 + \kappa)t)\sin(2\pi x) + \frac{\kappa}{d\lambda^2 - \kappa}\sin(\lambda x)\,. \tag{2}$$

To consider a multiscale model, we assume the reaction to be characterized by a typical time scale faster than the diffusion. Assuming problem (1) to be written in non-dimensional units, it corresponds to have $\|\kappa\| \gg \|d\|$.

**Lattice Boltzmann Method.** To solve numerically (1) we employ a lattice Boltzmann method (LBM) (see [1,2,9] and the references therein). The space interval is discretized with a regular grid of step size $\Delta x_h = h$, which we denote as $\mathcal{G}_h = \{0, \ldots, N_x - 1\}$. We use the so-called D1Q2 model [9], a one-dimensional scheme where each grid node $j \in \mathcal{G}_h$ holds two particle distributions, $\hat{f}_1$ and $\hat{f}_{-1}$ representing the population densities moving forward and backward, with discrete velocities $c_i \in \{-1, 1\}$. We denote the numerical solution with $\hat{\mathbf{f}} = \left(\hat{f}_1, \hat{f}_{-1}\right)$. The update rule has the form

$$\hat{f}_i^{t_{n+1}}(j + c_i) = \hat{f}_i^{t_n}(j) + \frac{1}{\tau}(f_i^{eq}(\hat{\rho}^{t_n}(j)) - \hat{f}_i^{t_n}(j)) + h^2\frac{1}{2}R(\hat{\rho}^{t_n}(j)),\ i = 1, -1\,, \tag{3}$$

where $R(\rho) = \kappa(\rho_\lambda - \rho)$ and $\hat{\rho} = \rho(\hat{\mathbf{f}}) := \hat{f}_1 + \hat{f}_{-1}$ is the numerical solution for the concentration field. Time steps are indexed by $n \in \mathbb{N}_0$ and have length $\Delta t_h$ such that, for all $h$:

$$\frac{\Delta t_h}{\Delta x_h^2} = \text{const.} \tag{4}$$

The equilibrium distribution in (3) is defined as $f_i^{eq}(\rho) = \frac{\rho}{2}$, for $i = 1, -1$.

The parameter $\tau$ is chosen according to the diffusion constant [2] as

$$\tau = \frac{1}{2} + d\frac{\Delta t_h}{\Delta x_h^2}. \tag{5}$$

Observe that $\tau$ is independent from $h$ in virtue of (4). It can be shown [2,8] that the above described algorithm yields a second order accurate approximation of the solution of (1).

The variable $\hat{\mathbf{f}}_h$ is an $h$-grid function, i.e. a real-valued function defined on a discrete grid: $\hat{\mathbf{f}}_h : \mathcal{G}_h \to \mathbb{R}^2$. Introducing the set $\mathcal{F}_h = \left\{\phi : \mathcal{G}_h \to \mathbb{R}^2\right\}$ we have $\hat{\mathbf{f}}_h \in \mathcal{F}_h$. Moreover, using the subscript $h$ for the operators acting from $\mathcal{F}_h$ to

itself, we introduce a *propagation* operator $P_h$, which acts on a grid function shifting the value on the grid according to $c_i$:

$$\left(P_h \hat{\mathbf{f}}_h\right)_i (j) = \hat{f}_{i,h}(j - c_i),$$

and the (local) reaction and diffusion operators $\Omega_{R_h}$ and (resp.) $\Omega_{D_h}$, defined in the right hand side of (3):

$$\left(\Omega_{R_h} \hat{\mathbf{f}}_h\right)_i = h^2 \frac{1}{2} R(\rho(\hat{\mathbf{f}}_h)), \quad \left(\Omega_{D_h} \hat{\mathbf{f}}_h\right)_i = \frac{1}{\tau}(f_i^{eq}(\rho(\hat{\mathbf{f}}_h)) - \hat{f}_{i,h}),$$

Finally, denoting with $I_h$the identity on $\mathcal{F}_h$, we can rewrite (3) in a compact form as a generalized CA-update rule:

$$\hat{\mathbf{f}}_h^{t_{n+1}} = P_h(I_h + \Omega_{D_h}(\tau) + \Omega_{R_h})\hat{\mathbf{f}}_h^{t_n} = \Phi_h \mathbf{f}_h^{t_n}, \tag{6}$$

where $\hat{\mathbf{f}}_h$ is the state variable and $\Phi_h = P_h(I_h + \Omega_{D_h}(\tau) + \Omega_{R_h})$ is the update rule.

Note that, since $\Omega_{D_h}$ depends linearly on $f^{eq}\left(\rho(\hat{\mathbf{f}}_h)\right) - \hat{\mathbf{f}}_h$ and $f^{eq}(\Omega_{R_h}(\hat{\mathbf{f}}_h)) = \Omega_{R_h}(\hat{\mathbf{f}}_h)$, we have

$$\forall \hat{\mathbf{f}}_h \in \mathcal{F}_h \,:\, \Omega_{D_h}\Omega_{R_h}(\hat{\mathbf{f}}_h) = 0. \tag{7}$$

Conversely, we observe that

$$\forall \hat{\mathbf{f}}_h \in \mathcal{F}_h \,:\, \Omega_{R_h}(I_h + \Omega_{D_h})(\hat{\mathbf{f}}_h) = \Omega_{R_h}(\hat{\mathbf{f}}_h). \tag{8}$$

*Proof.* It follows observing that $\Omega_{R_h}$ depends only on the moment of $\hat{\mathbf{f}}_h$, and the diffusion operator preserves the concentration, i.e. $\rho\left(\hat{\mathbf{f}}_h + \Omega_{D_h}\hat{\mathbf{f}}_h\right) = \rho(\hat{\mathbf{f}}_h)$.

Equation (7) allows to split the LB algorithm (6) in the equivalent form

$$\hat{\mathbf{f}}_h^{t_{n+1}} = P_h(I_h + \Omega_{D_h}(\tau))(I_h + \Omega_{R_h})\hat{\mathbf{f}}_h^{t_n} = \mathcal{D}_h \mathcal{R}_h \hat{\mathbf{f}}_h^{t_n}, \tag{9}$$

separating reaction $\mathcal{R}_h = I_h + \Omega_{R_h}$ and diffusion $\mathcal{D}_h = P_h(I_h + \Omega_{D_h}(\tau))$.

**CxA Formalism and Scale Separation Map.** We define a Complex Automaton, composed of two Automata $\mathrm{CA}_R$ (reaction) and $\mathrm{CA}_D$ (diffusion), introducing a coarser time step for the diffusion model, i.e.

$$\Delta x_D = \Delta x_R = h, \; \Delta t_D = M \Delta t_R = M h^2 \,, \tag{10}$$

for $M \in \mathbb{N}_0$. In practice, it corresponds to execute $M$ steps of the reaction ($\mathrm{CA}_R$), followed by a single diffusion step[1]. In this case the CxA model is analogous to an operator splitting approach [1].

[1] In this paper, we focus on time coarsened diffusion. Depending on the problem, the diffusion process can be coarsened both in space and time. With a more general formulation and more complicated calculations, the results we will present can be generalized to this case as well [4].

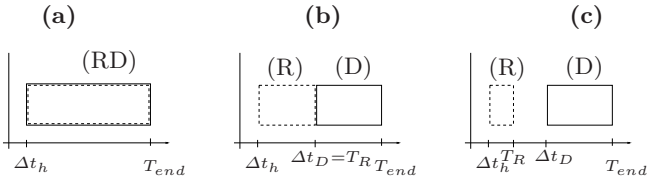



**Fig. 1.** The SSM for the reaction-diffusion problem. In **(a)** reaction (dashed line) and diffusion (solid line) are considered as a single multiscale algorithm. In **(b)** the diffusion time step $\Delta t_D$ is larger than the original $\Delta t_h$. **(c)** The two processes are completely separated in time, e.g. when a very fast reaction yields an equilibrium state in a time $T_R \ll \Delta t_D$.

The information regarding time and space scales of the single scale models, and the way these sub-processes are coupled together is contained in the so-called Scale Separation Map (SSM) [6,7] (shown in Fig. 1 for the considered example). Each single scale model of the CxA (in this case a $\mathrm{CA}_R$ and $\mathrm{CA}_D$) defines a box on the SSM, whose edges depend on the discrete resolutions, i.e. $\Delta t_R = \Delta t_h = h^2$ for the reaction and $\Delta t_D = Mh^2$ for the diffusion, and on the temporal domains. The reaction is run up to a time $T_R$, then re-initialized after a diffusion time step. If $T_R = \Delta t_D$ (Fig. 1b), the two processes are not completely separated. Fig. 1c sketches the case when reaction leads very quickly to an equilibrium state in a typical time which is even smaller than the discrete time step of the diffusion.

To describe formally the CxA model, we introduce a vector of parameters $H = (h_R, h_D)$, to which the corresponding discretizations $(\Delta x_R, \Delta t_R)$ and $(\Delta x_D, \Delta t_D)$, present in the CxA model, are related.

Denoting with $t_{R,n_R}$ and $t_{D,n_D}$ the time steps (with $t_{m,n_m} = n_m \Delta t_m$, for $m = R$ or $D$), of reaction and diffusion algorithms, we describe the evolution of the system with state variable $\hat{\mathbf{f}}_H = \left(\hat{\mathbf{f}}_{R,h_R}, \hat{\mathbf{f}}_{D,h_D}\right)$, whose components are updated according to

$$\begin{array}{ll} (\mathrm{CA}_R) & (\mathrm{CA}_D) \\ \hat{\mathbf{f}}^{t_{R,0}}_{R,h_R} = \hat{\mathbf{f}}^{t_{D,n_D}}_{D,h_D}, & \hat{\mathbf{f}}^{0}_{D,h_D} = \hat{\mathbf{f}}^{init}_{D,h_D}(\rho_0), \\ \hat{\mathbf{f}}^{t_{R,n_R+1}}_{R,h_R} = \mathcal{R}_{R,h_R} \hat{\mathbf{f}}^{R,t_{n_R}}_{R,h_R}, \; n_R = 0,\ldots,M-1 & \hat{\mathbf{f}}^{t_{D,n_D+1}}_{D,h_D} = \mathcal{D}_{D,h_D} \hat{\mathbf{f}}^{t_{D,n_D}+M\Delta t_R}_{D,h_D}. \end{array} \tag{11}$$

A few comments are necessary to explain the formalism introduced in (11). Initial conditions for the algorithm $\mathrm{CA}_R$, are obtained taking the results after a certain interation of $\mathrm{CA}_D$. As next, $M$ iterations are performed, according to an update rule depending only on the reaction process. On the right, the new state of the diffusion Automaton $\mathrm{CA}_D$ is computed starting from the output state of $\mathrm{CA}_R$. With $\hat{\mathbf{f}}^{init}_{D,h_D}(\rho_0)$ we denote the initial condition, which is function of the initial concentration in (11)[2].

[2] In more general cases, where both coarser grid in space and time are used, we need to introduce opportune mappings between the discrete spaces, like a projection (fine to coarse) and a lift (coarse to fine) operators.

Note also that in (11) we have introduced the operators $\mathcal{R}_{R,h_R}$ and $\mathcal{D}_{D,h_D}$, which depend on $\Delta x$ and $\Delta t$ of the different algorithms. In the considered case, $\Delta x_{D,h_D} = \Delta x_{R,h_R} = h$ yields $\mathcal{R}_{R,h} = \mathcal{R}_h$. The coarsening in time affects the diffusion operator, since it depends on $\tau$, which must be modified according to (5). If $\Delta t_{D,h} = M\Delta t_h$,

$$\tau_D = \frac{1}{2} + Md\,. \tag{12}$$

Finally, observe that this example is a special case where the two processes act on the same variable, and it is possible to write the algorithm only depending on $\hat{\mathbf{f}}_D$ as

$$\hat{\mathbf{f}}_{D,h}^{t_{n_D+1}} = \mathcal{D}_{D,h}\mathcal{R}_h^M \hat{\mathbf{f}}_{D,h}^{t_{D,n_D}}\,. \tag{13}$$

## 3 Error Investigations

A CxA model replaces a single multiscale algorithm ($\mathbf{A}_h$) related to a discretization $h$, with a collection ($\mathbf{CxA}_H$) of coupled single scale models, depending on coarser discretizations defined by $H$. The simplification in terms of complexity yields an improvement of the performance, which is paid by a possible loss of precision. Focusing on the benchmark (1), the algorithm ($\mathbf{A}_h$) is defined by (9) (a standard lattice-Boltzmann scheme), while $\mathbf{CxA}_H$ corresponds to the particular operator-splitting approach (11). We are interested in quantifying the difference between the numerical solutions of ($\mathbf{A}_h$) and (CxA$_H$), which we call *scale-splitting* error [4].

A formal definition of the scale-splitting error depends in general on the particular problem and on the specific quantities of interests. In this case, since our algorithm is designed to approximate the variable $\rho$, we define the scale-splitting error at time iteration $t_N$ taking the difference (after both reaction and diffusion have been executed) between $\rho(\hat{\mathbf{f}}_h)$ (numerical solution of the fine-grid algorithm (9)) and $\rho(\hat{\mathbf{f}}_{D,h_D})$ (the output of the CxA model (11)):

$$E(\rho; M, t_N) = \left\| \rho\left(\hat{\mathbf{f}}_h^{t_N}\right) - \rho\left(\hat{\mathbf{f}}_{D,h_D}^{t_N}\right) \right\|\,. \tag{14}$$

Note that the scale-splitting error is a measure of a difference between discrete systems (the two algorithms).

**Differences Between Discrete Operators.** In the following, we will use $h_R = h_D = h$. Since $\hat{\mathbf{f}}_h$ is the solution of (9) and $\hat{\mathbf{f}}_{D,h}$ is obtained from (11) (in the form (13)) we rewrite (14) as

$$\begin{aligned} E(\rho; M, t_N) &= \left\| \rho\left( \left(\mathcal{D}_h\mathcal{R}_h\right)^M \hat{\mathbf{f}}_h^{t_N - M\Delta t_h} - \mathcal{D}_{D,h}\mathcal{R}_h^M \hat{\mathbf{f}}_{D,h}^{t_N-\Delta t_D} \right) \right\| \\ &\leq \left\| \rho\left( \left(\mathcal{D}_h\mathcal{R}_h\right)^M - \mathcal{D}_{D,h}\mathcal{R}_h^M \right) \mathbf{f}_{D,h}^{t_N-\Delta t_D} \right\| + C(h,\mathcal{D},\mathcal{R}) E(\rho; M, t_N - \Delta t_D) \end{aligned} \tag{15}$$

(using $\mathcal{R}_{R,h} = \mathcal{R}_h$ and $t_N - M\Delta t_h = t_N - \Delta t_D$), where $C(h,\mathcal{D},\mathcal{R})$ is an opportune constant depending on the reaction and diffusion operators and $E(\rho; M, t_N - \Delta t_D)$ is the scale-splitting error at the previous iteration.

From (15), we conclude that the distance between the numerical solutions can be estimated by measuring the distance between the algorithms (in an operator-norm)

$$\begin{aligned} E^{A-\mathrm{CxA}}(M) :=& \left\| \rho \left( \left(\mathcal{D}_h \mathcal{R}_h\right)^M - \mathcal{D}_{D,h} \mathcal{R}_h^M \right) \right\| \\ &\leq \left\| \rho \left( \left(\mathcal{D}_h \mathcal{R}_h\right)^M - \mathcal{D}_h^M \mathcal{R}_h^M \right) \right\| + \left\| \rho \left( \left(\mathcal{D}_h^M - \mathcal{D}_{D,h}\right) \mathcal{R}_h^M \right) \right\| = \\ &= E^{(1)}(M) + E^{(2)}(M) \,. \end{aligned} \tag{16}$$

To analyze (16) we use an argument based on asymptotic analysis [8]. First, we assume that the numerical solution (a $h$-grid function) can be approximated by an ansatz of the form

$$\hat{\mathbf{f}}_h^{t_n}(j) \approx \mathbf{f}_h(t_n, x_j) = \mathbf{f}^{(0)}(t_n, x_j) + h\mathbf{f}^{(1)}(t_n, x_j) + h^2 \mathbf{f}^{(2)}(t_n, x_j) + \ldots, \tag{17}$$

where $\mathbf{f}_h(t, x)$ is an asymptotic expansion with smooth and $h$-independent coefficients. Estimates of (16) can be derived applying the different discrete operators to the ansatz (17) and using the smoothness hypothesis.

**Commutation Error.** The contribution $E^{(1)}$ can be estimated by a function of $[\mathcal{D}_h, \mathcal{R}_h] = \mathcal{D}_h \mathcal{R}_h - \mathcal{R}_h \mathcal{D}_h$, i.e. the commutator of the operators $\mathcal{R}_h$ and $\mathcal{D}_h$. It can be shown that, for any function $\mathbf{f}_h$ in the form (17)

$$[\mathcal{D}_h, \mathcal{R}_h](\mathbf{f}_h) = O\left(h^3 \kappa \partial_x (\rho(\mathbf{f}_h) - \rho_\lambda)\right) \,. \tag{18}$$

*Informal proof:* Using the properties (7)-(8), the commutator involves only the exchange of the operations $P_h$ and $I_h + \Omega_{Rh}$. Assuming the smoothness of the coefficients of $\mathbf{f}_h$, the difference can be estimated replacing the shifting operated by $P_h$ with a spatial derivation, and using $\partial_x R(\rho) = -\kappa \partial_x (\rho - \rho_\lambda)$ [4].

By counting the number of times the commutator appears in the difference $\left(\mathcal{D}_h \mathcal{R}_h\right)^M - \mathcal{D}_h^M \mathcal{R}_h^M$, we have (by induction)

$$\left\| \left(\mathcal{D}_h \mathcal{R}_h\right)^M - \mathcal{D}_h^M \mathcal{R}_h^M \right\| \leq M(M-1) \left\| [\mathcal{D}_h, \mathcal{R}_h] \right\| = O(M^2 h^3 \kappa) \,. \tag{19}$$

**Time-Coarsening Error.** The term $E^{(2)}$ in (16) derives from the time-coarsening of the diffusion part of the original lattice Boltzmann algorithm. Starting from ansatz (17) and analyzing the operators $\mathcal{D}_h^M(\tau)$ and $\mathcal{D}_{D,h}(\tau_D)$ (defined as in (12)) we obtain [4].

$$\rho \left( \mathcal{D}_h^M - \mathcal{D}_{D,h}(\tau_{(M)}) \right) \mathbf{f}_{D,h_D} = O(M^2 d^3 h^2) \,. \tag{20}$$

In conclusion, (19)-(20) yield

$$E^{A-\mathrm{CxA}}(M) = O\left(M^2 \kappa\right) + O\left(M^2 d^3\right) \,. \tag{21}$$

## 4 Numerical Results

Considering problem (1), we are primarily interested in the difference between the simulation results $\hat{\mathbf{f}}_h$ of algorithm (9) and $\hat{\mathbf{f}}_{D,h}$ of (11). In the following

numerical tests, both simulations are performed, evaluating the scale-splitting error (depending on the simulation parameters) as

$$E(\kappa; d; \lambda; M) = \max_N \left\{ \frac{1}{N_x(h)} \left\| \rho\left(\hat{\mathbf{f}}_h^{t_N}\right) - \rho\left(\hat{\mathbf{f}}_{D,h}^{t_N}\right) \right\|_1 \right\}. \tag{22}$$

To "measure" the scale separation in a particular setting, we introduce the additional non-dimensional parameter $\sigma = \frac{\kappa}{\lambda^2 d}$ (where $\lambda$ determines $\rho_\lambda(x)$ in (1)).

In Fig. 2a we show the results for the scale-splitting error as a function of $M$, fixing $h$ and $\kappa$, for different values of $d$. This choice is due to the fact that the grid size is related to the reaction rate for stability reasons. In general, for better scale separation (larger values of $\sigma$) we obtain lower scale-splitting error. The order

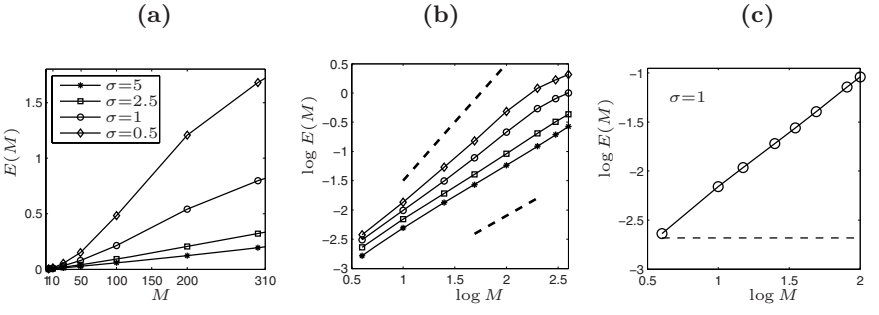



**Fig. 2.** **(a)**: Scale-splitting error (22) as a function of $M$ for different values of $\sigma$ ($h = 0.02$, $\lambda = 4\pi$, $\kappa = 10$, $d \in \{0.5, 1, 2.5, 5\}$). **(b)**: Fig. (a) in double logarithmic scale, showing the maximum scale-splitting error versus $M$. The dashed lines show reference slopes 1 (bottom) and 2 (top). **(c)**: Error (22) (range of moderate $M$), compared to the error of the original (fine-discretized) algorithm (3) (dashed line) with respect to the exact solution (2).

plot in Fig. 2b shows that the size of the splitting error increases superlinearly for moderate values $M$ (see (21)), becoming linear when $M$ increases too much. The transition starts earlier for larger $\sigma$ (if $\kappa$ is bigger with respect to $d$). Due to the simple derivation, (21) overestimates the errors. A deeper analysis [4] can be used to explain this effect.

Fig. 2c compares the scale-splitting error as a function of $M$ (for moderate values) with the error of the original LBM (9) with respect to the exact solution (2). We note that for small $M$, the scale-splitting error is of the same order as the discretization error, i.e. the loss of accuracy of the CxA model is comparable with the numerical error produced by the orginal algorithm.

**Conclusions and Outlook.** We proposed a basic formalism to describe mathematically a Complex Automata model for reaction-diffusion. For this problem, we discussed a methodology to analyze the CxA, which combines asymptotic expansion techniques with general properties of the considered discrete systems.

Although we focused on a particular benchmark, we believe that a more general formalism can be constructed, on the basis of the one proposed here. Detailed investigations of different types of CxA multiscale coupling will be topic of a forth-coming research.

**Acknowledgments.** This research is supported by the European Commission, through the COAST project [10] (EU-FP6-IST-FET Contract 033664).

## References

1. Alemani, D., Chopard, B., Galceran, J., Buffle, J.: LBGK Method Coupled to Time Splitting Technique for Solving Reaction-Diffusion Processes in Complex Systems. Phys. Chem. Chem. Phys. 7, 1–11 (2005)
2. Chopard, B., Droz, M.: Cellular Automata Modelling of Physical Systems. Cambridge University Press, Cambridge (1998)
3. Chopard, B., Falcone, J.-L., Razakanirina, R., Hoekstra, A.G., Caiazzo, A.: On the Collision-Propagation and Gather-Update Formulations of a Cellular Automata Rule. In: Umeo, H., et al. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 144–151. Springer, Heidelberg (2008)
4. Caiazzo, A., Falcone, J.L., Chopard, B., Hoekstra, A.G.: Asymptotic Analysis of Complex Automata Models for Reaction-Diffusion Systems (in preparation, 2008)
5. Hoekstra, A.G., Chopard, B., Lawford, P., Hose, R., Krafczyk, M., Bernsdorf, J.: Introducing Complex Automata for Modelling Multi-Scale Complex Systems. In: Proc. of Eur. Comp. Syst. Conf. 2006 (CD), Oxford, UK. Eur. Comp. Syst. Soc. (2006)
6. Hoekstra, A.G., Lorenz, E., Falcone, J.-L., Chopard, B.: Towards a Complex Automata Formalism for Multi-Scale Modeling. Int. J. Mult. Comp. Eng. 5, 491–502 (2007)
7. Hoekstra, A.G., Falcone, J.-L., Caiazzo, A., Chopard, B.: Multiscale Modeling with Cellular Automata: The Complex Automata Approach. In: Umeo, H., et al. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 192–199. Springer, Heidelberg (2008)
8. Junk, M., Klar, A., Luo, L.-S.: Asymptotic Analysis of the Lattice Boltzmann Equation. J. Comp. Phys. 210, 676–704 (2005)
9. Succi, S.: The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, Oxford (2001)
10. The COAST project, http://www.complex-automata.org

# Simulation of the Effect of Intermittent Flow in Polycrystals on the Basis of Cellular Automata and Relaxation Element Method

Ye.Ye. Deryugin[1], G.V. Lasko[1,2], and S. Schmauder[2]

[1] Institute of Strength Physics and Material Science, SB RAS (ISPMS SB RAS), pr. Akademicheskii 2/1, 634021 Tomsk, Russian Federation
[2] Institute for Material Testing, Material Science and Strength of Materials (IMWF) Universität Stuttgart, Pfaffenwaldring 32, D-70569, Germany
Galina.Lasko@mpa.uni-stuttgart.de

**Abstract.** The simulation of the effect of intermittent flow in polycrystals is performed on the basis of cellular automata and relaxation element method. Each cell (representing element of a structure) has an ability to switch its state by discrete jump in stresses (or plastic deformation), prescribed by a definite relaxation element which is put in the center of the cell. Interaction of the field of internal stresses from structural elements, undergone plastic deformation occurs automatically, following the superposition principle. The results are in a good agreement with known experimental findings.

## 1 Introduction

On the macrolevel one can distinguish three types of plastic strain localization: Lüders band propagation, the phenomenon of Portevin-Le Chatelier Effect and the neck formation at the stage of prefracture [1-5]. For the description and simulation of the above effects the more suitable and perspective is the Relaxation Element Method (REM), developed by the authors [6-10]. The definition of the change in the field of stresses in solid under loading as a result of the decreasing in elastic energy in the local volume, undergone plastic deformation is laid down on the basis of the method. For such a purpose a „tensor of relaxation", characterizing the changing field of elastic stresses in the local volume as a result of its plastic deformation is used.

In the present work the results of the simulation of the plastic strain localization in polycrystalline materials with the effect of Portevin-Le Chatelier is represented.

## 2 The Peculiarities of the Simulation by Relaxation Element Method

The developed models on the basis of REM operate on principle of cellular automata. The calculation fields is divided into a number of cells, playing the role of elements of structure (for example, grains in polycrystals). Each cell has an ability to switch its state by discrete jump in stresses (or plastic deformation), prescribed by a definite

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 268–273, 2008.

relaxation element which is put in the center of the cell. Decrease in the elastic energy in the cell is accompanied by stress increase beyond the cell, where plastic deformation took place. In such a manner, an element of structure is able to increase discretely the degree of plastic deformation and as stress concentrator effect the stress state of the nearest neighbors. The state of all structural elements changes simultaneously in a definite time interval. The involvement of the structural elements into plastic deformation is realized by the setting of criterion, for example critical shear stress at the center of the cell under the influence of external applied stress $\sigma$ (Mises-Tresca - criterion). Interaction of the fields of internal stresses from structural elements, undergone plastic deformation occurs automatically (following the superposition principle).

When simulating the process of crystallites involvement into plastic deformation it was assumed, that plastic deformation of crystallites occurs on the scheme of pure shear. The grain of polycrystal have been represented in the form of RE of round shape, in which the stress relaxation along the conjugate direction at an angle of 45° with respect to tensile axis is realized. The field of internal stresses of such a defect is characterized by stress of pure shear [10]

$$\tau(x,y)=\Delta\sigma\begin{cases}\dfrac{(\beta+1)a^2}{(\beta+3)r^2}\left[\dfrac{3(\beta+3)a^2}{(\beta+5)r^2}-2\right]\left[1-8\left(1-y^2/r_2\right)y^2/r_2\right], & \text{if } r^2\ge a^2,\\ -1+\left(\dfrac{r}{a}\right)^{\beta+1}\left\{\dfrac{\beta^2-1}{2(\beta+3)(\beta+5)}\left[1-8\left(1-y^2/r_2\right)y^2/r_2\right]+1\right\}, & \text{if } r^2\le a^2\end{cases}. \qquad (2.1)$$

Here $r$ - is the distance from the center of the family of RE to the point with the coordinates $(x, y)$, $a$ - is the radius of RE, $\Delta\sigma$ defines the value of stress relaxation of the pure shear inside the RE, $\beta$ - defines the value of the gradients of plastic deformation. The more $\beta$, the higher are the gradients. Shown in Fig. 1 are the distributions of the inhomogeneous stress fields of shear stresses $\tau(a)$ and shear strain $\gamma^p$ $(b)$ at $\beta = 6$.

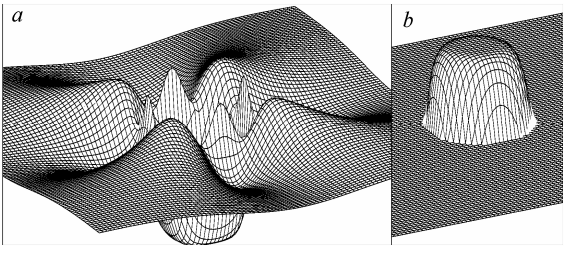



**Fig. 1.** Distributions of stress $\tau(a)$ and shear strain $\gamma^p$ $(b)$: $h = 0$, $\beta = 6$

Each act of element involvement into plastic deformation matches a definite time interval which we defined from physical reasoning. So, for example in the macroband of polycrystals Al-$Al_2O_3$ with the dimension of crystallite being equal to 40 μm there is not less than $4x10^3$ grains. The band is formed for 1-2s. Hence, a separate grain is involved into plastic deformation in the time of order $dt = 5x10^{-4}$s.

In this work with accounting of the rigidity of the testing device the increment of the external stress $d\sigma$ in the time interval $dt$ during which the stress relaxation in the value $\Delta\sigma$ occurs has been calculated. The following equation has been used

$$d\sigma = \frac{\mathrm{M}(v_0 \mathrm{E}\mathrm{b}dt - 3\pi a^2 \Delta\sigma)}{(\mathrm{SE} + \mathrm{M}l_0)\mathrm{b}}. \tag{2.2}$$

Here M - is the rigidity of the testing machine $v_0$ - is the velocity of the clamps of testing machine in unloaded state, $a$ - is the radius of crystallite, S, $l_0$ and b - are the cross-section, length and the width of the working part of the specimen correspondently.

The involvement of a separate grain into plastic deformation takes place when in its center the $\tau$ exceeds the critical shear stress $\tau_{cr}$, which for example one can connect with the stress of dislocation unpinning [1-3, 10]. Parameter $\Delta\sigma$ include in itself the mechanisms of plastic deformation, i.e. the ability of the material to plastic formchanging

$$\Delta\sigma = 2(\tau_{max} - \tau_0),$$

where $\tau_{max}$ - is the maximum shear stress in the direction at an angle of 45° with respect to tensile axis (according to Mises-Tresca criterion), $\tau_0$ - is the stress of plastic flow, which , for example can be connected with the resistance to the free movement of dislocations. $\sigma$-$\varepsilon$ diagram have been constructed by sum of the $d\sigma$ - increment.

Let us represent the update rules of stress state of structural elements in the given model of cellular automata:

1. In the initial state the structural elements in the calculation field are identical and obeys the law of elastic deformation. At this stage in the equation (2.2) left term in brackets, which is responsible for the contribution of plastic deformation is equal zero. In the modeled specimen exists a homogeneous stress field. At the stage of elastic deformation the increase in external stress is equal to

$$d\sigma = \frac{\mathrm{E}v_0 dt}{\mathrm{SE/M} + l_0}. \tag{2.3}$$

   The discrete increasing in external stress in the value (2.3) takes place till the instant of time, when external stress $\sigma$ doesn't exceed the $2\tau_{cr}$. value
2. At $\sigma > 2\tau_{cr}$ the coordinates of the point are prescribed, in which the relaxation element with its own field of internal stresses is placed (2.1). The value $a$ characterizes the dimension of structural element or RE. In the given model $a$ was taken to be equal one half of the distance between calculation points. The value $\Delta\sigma$ is defined according to the equation $\Delta\sigma = 2(\tau_{max} - \tau_0)$, where $\tau_0 < \tau_{cr}$. Since all elements are identical, then the coordinates of the first point are defined arbitrarily. In such a manner, simultaneously the change in the state of all elements of structure in a time interval $dt = 5\mathrm{x}10^{-4}$s takes place. The stress state in the calculation points starting from this instant of time became not identical.
3. Further the procedure is realize in cycle. In the changed stress field the coordinates of the point are defined, in which the maximum value of shear stress is observed in the direction of 45° with respect to the tensile axis. If $\tau_{max} > \tau_{cr}$, than the increase of the external stress is calculated according to the formulae (2.2) and a new relaxation element is placed in the given point. If $\tau_{max} < \tau_{cr}$, then the increment is

calculated following the formulae (2.3) and the internal stresses are summed until the situation arises when $\tau_{max} > \tau_{cr}$. And only after that in the given point the relaxation element is placed.

## 3 Results

The influence of the rigidity of the machine on the loading curve of the polycrystals of low-carbon steel with E = 210000 MPa is represented in Fig. 2. The curve 1 for «soft» mode of loading (M = $1.3x10^2$ kN/mm) has a stair-case type. As the rigidity of the machine M increases from $1.3x10^2$ to $1.3x10^8$ kN/mm the curve takes more serrated type. The flow tooth arises and increases. The amplitude of the oscillations of the external stress increases. At all curves the flow plateau is observed, then comes the stage of strain hardening, caused by the increasing of the stress fields from relaxation elements. Yield plateau is formed on the mechanism of Lüders band propagation, when the crystallites are involved into plastic deformation consequently filling the working part of the specimen. Repeated involvement takes place at the higher external applied stress.

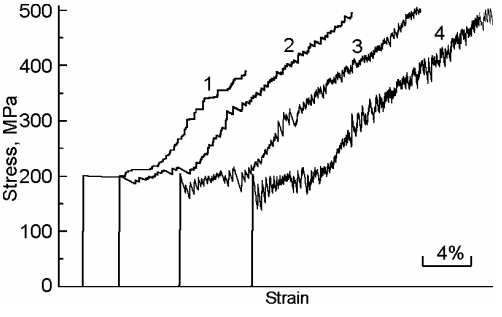



**Fig. 2.** The influence of the rigidity of the machine M on the type of loading curves: M (kN/mm) = 1.3 x102 (1), 1.3 x103 (2), 1.3 x105(3), 1.3 x108(4)

Loading rate effects essentially the $\sigma$–$\varepsilon$ curves. The less the rate of loading, the more pronounced is the effect of intermittent flow (Fig.3). Increase in the rate of loading result in decrease in the amplitude of oscillations of external stress. Starting from the definite rate of loading there is no oscillations on the loading curves. (curve5). Further increase in the rate of loading results in the decreasing of "tooth" and then the yield plateau (Fig.3). The flow stress increases and the effect of sharp yield flow disappears (curves 7 and 8 in Fig.3).

Along with the influence of the rigidity of the machine and loading rate which defines the boundary conditions of loading, the influence of the $\tau_{cr}$ characteristics of the material itself has been considered under the same other parameters of the model. Shown in Fig.4 are the $\sigma$–$\varepsilon$ - curves for the different $\tau_{cr}$ values. If the dislocations are not pinned by atmospheres of admixtures (when $\tau_{cr} = \tau_0$), then the phenomenon of interrupted flow is not observed (low curve). As the $\tau_{cr}$ increases the Portevin-Le Chatelier effect appears and enhances. At that time the flow stress, yield flow and amplitude of peaks at loading are increased on loading diagram.

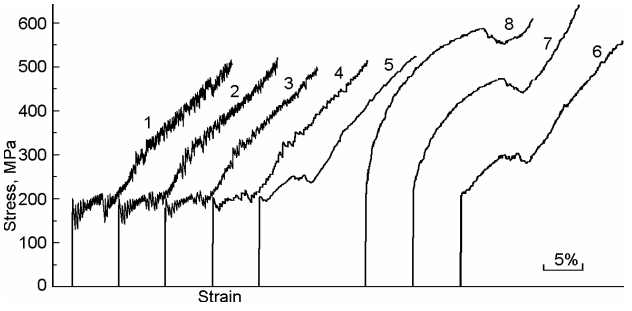



**Fig. 3.** The influence of the applied rate of the movement of the clamps on the type of loading diagrams: $v_0$, mm/s = 0.001 (1), 0.01 (2), 0.02 (3), 0.03 (4), 0.04 (5), 0.05 (6), 0.08 (7), 0.1 (8)

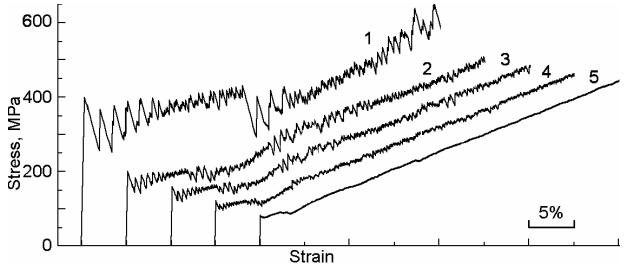



**Fig. 4.** The influence of $\tau_{cr}$ stress on the type of loading curves $\tau_{cr}$ = 200(1), 100(2), 80(3), 60(4), 40(5)

The performed simulation allowed revealing qualitative and quantitative changes of loading curve depending on the characteristics of material itself and boundary conditions of loading. The obtained regularities of the quantitative and qualitative changes in loading diagrams of polycrystals are in a good agreement with known experimental data [1, 3, 12-14].

## 4 Conclusions

In the present work the important result, testifying to the high predictive power and advantages of the simulation of plastic deformation on the basis of cellular automata in the form of relaxation elements are obtained. They are in a good agreement with known experimental findings:

- Effect of intermittent flow is the consequence of the formation of meso- and macrobands of localized deformation;
- In the changing field of stresses the jump-like movement of the process of the localization of deformation along the gauging section of the specimen takes place. Increase in the loading rate suppresses the effect of intermittent flow;
- Saw-tooth type of the $\sigma$-$\varepsilon$ curves is typical for the rigid mode of loading (device with the high value of modulus of rigidity), in the soft mode of loading the $\sigma$-$\varepsilon$ curves have stair case type;

The following peculiarities of the development of strain localization have been elucidated:

- The formation of the macroband occurs on the mechanism of Lüders band propagation and is accompanied by decreasing in the external stress
- The structure of the formed macroband is consists of a number of mesobands, oriented along the direction of the maximum tangent stresses.
- Accumulation of the fields of internal stresses in the volume of a polycrystal results in the effect of strain hardening.
- The reason of PLC-effect can be the difference in the stress of the start of dislocation unpinning and the lower stress of their free movement.

**Acknowledgements.** The work is funded by German Scientific Society (DFG), project DFG Schm 746/52-3 and the Russian foundation of fundamental researches, project № 07-08-00144.

## References

1. Estrin, Y., Kubin, L.P.: Spatial Cupling and Propagative Plastic Instabilities. In: Mühlhaus, H.-B. (ed.) Continuum Models of Materials with Microstructure, ch.2, pp. 395–450. John Wiley&Sons Ltd., Chichester (1995)
2. Klose, F.B., Ziegenbein, A., Weidenmüller, J., Neuhäuser, H., Hähner, P.: Portevin-Le Chatelier effect in strain and stress controlled tensile tests. Comput. Mat. Sci. 26, 80–86 (2003)
3. Zhang, Q., Jiang, Z., Jiang, H., Chen, Z., Wu, X.: On the propagation and pulsation of Portevin-Le Chatelier deformation bands: An experimental study with digital specle pattern metrology. Int. J. Plasticity 21, 2150–2173 (2005)
4. Deryugin, E.E., Panin, B.E., Schmauder, S., Storozhenko, I.V.: Effects of strain localization in composites on the basis of Al with Al2O3. Phys. mesomech. 4(3) (2001)
5. Krishtal, M.M.: Instability and mesoscopic inhomogenuity of plastic deformation (analitycal review). Phys. mesomech. V7(5), 5–45 (2001)
6. Deryugin, Ye.: Relaxation element method, Monograph. Novosibirsk, Nauka. Siberian book-publishing firm SB RAS (1998)
7. Deryugin, Ye. Ye., Lasko, G.V.: Relaxation element method in the problem of mesomechanics and calculations of band structures. In: Panin, V.E. (ed.) Physical mesomechanics and computer-aided designing of materials, Novosibirsk, Nauka, vol. 1, pp. 131–161. Siberian book-publishing firm SB RAS (1995)
8. Deryugin, Y.Y., Lasko, G.V., Schmauder, S.: Relaxation element method. Comput. Mat. Sci. 11(3), 189–203 (1998)
9. Lasko, G.V., Deryugin Ye, Y., Schmauder, S.: Plastic Deformation Development in Polycrystals based on Cellular Automata and relaxation Element Method. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 375–384. Springer, Heidelberg (2004)
10. Lasko, G.V., Deryugin Ye., Y., Schmauder, S.: Simulation of the Evolution of Band Structures in Polycrystals on the Basis of Relaxation Element Method and Cellular Automata. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 367–372. Springer, Heidelberg (2006)

# Lattice Gas Automata Simulation of 2D Site-Percolation Diffusion: Configuration Dependence of the Theoretically Expected Crossover of Diffusion Regime

Mehrdad Ghaemi[1,*], Nasrollah Rezaei-Ghaleh[2,3], and Yazdan Asgari[2]

[1] Chemistry Dept., Tarbiat moallem Univ., Tehran, Iran
ghaemi@tmu.ac.ir
[2] Center for Complex Systems Research, K.N. Toosi Univ. Technology, Tehran, Iran
yazdan1130@gmail.com
[3] Inst. Biochem. Biophys., Univ. Tehran, Tehran, Iran
nrezaeig@ibb.ut.ac.ir

**Abstract.** Theoretical analysis of random walk on percolation lattices has predicted that, at the occupied site concentrations of above the threshold value, a characteristic crossover between an initial sub-diffusion to a final classical diffusion behavior should occur. In this study, we have employed the lattice gas automata model to simulate random walk over a square 2D site-percolation lattice. Quite good result was obtained for the critical exponent of diffusion coefficient. The random walker was found to obey the anomalous sub-diffusion regime, with the exponent decreasing when the occupied site concentration decreases. The expected crossover between diffusion regimes was observed in a configuration-dependent manner, but the averaging over the ensemble of lattice configurations removed any manifestation of such crossovers. This may have been originated from the removal of short-scale inhomogeneities in percolation lattices occurring after ensemble averaging.

## 1 Introduction

To treat the static and dynamic properties of systems with inherent disorders, theory of percolation has proven useful in a large variety of areas. Biological evolution, protein diffusion in biological membranes, disease epidemics, forest fires and social phenomena are some relatively new examples of the wide applicability of this theory [1-5]. In spite of this, there exist some purely theoretical challenges in the area and much effort has been dedicated to solve them, with theoretical and computational tools [6].

The static and dynamic properties of site percolation lattices have been extensively investigated during the recent decades using theoretical, computational and even experimental methods [7-9]. It is well known that, as the concentration $P$ of the occupied sites approaches a threshold value of $P_c$, an infinite cluster of the occupied sites over which the unbounded diffusion or conduction can take place is formed [10]. For $P>P_c$, the probability of an occupied site to be on the infinite cluster, $P_\infty$ is given by the characteristic exponent $\beta$ through the scaling formula

* Corresponding author.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 274–281, 2008.

$$P_\infty \sim (P\text{-}P_c)^\beta \tag{1}$$

while $P_\infty$ is zero for $P<P_c$ [11]. There is also a percolation correlation length, $\xi$, which for length scales $r \geq \xi$, the percolating lattice appears homogeneous, but for $r<\xi$, it exhibits a self-similar fractal geometry. $\xi$ is zero for $P$=1, but as $P$ approaches $P_c$ from the right side, it diverges as [12]

$$\xi \sim (P\text{-}P_c)^{-\upsilon} \tag{2}$$

For site-percolation lattices with Euclidean dimension of $d$=2, $P_c$, $\beta$ and $\upsilon$ are known to be 0.592746 [13], 5/36 [11] and 4/3 [12], respectively.

Extensive efforts have been made to study the dynamic properties of percolation lattices such as diffusion or conduction over them and relate these properties to the static characteristics of such lattices. It is self-evident that below $P_c$, diffusion and conduction will be ultimately restricted by the perimeter of finite clusters, but at $P_c<P<1$ where $\xi \neq 0$, it has been conjectured [14,15] that the diffusing (or conducting) particle would undergo an anomalous sub-diffusive behavior, obeying the law

$$<R^2>^{1/2} \propto N^k \tag{3}$$

where $<R^2>^{1/2}$ is the ensemble average of the Euclidean displacement, $N$ is the number of time steps, and $k$ is an exponent equal to 1/2 for classical diffusion and less than 1/2 for the sub-diffusion regime. It is believed that for $<R^2>^{1/2} >> \xi$, the diffusing particle will behave classically (i.e. $k$=1/2), but the diffusion coefficient $D$ will scale through the following relation [16]

$$D \sim (P\text{-}P_c)^\mu \tag{4}$$

For $d$=2, the initial value of $k$ occurring when $<R^2>^{1/2} << \xi$ has been numerically calculated as 0.348 [17]. The Alexander-Orbach (AO) conjecture, pointing out that the spectral dimension (defined as $2kD_f$ where $D_f$ is the fractal dimension) is independent of $d$ and equal to 4/3 [18], suggests that the dynamic exponent $\mu$ must be related to the static ones through the following relation:

$$\mu=[(3d-4)\upsilon-\beta]/2 \tag{5}$$

so that for $d$=2, the dynamic exponent $\mu$ should be as 91/72 (i.e. about 1.264) [7,8]. However, the AO conjecture, although remains as a remarkably accurate estimate, is not precisely correct and the true value of spectral dimension is slightly smaller than 4/3 for $d<6$ [8]. Therefore, the dynamic exponent $\mu$ may be different from that proposed by the AO conjecture. In past, a variety of techniques have been employed to estimate $\mu$, including numerical methods, analytical approximations such as series expansions, small cell real-space renormalization technique, ε-expansion method and even experimental techniques and values ranging from 1.20-1.32 have been reported for $\mu$ in $d$=2 [8,16, 19-22]. It is also assumed that the change of diffusion regime occurs after some characteristic crossover time, $\tau$, which diverges at $P=P_c$, so that at the percolation threshold, the diffusion is anomalous for all $N$ [15,16].

This study aimed at simulating random walk on percolation lattices using the lattice gas automata (LGA) approach. LGA are discrete dynamical systems in regard with space, time, and the states of the system. Each point in a regular spatial lattice,

called a cell, can have a finite number of particles. The particles in the lattice move according to a local rule. That is, the movement of a particle at a given time depends only on its own state one time step previously, and the states of its nearby neighbors at the previous time step. All cells on the lattice are updated synchronously. Thus the state of the entire lattice advances in discrete time steps. Many LGA are two-dimensional due to visualization and computational concerns, but higher-dimensional lattices certainly exist [23,24]. Using LGA approach, we explored first how the diffusion coefficient ($D$) depends on the concentration of the occupied sites ($P$) in the lattice and obtained the related exponent of $\mu$. Since the obtained value of $\mu$ was reasonably consistent with the literature [8,16,19-22], we were convinced that our model has grasped the essential features of the studied phenomenon. Thereafter, the nature of diffusion regime and its dependence on $P$ was investigated. More specifically, we examined if and how the theoretically-expected transition between anomalous sub-diffusion and classical diffusion regimes occurred. It was found that the cross-over between two regimes occurs in a configuration-dependent manner and the averaging over the ensemble of configurations removes any manifestation of such crossovers. The time required for crossover to occur was also found to be configuration-dependent.

## 2 Results and Discussion

All simulations were conducted with the Lattice Gas Automata (LGA) of the square 2D lattices of size L=1000. This size is longer than the estimated correlation length of the lattice if $P$ is sufficiently higher than $P_c$ ($P$=0.61 and higher), so the lattice seems to be large enough to represent the theoretically-predicted crossover. Lattices of similar size have been previously employed to manifest such crossovers [e.g. 16]. The conducting sites where the random walker was permitted to enter were introduced randomly in specified concentrations, according to a uniform random distribution. Random numbers were generated by Mersenne Twister (MT19937) algorithm [25]. The walker is initially located at the center of the lattice. At each step, the walker must select one of the occupied sites within its von-Neumann neighborhood with equal probabilities to enter. No waiting time is allowed. The transition rule remains unchanged during the whole process of 150,000 time steps.

For each concentration of conducting sites ($P$), the random walk was simulated over 10000 random configurations and the ensemble average of the squared Euclidean displacement of the random walker ($\langle R^2 \rangle$, where $\langle\rangle$ indicates its ensemble-average) was plotted against time steps ($N$) and the effective diffusion coefficient ($D_{\text{eff}}$) was estimated through calculating the slope of this plot in double linear scale. Fig. 1 displays how $D_{\text{eff}}$ varies as $P$ is changed between 0 and 1. As expected, $D_{\text{eff}}$ was effectively zero at $P$ below a critical value of ($P'_c$) about 0.60 and started to smoothly increase above it. The more accurate examination of the concentration dependence of $D_{\text{eff}}$ between $P$s of 0.50 and 0.60 revealed that $D_{eff}$ was 0.0000 at $P$<0.57, 0.0005 at $P$=0.57, 0.0019 at $P$=0.58, 0.0096 at $P$=0.59 and 0.0193 at $P$=0.60 (see the Inset of Fig.1). It is well known that the percolation threshold $P_c$ is around 0.592746 where the single infinite cluster is first formed [13]. At $P<P_c$, it is expected that the diffusion will ultimately be restricted by the perimeter of finite clusters, so that providing the simulation is long

enough, the diffusion coefficient will be effectively zero. The non-zero value of $D_{eff}$ at $P$=0.57-0.59 may be caused by the insufficiency of 150,000 time steps for the random walker to reach at the boundary of the large finite clusters. Thereafter, considering the dynamic scaling law $D \sim (P\text{-}P'_c)^{\mu}$, $\log(D_{\text{eff}})$ was plotted against $\log(P\text{-}P'_c)$ (Fig. 2) to estimate the value of $\mu$. The calculated $\mu$ was 1.251±0.010, reasonably consistent with the values proposed by Alexander-Orbach conjecture [18] and reported in some references [8,16,19-22]. The obtained value for $\mu$ was found to be nearly independent of lattice size, within the range of $L$=1000-5000.

To investigate if the random walk on the percolation lattice obeyed the classical or anomalous regime of diffusion, the squared Euclidean displacement of the random walker ($\langle R^2 \rangle$) was plotted against time steps ($N$) in double logarithmic scale for each concentration of conducting sites ($P$) above $P'_c$. As manifested in Fig. 3, log $\langle R^2 \rangle$ seems to vary linearly with log $(N)$ at all studied $P$s above $P'_c$, although the slopes of

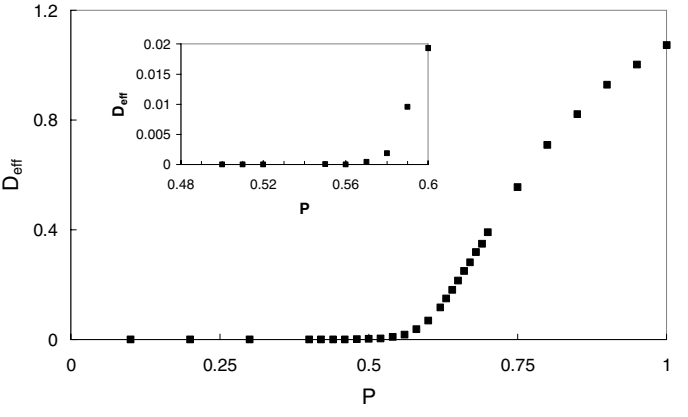



**Fig. 1.** Effective diffusion coefficient ($D_{eff}$) dependence on $P$, the concentration of the occupied sites. Inset shows this dependence for $P$ between 0.50 and 0.60

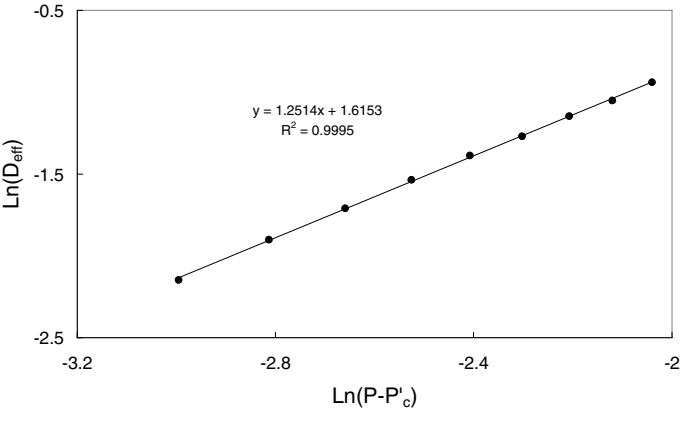



**Fig. 2.** The scaling behavior of $D$ with $(P\text{-}P'_c)$

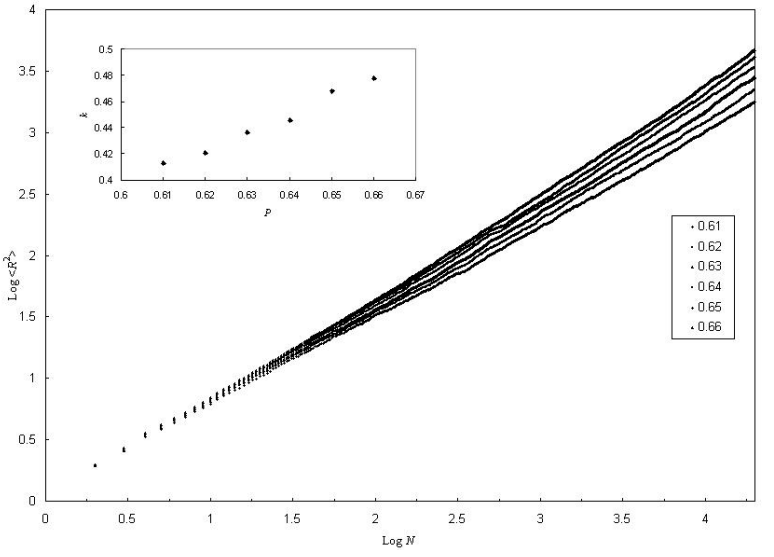



**Fig. 3.** Log-Log plot of the mean square displacement, $<R^2>$, versus $N$ for various concentrations of the occupied sites, $P$. $P$ varies between 0.66 and 0.61 from top to down. The inset shows how $k$ varies with $P$.

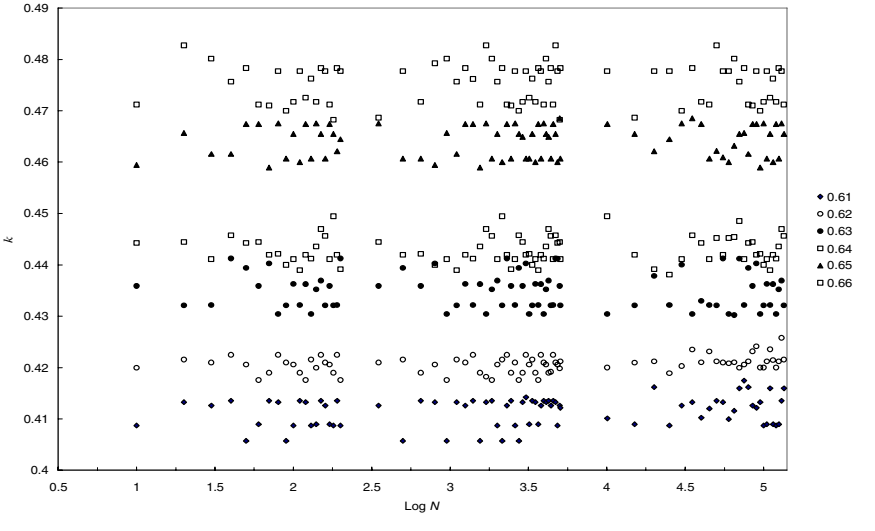



**Fig. 4.** The behavior of $k$ with Log$N$ for various concentrations of the occupied sites, $P$. $P$ varies between 0.66 and 0.61 from top to down.

the linear plots ($2k$) smoothly decrease with $P$ (see the inset of Fig. 3). Similar to $\mu$, the obtained values for $k$ were insensitive to lattice size within the range of $L$=1000-5000. These results may be taken to indicate that, when $P$ approaches $P'_c$ from the

right side, random walk on percolation lattice progressively demonstrates the sub-diffusive behavior. This was in contrast with a fixed value of about 1/3 expected according to the theory [15]. However, at $P$ sufficiently higher than threshold, deviation from the theoretically-expected $k$ of 1/3 and significant dependence on $P$ have been previously reported [e.g. in 16].

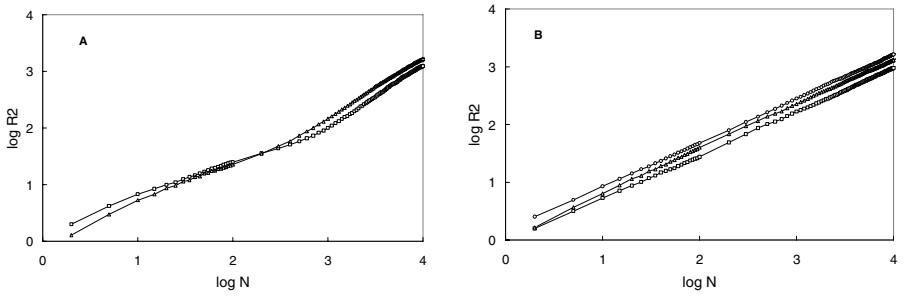



**Fig. 5.** Log-Log plot of $R^2$ versus $N$. A) for lattice configurations generated with random seeds of 1 or 3. B) for lattice configurations generated with random seeds of 2, 19 or 23

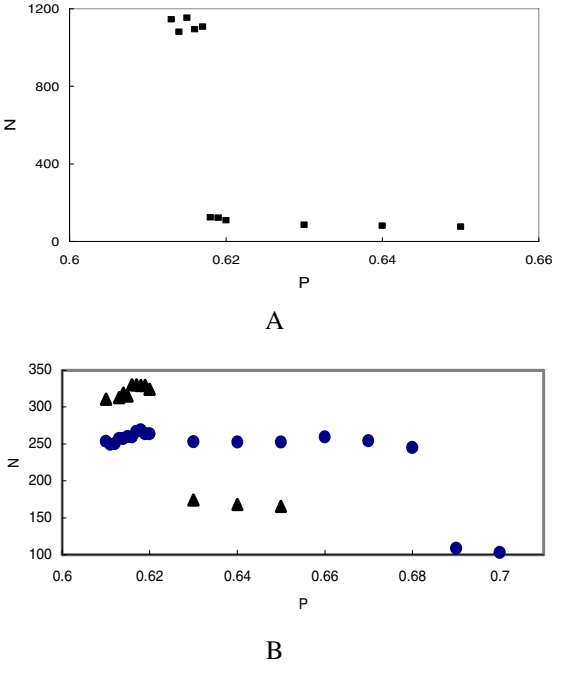



**Fig. 6.** The time step $N$ required for crossover to occur decreases sharply when $P$ increases A) For a random configuration generated with a specific seed B) For two other random configurations generated with two distinct seeds

Theoretical analysis of anomalous diffusion on percolation lattices have frequently shown that, at $P_c<P<1$ where $\xi\neq0$ ($\xi$ denotes the correlation length of the lattice) and the fractal cluster structure influences random walk on the lattice [26], the random walker would initially exhibit the anomalous sub-diffusion behavior (with $k<1/2$), but after a characteristic crossover time $\tau$, the random walker would behave classically (with $k=1/2$) [15,16]. To examine if such a crossover could be simulated by our LGA model, the local value of exponent $k$ was estimated for different $N$ from the slope of $\log R^2$ versus $\log (N)$ plot. Figure 4 demonstrates the evolution of $k$ in a wide range of $N$, where $1<\log(N)<5$. In contrast with theoretical prediction, $k$ was found to fluctuate around a nearly constant value along the course of walk at all $P$s examined, and neither the characteristic crossover nor any smooth change from anomalous sub-diffusion to normal diffusion regime was observed.

While no characteristic crossover could be represented in diffusion behavior when the Euclidean displacements were averaged over the ensemble of LGA configurations, it was observed that, at numerous LGA configurations but not all of them, a sharp transition can be found between two regimes of diffusion. Fig. 5A depicts two of the observed crossovers between various diffusion regimes. However, this phenomenon is strongly configuration-dependent and in several LGA configurations, this transition is completely disappeared (see Fig. 5B). Interestingly, the crossover time $\tau$ also revealed prominent configuration dependency. In order to obtain the crossover time $\tau$, we used the interception point of two fitted lines before and after the transition observed in the graph of $R^2$ versus time. As Fig. 6A-B demonstrates, for random configurations generated with specific seeds showing a characteristic crossover, the crossover time $\tau$ steps upward when $P$ approaches $P'_c$ from the right side although the extent and accurate position of this upward stepping are variable for different random seed numbers.

Finally, our results may be taken to indicate that the transition between two diffusion regimes is manifested in a configuration-dependent manner and the ensemble averaging over the lattice configurations disappears such transitions. This effect of ensemble averaging has previously been reported for the anisotropic nature of random walk over two-dimensional percolation clusters [27]. This effect may be originated from the removal of short-scale inhomogeneities in percolation lattices due to ensemble averaging.

## 3 Conclusion

We have employed the lattice gas automata model to simulate random walk over a square 2D site-percolation lattice. For the critical exponent of diffusion coefficient quite good result was obtained and the random walker was found to obey the anomalous sub-diffusion regime, with the exponent decreasing when the occupied site concentration decreases. The expected crossover between diffusion regimes was observed in a configuration-dependent manner, but the averaging over the ensemble of lattice configurations removed any manifestation of such crossovers. Finally, our results may be taken to indicate that the transition between two diffusion regimes is manifested in a configuration-dependent manner and the ensemble averaging over the lattice configurations disappears such transitions which may have been originated from the ensemble averaging.

## References


1. Jovanovic, B., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Phys. Rev. E 50, 2403 (1994)
2. Khan, S., Reynolds, A.M., Morrison, I.E.G., Cherry, R.J.: Phys. Rev. E 71, 041915 (2005)
3. Moore, C., Newman, M.E.J.: Phys. Rev. E 61, 5678 (2000)
4. Henley, C.L.: Phys. Rev. Lett. 71, 2741 (1993)
5. Solomon, S., Weisbuch, G., de Arcangelis, L., Jan, N., Stauffer, D.: Physica A 277, 239 (2000)
6. Martins, P.H.L., Plascak, J.A.: Phys. Rev. E 67, 046119 (2003)
7. Orbach, R.: Science 231, 814 (1986)
8. Nakayam, T., Yakubo, K., Orbach, R.L.: Rev. Mod. Phys. 66, 381 (1994)
9. Klemm, A., Metzler, R., Kimmich, R.: Phys. Rev. E 65, 021112 (2002)
10. Aharony, A.: J. Stat. Phys. 34, 931 (1984)
11. Nienhuis, B., Riedel, E.K., Schick, M.: J. Phys. A 13, L189 (1980)
12. den Nijs, M.P.M.: J. Phys. A 12, 1857 (1979)
13. Ziff, R.M.: Phys. Rev. Lett. 69, 2670 (1992)
14. de Gennes, P.G.: La Recherche 7, 911 (1976)
15. Gefen, Y., Aharony, A., Alexander, S.: Phys. Rev. Lett. 50, 77 (1983)
16. Poole, O.J., Salt, D.W.: J. Phys. A 29, 7959 (1996)
17. Havlin, S., Bunde, S.: In: Havlin, S., Bunde, A. (eds.) Fractals and Disordered Systems. Springer, Berlin (1991)
18. Alexander, S., Orbach, R.: J. Physique Lett. 43, L625 (1982)
19. Sahimi, M., Hughes, B.D., Scriven, L.E., Davis, H.T.: J. Phys. C 16, L521 (1983)
20. Yuge, Y., Onizuka, K.: J. Phys. C 11, L571 (1978)
21. Pandey, R.B., Stauffer, D., Margolina, A., Zabolitzky, J.G.: J. Stat. Phys. 34, 427 (1984)
22. Straley, J.P.: Phys. Rev. B 15, 5733 (1977)
23. Rothman, D.H., Zaleski, S.: Rev. Mod. Phys. 66, 1417 (1994)
24. Bar-Yam, Y.: Dynamics of Complex Systems. Perseus Books Group, U.S (1997)
25. Matsumoto, M., Nishimura, T.: ACM Trans. Model. Comput. Simul. 8, 3 (1998)
26. Kapitulnik, A., Aharony, A., Deutscher, G., Stauffer, D.: J. Phys. A 16, L269 (1983)
27. Muralidhar, R., Jacobs, D.J., Ramkrishna, D., Nakanishi, H.: Phys. Rev. A 43, 6503 (1991)

# Study on Acoustic Field with Fractal Boundary Using Cellular Automata

Toshihiko Komatsuzaki and Yoshio Iwata

Graduate School of Natural Science and Technology, Kanazawa University,
Kakuma-machi, Kanazawa, Ishikawa, 920-1192 Japan
{toshi,iwata}@t.kanazawa-u.ac.jp

**Abstract.** In the present study, characteristics of the acoustic field in an enclosure bounded by fractal walls are investigated using Cellular Automata (CA). CA is a discrete system which consists of finite state variables arranged on uniform grid. The dynamics of CA is expressed by temporary updating the states of cells according to the local interaction rules, defined among a cell and its neighbors. In this paper, the effect of fractal shaped boundary structure to the reverberation and sound absorption characteristics of an enclosure is investigated for two dimensional acoustic wave propagation model described by CA. Local rules are provided for the construction of fractal patterns as well as representation of wave propagation phenomena. It is known by the numerical simulations that the damping enhancement and also frequency-selective absorbing behavior is seen for specific fractal patterns and stage numbers.

## 1 Introduction

Among various kinds of sound dissipation schemes, the use of absorptive materials such as porous materials is the most common and significant technique which is widely used for room acoustics and various electric devices. However, the application of such dissipative materials may be limited by their weights, placement and costs. Basically, such absorbing materials can mitigate sound effectively at mid and high frequency range, whereas it is physically well known that they are not much effective for the low frequency regions.

On the other hand, the sound may be reduced by appropriately arranging the acoustic boundary or the sound transmission paths where the sound waves are well diffracted and interfered with each other so that the reverberation characteristic is changed. Several works have been done regarding this issue, where the wave is dissipated depending on the irregularity of the perimeter [2]-[7]. It is also reported that certain acoustic modes are trapped within a part of such irregular boundary which contributes to enhance the damping effect.

The fractal nature would be a measure for evaluating complexity of the boundary.The geometric definition of fractal structure, namely the fractal dimension was first proposed by Mandelbrot in 1975 [1], in order to describe the irregularity of object geometries. The self-similar patterns can be seen in many natural

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 282–290, 2008.

systems such as in coastlines, clouds, snow flakes and even in economic trends, where the fragments having similar patterns to the whole structures. Typical applications of fractal geometry include artistic designs of computer graphic pictures, engineering applications as the way to analyze spatio-temporal characteristic of images and information. Their intrinsic physical properties are also of general interest and various kinds of researches have also been done with the expectations that any peculiar phenomena may appear due to its self-similarity.

Among many physical processes, the wave responses in fractal structures have been attracted a wide interest and several studies have been done regarding the problem of wave localization in structural vibrations, acoustic waves and electromagnetic waves. The problems of vibrations and acoustic waves within the regions which is bounded by fractal patterns, namely a fractal drum or a fractal cavity were consistently studied by Sapoval et. al [2]-[4]. They have shown analytically that the waves are localized near the perimeter in these surface fractals. They have also shown from more general aspect that the surface irregularities play a role in damping characteristics of the cavity which is contributed by the wave localization. Gibiat et al. have shown specifically for one-dimensional acoustic wave propagation in Cantor-like waveguide that the self-similar structure appears in the frequency response characteristics and that the wave is trapped within such structure [5]. Moreover, the recent work of Kirihara et al.'s have shown experimentally that the incident microwave is confined in the central air cavity of the Menger-sponge fractal by measuring the intensity profile of electric field [7]. The more strong localization of microwaves can also be observed by increasing the fractal stage. As far as the acoustic problems are concerned, such peculiar effects originated by the fractal nature are worth studying with possible application to the room acoustics or the noise control engineering where the reverberation and attenuation characteristics can be controlled. However, the problems may seem rather pattern specific and more extensive research should be done in more effective way for modeling and performing simulations. The Cellular Automata would be a good candidate for that purpose, due to its simplicity but potentiality to express the complex phenomena.

In this paper, both the fractal boundary structure and the acoustic wave model are developed using Cellular Automata (CA), where the acoustic wave propagation is simulated in two dimensional cavity having fractal geometric pattern at the boundary. The wave model is based on past studies by authors where the two dimensional acoustic problems were solved [11,12]. In the present model, the energy dissipation takes place at the boundary where the wall of the cavity is assumed to be lightly damped without frequency dependence. Response characteristics of fractal patterns which are different in fractal dimension and length of perimeter are compared for the acoustical properties such as the reverberation and dissipation. The effect of the stage number which characterizes the elaborateness of each pattern is also investigated. It is shown that the CA approach is highly compatible with the problems involved. The results also showed that the frequency-selective absorbing behavior is seen for the different fractal patterns and stage numbers, as indicated by the past studies.

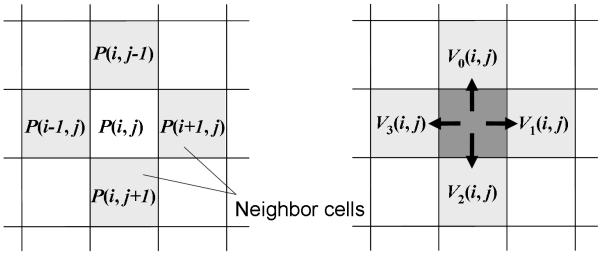



**Fig. 1.** Definition of neighbor in two dimensional acoustic model. Two state variables, sound pressure $P$ and particle velocity $V$, are placed in each cell.

## 2 The Cellular Automata Model for Wave Propagation

Cellular Automata model for simulating acoustic wave propagation is shown in this section. CA has been developed for modeling wide range of phenomena including many physical processes [8]. Specifically the wave propagation models have been studied by researchers based on Cellular Automata [9]-[12]. The simple finite difference scheme obtained by linear wave equation is referenced for developing local interaction rule, in a sense that discretized wave equation yields to an expression of local relationship of wave amplitudes. The rule is then extended to a more practical case, yet time and space are treated as discrete integers. Definitions for state variables and local interaction rules are presented in the following subsections.

### 2.1 Space Partitioning and State Definition

Two dimensional space is discretized into rectangular cells, where state of each cell is distinguished by two integers; i) zero for acoustic media, ii) 1 for lightly absorptive wall. Additionally, two variables which express the sound pressure and particle velocity in four neighbor directions are defined for the acoustic medium state. These variables are updated at each simulation step according to the local interaction rules which describe the relationship between a cell and its cross-located four neighboring cells as shown in Fig.1. Following Cellular Automata convention, time and space are treated as integers. In order for the model to be comparable with actual dimension, we assign unit cell length $dx = 0.001$[m], and also $c = 344$[m/$s$] for sound speed.

### 2.2 Definition of Local Rules

State parameters given in each cell is updated every discrete time step according to a local interaction rule. First, the particle velocities in four directions are updated in time with respect the difference of sound pressure between adjacent cells, whose update rule is described explicitly as,

$$V_a(\mathbf{x}, t+1) = V_a(\mathbf{x}, t) - \{P(\mathbf{x} + \mathbf{dx}_a, t) - P(\mathbf{x}, t)\} \ . \tag{1}$$

$V_a$ represents particle velocity of media and $P$ the sound pressure. Two dimensional cell position is expressed as a vector $\mathbf{x}$ and discrete time step as $t$. A suffix $a$ in (1) signifies index of four neighbors. The particle velocity further obeys the next (2), which expresses linear energy dissipation mechanism.

$$V_a(\mathbf{x}, t) = (1 - n \cdot d) \cdot V_a(\mathbf{x}, t) \ . \tag{2}$$

In the above (2), $n$ represents number of neighbor cells with absorbing wall state, $d$ a damping constant per unit cell. In the present study, $d$ is given as 0.001. The pressure is then updated according to the rule described by (3),

$$P(\mathbf{x}, t+1) = P(\mathbf{x}, t) - c_a^2 \sum_a V_a(\mathbf{x}, t+1) \ , \tag{3}$$

where $c_a$ denotes the wave traveling speed in CA space. Sound pressure and particle velocities are updated according to the local rule described by above three equations.

Since calculation is carried out between nearby cells that are separated only a unit length at every step, any physical quantities cannot have the transport speed exceed to this calculation limit. The maximum wave speed becomes $c_a = 1/\sqrt{2}$ for two dimensional space, therefore the wave front travels $1/\sqrt{2}$ of unit length per calculation step [11].

## 3 Generating Fractal Pattern with CA

The fractal geometries of two dimensional acoustic cavity are produced according to a set of rules within the Cellular Automata framework. Among many patterns that were tested, six fractal patterns which are different in fractal dimension are shown here. The elementary patterns consisting of $5 \times 5$ cells that are used for the boundary construction are shown in Fig.2. The filled square signifies that the cell is a part of wall, and the unfilled a vacant space.

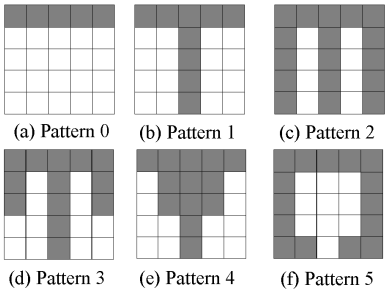



**Fig. 2.** Elementary patterns for generating fractal structure. Fractal dimension of respective pattern is: (a) 1.00, (b) 1.37, (c) 1.76, (d) 1.59, (e) 1.59, and (f) 1.68.

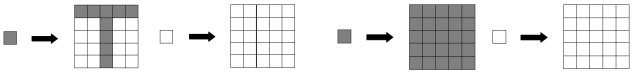



**Fig. 3.** Set of local rules for fractal pattern growth. Rule in (a) generates self-similar pattern according to pre-assigned elementary structure. In (b), a single cell is enlarged to a set of cells having the same state. In both rules a cell is amplified to larger area consisting of $5 \times 5$ grids.

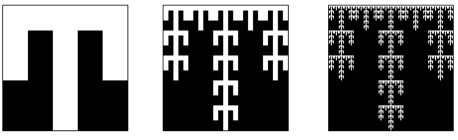



**Fig. 4.** An example of generated $125 \times 125$ cell pattern for respective fractal level based on elementary pattern no. 3

To generate gross structure of fractal walls with respect to each fractal stage, a set of rules consisting of self-similar growth and enlargement functions should be defined. The self-similar pattern growth rule is used to emphasize elaborateness of fractal pattern, whereas the enlargement rule is applied for simply magnifying the original pattern. These rules are schematically shown in Fig.3. Following each of these rules, a cell is enlarged to a set of cells consisting of $5 \times 5$ grids for single calculation. By combining these rules arbitrarily and calculating for three times, gross structures of $125 \times 125$ cells having fractal geometry will appear corresponding to respective fractal stage number. Examples of fractal picture compared by the stage number varying from stage 1 to 3 for elementary pattern 3 are shown in Fig.4. Starting from a single wall state cell, stage 1 pattern is obtained where the self-similar growth rule is applied once and subsequently the enlargement rule twice. Stage 2 can be calculated by applying the former rule twice followed by latter once. Additionally, stage 3 is obtained by the application of the latter rule three times. It is known from these figures that the self-similar structure is expressed at higher stage numbers.

## 4 Wave Propagation in a Cavity with Fractal Boundary

Simulations are performed for two-dimensional acoustic wave which propagates inside the cavity having fractal geometric patterns developed in Sect.3. Schematic of the acoustic field is shown in Fig.5. The simulation space is divided into $625 \times 625$ cells, where the unit size of a cell is assumed to be 1[mm] which is compatible with physical dimension of $6.25 \times 6.25$[m]. The twelve identical fractal blocks made of $125 \times 125$ cells clusters around the acoustic cavity and constitutes the boundary. A

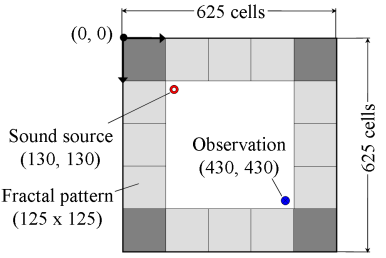



**Fig. 5.** Spatial arrangement of two-dimensional acoustic field

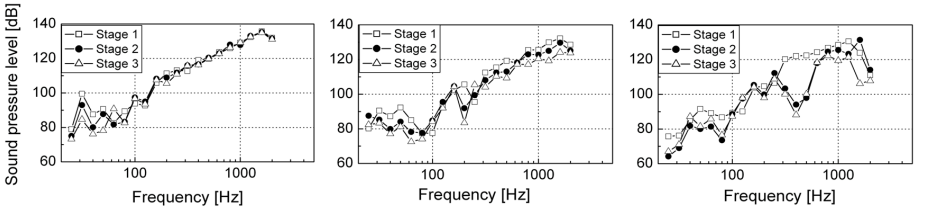


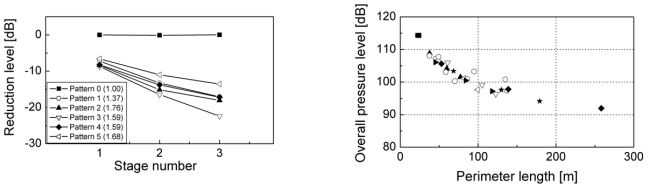

**Fig. 6.** 1/3 octave band sound pressure level of fractal acoustic cavity

sound source is located at (130, 130), whereas the time history of the sound inside the cavity is observed at (430, 430). Since the choice of random excitation is an appropriate way to know the approximate acoustical characteristics of an acoustic field, the white noise filtered with cutoff frequency of 12.5[kHz] is driven by assigning forced particle velocity to the cell. In the following subsections, the gross characteristics regarding reverberation and the damping of sound inside fractal bounded acoustic cavity are shown.

### 4.1 Frequency Response of Sound Pressure Inside Fractal Cavity

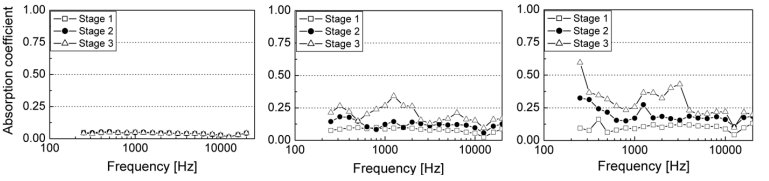
While generating random noise, the simulation is carried out for 400, 000 iterations which approximately corresponds to the actual time of 8 seconds. Based on the measured time history of the last one second at the observation point, the 1/3 octave band frequency analysis of the sound pressure level for respective fractal boundary pattern is performed. Among six patterns, the frequency responses for the first three patterns are shown in Fig.6. In each pattern the response is also compared by the stage number. As seen from Fig.6(a), the response for pattern 0 where the shape of the perimeter is simply consisted of straight lines is not influenced by the stage sequence, since the geometry is not changed by pattern transformation. The slight change in the pressure level is due to the alteration of the cavity volume. The pressure level is holistically decreased according to the stage number in Fig.6(b) and (c), since the length of perimeter as well as the gross damping effect increases. Also specifically in Fig.6(c), the selective damping

enhancement can be seen around 100 and 400[Hz] depending on the stage number. Though it remains conjectural, a certain effect may take place due to the coincidence between the wavelength of sound and the size of concaves in the perimeter. Additionally, the overall sound pressure level for respective fractal




(a) Sound reduction relative to pattern 0 (b) Overall level against perimeter length

**Fig. 7.** Overall sound pressure level




(a) Pattern 0 ($D$=1.00) (b) Pattern 1 ($D$=1.37) (c) Pattern 2 ($D$=1.76)

**Fig. 8.** Absorption coefficient for different boundaries

pattern is shown in Fig.7. In Fig.7(a), the overall pressure decreases according to the stage development in every pattern. Also it is known that the larger fractal dimension value does not always contribute to larger decrease in overall pressure. Results of all patterns and stages are plotted altogether in Fig.7(b). The figure signifies that the damping of sound pressure inside cavity strongly depends on the length of perimeter, rather than the fractal dimension of elementary pattern.

### 4.2 Equivalent Absorption Coefficient

In order to evaluate the sound dissipation in a room, statistical absorption coefficient is practically derived based on the Eyring's equation [14], written as:

$$T_{\mathrm{r}} = \frac{0.161V}{-S\log_e(1-\alpha)} \quad , \tag{4}$$

where the reverberation time $T_{\mathrm{r}}$ is defined as the amount of time in which the sound pressure decreases naturally over time as much as 60dB relative to the pressure when shutting down the source excitation. In (4), $V$ signifies the volume of the cavity and also $S$ the surface area, respectively. Using (4) and measured reverberation time, the absorption coefficient $\alpha$ is reversely calculated for each 1/3 octave band. The sound absorption characteristics for the fractal pattern

of 0, 1 and 2 are specifically shown in Fig.8. As easily predicted from Fig.6, significantly low and unchanged absorption characteristic is seen for the acoustic cavity with pattern 0 regardless of the stage number increment. On the other hand, the results of fractal pattern 1 and 2 as in Fig.8(b) and (c) signify that the sound is dissipated as large as the stage number increases. These results imply that even with the limited amount of absorptive treatment on walls, the sound is effectively damped by providing structural irregularity on the boundary. The frequency-selective behavior must depend on the geometric nature of fractal patterns, which should further be investigated.

## 5 Conclusions

In the present paper, the acoustic wave propagation in a cavity bounded by fractal walls are modeled and investigated using Cellular Automata. Local rules are provided for both the generation of fractal patterns and the representation of wave propagation phenomena. The reverberation and the damping characteristics of the cavity having fractal boundary with different patterns and stages are compared. It is known that the damping of sound inside cavity strongly depends on the length of perimeter, rather than the fractal dimension of elementary pattern. The results also showed that the frequency-selective absorbing behavior is seen for the different fractal patterns and stage numbers. Such pattern specific feature must further be investigated as the future issue.

## References

1. Mandelbrot, B.B.: The Fractal Geometry of Nature. Freeman, San Francisco (1982)
2. Sapoval, B., Gobron, T.: Vibrations of strongly irregular or fractal resonators. Phys. Rev. E 47(5), 3013–3024 (1993)
3. Sapoval, B., Haeberle, O., Russ, S.: Acoustical properties of irregular and fractal cavities. J. Acoust. Soc. Am. 102(4), 2014–2019 (1997)
4. Felix, S., Asch, M., Filoche, M., Sapoval, B.: Localization and increased damping in irregular acoustic cavities. J. Sound Vib. 299, 965–976 (2007)
5. Gibiat, V., Barjau, A., Castor, K., Chazaud, E.B.: Acoustical propagation in a prefractal waveguide. Phys. Rev. E 67, 066609 (2003)
6. Hobiki, Y., Yakubo, K., Nakayama, T.: Spectral characteristics in resonators with fractal boundaries. Phys. Rev. E 54(2), 1997–2004 (1996)
7. Kirihara, S., Takeda, M., Sakoda, K., Honda, K., Miyamoto, Y.: Strong localization of microwave in photonic fractals with Menger-sponge structure. J. Euro. Ceram. Soc. 26, 1861–1864 (2006)
8. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press, Cambridge (1998)
9. Chen, H., Chen, S., Doolen, G., Lee, Y.C.: Simple Lattice Gas Models for Waves. Complex Systems 2, 259–267 (1988)
10. Sudo, Y., Sparrow, V.W.: Sound Propagation Simulations Using Lattice Gas Methods. AIAA J. 33, 1582–1589 (1995)

11. Komatsuzaki, T., Sato, H., Iwata, Y., Morishita, S.: Simulation of Acoustic Wave Propagation using Cellular Automata. Trans. JSCES 1, 135–140 (1999)
12. Komatsuzaki, T., Iwata, Y.: Modeling of Sound Absorption by Porous Materials using Cellular Automata. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 357–366. Springer, Heidelberg (2006)
13. Wolfram, S.: A New Kind Of Science. Wolfram Media Inc. (2002)
14. Morse, P.M., Ingard, K.U.: Theoretical Acoustics. Princeton University Press, Princeton (1986)

# The Heart Pacemaker by Cellular Automata on Complex Networks

Danuta Makowiec

Institute of Theoretical Physics and Astrophysics, Gdańsk University,
80-952 Gdańsk, ul.Wita Stwosza 57, Poland
fizdm@univ.gda.pl

**Abstract.** Specially desinged network of Greenberg-Hastings (GH) cellular automata is shown to be a reliable approximation to cardiac pacemaker. A complex network is built basing on a square lattice where some edges are rewired locally and with preference to link to cells which are more connected to other cells. GH automata are systems with cyclic intrinsic dynamics where three possible cell's states are characterized by timings — time steps spent in each state. Complex networks of GH automata evolve rhythmically with periods determined by timings. Diversity of oscillations coexisting in the system depends on network topology. Large variety in oscillations is interpreted as effectiveness to response to actual needs of a body.

**Keywords:** Greenberg-Hastings cellular automata, complex networks.

## 1 Introduction

The regular impulses, that result in rhythmic contractions of the heart, begin at the cardiac pacemaker called the sinoatrial node (SA node) [1]. The sinoatrial (SA) node is a piece of the cardiac tissue (3 mm wide and 7 mm long) located on the right atrium. The action potential originates in the SA node and then travels across the wall of the atrium to the atrioventricular (AV) node. In the same time the activity of the SA node spreads throughout the atria causing the atrial contraction. Then, the specialized conduction pathways: bundle of His and Purkinje fibers conduct the impulse throughout the ventricles causing the ventricle's contraction in an unison way.

Cellular automata have been used to model biological systems of different types, see, e.g., [2] for the review. In particular, cellular automata are known to model excitable media very well. The simplest cellular automata which are used to model the cardiac tissue are known as Greenberg-Hastings (GH) [3,4]. In order to include the heterogeneity of the real cardiac tissue, different cell connections between cellular automata have to be considered [5,6].

In the present work, following facts known from the SA node physiology, we propose a network which reconstructs the heterogeneity of the canine cardiac tissue, Section 2. Then we discuss properties of the GH cellular automaton in order to model activity of a SA nodal cell, Section 3. Finally, in Section 4, we

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 291–298, 2008.

present results indicating that the specially constructed network of GH-type cellular automata leads to a reliable approximation of the cardiac pacemaker. The system evolves rhythmically with the period determined by intrinsic dynamics of the cellular automaton. Depending on the underlying network the diversity of oscillations varies. The large variety in oscillations we interpret as readiness of the system to the effective response to the actual needs of the body.

## 2 The Network Topology

For a long time the heart tissue was considered as syncytium - multi nucleated mass of cytoplasm that is not separated into individual cells [7]. Due to development of electron microscopy it became clear that cardiac cells — myocytes, were long and individual units, bounded on either end by intercalated discs. Then, it was shown that each disc was a gap which separated the opposed cell membranes. Each gap junction consists of many mechanisms which provide a pathway for direct cell-to-cell communication. Hence, in a simplified picture, one can see the cardiac tissue as a network consisting of branched chains of elongated cells which are connected by gap junctions — the only way to transmit the interactions.

There are known some network characteristics of cardiac tissue. It appears that a typical myocyte which builds the canine SA node has about $4.8 \pm 0.7$ nearest neighbors which are located as follows [8]: (a) about $0.7 \pm 0.5$ are side-to-side neighbors, (b) about $0.6 \pm 0.7$ are end-to-end neighbors , and (c) about $3.5 \pm 1.9$ are lateral neighbors. Hence about 74% connections are lateral. Moreover, it is known that side-to-side and lateral connections have relatively small gap junctions, and therefore, their efficiency in transmiting signals is considered as less effective than in the case of the end-to-end connections.

To replicate the nodal properties let us consider a square lattice and introduce preferences to askew connections in the following way:

**(I)** For a given probability $d$, a vertical or horizontal link is created with $d/2$ probability while any askew edge is created with $2 * d$ probability.

It is easy to check that the canine SA network is about to be restored if we work with $d = 0.45$, see Fig. 2. The cells from the leftmost column and rightmost column form the interface to crista terminalis cells — cardiac tissue which conducts signals from the SA node to the AV node. Therefore, we additionally link them via horizontal connections. Moreover, we connect unlinked cells cells to their nearest right cells. By these two extra rules, some additional horizontal connections are added. Finally, in case of $d = 0.45$, a resulting network has about 10% vertical, 11% horizontal and 79% askew connections.

The introduced structure is flat. To make the surface uneven we propose the local rewiring procedure which additionally sets the preference to the connectivity between a cell and the network. The rewiring algorithm consists of the following rules: see Fig. 2: let $p$ is *probability of rewiring* then

**(II)** for a given cell $A$ a less connected cell is unlinked. The probability to unlink a $B$ cell from a vertex $A$ is calculated as: $p_{unlink} = \frac{p}{\deg(B)}$ , where $\deg(B)$ is degree of a vertex $B$. The rewiring is local what means that a new cell $B'$ which

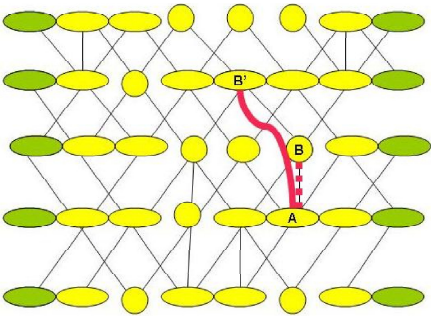



**Fig. 1.** Construction of a network: most of connections are diagonal because of **(I)** rule; the leftmost and rightmost cells of each row (input to crista terminalis) are connected end-to-end with their neighbors; red lines illustrate the rewiring rule **(II)**. Color on-line.

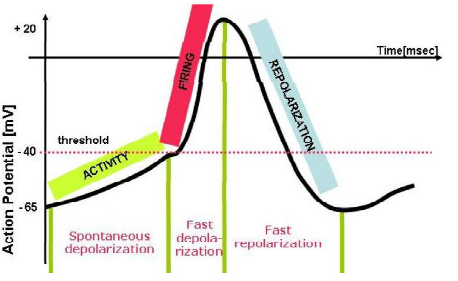



**Fig. 2.** Changes in the electrical potential on the membrane of a typical cell from the SA node, adapted from [1], color on-line

will be linked to the cell $A$ is chosen from actual neighbors of $B$ automaton. To preserve the line structure, any horizontal connection is forbidden to be rewired. Unlinking from a leaf is forbidden also.

In Fig.3 left we show vertex degree distributions in networks resulted after applying the above algorithm to each edge with $p = 0.01$ and repeating the procedure 0, 100 and 500 Monte Carlo steps (MCS). It appears that due to locality in rewiring, the network is only slightly modified. In Fig.3 right a typical example of connections is shown. A network is almost flat and slightly heterogeneous — there are several vertices with the vertex degree larger two times than the average vertex degree.

## 3 The FRA Cellular Automaton

In the simplified picture, there are three main stages in a cell dynamics which we will call *firing, repolarization* and *activity* as it is indicated in Fig. 2. Being in the firing state a cell initiates the action potential in the neighboring cells. But to read the signal from a neighbor, the cell has to be in the activity state. If the cell is in the repolarization state then there is no way neither to be excited nor to excite the other cell. If a cell does not receive a stimulus from a neighbor, then after some time the cell starts firing because of its intrinsic cycle dynamics.

The above description suggests considering a cellular automaton as a three-state automaton $F, R, A$ which performs intrinsic cyclic evolution $F \to R \to A \to F \to ....$ Let us assume that there are fixed numbers of steps $n_F, n_R, n_A$ during which each automaton stays in corresponding states. This proposition is related to the original GH cellular automata [3]. Our modifications follow considerations of Bub *et.al* [9] — we skip constraints for self-excitation of a cell. Here, each automaton after spending $n_A$ steps in $A$ state changes its state to $F$.

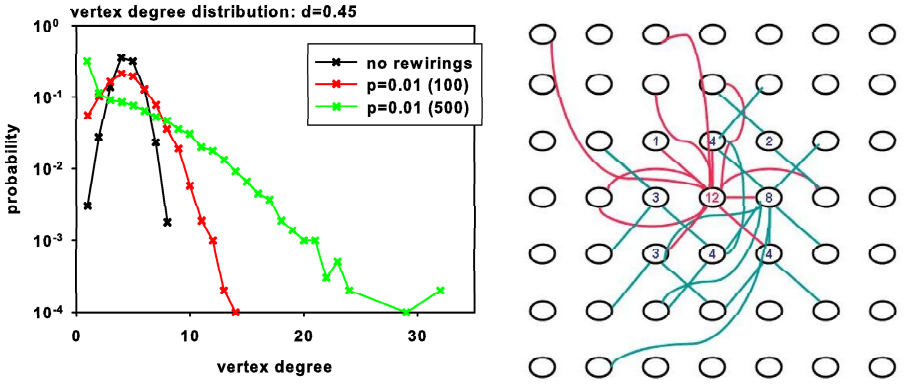



**Fig. 3.** Left figure: the vertex distribution in the networks considered. Right figure: a piece of a typical connections. Neighbors of a vertex with high degree are presented - red lines; green lines describe links of Moore neighbors of the vertex. Numbers are degrees of vertices. Color on-line.

The introduced intrinsic dynamic rule is deterministic what is closely related with the physiologically observed fact that the lengths of the fast depolarization and repolarization are strongly fixed. Hence any changes in timings $n_F, n_R$ and $n_A$ from a period to a period should be rather limited. Therefore, let us consider the little probable possibility of shortening each of $n_F, n_R$ and $n_A$ value. It effects in that there is a nonzero probability for a switch of the actual automaton state to the next state in its intrinsic cycle. Formally, we propose that cellular automaton performs the stochastic evolution governed by the following rule:

$$\binom{\sigma}{s}(t) \overset{t\to t+1}{\longmapsto} \begin{cases} \binom{next(\sigma)}{1} & \text{with probability } (\frac{s}{n_S})^{\xi} \\ \binom{\sigma}{s+1} & \text{with probability } 1-(\frac{s}{n_S})^{\xi} \end{cases} \tag{1}$$

where $\sigma$ is any of $F, R, A$ states, $s$ counts the number of steps which an automaton actually stays in a $\sigma$ state and *next* means the subsequent state in the intrinsic cell cycle. If $\xi >> 1$ then we restore the deterministic evolution. Notice that for $\xi > 1$ only very few last steps could be skipped and therefore the effective timings are closely determined by values of $n_F, n_R$ and $n_A$.

Finally, let as assume that the threshold for firing an automaton by its nearest neighbors equals to 1, i.e., at least two neighbors in the *firing* state are needed to switch an automaton from the state of *activity* to the *firing* state. However, since the horizontal connections are known to be more efficient than others, we additionally assume that the influence of the horizontal connections is doubled — only one left or right neighbor being in the *firing* state is able to activate the cell. We will refer to described cellular automata as *FRA-CA*.

## 4 Results

Our earlier studies of FRA-CA have provided that in case of the two FRA-CA evolving according to the deterministic version of 1, only the three stable solutions are possible, [10,11]:

• *the rules adjusted evolution*: if the result of both rules: intrinsic and interactions between the automata, is the same;

• *the alternating impacts evolution*: if within each period two events of impacts take place. The first event means **A** automaton is impacted by **B** automaton —**A** automaton is switched to *firing*. Then the second event occurs — **B** automaton is impacted by **A** automaton what switches **B** automaton to *firing*.

•*the quiet evolution*: there are not any impacts between the automata.

One can say that the alternating impacts evolution is the maximally active dynamics since both cells of a pair intensively interact with each other. Because of the intensity of impacts the intrinsic periods of both automata are shortened to the shortest period possible $T^* = n_F + n_R + 1$. The other two evolutions are also periodic but the period is equal to the intrinsic period of the FRA-CA, i.e., $T = n_F + n_R + n_A$.

Moreover, it has appeared that in case of a line of many FRA-CA with open boundary, the system always reaches the periodic state and only one of the two periods happens — either $T$ or $T^*$. Depending on the model parameters the probability to find which period occur is different. If $n_F \leq n_R$ then the solution with the period $T$ happens. If $n_F > n_R$ then the solution with the period $T^*$ is significantly more probable. It also has been found that if $n_R > n_F$ then all automata from a line follow the rules adjusted evolution. Case of $n_R > n_F$ is physiologically important because it is known that the time used by a myocyte for the fast depolarization is shorter than the time spent on repolarization processes or during the slow depolarization [12].

At present we ask whether stationary evolution of FRA-CA distributed on networks described in Section 2 with dynamics (1) is periodic. The network state is evaluated by number of FRA-CA staying in the *firing* state in the leftmost column $I_{left}$ and in the rightmost column $I_{right}$. These currents are assumed to imitate signals which arrive to outputs — the crista terminalis. Additionally, we also consider a total signal $I_{total}$ which counts all cells which are in *firing* state.

To identify oscillations in the signal we calculate the power spectrum $S(f)$ of signals by the Fast Fourier Transform. In Fig.4 we show representative signals obtained for various model parameters from networks consisting of $N = 10\,000$ FRA-CA. It appears that oscillations of the total signal $I_{total}$ are rather weak. But the signals from the borders of the networks $I_{left}$ and $I_{right}$ are noticeable periodic. There are evident maxima in all plots of power spectra. These maxima are located close to $1/T^*$ and / or $1/T$. They are moved a little to the right and wide because of the stochasticity of the intrinsic dynamics.

Let us assume that a given frequency $f$ is present in a signal if its power spectrum $S(f)$ value is greater than $10^{-7}$. In Fig. 5 we present all frequencies extracted in this way from the spectra of $I_{total}$, $I_{left}$ and $I_{right}$ when $n_F = 10$, $n_A = 20$ and for different values of $n_R$. The results are collected according to

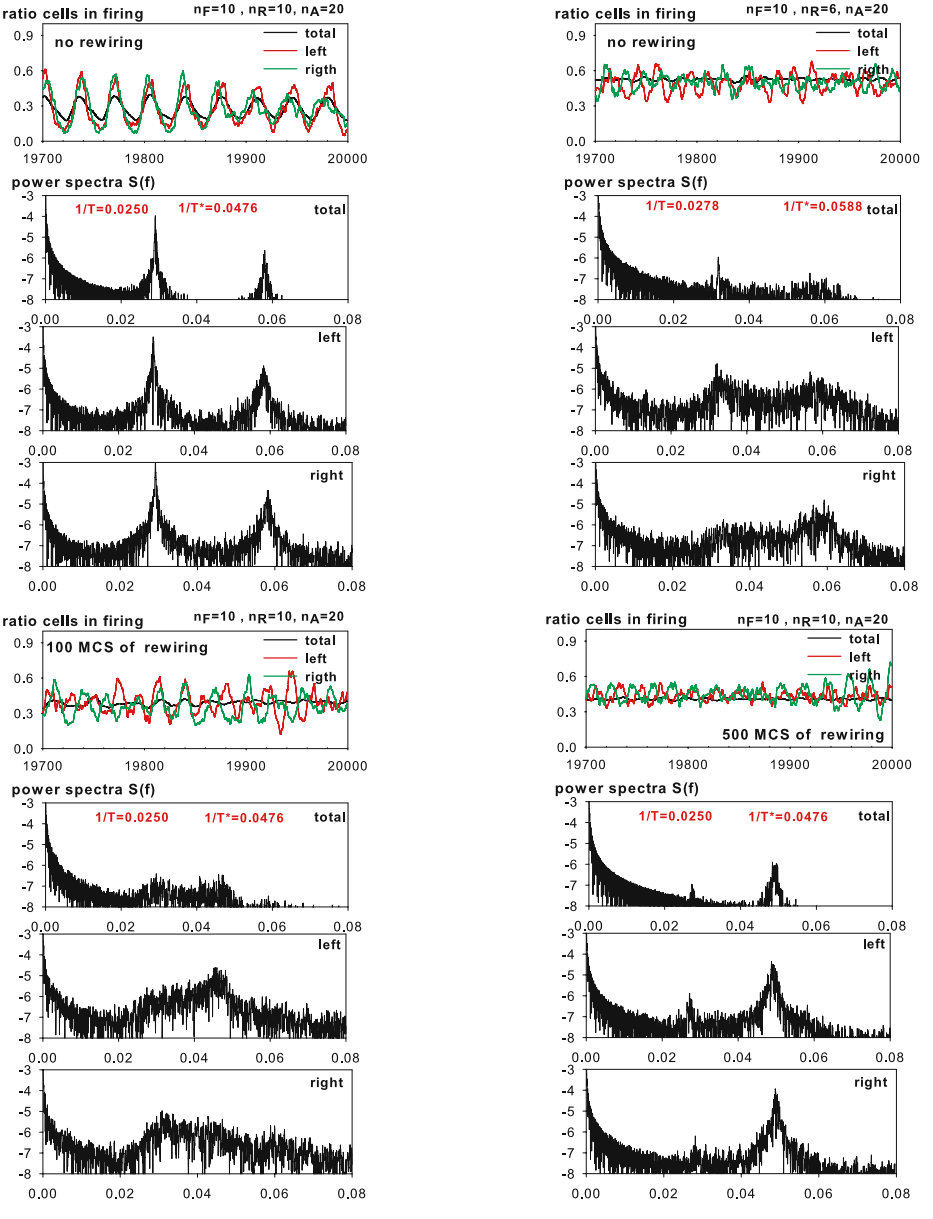



**Fig. 4.** Typical time series of the output signals $I_{total}$, $I_{left}$ and $I_{right}$ and their power spectra (log plots) for different networks. Timings are given above the plots of time series. Color on-line.

the three types of network settings: no rewiring, 100 MCS of rewiring and 500 MCS of rewiring.

It appears that in case of non rewired lattice the system oscillates with one dominant period. For $n_R$ significantly smaller than $n_F$ (here $n_R < 6$) the period

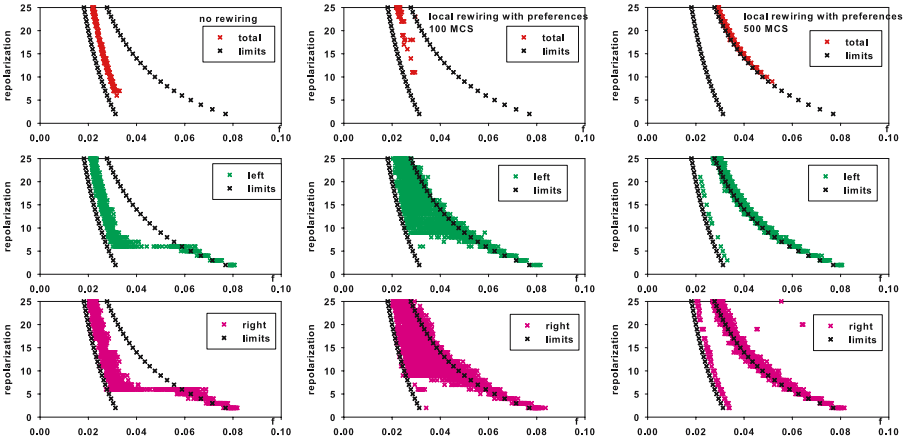



**Fig. 5.** Dominant frequencies identified in the output signals $I_{total}$, $I_{left}$ and $I_{right}$ in different networks. Black points correspond to $1/T$ and $1/T^*$ frequencies. Color on-line.

is about $T^*$. But then the evolution switches to oscillations with the period $T$ what indicates that most of FRA-CA is not influenced by neighbors and rather follow the intrinsic cyclic dynamics. This observation does not hold if the network is modified by rewiring. In case of a little modification (100 MCS) for any $n_R$ the active oscillations with $T^*$ period are always present. Moreover, the same time the wide spectrum of possible oscillations between the $T^*$ and $T$ periods is present in the system. Hence the interactions among cells vividly influence the system evolution. Finally, if the network is strongly modified (case of 500 MCS) then only two basic oscillations $T$ and $T^*$ are present and $T^*$ is dominant.

## 5 Conclusions

In the following by simulations we have investigated influence of topology of a network to oscillations of multi FRA-CA systems. We have found that all systems considered evolve rhythmically with the periods determined by the cell's intrinsic dynamics. However, in case of the heterogeneous network more oscillations coexists together. Moreover there is not present the transition in the dominant oscillation when timings are changed. Therefore the FRA-CA on the heterogeneous networks seem to better reconstruct properties of the real pacemaker — such systems can answer more flexible to external regulating systems.

The simulations were performed with 10 000 FRA-CA. In case of humans it is known that the SA node consists of about 70 000 cells[1]. Hence to obtain on-to-one mapping we should increase the size of simulated systems. Moreover, it is known that the SA nodal cells are not identical. Fortunately, the differences between cells are systematic — the further from the center of the sinus node the cell is located, then the difference between a center cell and a periphery

cell is more evident. Therefore the work on enlarging the system, should also incorporate the fact of the cell diversity.

The presented investigations are only preliminary and are not claimed to be completed. However, we hope that by comparing the physiologically known properties and our results we provide hints on the cellular network of the SA node.

**Acknowledgment.** Author thanks to University of Gdańsk for support: BW-5400-5-0090-08.

## References

1. Klabunde, R.E.: Cardiovascular Physiology Concepts, http://www.cvphysiology.com
2. Ermentrout, G.B., Edelstein-Keshet, L.: Cellular Automata Approaches to Biological Modeling. J. Theor. Biol. 160, 97–133 (1993)
3. Greenberg, J.M., Hastings, S.P.: Spatial Patterns for Discrete Models of Diffusion in Exitable Media. SIAM J. Appl. Math. 34, 515–523 (1978)
4. Greenberg, J.M., Greene, C., Hastings, S.P.: Remarks on a 25 Year Old Theorem on Two-dimensional Cellular Automata. Int. J. Unconv. Comp. 1, 399–402 (2005)
5. Markus, M., Hess, B.: Isotropic Cellular Automaton for Modeling Excital Media. Nature 347, 56–58 (1990)
6. Bub, G., Shrier, A., Glass, L.: Spiral Wave Generation in Heteroneeous Excitable Media. Phys.Rev.Lett. 88, 058101–1–4 (2002)
7. Saffitz, J.E., Lerner, D.L., Yamada, K.A.: Gap Junctions Distribution and Regulation in the Heart. In: Zipes, D.P., Jalive, J. (eds.) Cardiac Elecrophysiology. From Cell to Bedside, pp. 181–191. Saunders Co., Philadelphia (2004)
8. Luke, R.A., Saffitz, J.E.: Remodeling of Ventricular Conduction Pathways in Healed Canine Infarct Border Zones. J. Clin. Invest. 87(5), 1594–1602 (1991)
9. Bub, G., Shrier, A., Glass, L.: Global Organization of Dynamics in Oscillatory Heterogeneous Excitable Media. Phys. Rev. Lett. 94, 028105-1–4 (2005)
10. Makowiec, D.: On Cellular Automata Modeling of Cardiac Pacemaker. Journal of Cellular Automata (to appear, 2008)
11. Makowiec, D.: Cellular Automata Model of Cardiac Pacemaker. Acta Phys. Pol. B (to appear, 2008)
12. Kodama, I., Honyo, H., Boyett, M.R.: Are We Lost in the Labyrinth of the Sinoatrial Node Pacemaker Mechanism? Journal of Cardiovascular Electrophysiology 13(12), 1303–1305 (2002)

# A Proposal for a Japanese Keyboard on Cellular Phones

Maurice Margenstern[1], Benoît Martin[2], Hiroshi Umeo[3], Shogo Yamano[3], and Kazuhiro Nishioka[3]

[1] Université Paul Verlaine − Metz, IUT de Metz,
LITA, EA 3097, UFR MIM, Île du Saulcy,
57045 Metz, Cédex, France
margens@univ-metz.fr
[2] Université Paul Verlaine − Metz,
LITA, EA 3097, UFR MIM, Île du Saulcy,
57045 Metz, Cédex, France
benoit.martin@univ-metz.fr
[3] Faculty of Information Science and Technology,
Osaka Electro-Communication University,
Neyagawa-shi, Hatsu-cho, 18-8, Osaka, Japan
umeo@cyt.osakac.ac.jp

**Abstract.** In this paper, we propose a new way to display alphabetical signs in order to write messages on a cellular phone in the Japanese language.

## 1 Introduction

In [1], the second author tackled the problem of defining a good way to put messages on a cellular phone in a very new approach. The idea is to use the pentagrid, a tiling of the hyperbolic plane, thoroughly studied from an algortihmic point of view by the first author, see [5]. The pentagrid was already used by the first and the second author in order to define a colour chooser, see [2].

The approach of [1] was confirmed by tests, however performed on a small number of persons.

In the next section, we remind the minimal features of hyperbolic geometry in order the reader could understand the basic material of the paper. In the following section, we describe the proposal. Then, we describe the first results on the work for implementing the proposal on cell phones.

## 2 A Few Words on the Pentagrid

We use the model of the hyperbolic plane devised by Poincaré in the last third of the 19$^{\text{th}}$ century. In this model, the hyperbolic plane is the set of points which lie in the open unit disc of the Euclidean plane whose border is the unit circle. The lines of the hyperbolic plane in Poincaré's disc model are either the trace

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 299–306, 2008.

of diametral lines or the trace of circles which are orthogonal to the unit circle, see figure 1. We say that the considered lines or circles **support** the hyperbolic line, *h*-**line** for short and, most often, simply **line** when there is no ambiguity.

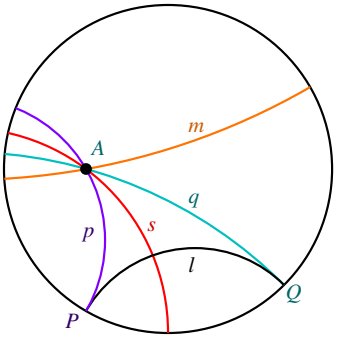



**Fig. 1.** The lines $p$ and $q$ pass through $A$ and are parallel to the line $\ell$, with points at infinity $P$ and $Q$, on the border of the unit disc. The $h$-line $m$ passes through $A$ too and is non-secant with $\ell$: it does not cut it, neither in the unit disc nor outside.

The angle between two $h$-lines are defined as the Euclidean angle between the tangents to their support. This is one reason for choosing this model: hyperbolic angles between $h$-lines are, in a natural way, the Euclidean angle between the corresponding supports. In particular, orthogonal circles support perpendicular $h$-lines.

A striking property of this geometry is that there is no rectangle. From this, one can prove that two lines of the hyperbolic plane are non-secant if and only if they have a common perpendicular.

Contrary to the Euclidean plane were there are only three kinds of tilings based on the recursive replication of a regular polygon by reflection in its sides and of the images in their sides, here there are infinitely many such tilings. This was proved by Poincaré in late $19^{\text{th}}$ century. In one of these tilings, called the **pentagrid**, the regular polygon is a pentagon with right angles.

Figure 2 sketchily remembers that the tiling is spanned by a generating tree. Now, as indicated in figure 3, five quarters around a central tile allows us to exactly cover the hyperbolic plane with the **pentagrid**.

In the right-hand side picture of figure 3, we remember the basic process which defines the coordinates in a quarter of the pentagrid, see [5]. We number the nodes of the tree, starting from the root and going on, level by level and, on each level, from the left to the right. Then, we represent each number in the basis defined by the Fibonacci sequence with $f_1 = 1$, $f_2 = 2$, taking the maximal representation, see [3,5].

The main reason of this system of coordinates is that from any cell, we can find out the coordinates of its neighbours in linear time with respect to the coordinate of the cell. Also in linear time from the coordinate of the cell, we can

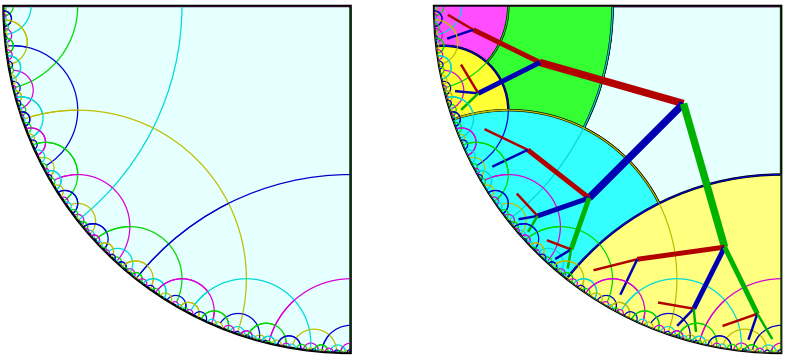


**Fig. 2.** On the left: the tiling; on the right: the underlying tree which spans the tiling

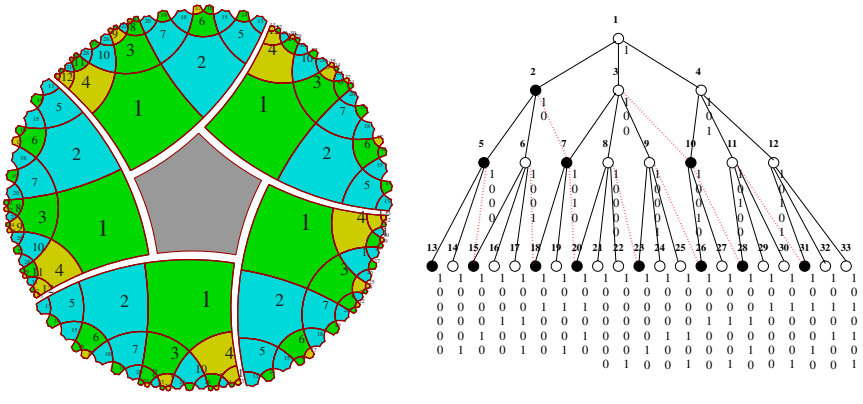


**Fig. 3.** On the left: five quarters around a central tile; on the right: the representations of the numbers attached to the nodes of the Fibonacci tree

compute the path which goes from the central cell to the cell. Further references on this system and on hyperbolic geometry can be found in [5] and [4].

Now, as the system coordinate is fixed, we can turn to the application to keyboards for cell phones.

## 3 The Proposal

The proposal is based on the idea to use the pentagrid to display **hiraganas** and **katakanas**, the syllabic alphabets of the Japanese language.

Looking at the pentagrid of figure 4, does not it seem that the pentagrid is fit for the Japanese language? Indeed, the phonetic writing of Japanese, using **hiraganas** and **katakanas** is based on ten series, in which the letters are ordered

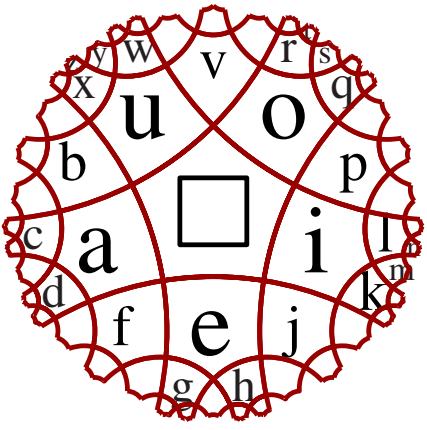



**Fig. 4.** The proposal of keyboard from [1]

in relation to the vowels they contain. Now, the Japanese language has five basic vowels, **a**, **i**, **u**, **e** and **o**, following the traditional Japanese order, the corresponding hiraganas being: あ , い , う , え and お .

This keyboard is displayed in figure 5, where each sector, spanned by a Fibonacci tree, is devoted to the syllables associated to the same vowel. The corresponding vowel is put at the root of the tree. The others are displayed level by level, following the traditional order while going from the left-hand side border to the right-hand side one of a sector.

However, the project of this keyboard, which we are developping does not stop on this representation of the hiraganas. Of course, a similar keyboard may display the katakanas, as their traditional order is exactly the same as that of the hiraganas. Now, standard Japanese language makes use of Chinese characters named **kanjis**. The main goal of the project is to give access to the kanjis themselves, using the display of the Japanese keyboard, see figure 5.

Indeed, each kanji has several pronunciations represented by a hiragana or a katakana. The idea is to go from the pronunciation to the kanjis. Now, several kanjis have the same pronunciation and the same syllabic sign represents several kanjis. Nevertheless, this system seems to be convenient: once a hiragana/katakana is selected, a new viewer displays the possible kanjis inside a pentagrid view. In such a display, the kanjis can be ordered by the number of strokes as in traditional dictionaries.

## 4 For the Cell Phones

The project is still under progress. Below, we can see pictures taken from an experiment performed in Japan under the supervision of the third author. In figure 6, we can see a snap shot of the screen. On the screen, the different parts

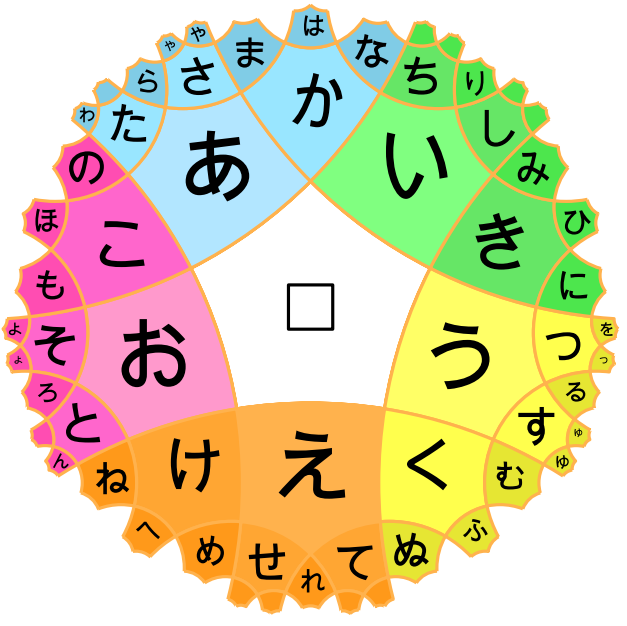



**Fig. 5.** The Japanese keyboard

of a mobile phone are represented, in particular the screen of the mobile phone. In the display of the figure, the hiraganas are displayed in the reverse order with respect to the order of figure 5.

But the work did not stop with tests on a computer. It is now on the implementation on an actual mobile phone. Figure 9 displays a picture of what can be seen by a user of an actual mobile phone on which the keyboard with the pentagrid appears on its screen.

Our VirHKey is implemented on a platform of *iappli Development Kit for DoJa (FOMA) Version 3.00* which is a Java-based software development workbench supplied with NTT Docomo Inc. for cell phone. Figure 6 is a screen snapshot of the application software.

Figure 7 illustrates some functions for inputting Japanese characters using the VirHKey implemented on the cell phone. It also gives a correspondence between a screen and keys on the cell hpone. Each key indicated by numbers operates as follows:

- **1**: **terminate** the VirHKey
- **2**: **input** a character displayed at center cell
- **3**: **change** a default input mode into a special character input mode
- **4, 5, 6, 8, and 9**: **move** a character around into the center cell
- **7**: **display** characters being input
- **10**: **delete** one character

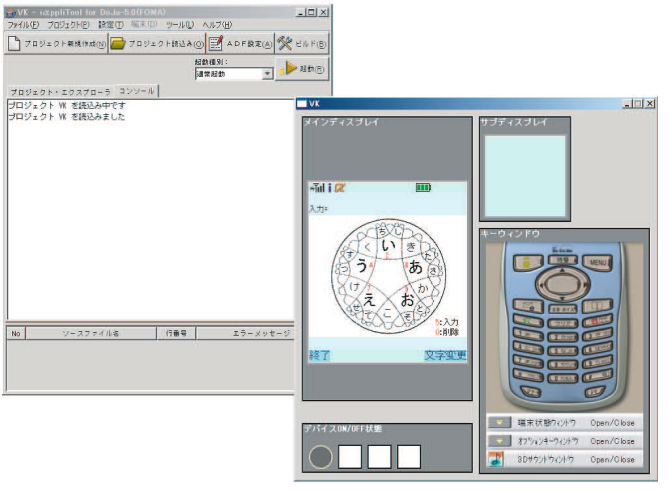


**Fig. 6.** Screen snapshot of *iappli Development Kit for DoJa* developing the VirHKey

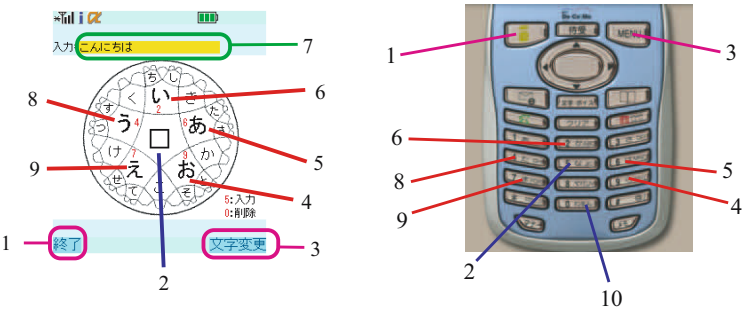



**Fig. 7.** Screen snapshot of *Functions for inputting Japanese characters using the VirHKey*

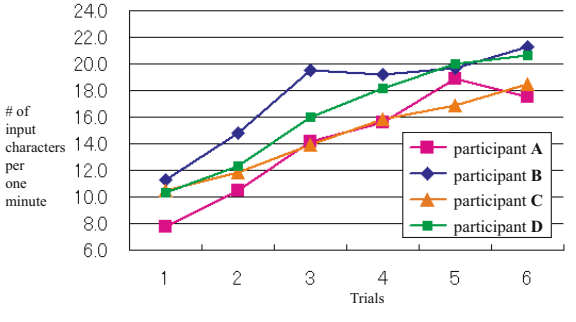



**Fig. 8.** Screen snapshot of *Usability test for inputting Japanese characters using the VirHKey*


**Fig. 9.** The Japanese keyboard on an actual cell phone, from an experiment at the University of Electro-communications of Osaka

We have made a small usability experiment for inputting Japanese character using the VirHKey. See figure 8. The vertical axis shows the number of characters input for one minute by each testee. It is noted that input errors are excluded. The horizontal axis means the trials. As can be seen, input performance at each testee is growing progressively to about twice in comparison with the initial performance, however, its absolute performance is not so good in comparison with the standard Japanese character input scheme. The biggest reason is that all of the testees are not familiar with the character arrangement in VirHKey. The Japanese consists of many characters including alphabets in a usual communications on mobile phones. We usually have to hit several keys to get a desired

character. For example, to select "(ho)" one has to press a key at six times in a standard input scheme. But on VirHkey one has to press a key 9, 2, 4 and key 5 at each stroke, totally four times. The number of strokes is smaller than the standard way, however, we have to change the key at each stroke. The input performance should be improved after trials. We are convinced the VirHkey would be efficient useful for general-purpose intelligent terminals with small display areas and devices where relatively large number of characters must be tackled.

This is very interesting! However, the work does not stop here. As previously mentioned, the project deals with kanjis too. For this, the big problem is the input of all the kanjis and their associations to the hiraganas and katakanas. There are data bases for such associations but the related problems are, of course, beyond the scope of this paper.

## 5 Conclusion

It seems to us that this project is very promising. The tests on users seems to indicate that the proposed tool is very efficient. We are convinced that the project, completed with the kanji version will meet a great success.

## References

1. Martin, B.: VirHKey: a VIRtual Hyperbolic KEYboard with gesture interaction and feedback for mobile devices. In: Proceedings of MobileHCI 2005, Salzburg, Austria, September 2005, pp. 99–106 (2005)
2. Chelghoum, K., Margenstern, M., Martin, B., Pecci, I.: Palette hyperbolique : un outil pour interagir avec des ensembles de données. In: IHM 2004, Hyperbolic chooser: a tool to interact with data, Namur, Belgium (September 2004)
3. Margenstern, M.: New Tools for Cellular Automata of the Hyperbolic Plane. Journal of Universal Computer Science 6(12), 1226–1252 (2000)
4. Margenstern, M.: Cellular Automata and Combinatoric Tilings in Hyperbolic Spaces, a survey. In: Calude, C.S., Dinneen, M.J., Vajnovszki, V. (eds.) DMTCS 2003. LNCS, vol. 2731, pp. 48–72. Springer, Heidelberg (2003)
5. Margenstern, M.: Cellular Automata in Hyperbolic Spaces, vol. 1, Theory, 422 p. Old City Publishing, Philadelphia (2007)

# Quick Energy Drop in Stochastic 2D Minority

Damien Regnault

Université de Lyon, IXXI-LIP, École Normale Supérieure de Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France
http://perso.ens-lyon.fr/damien.regnault

**Abstract.** Cellular automata are usually updated synchronously and thus deterministically. The question of stochastic dynamics arises in the development of cellular automata resistant to noise [1] and in simulation of real life systems [2]. Synchronous updates may not be a valid hypothesis for such simulations and most of these studies use stochastic versions of cellular automata.

In [3,4,5,6], the authors study different classes of cellular automata under fully asynchronous dynamics (only one random cell fires at each time step) and $\alpha$-asynchronous dynamics (each cell has a probability $\alpha$ to fire at each time step). They develop tools and methods to ease the study of other cellular automata. In [4,6], they analyze 2D Minority under fully asynchronous dynamics for Von Neumann and Moore neighborhoods. The behavior of this cellular automaton under these dynamics is surprisingly rich. The energy of a configuration is an useful information. In [4], it is proved that configurations of energy greater than $\frac{5mn}{3}$ (where $m$ and $n$ are the length and the width of the configuration) will not appear in the long range behavior of 2D minority for Von Neumann neighborhood. In this paper we improve this bound to $18\lceil\frac{m}{4}\rceil\lceil\frac{n}{4}\rceil$. The proof is based on an enumeration of cases made by computer. This method could be easily tuned for other cellular automata or neighborhoods.

## 1 Introduction

Cellular automata are attractive models for complex systems in various fields, like physics, biology or social sciences. An example of challenging issue in biology is to predict the expression of genes in a set of cells which share the same gene regulatory network. Cellular automata can be used to model such systems [2,7]. Classically cellular automata update synchronously. However models for natural phenomena rather update asynchronously.

Empirical studies [8,9,10] have shown how widely the behavior can change when introducing asynchronism. However only few mathematical analyses are available and they mainly concern one-dimensional stochastic cellular automata [3,5,11]. Providing analyses of 2D rules remains a real challenge. For instance the mean-field approach does not succeed in approximating tightly such stochastic dynamics [12]. The cellular automaton 2D Minority is studied under fully asynchronous dynamics (at each time step only one random cell chosen uniformly

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 307–314, 2008.

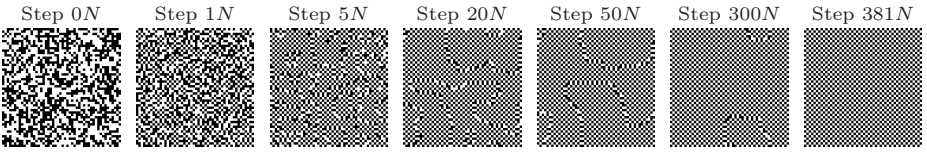



**Fig. 1.** A typical evolution of 2D Minority under fully asynchronous dynamics on a configuration of $N = 50 \times 50$ cells

fires) in [4,6]. 2D Minority is a rule with negative feedback. Such rules are acknowledged to be harder to analyze [13]. In [4,6] the authors develop tools for studying 2D asynchronous cellular automata. Ongoing works show their results hold for the classes of 2D Threshold cellular automata. This class has been intensively studied under synchronous dynamics [14] and exhibits interesting behaviors under fully asynchronous dynamics.

In this paper, we continue the study of 2D Minority under fully asynchronous dynamics with Von Neumann neighborhood. Figure 1 shows a classical evolution of fully asynchronous 2D Minority. In [4], it is proved that any initial configuration will reach almost surely a stable configuration after $O(N^{2N+1})$ steps on expectation (where $N$ is the number of cells of the configuration). Nevertheless it is conjectured that this time is polynomial in $N$. The energy of a configuration is useful to describe the behavior of 2D Minority. This notion was first introduced in Ising model [15] or Hopfield networks [13]. In [4], it is shown that:

1. The energy of a configuration is between 0 and $4N$.
2. The energy of a stable configuration is between 0 and $N$.
3. The energy is non-increasing according to time.
4. The energy drops below $5N/3$ after $O(N^2)$ steps on expectations.
5. The dynamics converges almost surely to a stable configuration from an initial *bounded* configuration in $O(N^2)$ steps on expectation. Bounded configurations correspond to final steps of a classical execution of $2D$ Minority (After step $50N$ the configurations of figure 1 are bounded configurations).

In this paper we improve point 4: the energy drops below $18\lceil\frac{m}{4}\rceil\lceil\frac{n}{4}\rceil$ after $O(N^2)$ steps on expectation (where $m$ and $n$ are the length and the width of the configuration: $N = mn$). The proof relies on an enumeration of cases by computer. Nevertheless, our method could be easily adapted for other cellular automata or neighborhoods. Moreover, a more accurate bound could be found with more precise computations (but cannot be lower than $N$ because of point 2). Finally our results is of interest to prove the conjecture. Only a small gap separates configurations of energy less than $18\lceil\frac{m}{4}\rceil\lceil\frac{n}{4}\rceil$ and bounded configurations.

## 2 Definitions

We consider in this paper the 2D 2-states cellular automaton Minority under fully asynchronous dynamics over finite configurations with periodic boundary

conditions. Except for the main theorem, all notations and results are introduced or proved in [4].

**Definition 1 (Configuration).** *We are given two positive integers $n$ and $m$, let $N = nm$. We denote by $\mathbb{T} = \mathbb{Z}_n \times \mathbb{Z}_m$ the set of* cells *and $Q = \{0, 1\}$ the set of* states *(0 stands for white and 1 for black in the figures). We consider the* Von Neumann neighborhood*: the neighbors of each cell $(i, j)$ are the four cells $(i, j \pm 1)$ and $(i \pm 1, j)$. A $n \times m$-*configuration *$c$ is a function $c : \mathbb{T} \to Q$; $c_{ij}$ is the* state *of the cell $(i, j)$ in configuration $c$.*

**Definition 2 (Stochastic 2D Minority).** *We consider the* fully asynchronous dynamics of 2D Minority*. Time is discrete and let $c^t$ denote the configuration at time $t$; $c^0$ is the* initial configuration*. The configuration at time $t + 1$ is a random variable defined by the following process: a cell $(i, j)$ is selected uniformly at random in $\mathbb{T}$ and its state is updated to the minority state in its neighborhood (we say that cell $(i, j)$* fires *at time $t$), all the other cells remain in their current state:*

$$c_{ij}^{t+1} = \begin{cases} 1 \text{ if } (c_{ij}^t + c_{i-1,j}^t + c_{i+1,j}^t + c_{i,j-1}^t + c_{i,j+1}^t) \leq 2 \\ 0 \text{ otherwise} \end{cases}$$

*and $c_{kl}^{t+1} = c_{kl}^t$ for all $(k, l) \neq (i, j)$. A cell is said* active *if its state would change if fired.*

**Definition 3 (Potential).** *The* potential *$v_{ij}$ of cell $(i, j)$ is the number of its neighboring cells in the same state as itself. By definition, if $v_{ij} \leq 1$, then the cell is in the minority state in its own neighborhood and is thus* inactive *(its state will not change when fired); whereas, if $v_{ij} \geq 2$ then the cell is* active *and its state will change if fired.*

**Definition 4 (Energy).** *The* energy *$E$ of configuration $c$ is defined as: $E = \sum_{(i,j)\in\mathbb{T}} v_{ij}$.*

Thus the energy of a configuration is positive and less than $4N$. The next fact shows that an energy drop is irreversible.

**Proposition 1 (Energy is non-increasing).** *Under fully asynchronous dynamics, the energy is a non-increasing function of time and decreases each time a cell with potential $\geq 3$ fires.*

The next theorem shows that configurations of high energy will not appear in the long range behavior of 2D Minority. The next parts are dedicated to the proof of this theorem.

**Main Theorem 1 (Initial energy drop).** *The random variable $T = \min\{t : E(c^t) < 18\lceil \frac{m}{4} \rceil \lceil \frac{n}{4} \rceil\}$ is almost surely finite and $E[T] = O(N^2)$.*

## 3 Proof

The proof is based on the correlation between high energy configurations and local patterns. For example a configuration of energy higher than $3N$ has at least a cell of potential greater than 3. Firing such a cell decreases the energy of the configuration. The bound of $5N/3$ proved in [4] was obtained by considering two facts. Firing two adjacent cells in opposite states of potential 2 decreases the energy of the configuration in two steps. Configurations of energy higher than $5N/3$ admit at least a cell of potential 3 or two adjacent cells in opposite states of potential 2. Here we formalize this approach.

### 3.1 Decreasing Sequences

**Definition 5 (decreasing sequence).** *Given a configuration $c$, a finite sequence $\mathcal{S} = (c_i)_{1\leq i\leq j}$ of cells ($j$ is the* length *of the sequence) is a decreasing sequence if firing the $j$ cells $c_1$ to $c_j$ leads to a configuration of lower energy after $j$ steps. The neighborhood $\mathcal{N}(\mathcal{S})$ of a sequence is a set containing the cells of the sequence and their neighbors.*

Note that if for a configuration $c$ a decreasing sequence of length $j$ exists then there exist decreasing sequences of length $k \geq j$. A sequence remains decreasing by adding any cell at the end of the sequence.

**Fact 2 (evolution of a decreasing sequence).** *Given a configuration $c^t$ and a decreasing sequence $\mathcal{S} = (c_i)_{1\leq i\leq j}$ of length $j$. then :*

- *with probability $\frac{1}{N}$ the cell $c_1$ fires : either the energy decreases or $\mathcal{S}' = (c_i)_{2\leq i\leq j}$ is a decreasing sequence of $c^{t+1}$.*
- *with probability $\frac{|\mathcal{N}(\mathcal{S})|-1}{N}$ a cell $c_0$ of $\mathcal{N}(\mathcal{S})$ different from $c_1$ fires : either the potential of $c_0$ is $\geq 3$ and the energy drops or the potential of $c_0$ is 2 and $\mathcal{S}' = (c_i)_{0\leq i\leq j}$ is a decreasing sequence or the potential of $c_0$ is $\leq 1$ and $\mathcal{S}$ is still a decreasing sequence.*
- *with probability $\frac{N-|\mathcal{N}(\mathcal{S})|}{N}$ a cell which is not in $\mathcal{N}(\mathcal{S})$ fires: the energy may drop but $\mathcal{S}$ is still a decreasing sequence.*

**Definition 6 (hypothesis $\mathcal{H}(E, j)$).** *We call $\mathcal{H}$(E,j) the hypothesis that all configurations of energy $E$ admit a decreasing sequence of size less $j$.*

**Definition 7 (random walks $\mathcal{RW}_j$).** *Given $j \in \mathbb{N}^*$ the random walk $\mathcal{RW}_j$ is a sequence of random variables $(X^t)_{t\geq 0}$ taking their value in $\{0, \ldots, j\}$ such that $X^0 = j$ and :*

| | $P(X^{t+1} = X^t - 1)$ | $P(X^{t+1} = X^t)$ | $P(X^{t+1} = X^t + 1)$ |
|---|---|---|---|
| *if* $X^t = 0$ | *0* | *1* | *0* |
| *if* $X^t \in \{1, \ldots, j-1\}$ | $\frac{1}{N}$ | $\frac{N-5j}{N}$ | $\frac{5j-1}{N}$ |
| *if* $X^t = j$ | $\frac{1}{N}$ | $\frac{N-1}{N}$ | *0* |

**Fact 3.** *Given $j \in \mathbb{N}^*$ and the random walk $\mathcal{RW}_j = (X^t)_{t\geq 0}$ then $T' = \min\{t | X^t = 0\}$ is almost surely finite and $E[T'] = O(j^j N)$.*

**Lemma 1.** *Consider a configuration $c^0$ of energy $E$ and $j \in \mathbb{N}$, suppose that $\mathcal{H}(E,j)$ is true then the random variable $T = \min\{t : E(c^t) < E\}$ is almost surely finite and $E[T] = O(j^j N)$.*

*Proof.* The proof is based on a the coupling between $(c^t)_{t\geq 0}$ and $\mathcal{RW}_j = (X^t)_{t\geq 0}$. To define the coupling we need that at each time step $t$ either $E(c^t) < E$ or $X^t = 0$ or there exists a decreasing sequence $\mathcal{S} = (c_i)_{1\leq i\leq X^t}$ for configuration $c^t$. At each time step $t$ we update $X^t$ and $c^t$ according to the following coupling : if $E(C^t) < E$ or $X^t = 0$ then $X^t$ updates according to the rule of the random walk and independently fire a random cell of $c^t$ chosen uniformly. Otherwise $X^t = k$ with $k \neq 0$ and we consider a decreasing sequence $S^t = (c_i)_{1\leq i\leq k}$ of $c^t$ of size $k$ and :

- if $X^t = j$, with probability $\begin{cases} \frac{1}{N} & X^{t+1} = X^t - 1 \text{ and } c_1 \text{ fires.} \\ \frac{N-1}{N} & X^{t+1} = X^t \text{ and one cell is selected uniformly at random among the } N-1 \text{ other cells.} \end{cases}$
- if $0 < X^t < j$, with prob. $\begin{cases} \frac{1}{N} & X^{t+1} = X^t - 1 \text{ and } c_1 \text{ fires.} \\ \frac{|\mathcal{N}(\mathcal{S})|-1}{N} & X^{t+1} = X^t+1 \text{ and one cell is selected uniformly at random among } \mathcal{N}(\mathcal{S}). \\ \frac{5j-|\mathcal{N}(\mathcal{S})|}{N} & X^{t+1} = X^t+1 \text{ and one cell is selected uniformly at random among the } N - |\mathcal{N}(\mathcal{S})| \text{ other cells.} \\ \frac{N-5j}{N} & X^{t+1} = X^t \text{ and one cell is selected uniformly at random among the } N - |\mathcal{N}(\mathcal{S})| \text{ other cells.} \end{cases}$

According to this coupling, each cell of $c^t$ is updated uniformly and $X^t$ evolves according to the rule of the random walk. Now we prove by recurrence over $t$ that this coupling is coherent that is to say that either $E(c^t) < E$ or $X^t = 0$ or there exists a decreasing sequence $\mathcal{S} = (c_i)_{1\leq i\leq X^t}$ for configuration $c^t$. At time $t = 0$, $X^0 = j$ and since $E(c^0) = E$ then by $\mathcal{H}(E,j)$ there exists a decreasing sequence of size $j$. Now if at time $t$ :

- $E(c^t) < E$ then $E(c^{t+1}) < E$ since energy is non-increasing (see Proposition 1).
- $X^t = 0$ then $X^{t+1} = 0$.
- $X^t = j$ and $E(c^t) = E$ then by hypothesis of induction there exists a decreasing sequence $\mathcal{S} = (c_i)_{1\leq i\leq j}$:
  If $X^{t+1} = X^t$ then either $E(c^{t+1}) < E$ or $E(c^{t+1}) = E$ and by $\mathcal{H}(E,j)$ there exists a decreasing sequence $\mathcal{S}'$ of size $j$.
  If $X^{t+1} = X^t - 1$ then cell $c^0$ fires and according to fact 2 either $E(c^{t+1}) < E$ or $E(c^{t+1}) = E$ and $(c_i)_{2\leq i\leq j}$ is a decreasing sequence.
- $X^t = k$ where $0 < k < j$ and $E(c^t) = E$ then by hypothesis of induction there exists a decreasing sequence $\mathcal{S} = (c_i)_{1\leq i\leq k}$:
  If $X^{t+1} = X^t$ then the fired cell is not in $\mathcal{N}(\mathcal{S})$ and either $E(c^{t+1}) < E$ or $E(c^{t+1}) = E$ and $\mathcal{S}$ is still a decreasing sequence.

If $X^{t+1} = X^t - 1$ then cell $c^1$ fires and according to fact 2 either $E(c^{t+1}) < E$ or $E(c^{t+1}) = E$ and $(c_i)_{2 \leq i \leq j}$ is a decreasing sequence.
If $X^{t+1} = X^t + 1$ then cell $c^0$ fires and according to fact 2 either $E(c^{t+1}) < E$ or $E(c^{t+1}) = E$ and $(c_i)_{0 \leq i \leq j}$ is a decreasing sequence.

Thus the coupling is well defined. We call $T' = \min\{t | X^t = 0\}$. Either $E(c^{T'-1}) < E$ or $E(c^{T'-1}) = E$ and at time $T' - 1$ a cell of potential $\geq 3$ (a decreasing sequence of size one) fires. Thus $T \leq T'$ and moreover $E[T] \leq E[T']$. According to lemma 3, $E[T'] = O(j^j N)$ and thus $E[T] = O(j^j N)$.

**Theorem 4.** *If for all configurations of energy $E$ there exists $j \in \mathbb{N}^*$ such that for all $E' > E$ the hypothesis $\mathcal{H}(E', j)$ are true then given a configuration $c^0$ the random variable $T = \min\{t : E(c^t) < E\}$ is almost surely finite and $E[T] = O(j^j N^2)$.*

*Proof.* We call $t_0 = 0$ and for $i \in \mathbb{N}^*$ if $E(c^{t_{i-1}} > E)$ we define $t_i$ as $t_i = \min\{t | E(c^{t_i}) < E(c^{t_{i-1}})\}$. We define $k \in \mathbb{N}$ such that $T = t_k$. According to lemma 1 for $i \leq k$, $E[t_i - t_{i-1}] = O(j^j N)$. And since the energy of a configuration is between 0 and $4N$ : $k < 4N$ and then $E[T] = O(j^j N^2)$

In the next part, we present a method to prove $\mathcal{H}(E, j)$ by considering finite local patterns.

### 3.2 Enumeration of Acceptable Coloring

We consider finite patterns of cells and colorings (i.e. the state of the cells) of a pattern. We present an algorithm which enumerates all the possible colorings of a pattern and eliminates colorings which imply a quick energy drop. The remaining colorings are called acceptable. Finally, we compute the maximum energy contained by a pattern matching an acceptable coloring.

**Definition 8 (Pattern).** *A pattern $\mathcal{P}$ is a subset of $\{0, \ldots, n\} \times \{0, \ldots, m\}$. The pattern $\mathcal{P}$ centered on $c_{ij}$ is the set of cells $\cup_{(k,l) \in \mathcal{P}} \{c_{i+k,j+l}\}$.*

**Definition 9 (Interior).** *The interior of $\mathcal{P}$ is the pattern $\{(i,j) \in \mathcal{P} | (i \pm 1, j) \in \mathcal{P}$ and $(i, j \pm 1) \in \mathcal{P}\}$.*

**Definition 10 (Coloring).** *A coloring $f$ of $\mathcal{P}$ is a function $f : \mathcal{P} \to Q$. We say that the pattern $\mathcal{P}$ centered on cell $c_{i,j}$ matches coloring $f$ if for any $(k,l) \in \mathcal{P}$, we have $f(k,l) = c_{i+k,j+l}$. We denote by $\mathcal{C}^{\mathcal{P}}$ the set of all the colorings of pattern $\mathcal{P}$.*

**Definition 11 ($k$-acceptable).** *A coloring $f$ of $\mathcal{C}^{\mathcal{P}}$ is $k$-acceptable if there exists a configuration $c$ such that there is no decreasing sequence of length $k$ in $c$ and there is a cell $c_{i,j}$ such that the pattern $\mathcal{P}$ centered on $c_{i,j}$ matches $f$.*

If a coloring $f$ is not $k$-acceptable then $f$ is not $k'$-acceptable for all $k' > k$.

**Definition 12 (Relative Potential).** *Consider a pattern $\mathcal{P}$ centered on cell $c_{ij}$ which matches a coloring $f$ of $\mathcal{C}^{\mathcal{P}}$ and $(k,l) \in \mathcal{P}$, the relative potential $v'_{kl}(f)$ is defined as the number of neighboring cells of cell $c_{i+k,j+l}$ which are in the same state as $c_{i+k,j+l}$ and which are in the pattern $\mathcal{P}$ centered on cell $c_{ij}$.*

**Fact 5.** *Consider a coloring $f$ of $\mathcal{C}^{\mathcal{P}}$, $(k,l) \in \mathcal{P}$ and a pattern $\mathcal{P}$ centered on cell $c_{i,j}$ matching $f$. If $(k,l)$ is in the interior of $\mathcal{P}$ then $v'_{kl}(f) = v_{(k+i)(l+j)}$, otherwise $v'_{kl}(f) \leq v_{(k+i)(l+j)}$.*

A coloring $f$ of $\mathcal{C}^{\mathcal{P}}$ admitting $(i,j) \in \mathcal{P}$ such that $v'_{ij}(f) \geq 3$ is not 1-acceptable.

**Definition 13.** *For each pattern $\mathcal{P}$, we define the oriented graph $G^{\mathcal{P}} = (\mathcal{C}^{\mathcal{P}}, E)$. There is an oriented edge between coloring $f_1$ to coloring $f_2$ if :*

- *there is $(i,j) \in \mathcal{P}$ such that $f_1(i,j) \neq f_2(i,j)$*
- *for all $k,l \neq (i,j)$, $f_1(k,l) = f_2(k,l)$*
- *$v'_{i,j}(f_1) = 2$*

**Fact 6.** *If there is a path of length $i$ from a coloring $f_1$ to a coloring $f_2$ in $G^{\mathcal{P}}$ and if $f_2$ is not $k$-acceptable then $f_1$ is not $(k+i)$-acceptable.*

**Algorithm 1.** *Given a set of coloring $\mathcal{C}_0$, this algorithm computes a set of coloring $\mathcal{C}_1$ such that $\mathcal{C}_1 \subset \mathcal{C}_0$ and for all coloring $f$ of $\mathcal{C}_0 \backslash \mathcal{C}_1$ there exists a path in $G^{\mathcal{P}}$ from $f$ to $f'$ such that $f' \in \mathcal{C}^{\mathcal{P}} \backslash \mathcal{C}_1$ or $f'$ is not 1-acceptable:*

*1 - Compute $\mathcal{C'}_0 = \{f \in \mathcal{C}_0 | \forall (i,j) \in \mathcal{P}, v'_{ij}(f) \leq 2\}$.*

*2 - Compute the graph $G' = (\mathcal{C}'_0 \cup \{d\}, E)$ where there is an edge between two colorings $f$ and $f'$ if and only if there is an edge between $f$ and $f'$ in $G^{\mathcal{P}}$; there is an edge between coloring $f$ and $d$ if there exists a coloring $f' \in \mathcal{C}^{\mathcal{P}} \backslash \mathcal{C}'_0$ such that there is an edge between $f$ and $f'$ in $G^{\mathcal{P}}$.*

*3 - Compute $C_1 = \{f \in \mathcal{C}'_0 |$ there is no path between $f$ and $d$ in $G'\}$.*

The complexity of this algorithm is $O(|\mathcal{C'}_0|^2 + |\mathcal{C}_0|)$.

**Fact 7.** *Consider a set of colorings $\mathcal{C}_0 \subset \mathcal{C}^{\mathcal{P}}$ and $k'$ such that colorings of $\mathcal{C}^{\mathcal{P}} \backslash \mathcal{C}_0$ are not $k'$-acceptable. Consider $\mathcal{C}_1$ the result of algorithm 1 applied to set $\mathcal{C}_0$, then there exists $k$ such that all colorings of $\mathcal{C}^{\mathcal{P}} \backslash \mathcal{C}_1$ are not $k$-acceptable. In particular, this result holds for $\mathcal{C}_0 = \mathcal{C}^{\mathcal{P}}$.*

We apply this algorithm to pattern $\mathcal{P} = \{0, \ldots, 3\}^2$ in order to prove the main theorem. All enumerations of cases in the next proofs are made by computer.

**Theorem 8.** *Consider a configuration $c$. There exists $k$ such that if there exists $(i,j) \in \mathbb{T}$ such that $\sum_{(k,l) \in \{0,\ldots,3\}^2} v_{(i+k)(j+l)} > 18$ then there exists a decreasing sequence of length $k$ in $c$.*

*Proof.* Consider the pattern $\mathcal{P} = \{0, \ldots, 3\}^2$. Applying algorithm 1 to $\mathcal{C}_0 = \mathcal{C}^{\mathcal{P}}$ leads to a set $\mathcal{C}_1$ which contains 1092 colorings whereas $|\mathcal{C}_0| = 65536$. We now consider the pattern $\mathcal{P}' = (\{-1, \ldots, 4\}^2 \backslash \{(-1,-1), (-1,4), (4,-1), (4,4)\})^2$. The interior of $\mathcal{P}'$ is $\mathcal{P}$. Applying algorithm 1 to set $\mathcal{C}'_0 = \{f \in \mathcal{C}^{\mathcal{P}'} | f_{|\mathcal{P}} \in \mathcal{C}_1\}$ leads to a set $\mathcal{C}'_1$. According to fact 7, there exists $k$ such that all colorings of $\mathcal{C}^{\mathcal{P}'} \backslash \mathcal{C}_1$ are not $k$-acceptable. Computing $\mathcal{C}'_1$ using $\mathcal{C}_1$ is much faster than computing $\mathcal{C}'_1$ from $\mathcal{C}^{\mathcal{P}'}$. For each coloring $f$ of $\mathcal{C}_1$, we have $\max_{f' \in \mathcal{C}'_1 | f'_{|\mathcal{P}} = f} (\sum_{(i,j) \in \mathcal{P}} v'_{ij}(f')) \leq 18$. According to fact 5, if the pattern $\mathcal{P}'$ centered on cell $(i,j)$ matches the coloring $f'$ of $\mathcal{C}^{\mathcal{P}'}$ then $\sum_{(k,l) \in \mathcal{P}} v'_{kl}(f') = \sum_{(k,l) \in \mathcal{P}} v_{(i+k)(j+l)}$. Thus if a pattern $\mathcal{P}$ centered on $c_{ij}$ is such that $\sum_{(k,l) \in \mathcal{P}} v_{(i+k)(j+l)} > 18$, then the corresponding coloring is not $k$-acceptable. Thus there is a decreasing sequence of length $k$ in $c$.

**Theorem 9.** *There exists $j$ such that $\mathcal{H}(E,j)$ is true for every $E > 18\lceil\frac{m}{4}\rceil\lceil\frac{n}{4}\rceil$.*

*Proof.* Consider a configuration $c$ and the pattern $\mathcal{P} = \{0,\dots,3\}^2$. There exists $j$ such that if $c_{ij}$ is such that $\sum_{(k,l)\in\mathcal{P}} v_{(i+k)(j+l)} > 18$ then there exists a decreasing sequence of length $j$ in $c$. Suppose that there is no cell $c_{ij}$ such that $\sum_{(k,l)\in\mathcal{P}} v_{(i+k)(j+l)} > 18$. If $n = 0 \bmod 4$ and $m = 0 \bmod 4$ then $E(c) = \sum_{(i,j)\in\mathcal{T}} v_{ij} = \sum_{0\leq i<n/4, 0\leq j<m/4}(\sum_{(k,l)\in\mathcal{P}}(v_{4i+k,4j+l})) \leq \frac{18nm}{16}$. If $n \neq 0 \bmod 4$ and $m = 0 \bmod 4$ then $E(c) \leq \sum_{0\leq i\leq n/4, 0\leq j<m/4}(\sum_{(k,l)\in\mathcal{P}}(v_{4i+k,4j+l})) \leq 18\lceil\frac{n}{4}\rceil\frac{m}{4}$. In the two other cases we also have $E(c) \leq \lceil\frac{m}{4}\rceil\lceil\frac{n}{4}\rceil$. This imply that $\mathcal{H}(E,j)$ is true for $E > 18\lceil\frac{m}{4}\rceil\lceil\frac{n}{4}\rceil$.

Applying this result to theorem 4 proves the main theorem.

## References


1. Gács, P.: Reliable computation with cellular automata. Journal of Computer andSystem Sciences 32(1), 15–78 (1986)
2. Ermentrout, G.B., Edlestein-Keshet, L.: Cellular automata approaches to biological modelling. Journal of Theoretical Biology 160, 97–133 (1993)
3. Fatés, N., Morvan, M., Schabanel, N., Thierry, E.: Asynchronous behaviour of double-quiescent elementary cellular automata. In: Jedrzejowicz, J., Szepietowski, A. (eds.) MFCS 2005. LNCS, vol. 3618, pp. 316–327. Springer, Heidelberg (2005)
4. Regnault, D., Schabanel, N., Thierry, E.: Progresses in the analysis of stochastic 2d cellular automata: A study of asynchronous 2d minority. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 320–332. Springer, Heidelberg (2007)
5. Fatés, N., Regnault, D., Schabanel, N., Thierry, E.: Asynchronous behaviour of double-quiescent elementary cellular automata. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887. Springer, Heidelberg (2006)
6. Regnault, D., Schabanel, N., Thierry, E.: On the analysis of "simple" 2d stochastic cellular automata. In: Proc. of LATA 2008. Springer, Heidelberg (Volume to appear, 2008)
7. Demongeot, J., Aracena, J., Thuderoz, F., Baum, T.P., Cohen, O.: Genetic regulation networks: circuits, regulons and attractors. C.R. Biologies 326, 171–188 (2003)
8. Bersini, H., Detours, V.: Asynchrony induces stability in cellular automata based models. In: Proceedings of Articial Life IV, pp. 382–387. MIT Press, Cambridge (1994)
9. Fatés, N., Morvan, M.: An experimental study of robustness to asynchronism for elementary cellular automata. Complex Systems 16(1), 1–27 (2005)
10. Schónsch, B., de Roos, A.: Synchronous and asynchronous updating in cellular automata. BioSystems 51, 123–143 (1999)
11. Fukś, H.: Probabilistic cellular automata with conserved quantities. Nonlinearity 17(1), 159–173 (2004)
12. Balister, P., Bollobás, B., Kozma, R.: Large deviations for mean fields models of probabilistic cellular automata. Random Structures & Algorithms 29, 399–415 (2006)
13. Rojas, R.: Neural Networks: A Systematic Introduction, ch. 13 - The Hopfield Model. Springer, Heidelberg (1996)
14. Goles, E., Martinez, S.: Neural and automata networks, dynamical behavior and applications. Maths and Applications, vol. 58. Kluwer Academic Publishers, Dordrecht (1990)
15. McCoy, B., Wu, T.T.: The Two-Dimensional Ising Model. Harvard University Press (1974)

# The Diffusion of Perturbations in a Model of Coupled Random Boolean Networks

Roberto Serra[1], Marco Villani[1], Chiara Damiani[1], Alex Graudenzi[1], and Annamaria Colacci[2]

[1] Dipartimento di scienze sociali, cognitive e quantitative
Università di Modena e Reggio Emilia, Via Allegri 9, 42100 Reggio Emilia, Italia
{rserra,mvillani,chiara.damiani,alex.graudenzi}@unimore.it
[2] Excellence Environmental Carcinogenesis, Environmental Protection and Health Prevention Agency Emilia-Romagna, viale Filopanti 22, Bologna, Italia

**Abstract.** Deciphering the influence of the interaction among the constituents of a complex system on the overall behaviour is one of the main goals of complex systems science. The model we present in this work is a 2D square cellular automaton whose of each cell is occupied by a complete random Boolean network. Random Boolean networks are a well-known simplified model of genetic regulatory networks and this model of interacting RBNs may be therefore regarded as a simplified model of a tissue or a monoclonal colony. The mechanism of cell-to-cell interaction is here simulated letting some nodes of a particular network being influenced by the state of some nodes belonging to its neighbouring cells. One possible means to investigate the overall dynamics of a complex system is studying its response to perturbations. Our analyses follow this methodological approach. Even though the dynamics of the system is far from trivial we could show in a clear way how the interaction affects the dynamics and the global degree of order.

**Keywords:** genetic network model, random Boolean network, cellular automaton, interaction, cell-criticality.

## 1 Introduction

One of the main goals of complex systems science is deciphering the influence of the interactions among the system components on the global dynamics. In this work we introduce a model in which another level of complexity is added: the system components are themselves well-know models of complex systems, i.e. random Boolean networks.

Random Boolean networks are a simplified model of genetic networks [1] and the statistical analysis of their dynamical properties has proven fruitful in the description of general emerging features of real networks [2,3,4,5,6].

The theme of interaction turns out to be deeply relevant when speaking of the so-called "criticality" of living organisms. The idea that evolution would drive living organism in a region of the space of parameters close to the boundary between order and chaos [2,7] is an intriguing general hypothesis to investigate

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 315–322, 2008.

and, since it applies to organisms as a whole rather than to their individual constituents, it makes particular sense to study the way by which the single elements constituting the whole system (i.e. single cells) interact and how their interaction affects the overall dynamical regime. Therefore, the aim of this work is to analyse the relationship between the dynamics of a single, isolated RBN (which has been extensively studied in the past) and that of a collection of interacting networks: the model we are going to present is a 2-D lattice cellular automaton in which of each cell (which is meant to simulate a biological cell) hosts a complete RBN. This model could be regarded as a simplified description of a tissue in a multicellular organism, or of a colony of unicellular organisms: at this level of modelling the two cases are rather similar, since all that matters is that neighbouring cells influence each other.[1] The particular kind of cell-to-cell interaction we want to represent implicates that the state of a cell is determined by both its own genetic network and the state and of its neighbouring cells. Cellular automata are particularly appropriate to simulate the evolution of phenomena that depends on local rules, since every entity of the system change its state taking into account what happens in its neighbourhood[12].

A particularly effective means to examine the dynamical regimes of complex systems is studying their response to perturbations. The analyses presented in this work follow this approach, in order to root out the different responses to small perturbation in case of either isolated or coupled networks.

## 2 Random Boolean Networks (RBN)

For an exhaustive description of the model of random Boolean networks please refer to [2,8,9]. Here we will only outline its main features.

A RBN is an oriented graph constituted of $N$ Boolean nodes, which represent the genes of a specific genetic network. A node is active (value = 1) if its corresponding gene synthesises its protein, inactive (value = 0) otherwise. The direct or indirect influences of genes on the expression of other genes in real networks are represented in the model by directed links (if the activation of gene $A$ influences the activation of gene $B$, node $A$ will be an input of node $B$). Therefore, the activation of a certain node depends on the value of its input nodes, according a specific Boolean function. The updating of the network is synchronous, the time is discrete and both the topology and the Boolean function associated to each node do not change in time (this is the so-called *quenched* model [9]. In "classical" RBNs each node has the same number of ingoing connections kin and its input nodes are chosen at random with uniform probability among the remaining $N - 1$ nodes (self-coupling and multiple connections are forbidden). The analysis of the dynamics that show up in RBNs reveals the presence of two typical dynamical regimes, which can be defined as "ordered" and "chaotic"

[1] Note that, according to the usual biological interpretation of RBNs, the attractor of a given network is associated to the cell type: therefore a tissue should be composed by cells which are all in the same attractor. This condition is not imposed in our model.

[2,9]. The dynamical regime of a RBN depends primarily on two parameters, the average connectivity of the network $K =< k_{in} >$ and the bias $p$.[2]

## 3 Multi Random Boolean Network (MRBN)

The model of *Multi random Boolean network* (MRBN) has been introduced in a previous work with different features and a different name [10]. In this section we will briefly describe its most important characteristics.

A Multi random Boolean network is a cellular automaton in each of whose cells is hosted a complete random Boolean network. In our case, we have a 2D square lattice automaton with $M^2$ cells. The neighbourhood we consider is of the von Neumann type (composed by the cell itself and its N, E, S, W neighbours) and the overall topology is toroidal. Every RBN of the MRBN is identical in

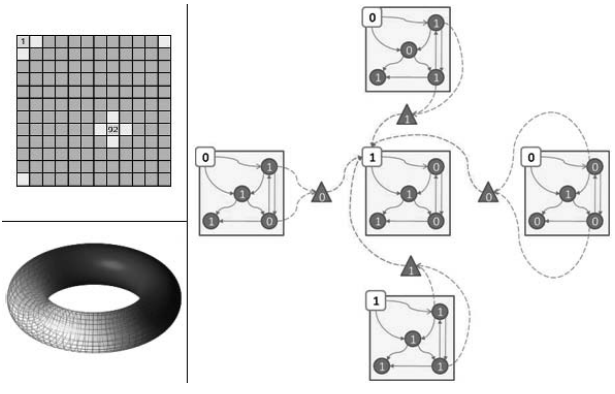


**Fig. 1.** (Left) The spatial shape of the CA is that of a torus, the neighbourhood is of the Von Neumann type. (right) The interaction mechanism for the central cell in the figure (nodes are numbered row by row from the top-left corner): nodes 2 and 5 of all the neighbouring cells can produce a specific signal molecule (triangle-shaped) according a specific Boolean function. If at least one of the neighbours of the central cell produces its signal molecule then the receptor of the central network (white-coloured) is active. Afterwards, the activation value of the receptor influences its output nodes, i.e. nodes 2 and 3 inside the central cell.

terms of number of Boolean nodes ($N$), topology (i.e. the ingoing and outgoing connections per node) and Boolean functions associated to each node, while the initial activation states of the genes are assigned randomly. The choice of homogeneous RBNs for all the cells of the automaton is due to the fact that the cells of a given multicellular organism or monoclonal colony share the same

[2] $p$ is the probability that the output of a certain Boolean function associated to a node is equal to 0. In this case we refer to the average value of $p$ for the whole set of $N$ nodes.

genetic material. In our model, the common structural features of the RBNs, i.e. number of nodes, topology and Boolean functions per node, define their common *genome*.

Real cells interact in many different ways. In this work we only consider the particular kind of interaction that occurs when certain molecules synthesized in a particular cell bind to specific membrane receptors of its neighbouring cells, so influencing the internal dynamic of the cells which host the receptors. In our model, every cell of the MRBN owns a certain number of nodes defined as *receptors*, whose ratio on the total number of nodes is defined as *interaction* strength, $f$. The activation state of a receptor is determined by the presence of a certain entity defined as *signal molecule*, produced by some genes in the neighbouring cells (according a specific Boolean function), but it is not affected by the presence of the signal molecule synthesized by the cell itself. The receptor has its own outputs and they can be either receptors or *internal* nodes of the network (Fig. 1). The interaction mechanism depends on the choice of an *interaction rule*. In this work we suppose that if at least one of the cells in the neighbourhood of cell A produces a signal molecule, then the correspondent receptor in cell $A$ is active (value = 1), inactive (value = 0) otherwise. Note that receptors are considered as actual nodes of the network (e.g. a network with $N = 100$ and $f = 10\%$ owns 90 internal nodes and 10 receptors).

It is important to specify that, in order to isolate the influence of the interaction strength, it is necessary to keep the other structural features of the MRBN fixed, i.e. dimension of the lattice $M$, topology of interaction, interaction rule and genome of each RBN.

## 4 Experiments

The simulations have been made on 100 distinct MRBNs, different in terms of the genome of their characterizing RBNs. All the MRBNs are 20*20 square lattices and the RBNs are composed of $N = 100$ nodes. [3] Past researches demonstrated how MRBNs whose RBNs are characterized by different genomes may show deeply different behaviours and this is the reason why it is important to make specific analysis on single MRBNs. The parameters of the RBNs in the cells are chosen in such a way to be "critical" [9], even though the dynamical behaviour of each single realization may be highly different from the average critical dynamics [11]. The networks are "classical" RBNs, with an equal number of incoming connections per node ($k_{in} = K = 2$). The input nodes are chosen at random with uniform probability excluding self-couplings and multiple connections. The Boolean functions are assigned with uniform likelihood on the set of all the possible functions. The initial states of the nodes are chosen at random for every RBN, independently from those of the other cells. In order to investigate the influence of the interaction strength on the dynamic, we analysed the differences

[3] Obviously, real networks are much larger in terms of number of nodes and interacting cells. Future researches will be aimed to study lager simulated networks and to investigate on possible scale properties of such systems.

in the behaviour of each single MRBN in presence of different values of the interaction strength.[4]

One possible method to investigate the dynamical regime of complex systems in general is to analyse their response to perturbation. A large sensitivity to the initial conditions of the system is usually related to disordered (or chaotic) systems, while, vice versa, a low sensitiveness (higher robustness) refers to ordered systems. For what concerns RBNs one possible means to discriminate the dynamical regimes is to observe the variation in time of the Hamming distance between a "control" network (wild type $WT$) and a perturbed network ($PN$). In our case the perturbation is the *flip* of one node chosen at random, i.e. the change of the activation value of one node in the initial condition of the network.[5] The variation of the Hamming distance in time is a relevant parameter since it provides a clear indication on how the dynamics of two systems diverge. In the case of a flip perturbation, a Hamming distance tending to 0 is peculiar of ordered networks, a value close to 1 is related to critical networks, while values higher than 1 refers to chaotic ones [9].In the simulations on MRBNs a node is chosen at random in a random cell and its initial activation value is flipped. It is then possible to calculate the variation of the Hamming distance of the whole automaton and the number of cell affected by the perturbations, i.e. the number of cells whose Hamming distance is higher than 0 after a certain transient.

## 5 Results

As clarified in the introduction, the primary aim of this research is deciphering the relation between the dynamical behaviour of a single random Boolean network and the emerging dynamics of a collection of coupled RBNs. We firstly analysed the variation of the Hamming distance in time of perturbed isolated RBNs with "critical" structural parameters. Although the average variation of the Hamming distance on the whole set of networks closely resembles the results of past studies of the same kind [9], there is an interesting aspect which is hidden by an analysis on average values. RBNs with identical structural parameters can indeed behave in a highly different way in response to perturbations and it is actually possible to group networks on the basis of the variation of the Hamming distance in time (Fig. 5(left)):

- *Ordered* behaviour: networks with average Hamming distance tending to values lower than 0.5
- *Chaotic* behaviour: networks with average Hamming distance tending to values higher than 1.5
- *Critical* behaviour: networks with average Hamming distance tending to values in the range 0.5 and 1.5.

[4] Every simulation on every MRBN is repeated 150 times. The simulation runs differ for the choice of the set of the receptors and for the initial condition of the nodes of the RBNs constituting the MRBN.

[5] Note that receptors and signal molecules also can be chosen for the flip.

Grouping the networks in the three classes above, we found out that: about 50% of the networks belong to the ordered group, 25% of the networks to the chaotic one and the remaining 25% belongs to the critical class. It is interesting to notice how the Hamming distance reaches its asymptotic value after a relatively small number of time steps for all the analysed RBNs. Coupling RBNs into MRBNs,

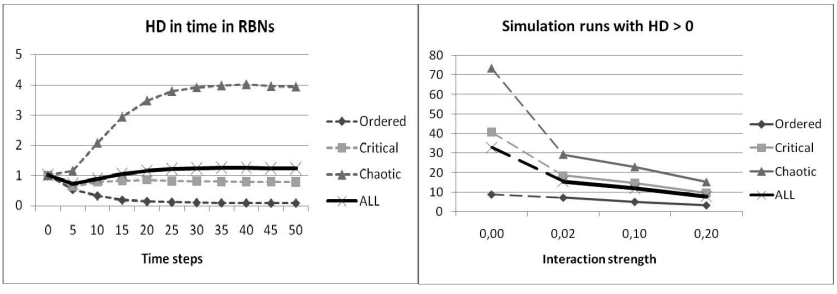



**Fig. 2.** (Left) Variation of the average Hamming distance in time for the three classes and for the complete set of 100 isolated RBNs (on 150runs). (right) Variation of the average simulation runs for the three classes and for the whole set of MRBNs in which the overall Hamming distance after 100 time steps is higher than 0, for different values of the interaction strength. The values in correspondence of $f = 0$ refer to the isolated RBNs.

we could study the changes of the dynamics in response to different value of the interaction strength. We focused our attention on some significant values of the interaction strength, i.e. 2%, 10%, 20%.[6] Once more it turns out to be fruitful to observe the behaviour of each MRBN separatedly; furthermore, we decided to keep the distinction in three groups on the basis of the dynamics of the isolated characterizing RBNs[7] (e.g. MRBNs whose characterizing RBNs have been signed as critical when analysed sigularly will be defined as *critical MRBNs*). Since for a certain number of MRBNs the Hamming distance reaches its asymptotic value after a large number of time steps, we decided to analyse the system after a precise transient (i.e. 100 time steps). In Fig. 5(right) we can notice how the number of runs in which the Hamming distance is higher than 0 after the transient (in other words, the number of runs in which the perturbation is not completely absorbed by the system) dramatically decreases when the RBNs begin to interact and, then, it diminishes monotonously as the interaction strength raises. The trend is analogous for all the three classes of MRBNs, even if the magnitudes are substantially different and coherent with the specific degree of order.

Apparently, the system would tend to more ordered states in correspondence of higher values of the interaction strength, for all the three classes of behaviour.

[6] Higher values of the interaction strength would entail a too large ratio of receptors over the number of internal nodes.

[7] Note that this classification of the MRBNs is possible because their characterizing RBNs are exactly those previously analyzed and grouped in three classes.

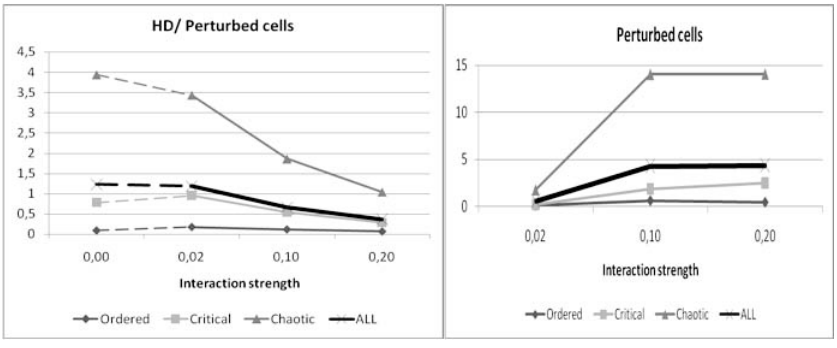



**Fig. 3.** (Left) Variation of the average ratio [Hamming distance/number of perturbed cells] after 100 time steps for the three different classes and for the whole set of 100 MRBNs (on 150 runs) for different values of $f$. The values in corrispondence of $f = 0$ refer to the average Hamming distance of the isolated RBNs. (right) Variation of the average number of perturbed cells after 100 time steps for the three different classes and for the whole set of MRBNs (150 runs) for different values of the interaction strength.

Nevertheless, the dynamics is far more complex. Figure 3 shows that if we consider an isolated RBN and we perturb it, the average Hamming distance reaches a value slightly higher than the critical value 1, while when the cells begin to interact within a MRBN the average value of the Hamming distance over the number of perturbed cells decrease and it continues to decrease as the interaction strength raises. On the other hand, in correspondence of higher values of the interaction strength the average number of cells involved in the perturbation plainly increases. Note that, even though the typical trends in the observed variables are analogous for the three classes, also in this case the differences in the magnitude are remarkable. Hence, the distinction in classes turns out to be robust also when RBNs interact and three distinct dynamical behaviours can indeed be identified.

From these results we can notice how a higher interaction strength may either contribute to brake the perturbation, or allow the diffusion of perturbations that hit a larger number of cells, although involving fewer nodes within each of them (on the average). We may ascribe this peculiar behaviour to the features of the specific interaction mechanism we adopted and, in particular, to the role of receptors in the overall dynamics. Nevertheless, further analyses on the model are needed to reach a definitive conclusion.

## 6 Conclusions

A first remark is about the intrinsic complexity of the model. The dynamical behaviour that shows up is far from trivial and so is its interpretation. Nevertheless, the analyses made on the model provided some interesting cues.

The approach that involves perturbations to examine the response of the system has proven to be effective for different reasons. First of all, it allows to

clearly discriminate three classes of RBNs on the basis of their actual dynamical regime, i.e. ordered, critical or chaotic. Once again, it is possible to demonstrate how networks with identical critical structural parameters can indeed show substantially different dynamic behaviours when analysed singularly. Besides, the dynamic behaviour of isolated RBNs is actually confirmed and enhanced when RBNs are coupled, even though we observe a general shift toward the ordered regime region in correspondence of a stronger interaction.

Yet, as we have shown, interaction leads to dynamic behaviours indeed hard to interpret. Therefore, further development of the model are needed, primarily aimed to grasp the relation among the features of the interaction mechanism and the overall dynamics: for instance, the study of the effect of different interaction rules, or the analysis of MRBNs whose RBNs are characterized by a constant number of internal nodes for different values of the interaction strength.

**Acknowledgments.** This work has been partially supported by the Italian MIUR-FISR project nr. 2982/Ric (Mitica).

## References

1. Kauffman, S.A.: Gene regulation networks: A theory of their global structure and behaviour. Top. Dev. Biol. 6, 145–182 (1971)
2. Kauffman, S.A.: The origins of order. Oxford University Press, Oxford (1993)
3. Kauffman, S.A.: At home in the universe. Oxford University Press, Oxford (1995)
4. Serra, R., Villani, M., Semeria, A.: Genetic network models and statistical properties of gene expression data in knock-out experiments. J. Theor. Biol. 227, 149–157 (2004)
5. Serra, R., Villani, M., Graudenzi, A., Kauffman, S.A.: Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. J. Theor. Biol. 246, 449–460 (2007)
6. Ramo, P., Kesseli, J., Yli-Harja, O.: Perturbation avalanches and criticality in gene regulatory networks. J. Theor. Biol. 242, 164–170 (2006)
7. Kauffman, S.A.: Investigations. Oxford University Press, Oxford (2000)
8. Harvey, I., Bossomaier, T.: Time out of joint: Attractors in asynchronous random Boolean networks. In: Husbands, P., Harvey, I. (eds.) Proceedings of the Fourth European Conference on Artificial Life (ECAL 1997), pp. 67–75. MIT Press, Cambridge (1997)
9. Aldana, M., Coppersmith, S., Kadanoff, L.: Boolean dynamics with random couplings. In: Kaplan, E., Marsden, J.E., Sreenivasan, K.R. (eds.) Perspectives and Problems in Nonlinear Science. Springer Applied Mathematical Sciences Series (2003)
10. Villani, M., Serra, R., Ingrami, P., Kauffman, S.A.: Coupled random Boolean network forming an artificial tissue. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 548–556. Springer, Heidelberg (2006)
11. Bastolla, U., Parisi, G.: The modular structure of Kauffman networks. Physica D 185, 45–66 (2003)
12. Yacoubi, S.E., Chopard, S., Bandini, S.: ACRI 2006. LNCS, vol. 4173. Springer, Heidelberg (2006)

# Research into the Generation of Sound Effects Using a Cellular Automaton

Seita Toguchi, Yuhei Akamine, and Satoshi Endo

Department of Information Engineering, Univesity of the Ryukyus
1 Senbaru, Nishihara-cho, Okinawa, 903-0213 Japan
seita@eva.ie.u-ryukyu.ac.jp,
{yuhei,endo}@ie.u-ryukyu.ac.jp

**Abstract.** The purpose of this research is to generate impact sounds often used as sound effects through physical simulation using cellular automaton (CA). The authors designed a model based on the wave propagation CA model. In the model, desired sound is generated by setting several parameters. This paper shows the effects of each parameter on the sound, and shows that various impact sound can be generated by the model. Production cost of sound effects can be reduced by using this model.

**Keywords:** sound generation, physical simulation, cellular automaton.

## 1 Introduction

Sound effects are used to emphasize artistic and other content in movies, video games, music, or other media. The most commonly-used production methods for sound effects are: (1) to record real sound, edit and then use it, (2) to generate sound effects using a synthesizer, (3) to process recorded sound similar to the desired sound, or (4) to use prerecorded sample collections for sound effects.

However, there are a number of problems. In the first part, recording real sounds can be expensive since a sound source must be prepared. Second, synthesizers are unsuitable for generating realistic sounds. Finally, prerecorded sample collections do not contain enough sound varieties. With these considerations in mind, if real world sounds are reproducible on a computer it should lead to cost reductions.

The purpose of this research is development of a tool generating sound effects through physical simulation. There are a number of different types of sound effect. For example, impact sound, aerodynamic sound, explosion sound, and friction sound. In this research, the authors have modeled impact sounds often used as sound effects as the first step. An object is vibrated when a shock is applied to the object. This is recognized as an impact sound by the propagation of it via the air. Therefore, impact sounds are generable by modeling the vibration of an object.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 323–328, 2008.

## 2 Related Study

Digital waveguide is known as a method of generating sound through physical simulation [2]. This is a efficient method for modeling one-dimensional waveguide such as string or air column. A lossless digital waveguide is realized as discrete form of d'Alembert's solution of the one dimensional wave equation, and the vibration is obtained by superposition of a right-going wave and left going wave. A major advantage is high calculation speed because this method consist of multiple delay lines. A delay line can be operated with just two fetch, store and arithmetic operations. This method is adopted by most modern physical modeling synthesizer. However, nonlinear vibration can not be dealt directly because this method is premised on commutativity of linear time-invariant system. In addition, modeling with this method is not simple because it has complicated parameters.

In this research, the authors designed a physical simulation model based on wave propagation CA model [1] to model vibration of objects. This model can express complex shaped object and can introduce nonlinearity with a simple parameter. This model can be set parameters and extended easily because of simple structure of the model.

## 3 Wave Propagation CA Model

The wave propagation CA model proposed by Morisita and others is a CA application for solving wave propagation problems [1]. In this model, the space is divided with a square lattice. Each divided space has two state values. The state values are defined as sound pressure $p$ and its rate of time change $G$ by a real value. In a one-dimensional case, as shown in Fig. 1, two directions are considered as the neighborhood. The rule for wave propagation is defined as follows.

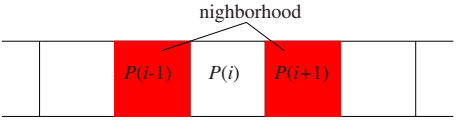



**Fig. 1.** The neighborhood of the One-dimensional wave propagation CA model

$$\begin{aligned} & p(i+1,t) - 2p(i,t) + p(i-1,t) \\ &= G(i,t+1) - G(i,t)\ , \end{aligned} \tag{1}$$

$$p(i,t+1) = p(i,t) + G(i,t+1)\ . \tag{2}$$

This model is mathematically equivalent of mass spring model whose mass points can move only vertically (Fig. 2) This interpretation allows definitions of fixed ends, free ends, dampers and others. The property of the vibrated object

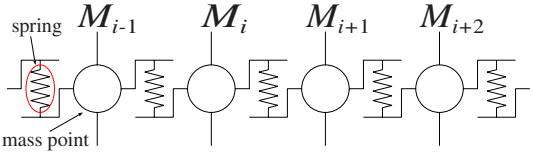



**Fig. 2.** One-dimensional mass spring model. Circles represent mass points, and polygonal lines represent springs. In this model, the mass points vibrate vertically, not horizontally.

can be set more in detail. In addition, it has also a big advantage that nonlinearity can be introduced.

In a general physical simulation of sound generation, the sound of plucked string is generated from an initial waveform as in the left figure of Fig. 3 [2]. Similarly, the sound of a plucked string is generable by giving a form as an initial state of a one-dimensional wave propagation CA model as shown in the left figure of Fig. 3. The sound of a striking string is generated by giving it an initial state like the thick solid line of the right figure of Fig. 3. The Impact sound can be generate by the two-dimensional model. The sound is generated by sampling vibration of a certain mass point at each time step.

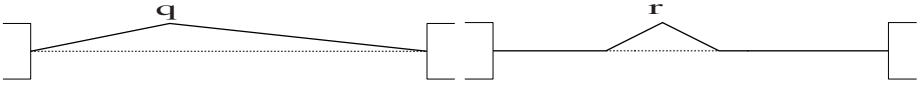



**Fig. 3.** Initial states of plucked string and struck string. Dashed lines represent state of a string in equilibrium, and thick solid lines represent initial state of plucked string or struck string. $q$ is the plucking position and $r$ is striking position.

## 4 Proposed Method

A two-dimensional object can be expressed by the mass spring model as shown in the left figure of Fig. 4. However, the computational effort is very large because there are a lot of mass points and springs. In three dimensional-model, computational effort is huge. Therefore, in this research, the authors designed a model based on wave propagation CA model (the right figure of Fig. 4.) In this model, computational effort can be greatly reduced because the object is expressed by a wire frame composed of mass points and springs.

However, the sound quality depends on the number of mass points and springs that express the object. Higher quality sound can be generated by more mass points and springs. Fig. 5 shows it. However, computational effort increase with number of points. There are trade-offs between quality and computational effort. When there are enough of mass points and springs, even if those are increased, sound quality is hardly changed. Therefore, it is important to find the adequate number of mass points and springs.

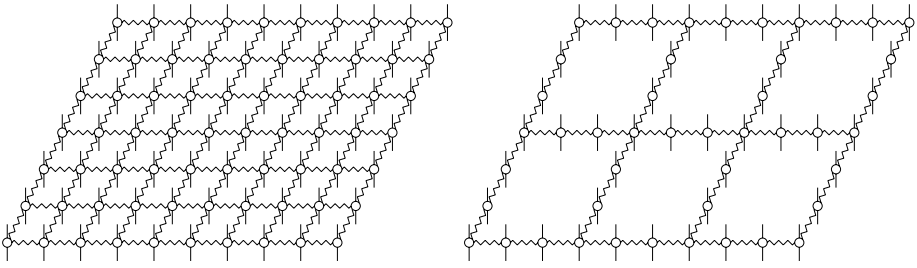


**Fig. 4.** Two-dimensional object. The left figure shows the object expressed by mass spring model, and the right figure shows the expressed by the model designed by the authors.

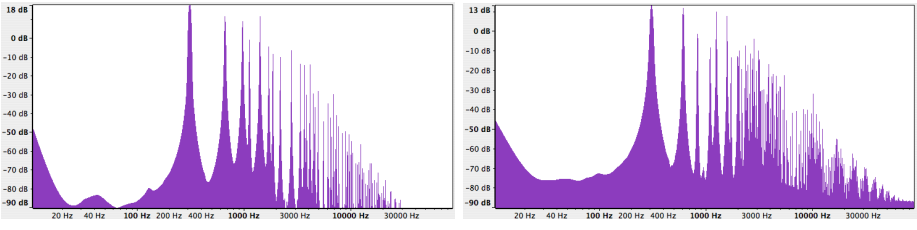



**Fig. 5.** The results of spectrum analysis of the generated sounds. A horizontal axis represents frequency and a vertical axis represents energy. The right figure shows the result of the sound generated by more mass points and springs than the left one.

## 5 Parameters

Many parameters are needed for generating an impact sound using the model. In order to generate a desired sound, it is important to understand the effect of each parameter on the sound. The authors defined the parameters which do not affect tone but affects the pitch and length of sound as the basic parameter, and the parameters which affect tone as the tonal parameter. The effect of each parameter on the sound are described below.

### 5.1 Basic Parameters

An explanation of basic parameters is described below. Each mass points has the basic parameters.

**Coefficient of damping**
Coefficient of damping affect decay of sound. Higher coefficient of damping makes shorter sound. Lower value makes long sound. Zero value makes a sound without decay. For example, resonant object is expressed by setting low value.

**Spring modulus and mass**
Spring modulus and mass affect the pitch of the sound. The pitch of the generated sound is control by setting of sprimg modulus and mass.

### 5.2 Tonal Parameters

An explanation of tonal parameters is described below.

**Shape of the object**
The sound of plucked string is generated using a one-dimensional model. The sound of striking a plate is generated using a two-dimensional model. Bigger object makes lower pitch sound, and smaller object makes higher pitch sound. If the object size is very small and the spring modulus is very low, the sound quality would be low because high frequency tone is lost. However, long time is required for generating a sound from large size object. The adequate size object should be used with consideration for balance of quality and calculation time.

**Nonlinearity**
The proposed model uses Hooke's law to calculate the wave propagation. In Hooke's law, force $F$ is calculated by the product of displacement $x$ and the spring modulus $k$. Nonlinearity can be introduced by using the cube of displacement $x$ to calculate force $F$ like the following expression.

$$F = -kx(1 + ax^2) \ . \tag{3}$$

Where, $a$ is a value near zero. The nonlinear model can generate a sound ungenerable in a linear model. Although the vibration of a object is nonlinear, most vibrations can be approximated linearly. However, some objects can not be expressed without nonlinearity.

**High cut rate**
In this model, calculation results is unstable when nonlinearity is introduced. Therefore, Runge-Kutta method is used for stabilizing the calculation. However, this method reduces high frequency waves as a result of calculation error. Proposed model uses this reduction for controlling attenuation of high frequency waves. In this paper, this control parameter is called "high cut rate." The high frequency wave is reduced when high cut rate is set high. A soft sound is generated as a result of cutting high frequency wave. Fig. 6 shows the temporal variation of the spectrum of generated sound made with different values.

## 6 Generated Sound Examples

Various impact sounds can be generated by setting each parameter in the model. Table 1 show generated sound example and setting of each parameter. The blank means setting any value. These sound are put on the following web page.
http://www.ie.u-ryukyu.ac.jp/~e045737/oto.html

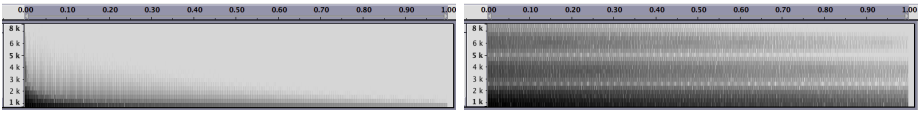


**Fig. 6.** The temporal variation of the spectrum of generated sound made with different values. A horizontal axis represents time and a vertical axis represents frequency. The left figure shows the sound made with high value of high cut rate, and the right figure shows the sound made with the low value.

**Table 1.** Generated sound examples and setting of each parameter

| | pitch | damping | shape | nonlinearity | high cut rate |
|---|---|---|---|---|---|
| guitar | | | one-dimention | linear | |
| sanshin[1] | | | one-dimention | nonlinear | |
| metal plate | | | two-dimention | linear | low |
| wooden board | | high | two-dimention | linear | high |
| glass | high | | two-dimention | linear | high |
| cymbal | | | two-dimention | nonlinear | low |

## 7 Conclusion

In this paper, the authors discussed the method generating impact sound based on wave propagation CA model for sound effect used by various contents. By the result of examination for effect of the parameter of the proposed method on the generated sound, the possibility to generate various impact sound has been shown.

The proposed method has many parameters. It is difficult to find adequate parameters for desired sound. Therefore, the authors consider to develop a tool generating user's desired sound from simple parameters.

## References

1. Morishita, S.: Cellular Automata (in Japanease). Yokendo Ltd., Tokyo (2003)
2. Smith III, J.O.: Physical Modeling using Digital Waveguides. Computer Music Journal 16(4), 74–91 (1992)
3. Kato, Y., Mitsunari, T., Tsukiyama, H.: Cellular Automaton (in Japanease). Morikita Publishing Co., Ltd., Tokyo (1998)

[1] Japanese traditional instrument.

# Modelling Combined Subaerial-Subaqueous Flow-Like Landslides by Cellular Automata

Maria Vittoria Avolio[1], Valeria Lupiano[1,2], Paolo Mazzanti[3], and Salvatore Di Gregorio[1]

[1] University of Calabria, Department of Mathematics, 87036 Rende (CS), Italy
{avoliomv,dig}@unical.it
[2] University of Calabria, Department of Earth Sciences, 87036 Rende (CS), Italy
lupianov@unical.it
[3] University of Rome "La Sapienza", Department of Earth Sciences, P.le Aldo Moro, 00185, Roma, Italy
paolo.mazzanti@uniroma1.it

**Abstract.** Macroscopic Cellular Automata characterize a methodological approach for modelling large scale (extended for kilometres) complex acentric phenomena, e.g. surface flows as lava flows, debris flows etc.. This paper concerns the extension of such a method in order to simulate combined subaerial-subaqueous flow-like landslides. The occurrence of heterogeneous interacting processes requires a more physical description of the energy balance and an explicit velocity management. The model SCIDDICA-SS2 proposes some empirical solutions, such as the computation at each step and inside each cell, of departing flows which are characterized by their mass centre position and velocity. An application to combined subaerial-subaqueous landslide is exhibited together with simulation results of the 1997 Albano lake (Rome, Italy) debris flow.

## 1 Introduction

An extension of classical Cellular Automata (*CA*), the Macroscopic *CA* [1], were developed in order to model many complex macroscopic fluid-dynamical phenomena, that seem difficult to be modelled in other *CA* frames (e.g. the lattice Boltzmann method), because they take place on a large space scale.

Macroscopic *CA* can need a large amount of states, that describe properties of the cells (e.g. temperature); such states may be formally represented by means of substates, that specify the characteristics to be attributed to the state of the cell and determining the *CA* evolution. The Cartesian product of the sets of all substates constitutes the set of states. It involves a large amount of states more a complicated transition function, not reducible to a lookup table.

In the case of surface flows, quantities concerning the third dimension, i.e. the height, may be easily included among the *CA* substates (e.g. the altitude), permitting models in two dimensions, working effectively in three dimensions. Furthermore, an algorithm for the minimisation of the differences (in height) [1] was found in this context in order to determine the outflows from a cell

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 329–336, 2008.

toward the remaining cells of its neighbourhood, giving rise to several models for different macroscopic phenomena: lava flows [1], debris/mud flows [1] and rain soil erosion [2].

This empirical method has a significant restriction, because it doesn't permit to make velocity explicit with a radius 1 neighbourhood: a fluid amount moves from a cell to another one in a *CA* step, which corresponds usually to a constant time. This implies a constant "velocity" in the *CA* context of discrete space/time. Nevertheless, velocities can be deduced by analysing the global behaviour of the system in time and space. In such models, the flow velocity emerges by averaging on the space (i.e. considering clusters of cells) or by averaging on the time (e.g. considering the advancing flow front average velocity in a sequence of *CA* steps).

Constant velocity could be a limit for modelling finely macroscopic phenomena, because it is difficult sometime to introduce physical considerations in the modelling at a local level. Furthermore, the time corresponding to a step of the *CA* is often deduced "a posteriori" by the simulation results and parameters of the transition function must be again verified when the size of the cell is changed.

A solution was proposed: moving flows toward the neighbouring cells are individuated by the substates mass, velocity and barycentre co-ordinates. The resulting new mass, barycentre and velocity are computed by composition of all the inflows from the neighbours and the residual quantities inside the cell. This overall method was applied first to lava flows [3]; the extension to debris flows involved more complex criteria to be considered.

The next section defines the criteria, adopted in modelling combined subaerial-subaqueous debris flows i.e. SCIDDICA-SS2, the simulation results of the 1997 Albano lake (Rome, Italy) debris flows [4] are shown in the third section, conclusions are reported at the end.

## 2 A *CA* Model for Combined Subaerial-Subaqueous Flow-Like Landslides

*CA* represent an alternative to differential equations for complex systems evolving by local interactions. Some researchers proposed *CA* models for flow type landslides.

Di Gregorio et al. [5] developed a simple two-dimensional *CA* model (first release of SCIDDICA) and validated it by simulating the Mt Ontake landslide. Many extensions of SCIDDICA were afterwards developed in order to improve the model and/or capture the characteristics of different or more complex landslides, performing also susceptibility analysis [6,7,8,9].

Segre and Deangeli [10] presented a three-dimensional numeric model, based on *CA* , for debris flows, using difference equations. The model was validated on the M. XiKou landslide, capturing its main characteristics.

Clerici and Perego [11] simulated the Corniglio landslide using a simple *CA* model in order to capture the blockage mechanisms for that type of landslide.

Salles et al. [12] developed recently a first interesting *CA* model for subaqueous flows, in order to simulate density currents.

### 2.1 The Model SCIDDICA-SS2

This version of SCIDDICA is an extension of the model applied to the landslides of Sarno [8] improved by the method of explicit velocities of the SCIARA model [3]. Such an extension involves more substates, processes and parameters because the phenomenon is more complex: in fact, subaqueous part of Albano landslide needs to be first modelled, especially the air-water transition.

The hexagonal $CA$ model SCIDDICA-SS2 is the quintuple $< R, X, S, P, \tau >$:

- $R$ is the set of regular hexagons covering the region, where the phenomenon evolves.
- $X$ identifies the geometrical pattern of cells, which influence any state change of the central cell: the central cell (index 0) itself and the six adjacent cells (indexes 1,...,6).
- $S$ is the finite set of states of the finite automaton, embedded in the cell; it is equal to the Cartesian product of the sets of the considered substates:

$$S = S_A \times S_D \times S_{TH} \times S_X \times S_Y \times S_{KH} \times S_E^6 \times \\ \times S_{XE}^6 \times S_{YE}^6 \times S_{KHE}^6 \times S_I^6 \times S_{XI}^6 \times S_{YI}^6 \times S_{KHI}^6$$

  - $S_A$ is the cell altitude, $S_D$ is the maximum depth of detrital cover, that could be transformed by erosion in landslide debris;
  - $S_{TH}$ is the average thickness of landslide debris inside the cell, $S_X$ and $S_Y$ are the co-ordinates of the debris barycentre with reference to the cell centre, $S_{KH}$ is the debris kinetic head;
  - $S_E$ is the part of debris flow, the so called "external flow", (normalised to a thickness) that penetrates the adjacent cell from central cell, $S_{XE}$ and $S_{YE}$ are the co-ordinates of the external flow barycentre with reference to the adjacent cell centre, $S_{KHE}$ is the debris kinetic head (six components for all the substates);
  - $S_I$ is the part of debris flow toward the adjacent cell, the so called "internal flow", (normalised to a thickness) that remains inside the central cell, $S_{XI}$ and $S_{YI}$ are the co-ordinates of the internal flow barycentre with reference to the central cell centre, $S_{KHI}$ is the debris kinetic head (six components for all the substates).
- $P$ is the set of the global physical and empirical parameters, which account for the general frame of the model and the physical characteristics of the phenomenon; the next section provides a better explication of the elements in the following list:

$$P = \{p_a, p_t, p_{adhw}, p_{adha}, p_{fcw}, p_{fca}, p_{tdw}, p_{tda}, \\ p_{edw}, p_{eda}, p_{ml}, p_{mtw}, p_{mta}, p_{pew}, p_{pea}, p_{wr}\}$$

  - $p_a$ is the cell apothem; $p_t$ is the temporal correspondence of a $CA$ step;
  - $p_{adhw}$, $p_{adha}$ are the water/air adhesion values, i.e. the debris thickness, that may not removed;
  - $p_{fcw}$, $p_{fca}$ are the water/air friction coefficient for debris outflows;

  - $p_{tdw}$, $p_{tda}$, $p_{edw}$, $p_{eda}$ are water/air parameters for energy dissipation by turbulence, water/air parameters for energy dissipation by erosion;
  - $p_{ml}$ is the matter loss in percent when the debris enters into water;
  - $p_{mtw}$, $p_{mta}$ are the water/air activation thresholds of the mobilisation;
  - $p_{tmt}$, is the activation threshold of the mobilisation for the transept;
  - $p_{pew}$, $p_{pea}$ are the water/air progressive erosion parameters;
  - $p_{wr}$ is the water resistance parameter.
- $\tau : S^7 \rightarrow S$ is the deterministic state transition for the cells in $R$. The basic elements of the transition function will be sketched in the next section.

At the beginning of the simulation, we specify the states of the cells in $R$, defining the initial $CA$ configuration. The initial values of the substates are accordingly initialised. In particular, $S_A$ assumes the morphology values except for the detachment area, where the thickness of the landslide mass is subtracted from the morphology value; $S_{TH}$ is zero everywhere except for the detachment area, where the thickness of landslide mass is specified; $S_D$ assumes initial values corresponding to the maximum depth of the mantle of detrital cover, which can be eroded; all the values related to the remaining substates are zero everywhere.

At each next step, the function $\tau$ is applied to all the cells in $R$, so that the configuration changes in time and the evolution of the $CA$ is obtained.

### 2.2 The SCIDDICA-SS2 Transition Function

Four local processes may be considered for the release SS2 of SCIDDICA:

- altitude, kinetic head, debris thickness variation by detrital cover erosion;
- kinetic head variation by turbulence dissipation;
- debris outflows (thickness, barycentre co-ordinates, kinetic head) determination and their shift deduced by the motion equations;
- composition of debris inside the cell (remaining debris more inflows) and determination of new thickness, barycentre co-ordinates, kinetic head.

In the following, a sketch of the local elementary processes will be given, which is sufficient to capture the mechanisms of the transition function; the execution of an elementary process updates the substates. Variables concerning substates and parameters are indicated by their abbreviation. When substates need the specification of the neighbourhood cell, its index is indicated between square brackets. $\Delta Q$ means variation of the substate $S_Q$. Parameter final letter $(w, a)$ is omitted when the formula is valid both in water and air.

**Mobilisation Effects.** When the kinetic head value overcomes an opportune threshold ($KH > mt$) depending on the soil features and its saturation state then a mobilisation of the detrital cover occurs proportionally to the quantity overcoming the threshold: $pe \cdot (KH - mt) = \Delta TH = -\Delta D$ (the detrital cover depth diminishes as the debris thickness increases), the kinetic head loss is: $-\Delta KH = ed \cdot (KH - mt)$. The mixing of the eroded detrital cover with the earlier debris involves that the earlier debris kinetic energy becomes the kinetic energy of all the mass, it implicates trivially a further kinetic head reduction.

The activation of secondary sources is managed by a "transept" for each secondary source. A threshold $tmt$ for the thickness of the debris flow crossing the transept is specified and secondary soil slips are actived for $TH > tmt$.

**Turbulence Effect.** The effect of the turbulence is modelled by a proportional kinetic head loss at each SCIDDICA step: $-\Delta KH = td \cdot KH$. This formula involves that a velocity limit is imposed de facto. A generic case with a maximum value of slope may be always transformed in the worst case of an endless channel with constant maximum value slope. In this case an asymptotic value of kinetic head is implied by infinite formula applications and, therefore, a velocity limit is deduced.

**Debris Outflows.** Outflows computation is performed in two steps: determination of the outflows minimising the "height" differences in the neighbourhood [1] and determination of the shift of the outflows.

Rapid debris flows imply often a run up effect, depending on the kinetic head associated to debris flow. As a consequence, the height minimisation algorithm [1] is applied, considering for the central cell the fixed part $b[0] = A[0] + KH[0] + adh$ and the movable (distributable) part $m[0] = TH[0] - adh$. The fixed part for the adjacent cells is: $b[i] = A[i] + TH[i]$, $1 \leq i \leq 6$; note that $KH[0]$ accounts for the ability of the flowing debris of climbing a slope. The minimisation algorithm determines the flows $f[i]$, $0 \leq i \leq 6$ toward the neighbouring cells ($f[0]$ is the part of $m[0]$ which is not distributed); such flows minimise the expression: $\sum \mid (b[i] + f[i]) - (b[j] + f[i]) \mid$ for $\{(i, j) \mid 0 \leq i < j \leq 6\}$.

The barycentre co-ordinates $x$ and $y$ of moving quantities are the same of all the debris inside the cell and the form is ideally a "cylinder" tangent the next edge of the hexagonal cell. A preliminary test is executed in order to account the friction effects, that prevent debris outflows, when the difference in height, $b[0] + m[0] - b[i]$, that determines an ideal slope "$\theta[i]$" between the two cells 0 and i, is insufficient: the condition is expressed by the formula $\tan\theta[i] < fc$. A ideal distance "$d$" is considered between the central cell debris barycentre and the centre of the adjacent cell $i$ including the slope $\theta[i]$.

The $f[i]$ shift "$sh$" is computed for subaerial debris flow according to the following simple formula, that averages the movement of all the mass as the barycentre movement of a body on a constant slope with a constant friction coefficient: $sh = v \cdot t + g(\sin\theta - fca \cdot \cos\theta)t^2/2$, with "$g$" the gravity acceleration, the initial velocity $v = \sqrt{(2g \cdot HK)}$. The shift formula for subaqueous debris considers also the water resistance, using modified Stokes equations with a form factor proportional to mass: $sh = = (1 - exp(-wr \cdot t))(v - (g(\sin\theta - fcw \cdot \cos\theta)/wr)) + g(\sin\theta - fcw \cdot \cos\theta) \cdot t/wr$.

The motion involves three possibilities: (1) only internal flow, the shifted cylinder is completely internal to the central cell; (2) only external flow, all the shifted cylinder is external to the central cell inside the adjacent cell; (3) the shifted cylinder is partially internal to the central cell, partially external to the central cell, the flow is divided between the central and the adjacent cell, forming two cylinders with barycentres corresponding to the barycentres of the internal debris flow and the external debris flow. Kinetic head variation is computed

according to the new position of internal and external flows, while energy dissipation was considered as a turbulence effect in the previous elementary process.

**Flows Composition.** When debris outflows are computed, the new situation involves that external flows left the cell, internal flows remain in the cell with different co-ordinates and inflows (trivially derived by the values of external flows of neighbour cells) could exist. The new value of $TH$ is given, considering the balance of inflows and outflows with the remaining debris in the cell. A kinetic energy reduction is considered by loss of flows, while an increase is given by inflows: the new value of the kinetic head is deduced from the computed kinetic energy. The co-ordinates determination is calculated as the average weight of $X$ and $Y$ considering remaining debris in the central cell, internal flows and inflows.

**Air-Water Interface.** Air-water interface is managed only for external flows from air to water and not vice versa. An external flow from an air cell (altitude higher than water level) to water cell (altitude lower than water level) implies always a loss of matter (water inside debris and components lighter than water) proportional to debris mass, specified by $ml$, it implies a correspondent loss of kinetic energy, determined by kinetic head decrease.

## 3 Simulations of the 1997 Albano Lake Debris Flow

SCIDDICA-SS2 was validated against the 1997 Albano lake (Rome, Italy) event (fig. 1) which is a rare case of combined subaerial-subaqueous debris-flow [4]. This landslide occurred in the eastern slope of the Albano lake on the 7th of November 1997 after an intense rainfall event (128 mm in 24 hours), and it began as a soil slide, mobilizing about 300 $m^3$ of eluvial material. The mobilized mass was channelled within a steeply dipping impluvium (about 40°) and thus evolved as a debris flow which entrained a large amount of debris material along the bottom of the channel and reached an estimated volume of some thousands of $m^3$ at the coastline. A few amount of material was deposited at the coastline while a greater quantity entered in water generating a little tsunami wave.

Detailed subaerial and submerged topographic data was acquired in 2005 and 2006 through aerial LiDAR and sonar multibeam swath bathymetric surveys, performed by the INGV (Italian National Institute of Geophysics and Volcanology) in the frame of a project sponsored by the Italian National Civil Protection Department. Consequently, a 1m cell size DTLM (Digital Terrain and Lacustrine Model) was produced and allowed to map in detail both the subaerial and the submerged detachment and deposition areas and to estimate their volumes [4].

Simulations permit to validate the general model and to calibrate adequately its parameters; results show a good agreement concerning erosion and deposits on both subaerial and subaqueous part (fig. 2). A partial, but very significant (quantitative) comparison between the real event and the simulated one for both the environments may be performed by a fitness function $\boldsymbol{f}$, that considers a normalised value between 0 (complete failure) and 1 (perfect correspondence), computed by the following formula $\sqrt{((\boldsymbol{R} \cap \boldsymbol{S})/(\boldsymbol{R} \cup \boldsymbol{S}))}$, where $\boldsymbol{R}$ is the set

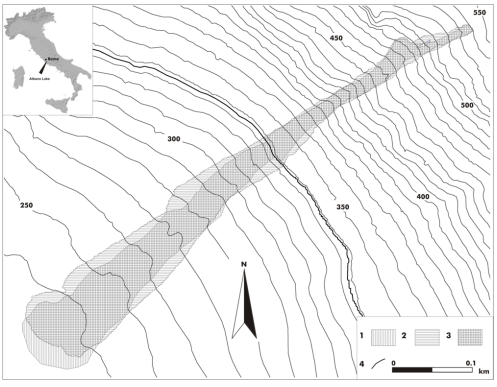



**Fig. 1.** The 1997 Albano lake subaerial-subaqueous debris flow. Key: (1) real event, (2) simulated event, (3) intersection between real and simulated event, (4) water level.

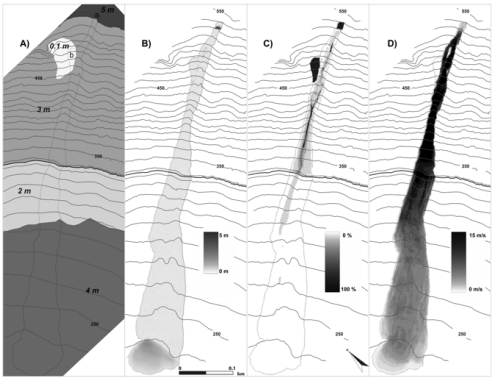



**Fig. 2.** A) regolith thickness (value is in italic) with pre-event reconstructed contour lines; "a" and "b" respectively principal and secondary source. The results regarding the best simulation are shown: B) deposit thickness; C) erosion depth; D) max occurred velocity. The thick contour line represents the water level.

of cells affected by the landslides in the real event and $\boldsymbol{S}$ the set of cells affected by the landslides in the simulation.

We obtain an encouraging $\boldsymbol{f}$ value 0.85 (fig. 1), that confirms a satisfying reliability in reproducing the phenomenon. The performance is positive compared to results of other models, that use almost more qualitative than quantitative evaluation tools.

## 4 Conclusions

Simulations of the 1997 Albano lake debris flow proved to be consistent with the observed path, deposit and erosion of the actual landslide suggesting that

SCIDDICA-SS2 could be usefully used in hazard analyses for subaerial, subaqueous and combined flow-like landslides; applications to other cases are planned.

SCIDDICA-SS2 is the first model for subaerial-subaqueous flow-like landslides (there are many only-subaerial models and few only-subaqueous models) to our knowledge. The problematical air-water transition gave rise to significant improvements in comparison with other SCIDDICA versions, i.e. the introduction of the explicit velocity with a more precise computation of shift for internal and external flows, while a limit is a possible underestimation of inertial effects.

**Acknowledgments.** This research was funded by the Italian Government, MIUR, PRIN-06, 2006042319, Project "Integration of inshore and offshore geological and geophysical innovative techniques for coastal landslides studies".

## References

1. Di Gregorio, S., Serra, R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. Future Generation Computer Systems 16, 259–271 (1999)
2. D'Ambrosio, D., Di Gregorio, S., Gabriele, S., Gaudio, R.: A cellular automata model for soil erosion by water. Physics and Chemistry of the Earth - Part B 26, 33–39 (2001)
3. Avolio, M.V., Crisci, G.M., Di Gregorio, S., Rongo, R., Spataro, W., Trunfio, G.A.: SCIARA $\boldsymbol{\gamma 2}$: an improved cellular automata model for lava flows and applications to the 2002 Etnean crisis. Computers & Geosciences 32, 897–911 (2006)
4. Mazzanti, P., Bozzano, F., Esposito, C.: Submerged landslides morphologies in the Albano lake (Rome, Italy). In: Lykousis, V., Sakellariou, J.L.D. (eds.) Proc. of 3rd Intern. Symp. Submarine Mass Movements and Their Consequences, Lugano, Switzerland, 2007. Advances in Natural and Technological Hazards Research, vol. 27, pp. 243–250. Springer, Heidelberg (2007)
5. Di Gregorio, S., Nicoletta, F., Rongo, R., Sorriso-Valvo, M., Spezzano, G., Talia, D.: Landslide simulation by cellular automata in a parallel environment. In: Furnari, M.M. (ed.) Proceedings of 2nd International Workshop Massive Parallelism: Hardware, Software and Applications, pp. 392–407. World Scientific, Singapore (1995)
6. Di Gregorio, S., Rongo, R., Siciliano, C., Sorriso-Valvo, M., Spataro, W.: Mt. Ontake landslide simulation by the cellular automata model SCIDDICA-3. Physics and Chemistry of the Earth 24, 97–100 (1999)
7. Avolio, M.V., Di Gregorio, S., Mantovani, F., Pasuto, A., Rongo, R., Silvano, S., Spataro, W.: Simulation of 1992 Tessina landslide by a cellular automata model and future hazard scenarios. JAG 2, 41–50 (2000)
8. D'Ambrosio, D., Di Gregorio, S., Iovine, G.: Simulating debris flows through a hexagonal cellular automata model: SCIDDICA S3-hex. NHESS 3, 545–559 (2003)
9. Iovine, G., D'Ambrosio, D., Di Gregorio, S.: Applying genetic algorithms for calibrating a hexagonal cellular automata model for the simulation of debris flows characterised by strong inertial effects. Geomorphology 66, 287–303 (2005)
10. Segre, E., Deangeli, C.: Cellular automaton for realistic modelling of landslides. Nonlinear Processes in Geophysics 2, 1–15 (1995)
11. Clerici, A., Perego, S.: Simulation of the Parma river blockage by the Corniglio landslide (Northern Italy). Geomorphology 33, 1–23 (2000)
12. Salles, T., Lopez, S., Cacas, M.C., Mulder, T.: Cellular automata model of density currents. Geomorphology 88, 1–20 (2007)

# Unstructured Cellular Automata and the Application to Model River Riparian Vegetation Dynamics

Qiuwen Chen and Fei Ye

State Key Lab for System Ecology, Research Centre for Eco-Environmental Sciences, Chinese Academy of Sciences, China
{qchen,fye}@rcees.ac.cn

**Abstract.** Cellular automata (CA) have proved to be a robust approach to spatially-explicit modeling of ecosystems. Conventionally the CA models applied a structured square grid. However, due to the anisotropic properties of environmental conditions, the capability of CA method was not fully explored when using the regular squared lattice. This research investigated the unstructured cellular automata (UCA) by implementing a irregular triangular grid and used it to develop a vegetation dynamics model. The model was then coupled with a two-dimensional hydrodynamic model to simulate the riparian vegetation dynamics due to flow modifications by the reservoirs operations. The integrated model was applied to a compound channel of the Lijiang River in the Southwest China, which has been affected by the flow regulations for navigation purpose. Through the simulations, the previous evolutions of the riparian vegetations were recaptured and their future developments under the new flow regulation scheme were predicted. In particular, the potentials of UCA in ecosystem modeling were illustrated in the research.

## 1 Introduction

Cellular automata constitute a mathematical system, in which the simple local components interact together to produce global complicated dynamics. Cellular automata have become a viable alternative approach to ecological modelling [1, 2], in particular after the emergence and application of object-oriented programming language.

Conventionally cellular automata models apply a structured grid, for example the regular squared lattice. However, ecosystems are mostly characterised by the spatial anisotropy [3, 4], which indicates that some important local variations might be lost in these models. Although, the hexagon grid had been investigated [5, 6], they are still structured. Besides, they have not been widely used in the ecological models.

The riparian zones are highly dynamic systems governed by interrelating physical and biological processes. The physical template of riparian zone is characterized by several typical geomorphic features, mainly including channel bed, channel shelf, floodplain and terrace [7]. Plant communities of different characteristics are distributed along the river side according to the ecological gradients. However, it is challenging to model the riparian successions, especially when flow patterns are largely modified by river regulations, for instance, the reservoir operations.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 337–344, 2008.

Some previous attempts to this subject include the use of cellular automata approach [1, 8]. However, these studies all applied a squared grid, thus some key local features such as turbulence intensity and microscopic topography of riverbed were not well presented. The simulated spatial vegetation patterns on the shoals and floodplains are different from the surveys.

This research developed an UCA model for vegetation dynamics by implementing the irregular triangular grid and integrated it with a two-dimensional hydrodynamic model. The integrated models were applied to simulate the impacts of flow modifications on riparian vegetation evolutions. The following sections will illustrate the development and application of the UCA model through a case study.

## 2 Study Site

The Lijiang River is a famous tourism resort in Southwest China. The flow in the river has dramatic seasonal fluctuations that can reach 12000 $m^3/s$ in rainy season and down to only $12m^3/s$ in dry season. For decades, river regulations have been implemented to ameliorate the flows for tourism cruise. Several reservoirs have been constructed or under construction in the main stem and branches. When all the reservoirs are in operation, the low flow during dry season is expected to reach $60m^3/s$. Since the flow regimes have been and will be further modified by reservoir operation, it is important to quantitatively evaluate the influences on the downstream ecosystem.

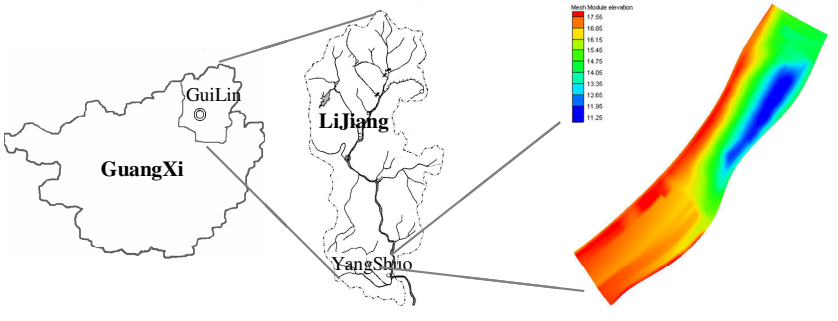



**Fig. 1.** Study site and the channel bed elevation

The study selected a typical compound channel in the downstream of the Lijiang River near the Yangshuo hydrological station, where the influences of all the upstream reservoirs can be perceived (Fig. 1). Field survey was conducted with concerns on both physical and biological features.

## 3 Model Development

### 3.1 Model Framework

To deal with the multiple processes described above, an integrated model is developed. The model framework is illustrated in Fig. 2 where the flow and vegetation

dynamics are solved by hydrodynamic module and UCA vegetation module respectively. The "global change" in Fig. 2 may be caused by either natural or anthropogenic influences, such as river regulations.

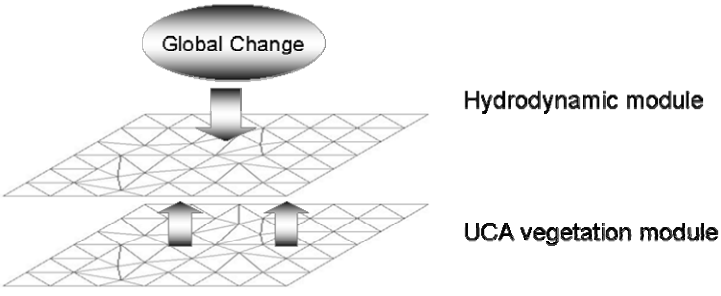



**Fig. 2.** The framework of the integrated hydrodynamic and UCA model

### 3.2 Hydrodynamic Module

The hydrodynamic model solves the depth-integrated equations (1-3) of fluid mass and momentum conservation in two horizontal directions.

$$h\frac{\partial u}{\partial t}+hu\frac{\partial u}{\partial x}+hv\frac{\partial u}{\partial y}-\frac{h}{\rho}\left[E_{xx}\frac{\partial^2 u}{\partial x^2}+E_{xy}\frac{\partial^2 u}{\partial y^2}\right]+gh\left[\frac{\partial a}{\partial x}+\frac{\partial h}{\partial x}\right]+\frac{gun^2}{h^{1/3}}\left(u^2+v^2\right)^{1/2}=0 \quad (1)$$

$$h\frac{\partial v}{\partial t}+hu\frac{\partial v}{\partial x}+hv\frac{\partial v}{\partial y}-\frac{h}{\rho}\left[E_{yx}\frac{\partial^2 v}{\partial x^2}+E_{yy}\frac{\partial^2 v}{\partial y^2}\right]+gh\left[\frac{\partial a}{\partial y}+\frac{\partial h}{\partial y}\right]+\frac{gvn^2}{h^{1/3}}\left(u^2+v^2\right)^{1/2}=0 \quad (2)$$

$$\frac{\partial h}{\partial t}+h\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+u\frac{\partial h}{\partial x}+v\frac{\partial h}{\partial y}=0 \quad (3)$$

In which: $h$-water depth, $u$, $v$ - velocities in the Cartesian directions, $x$, $y$, $t$ - Cartesian coordinates and time, $\rho$ - density of fluid, $E$ - Eddy viscosity coefficient, $g$ - acceleration due to gravity, a - elevation of bottom, n - Manning's roughness.

The boundary conditions were defined using the data from the Yangshuo hydrological station. Daily averaged discharge was applied at the upstream boundary and daily averaged water level was applied at the downstream boundary. The finite element method (FEM) was applied to solve the equations numerically. The simulation time step is 15minutes and the output data is daily-averaged in order to consist with the time step of vegetation module. Six scenarios were simulated that are high, even and low flows for with and without reservoir regulations respectively.

### 3.3 Vegetation Module

Three annual herbs that are *Rumex Maritimus* (*R. Maritimus*), *Polygonum Hydropiper* (*P. Hydropiper*) and *Leonurus Heterophyllus* (*L. Heterophyllus*) were investigated on the channel shelf. The annual herbaceous species were selected in the present study because they complete their life cycle in relatively short time. Field data were collected to reflect the present distribution patterns of these species.

The physical template of the riparian zone is represented by a UCA grid, with physical and biological properties specifying bed morphology and vegetation condition. The simulated plants can "grow" by updating cell properties at every time step. The updating rules are based on several dominant processes of the plant's life cycle. External forcing was introduced by reading the output of hydrodynamic module. The characteristics of each species are obtained from experiments and existing literatures. Table 1 gives the parameters used in the vegetation model. The typical responses of the modelled species to the inundations are presented in Fig 3.

**Table 1.** Empirical values for the parameters of the vegetation module

| Items | *R. Maritimus* | *P. Hydropiper* | *L. Heterophyllus* |
|---|---|---|---|
| Seed weight | 0.0002g | 0.0002g | 0.0002g |
| Max. growth rate | 0.13 | 0.12 | 0.13 |
| Max. biomass per plant | 2.70g | 1.65g | 3.00g |
| Biomass loss rate during inundation | 0 * | 0.02/d | / ** |
| Mortality rate during inundation | 0.05/d (i.d. >40 days) *** | 0.025/d (i.d.>10 days) | 0.8/d (i.d.>5 days) |
| Growth rate decrease during drought | 63% (d.d. >5 days) **** | 27% (d.d.>15 days) | 0 |
| Mortality rate during drought | 0.05/d (d.d.>10 days) | 0.05/d (d.d.>20 days) | 0 |

Note: (*) "0" indicates no biomass loss but not normal growth; (**) *L. Heterophyllus* suffers great biomass loss during inundation, with an assumed mortality rate of 0.1~0.2/d during short period of inundation; (***) i.d.: inundation duration; (****) d.d.: drought duration.

Based on these parameters, the vegetation dynamics were simulated in a UCA grid with main concerns on the flow disturbances during the growing season. The main CA rules are illustrated bellow: (1) Germination: in riparian zone, seed germination takes place along with the recession of the first floods after winter dormancy. Due to relatively high dispersal ability and long dormancy of herbaceous seeds, the compositions of herbaceous seed banks are similar in different areas of a riparian zone and abundant [9]. Therefore, the seed availability was not considered as a limiting factor for the herbaceous species. During the initialization of the model, seeds of the three species are scattered evenly in simulation space, and the amount of geminated seeds is equal to the maximum number of mature plants allowed in a cell.

(2) Growing period: the juvenile plants are most prone to adverse environment and disturbances. The susceptibility (or tolerance) differs among the species, which contributes to the species differentiation along the water level gradients. *R.Maritimus* and *P.Hydropiper* are typical riparian species, while *L.Heterophyllus* is not adapted to inundation and mostly presents on the upper land where floods seldom reach.

(3) Mature period: mature plants of the three species show much more tolerance to adverse environment. They all are able to survive longer inundation or drought stress, and the survival rates are similar. The seeds are produced in this period.

(4) Winter loss: all the annual herbaceous plants die in the winter, and the seeds have a loss rate.

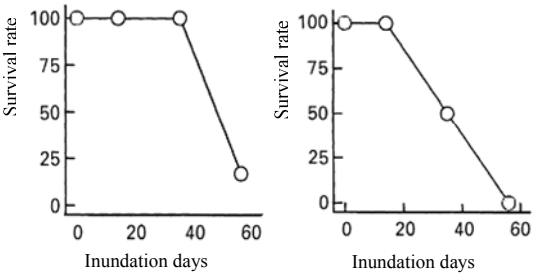



**Fig. 3.** Response curve of *R. Maritimus* and *P. Hydropiper* to inundation stress

(5) The local interactions and species competitions are formulated according to the field survey and lab experiments. Suppose that the resources of per unit area is R=1 (resources/m$^2$), and the resources consumption of species $i$ is $c_i = S/(n_i \times M^i_{\max})$, where $S$-area of sample site, $n$-the number of the species $i$ under optimal conditions in the sample site, $M_{max}$ - the maximum biomass of an individual species $i$ under optimal conditions. Taking *R. Maritimus* for example, the filed survey found that the $n$ =3 and $M_{max}$ =2.7, so the corresponding $C_i$ = 1/(3×2.7). Therefore, the available resource of the cell $k$ is defined as:

$$R_k = 1 - c_i \times B_i^k - c_i \times B_i^n \tag{4}$$

where the $R_k$ – available resource in cell $k$, $B_i^k$ -is the biomass of species $i$ in cell $k$, $B_i^n$ -is the biomass of species $i$ colonizing the cell $k$ from the neighbors. If $R_k$ >0, all species grow normally. Otherwise, two different options are defined: if *L. Heterophyllus* have not reached to the saturated biomass, *L. Heterophyllus* keeps normal growth due to the strong competition capability, while *R. Maritimus* and *P. Hydropipr* decrease. If *L. Heterophyllus* have reached to saturated biomass, it will colonise if there is available resources in the neighbours, or it will sustain saturation.

*R. Maritimus* and *P. Hydropipr* follow the similar rules as *L. Heterophyllus* except that they can only utilise the remained resources due to their low competition capability. In reality, *R. Maritimus* and *P. Hydropipr* hardly reach to saturated level.

The simulation time step of the CA based vegetation model is 1 day. The model reads the outputs (water level and velocity) from the hydrodynamic module and calculates its influence on the plants growth. Although only the water level and flow velocity were taken into account in the present vegetation module, other variables such as water quality could also be incorporated in the similar way.

## 4 Model Results

### 4.1 Hydrodynamic Model Outputs

Fig. 4 shows the daily variations of water levels with and without the reservoir operations in the typical dry year (2004, post-dam).

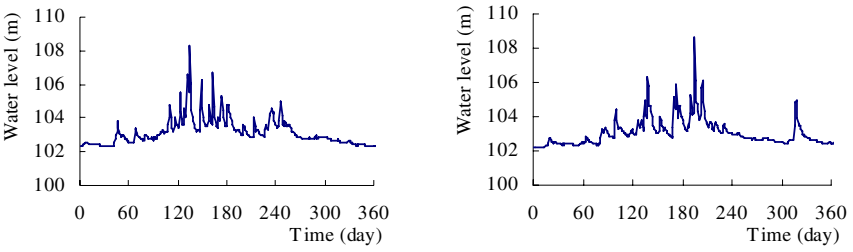



**Fig. 4.** Water levels in typical dry year without (left) and with (right) regulations

Without the reservoir regulation, the floods were frequent and intense, with high peak water level, but relatively short duration of single flood. Under the reservoir regulation, the flood frequency and intensity decreased, but the recession of flood is slower, resulting in prolonged duration of single flood event. Fig 5 is the snapshot of the simulated water depth during low flow and high flow periods within a year. The unsubmerged area of the channel shelf during low flow period provides the habitat for vegetation establishment.

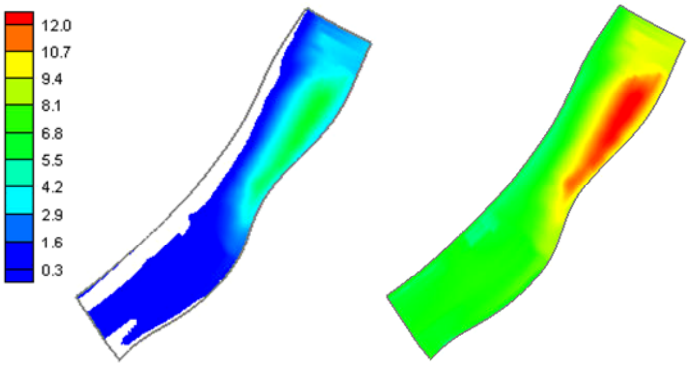



**Fig. 5.** Flow patterns during base flow (left) and high flow (right) seasons

### 4.2 Vegetation Model Outputs

Fig. 6 shows the comparisons of the species distributions with and without reservoir regulations in the typical dry year (2004, post-dam). From the results, the changes of distribution could be perceived. Without flow regulation, the habitat ranges of *R. Maritimus* and *P. Hydropiper* were significantly wider than with regulations. When river regulation was implemented, they both shrunk largely. At the same time, *L. Heterophyllus*, which is more susceptible to flood and tolerant to drought, expanded toward the river side and colonized the habitats that were previously occupied by the *R. Maritimus* and *P. Hydropiper*. The results are in general consistent with both the observations and the statistical niche models.

Fig.7 shows the elevation averaged biomass of each species along the transect. The "elevation averaged biomass" refers to the average biomass of all the plants growing on the same elevation. Because the absolute weights of the species are significantly different, the relative biomass was used instead. The relative biomass was defined as

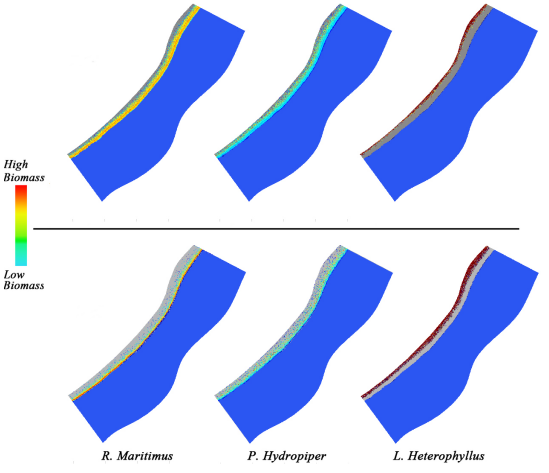



**Fig. 6.** Species distribution in dry year for without (top) and with (bottom) regulations

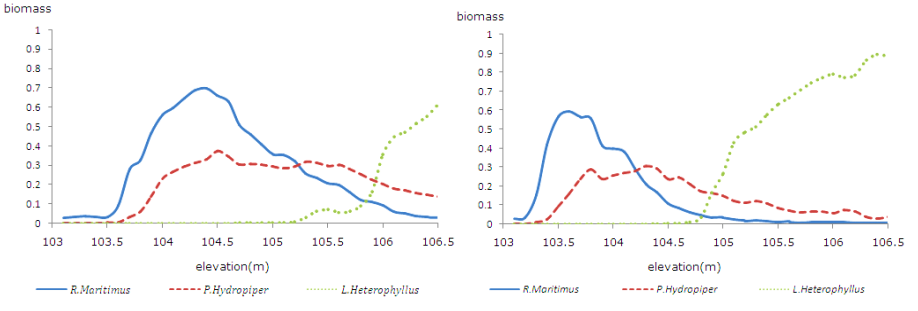



**Fig. 7.** Species gradients in dry year for without (left) and with (right) regulations

the proportion of average biomass to maximum biomass of a single mature plant. The values are then in between 0 and 1.

As a semi-aquatic species, *R. Maritimus* mainly grow in the area close to the river bank. The peak biomass appears in the strip with an elevation a little higher than the water level under base flow. As the elevation increases, the species disappears gradually for the drought stress and the competition from upland species. The distribution of *P. Hydropiper* has a similar pattern as *R. Maritimus*, but the peak biomass appears at a higher elevation because of the difference of flood tolerance. *L. Heterophyllus* start to appear near the peak biomass of *P. Hydropiper* and become dominant gradually at higher elevation where floods can seldom reach.

After the reservoir operation, the distribution of *R. Maritimus* shrunk, but the peak biomass has not been strongly affected. However, *P. Hydropiper* suffered a significant decrease in both the distribution and the peak biomass. Comparing to *R. Maritimus*, *P. Hydropiper* is less tolerant to flood inundation, and it consumes more stored carbohydrate during inundation. But on the other hand, it does require frequent flood disturbances to support the competition with the upland species *L. Heterophyllus*. Therefore, after reservoir operation, less flood frequency and longer flood duration

made the riparian habitat less suitable for the growth of *P. Hydropiper*. *L. Heterophyllus* obviously benefits from the flow modification, with an expanded habitat range and an increased peak biomass.

## 5 Discussions

The research demonstrated that UCA approach is powerful to implement spatial anisotropy and local interactions that are critical to understand riparian ecosystem. In the developed model, the microscopic riverbed topography was well represented by the UCA grid. The species local interactions were implemented at multiple levels, including internal interactions such as competitions and colonisations among the plants, and external interactions between the species and the flow conditions.

Although there are many factors influencing riparian vegetation dynamics, it is impractical and unnecessary to incorporate all of them into the model. The seed dynamics and the flow disturbances during juvenile period had been identified as the key processes of the vegetation models [3, 7, 9]. They determine whether a species is able to establish and its chance to survive after establishment. When the plants become mature, they are less susceptible to adverse environment except for prominent disturbances such as destructive high flows and extremely long inundation duration.

## Acknowledgement

The research is funded by National Nature Science Foundation of China (50639070).

## References

1. Chen, Q., Mynett, A.E., Minns, A.W.: Application of cellular automata to modelling competitive growths of two underwater species Chara aspera and Potamogeton pectinatus in Lake Veluwe. Eco. Mod. 147, 253–265 (2002)
2. Perry, G.L.W., Enright, N.J.: Contrasting outcomes of spatially implicit and spatially explicit models of vegetation dynamics in a forest-shrubland mosaic. Eco. Mod. 207, 327–338 (2007)
3. Aassine, S., Jaï, M.C.E.: Vegetation dynamics modelling: a method for coupling local and space dynamics. Eco. Mod. 154, 237–249 (2002)
4. Blackwell, P.G.: Heterogeneity, patchiness and correlation of resources. Eco. Mod. 207, 349–355 (2007)
5. Birch, C.P.D., Oom, S.P., Beecham, J.A.: Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. Eco. Mod. 206, 347–359 (2007)
6. Trunfio, G.A.: Predicting wildfire spreading through a hexagon cellular automata model. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 385–394. Springer, Heidelberg (2004)
7. Hupp, C.R., Osterkamp, W.R.: Riparian vegetation and fluvial geomorphic processes. Geomorphology 14, 277–295 (1996)
8. Chen, Q., Ye, F., Li, R.: Modelling of the Riparian Vegetation Dynamics due to Reservoir Operations through an Integrated Cellular Automata. In: IAHR 2008, Nanjing, China (2008)
9. Schneider, R.L., Sharitz, R.R.: Seed bank dynamics in a South-eastern riverine swamp. American Journal of Botany 73, 1022–1030 (1986)

# Improving the Behavior of Creatures by Time-Shuffling

Patrick Ediger and Rolf Hoffmann

Technische Universität Darmstadt
FB Informatik, FG Rechnerarchitektur
Hochschulstr. 10, 64289 Darmstadt, Germany
{ediger,hoffmann}@ra.informatik.tu-darmstadt.de

**Abstract.** The goal is to optimize the behavior of moving creatures by using "time-shuffling" techniques. The "creatures' exploration problem" is used as an example for a multi-agent problem modeled by cellular automata. The task of the creatures is to visit all empty cells in an environment with a minimum number of steps. The behavior of a creature is modeled by an automaton taking care of the collisions. Time-shuffling means that two behaviors (algorithms X and Y) are sequentially alternated with a certain time period. Ten different "uniform" (non-time-shuffled) algorithms with good performance from former investigations were used. We defined three time-shuffling modes differing in the way how the algorithms are interchanged. New metrics are used for such multi-agent systems, especially the *success rate* (number of successful explored environments) and the *mean normalized work* (cost). Time-shuffled systems with a time period of around 100 have resulted in much better success rates and lower cost compared to the uniform systems.

**Introduction.** The general goal of our project is to optimize the cooperative behavior of moving creatures modeled in the cellular automata model. The creatures have to fulfil a certain global task in an artificial environment. A creature or agent behaves according to an algorithm (rule, behavior) which is stored in the creature. We distinguish *uniform* and *time-shuffled* systems of creatures. A uniform system comprises creatures in which all creatures behave uniform (similar and time independent). A time-shuffled system comprises creatures which change their algorithm depending on the time (generation counter). In particular we have focused on time-shuffled systems in which two algorithms are activated alternatingly for a period of $T$ generations. The goal of this investigation was to find out with respect to the creatures' exploration problem (explained below): (1) what is the best mode (technique) to shuffle algorithms, (2) what is the optimal period T to alternate the algorithms, (3) how much more efficient are the time-shuffled systems compared to the uniform systems, and (4) which algorithms "harmonize" best, meaning which combinations of algorithms with how many creatures are the most efficient. – The general question behind this

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 345–353, 2008.

particular investigation is: What are the best techniques to combine good CA rules in order to get better ones with a certain probability?

We are modeling the behavior by a finite state machine. In the past we have tried to find out the best algorithm for one creature by enumeration. The number of state machines which can be coded using a state table is $M = (\#s\#y)^{(\#s\#x)}$ where $n = \#s$ is the number of states, $\#x$ is the number of different input values and $\#y$ is the number of different output actions. Note that $M$ increases dramatically, especially with $\#s$, which makes it very difficult or even impossible to check the quality of all algorithms by enumeration in reasonable time. By hardware support (FPGA technology) we were able to simulate and evaluate all $12^{12}$ 6-state algorithms (including algorithms with less than 6 states and including redundant ones) for a test set of 5 initial configurations [1]. The 10 best algorithms (with respect to percentage of visited cells) were used in further investigations to evaluate the robustness (using additional 21 environments) and the efficiency of $k > 1$ creatures. It turned out that several creatures can solve the problem with less cost than a single one [2]. We have recently investigated time-shuffled systems [3] in which the two algorithms were exchanged after each generation (time period $T = 1$). It turned out that only a few of the time-shuffled systems resulted in a better performance compared to the uniform systems. We were not satisfied with these results and therefore we tried to improve our techniques by choosing a longer time period $T > 1$.

Modeling the behavior with a state machine with a restricted number of states and evaluation by enumerations was also undertaken in SOS [4]. Additional work was done by these authors using genetic algorithms. The creatures' exploration problem based on our model was further investigated in [5]. Randomness was added which led to a higher degree of success. The problem is also related to the *first passage problem*. Our research in general is also related to works like: Evolving optimal rules for cellular automata (CA) [6,7] or finding out the center of gravity by marching pixels [8].

**CA model for the creatures' exploration problem.** The problem is the following: $p$ creatures are moving around in an environment that consists of empty cells and obstacle cells in order to visit all reachable empty cells in shortest time. Creatures cannot move on obstacle cells, and only one creature can be located on an empty cell at the same time. Creatures can look forward one cell ahead in their moving direction. The creatures may perform four different actions: $R$ (turn right only), $L$ (turn left only), $Rm$ (move forward and simultaneously turn right), $Lm$ (move forward and simultaneously turn left). The rules for executing these actions are:

(a1) if (front cell == obstacle OR creature) then $R/L$
(a2) if (two or more creatures want to move to the front cell) then $R/L$
(b) else $Rm/Lm$

The action $R$ or $L$ is performed if the "front cell" (the cell ahead) is not free, because it is an obstacle, another creature is located on it, or a collision conflict is anticipated. In all other cases the action $Lm$ or $Rm$ is performed. The detection

of anticipated conflicts is realized by an arbitration signal received from the destination cell. Each creature sends a request to its front cell, which sends back a grant signal if no other creature has sent a request [1]. – The state of a cell comprises the type, the direction $r$ and the control state $s$ (Fig. 1a). The bahavior is defined by a Mealy automaton with inputs $(m, s)$, next state $s'$ and output $d$. An algorithm is defined by the contents of a state table assigned to the state machine. We are coding an algorithm into a string representation or a simplified string representation by concatenating the contents line by line to a string, e. g.: 1L2L0L4R5R3R-3Lm1Rm5Lm0Rm4Lm2Rm or in a simplified form 1L2L0L4R5R3R-3L1R5L0R4L2R.

The state table can be represented more clearly as a state graph (Fig. 1b). If the state machine uses $n$ states, we call such an algorithm $n$-state algorithm. If the automaton is considered as a Moore automaton instead of a Mealy automaton, the number of states will be the product $n \times \#r$, where $\#r$ is the number of possible directions (4 in our case).

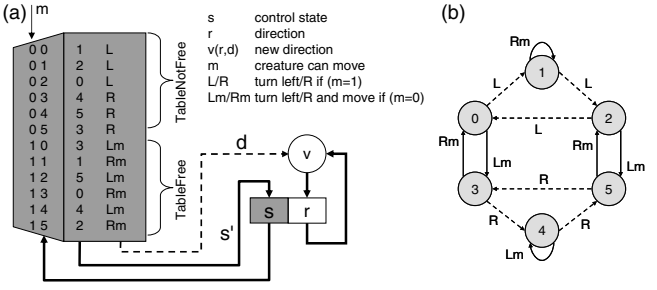



**Fig. 1.** A state machine (a) models a creature's behavior. Corresponding 6-state algorithm (b).

In preceding investigations [1] we could discover and evaluate the best 6-state algorithms for one creature by the aid of special hardware. All relevant algorithms were simulated and evaluated for 26 initial test configurations (they are different from the ones we are using here). The following 10 best algorithms were ranked using a dominance relation with the criteria (1.) success, (2.) coverage and (3.) speed:

1. G: 1L2L0L4R5R3R-3L1R5L0R4L2R
2. B: 1R2R0R4L5L3L-3R1L5R0L4R2L
3. C: 1R2R0R4L5L3L-3R4R2L0L1L5R
4. A: 0R2R3R4L5L1L-1R5R4R0L2L3L
5. D: 1R2R3R1L5L1L-1R0L2L4R3L1L
6. E: 1R2L0R4L5L3L-3R4R5R0L1L2R
7. F: 1R2L0L4R5R3R-3L4L5L0R1L2R
8. H: 1L2L3R4L2R0L-2L4L0R3L5L4R
9. I: 1L2L3L4L2R0L-2L4L0R3R5L4R
10. J: 1R2R3R0R4L5L-4R5R3L2L0L1L

In this investigation we are using the ten previously found algorithms (A to J). In order to find algorithms or combinations of algorithms that are robust against changes of the environment, we have created a new set of 16 environments of size $35 \times 35$ with a fixed number of obstacles (129) (Fig. 2). The grid size allows us to place the creatures at the beginning in regular formations and the constant number of obstacles simplifies the analysis.

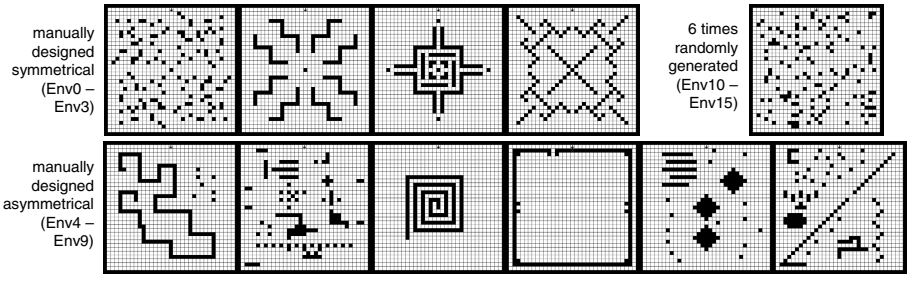



**Fig. 2.** The 16 environments with 35 × 35 cells, manually designed or randomly generated. Each environment comprises $R = 960$ empty cells and 129 obstacles.

**System configurations and metrics.** In [2,9] we have evaluated that increasing the number of creatures can lead to synergy effects, i. e., the uniform creatures can work more efficiently together than by their own. Initially 1 to 64 creatures were distributed with equal distances (if possible) at the borders with directions away from the borders. In order to investigate the performance of *time-shuffled systems* (multi-creature system with time-shuffled algorithms), we have simulated all pairs $(X, Y)$ of the former best single algorithms (A to J) on all 16 environments (Fig. 2) whereas the uniform pairs $(X, X)$ are included for comparison. The two algorithms in the time-shuffled system were interchanged after a period of $T$ generations. The tested values for $T$ were $T = 1$, 3, 5, 6, 12, 24, 48, 96, 384, 1536. We used three different modes of time-shuffling (Fig. 3):

– (c) *common.* A creature interchanges algorithm $X$ with algorithm $Y$ after every time period $T$. But it uses only one common state $s$ which is shared by the algorithms.
– (s) *simultaneous.* A creature works with both algorithms simultaneously. Each of the algorithms has its own state ($s_x$, $s_y$). The output is taken alternatingly from $X/Y$ for $t =$*(generation/T) mod 2 = 0/1.* Both states are updated in every generation.
– (a) *alternate.* An individual state is used for each of the algorithms. For $t = 0$ the output is taken from $X$ and only the state $s_x$ is updated. For $t = 1$ the output is taken from $Y$ and only the state $s_y$ is updated.

The shuffling modes c and s for all pairs of equal algorithms $(X, X)$ are special cases which are in fact uniform systems. In contrast the shuffle mode with a pair $(X, X)$ of equal algorithms does not result in a uniform case because if two algorithms of the type $X$ are active, they can be in a different control state. For reasons of preciseness and comparability we will divide the alternate shuffle mode into *alternate XX*, which only consists of the ten combinations $(X, X)$ and *alternate XY* (90 combinations of the ten algorithms A-J, $X \neq Y$), disregarding the ten pairs $(X, X)$. – We will denote a multi-creature system by *XY-k/pT*, where $XY$ is the pair of algorithms that is shuffled periodically every $T$

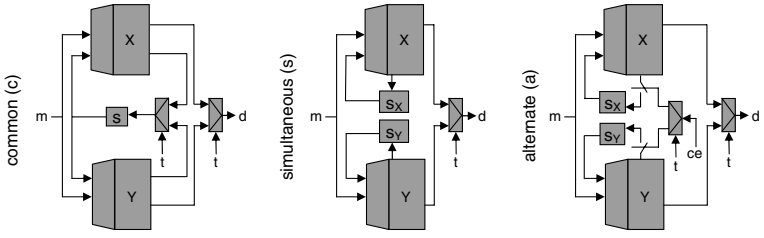



**Fig. 3.** Types of time-shuffling state machines. $X$ is the first algorithm, $Y$ the second. *t =(generation/T) mod 2 = 0/1* is a function of the generation counter and controls the outputs and also the state transition in the cases *c* and *a*. *ce* (clock enable) is a signal that enables the state transition.

generations, $k$ is the number of creatures and $p$ the time-shuffling mode. E. g., AB-8/a48 is a system with 8 creatures using the time-shuffling mode "alternate", periodically switching between the algorithms A and B every 48 generations.

In order to evaluate multi-creature systems the following definitions and metrics were used:

– $R :=$ number of empty cells; – $g :=$ generation (time steps)
– $r_{\max} :=$ the maximum number of cells which can be visited for $g \to \infty$
– $g_{\max} :=$ the first generation in which $r_{\max}$ is achieved
– $e := r_{\max}/R[\%]$, the coverage or exploration rate, i. e. $\frac{\text{visited cells}}{\text{all empty cells}}$
– *successful* := true, if $e = 100\%$
– *number of successes* := number of successful visited environments
– *success rate* := number of successes/no. of environments
– *completely successful* := successful for all environments under test
– *speed* := $R/g_{\max}$ (only defined for successful algorithms)
– $k :=$ number of creatures
– *mean speed per creature* $= ms(k) := \frac{\sum r_{\max,i}}{k \cdot \sum g_{\max,i}}$. The speed $ms(k)$ is an average over all environments $i$ and is normalized to one creature. This measure expresses how fast a creature can visit a cell on average (maximum is 100%). This measure should not be used if any of the environments cannot be visited successfully because otherwise the mean speed might be believed higher than it is reasonable.
– *mean normalized work* $= mw(k) := \frac{k \cdot \sum g_{\max,i}}{\sum r_{\max,i}} = \frac{1}{ms(k)}$. This value represents the work which is necessary, or the *cost* which has to be paid for one creature to visit a cell. Note that $mw$ is the reciprocal of $ms$. We call a system *efficient* if its mean speed is high or its mean normalized work is low.

**Success rates.** The results of simulating the uniform systems with one creature showed that none of them was *completely successful* (capable of solving all environments successfully), while some time-shuffled systems with one creature and a certain time period $T$ were completely successful. In general the success

**Table 1.** The number of completely successful systems (all 16 environments solved) with "alternate" shuffle mode and non-equal algorithm pairs (alternate XY) for each combination of the number of creatures $k$ and the shuffle time period $T$.

| | | k=1 | k=2 | k=4 | k=8 | k=12 | k=16 | k=28 | k=32 | k=60 | k=64 | SUM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNIFORM | | 0/10 | 0/10 | 0/10 | 1/10 | 3/10 | 5/10 | 6/10 | 8/10 | 9/10 | 9/10 | **41/100** |
| ALTERNATE XY | **T=1** | 0/90 | 0/90 | 0/90 | 5/90 | 3/90 | 7/90 | 11/90 | 9/90 | 17/90 | 21/90 | 73/900 |
| | **T=3** | 0/90 | 0/90 | 3/90 | 17/90 | 25/90 | 32/90 | 41/90 | 43/90 | 59/90 | 62/90 | 282/900 |
| | **T=5** | 0/90 | 2/90 | 6/90 | 23/90 | 28/90 | 36/90 | 52/90 | 59/90 | 70/90 | 73/90 | 349/900 |
| | **T=6** | 1/90 | 10/90 | 20/90 | 33/90 | 47/90 | 55/90 | 69/90 | 76/90 | 84/90 | 84/90 | 479/900 |
| | **T=12** | 7/90 | 24/90 | 30/90 | 48/90 | 66/90 | 72/90 | 81/90 | 79/90 | 84/90 | 84/90 | 575/900 |
| | **T=24** | 8/90 | 29/90 | 39/90 | 70/90 | 77/90 | 77/90 | 84/90 | 84/90 | 84/90 | 84/90 | 636/900 |
| | **T=48** | **10/90** | 34/90 | **58/90** | 72/90 | 77/90 | 82/90 | 84/90 | 86/90 | 87/90 | 87/90 | 677/900 |
| | **T=96** | 8/90 | **40/90** | 55/90 | **76/90** | **86/90** | 83/90 | **89/90** | 88/90 | 89/90 | **90/90** | 704/900 |
| | **T=384** | 3/90 | 30/90 | 48/90 | 64/90 | 79/90 | **84/90** | **89/90** | **90/90** | **90/90** | **90/90** | 667/900 |
| | **T=1536** | 2/90 | 12/90 | 34/90 | 69/90 | 74/90 | 75/90 | **89/90** | 88/90 | **90/90** | **90/90** | 623/900 |

rates are increasing with the number of creatures due to synergy effects (Tab. 1). This is true for uniform as well as for time-shuffled systems (only the alternate shuffle mode is shown in that table). The best success rates can be achieved for $T$ between 48 and 1536. The other shuffle modes have shown similar results.

41% of the uniform systems were completely successful (for all 16 test environments) averaged over the number $k$ of creatures. By comparison the time-shuffled systems reach a significant higher percentage for $T > 6$ or $T > 12$, whereas the alternate mode is slightly better than the simultaneous mode and the common mode (Fig. 4). Even the simple mode "alternate XX", using the same algorithm twice but with different control states, reaches a success rate of 68% for $T = 48$. The highest success rate which can be reached for an individual system $XY$-$k$/$pT$

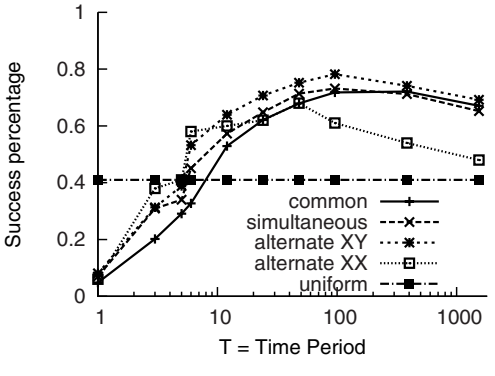



**Fig. 4.** Percentage of complete successes, depending on period $T$ (x-axis) and shuffle mode (different curves). The values are averaged over all pairs $XY$ and over the number of creatures $k$. A logarithmic scale was chosen for $T$ to achieve a better representation.

obviously depends on the individual parameters. In general the success rates are increasing with the number of creatures. E. g., for $k = 64$ the nine uniform systems (B..J)-64 and the 279 systems $XY$-64/(c,s,a)96 with $X$,$Y \in$ A..J are

**Table 2.** The top 5 most efficient (lowest total cost) uniform multi-creature systems (a) and the top 5 overall most efficient time shuffled multi-creature systems (b). The value "mean $g_{\max}$" is averaged over all environments.

| (a) Uniform System | No. of successes | mean g_max | mean speed per creature | mean normalized work |
|---|---|---|---|---|
| J-32 | 16 | 382 | 7.86% | 12.72 € |
| J-16 | 16 | 768 | 7.81% | 12.80 € |
| B-28 | 16 | 456 | 7.52% | 13.30 € |
| J-12 | 16 | 1069 | 7.48% | 13.36 € |
| J-28 | 16 | 463 | 7.41% | 13.49 € |

| (b) Time-shuffled System | No. of successes | mean g_max | mean speed per creature | mean normalized work |
|---|---|---|---|---|
| BJ-28/a96 | 16 | 370 | 9.27% | 10.79 € |
| BC-32/a96 | 16 | 338 | 8.88% | 11.26 € |
| JG-28/c96 | 16 | 388 | 8.85% | 11.30 € |
| GJ-8/a48 | 16 | 1376 | 8.72% | 11.47 € |
| JJ-16/a384 | 16 | 688 | 8.72% | 11.47 € |

completely successful. For $k = 1$, 23 time-shuffled systems out of 280 $XY$ combinations are completely successful, but none of the uniform ones. The optimal time period to switch between algorithms that is unsensitive against changes of the number of creatures is approx. $T \approx 100$, and it is relatively independent of the time shuffle mode.

**Efficiency of the systems.** In the following investigation about efficiency we will consider only systems that were completely successful because the inclusion of not successful systems would either falsify the results (affecting/pretending higher efficiency) or would require other metrics which will not be introduced here. The most efficient uniform systems for the 16 environments were the systems J-32, J-16 and B-28 (Tab. 2). The system J-32 has a mean speed per creature of 7.86%. Assuming that each step of a creature costs 1 €, the mean normalized work is 12.72 €, meaning that each creature needs 12.72 steps to visit a new empty cell on average. If you think of cleaning a room, this might be the price for "cleaning" one "square meter". The overall most efficient system is the time-shuffled system BJ-28/a96 with a mean normalized work of 10.79 € (Tab. 2), which is 16.7% more efficient than the best uniform system J-32.

**Table 3.** The costs for the completely successful systems BJ-$k$/a$T$

| | k=1 | k=2 | k=4 | k=8 | k=12 | k=16 | k=28 | k=32 | k=60 | k=64 |
|---|---|---|---|---|---|---|---|---|---|---|
| **T=3** | | | | | | 32.29 | 33.15 | 30.38 | 30.32 | 34.89 |
| **T=5** | | | | 23.43 | 25.18 | 17.31 | 27.78 | 24.63 | 24.79 | 27.73 |
| **T=6** | | | 20.16 | 20.40 | 18.45 | 19.15 | 18.35 | 18.51 | 20.36 | 22.37 |
| **T=12** | 15.47 | 15.81 | 14.72 | 16.14 | **13.28** | 15.74 | 18.90 | 14.30 | 18.30 | 17.65 |
| **T=24** | | | 14.90 | 15.97 | 14.79 | 15.80 | 15.81 | 15.31 | 14.75 | 18.37 |
| **T=48** | | **13.40** | **12.60** | **13.07** | 14.32 | **12.56** | 12.82 | 15.59 | 16.82 | **13.84** |
| **T=96** | **14.48** | 15.94 | 16.18 | 15.42 | 18.25 | 13.88 | **10.79** | 13.03 | 17.01 | 14.78 |
| **T=384** | | 14.16 | 14.91 | 13.30 | | 17.02 | 15.42 | **12.43** | **13.82** | 14.80 |
| **T=1536** | | | | 13.61 | 17.18 | 15.49 | 13.30 | 14.49 | 17.25 | 17.38 |

The systems BJ-$k$/a$T$ were analyzed in more detail (Tab. 3). Depending on $k$, the optimal time period lies in the range from $T = 12$ to 384, with costs between 10.79 € and 14.49 €. Further we had a closer look to the top 25 most efficient systems (all of them are time-shuffled). The occurrence of the algorithms as part of

a combination is: B (10 times), J (19), G (9), C (7), A (3), F (1), E (1). The leading four algorithms B, J, G, C are also the most efficient in uniform systems. This means that when good uniform algorithms are used in a time-shuffled combination, the resulting system is likely to be a good or a better system. The occurrences of the shuffle modes in the top 25 are: a (11 times), s (9), c (5). The occurrences of the time periods are: $T = 96$ (10 times), $T = 48$ (8), $T = 384$ (5), $T = 24$ (2). The occurrences for a certain number of creatures are: $k = 8$ (7 times), $k = 28$ (5), $k = 12$ (4), $k = 16$ (4) and for $k = 1, 2, 4, 32, 60$ only once.

In addition we have checked for comparison the quality of 32 random walkers (300 runs), which randomly choose the actions $R/L$ respectively $Rm/Lm$. The random walkers needed 1,833 generations to explore all 16 environments, averaged over 300 runs in each environment and averaged over the environments. In comparison the system J-32 needed only 382 generations (4.8 times faster) and the system BC-32/a96 only 338 (5.4 times faster). Furthermore the random walkers were very slow in solving successfully the environment 7 (spiral). Only 238 runs out of 300 were successful (within a time limit of 10,000 generations after a new empty cell was visited).

**Conclusion.** We could show that algorithms which have a good performance in uniform multi-creature systems can be combined by time-shuffling in order to produce a better performance. Three shuffling modes combining two algorithms were defined: (c) "common" using a common state, (s) "simultaneous" using two running states, and (a) "alternate" using two states of which only one is updated. It turned out that the alternate mode is the best choice in order to build successful and efficient systems. The best success rates were achieved for a time period $T$ between 48 and 1536. The optimal time period to switch between algorithms that is robust against changes of the number of creatures is approx. $T \approx 96$, and it is relatively independent of the time-shuffle mode. Uniform systems need at least 8 creatures to be completely successful, whereas time-shuffled systems are already completely successful with one creature only. The efficiency measures (mean speed, mean normalized work) are only reasonable for systems which are completely successful. The top 10 most efficient time-shuffled systems are 17.8 to 34.4% more cost efficient than the top 10 uniform ones.

Further work is in progress to use more general shuffling modes and to optimize them by heuristics, to use other basic algorithms with less and more states, to detect relations between certain cycles in the state graphs and the performance, to use efficiency measures which include unsuccessful systems and to discover relations between conflicts and synergy.

## References

1. Halbach, M., Hoffmann, R., Both, L.: Optimal 6-state algorithms for the behavior of several moving creatures. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 571–581. Springer, Heidelberg (2006)
2. Halbach, M., Hoffmann, R.: Solving the exploration's problem with several creatures more efficiently. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 596–603. Springer, Heidelberg (2007)

3. Ediger, P., Hoffmann, R., Halbach, M.: How efficient are creatures with time-shuffled behaviors? In: Nagel, W.E., Hoffmann, R., Koch, A. (eds.) PASA. Lecture Notes in Informatics, vol. 124, pp. 93–103. GI (2008)
4. Mesot, B., Sanchez, E., Peña, C.A., Perez-Uribe, A.: SOS++: Finding smart behaviors using learning and evolution. In: Standish, R., Bedau, M., Abbass, H. (eds.) Artificial Life VIII: The 8th International Conference on Artificial Life, pp. 264–273. MIT Press, Cambridge (2002)
5. Di Stefano, B.N., Lawniczak, A.T.: Autonomous roving object's coverage of its universe. In: CCECE, pp. 1591–1594. IEEE, Los Alamitos (2006)
6. Sipper, M.: Evolution of Parallel Cellular Machines. LNCS, vol. 1194. Springer, Heidelberg (1997)
7. Sipper, M., Tomassini, M.: Computation in artificially evolved, non-uniform cellular automata. Theor. Comput. Sci. 217(1), 81–98 (1999)
8. Komann, M., Mainka, A., Fey, D.: Comparison of evolving uniform, non-uniform cellular automaton, and genetic programming for centroid detection with hardware agents. In: Malyshkin, V.E. (ed.) PaCT 2007. LNCS, vol. 4671, pp. 432–441. Springer, Heidelberg (2007)
9. Hoffmann, R., Halbach, M.: Are several creatures more efficient than a single one? In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 707–711. Springer, Heidelberg (2006)

# Contact Network Modeling of Flu Epidemics

Ian X.Y. Leung[1], Gareth Gibbs[1], Franco Bagnoli[2], Anil Sorathiya[1], and Pietro Liò[1]

[1] The Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue, Cambridge CB3 0FD, U.K.
[2] Department of Energy, University of Florence, Via S. Marta 3, 50139 Firenze, Italy; also CSDC and INFN, sez. Firenze

**Abstract.** Using actual census, family and age structure, land-use and population-mobility data, we develop a stochastic cellular automata on a social contact network to study the propagation of influenza epidemics in the UK. In particular, we address age dependency and obtain the contact networks through the analysis of location co-presence. We analyze infection propensities as well as vaccination techniques. The results indicate the relative merits of different vaccination strategies combined with early detection without resorting to mass vaccination of a population.

## 1 Introduction

The effective response to an epidemic requires an interdisciplinary and multiscale approach, exploiting aspects from individual health to population spreading patterns. Modeling epidemics has therefore been one of the most valuable tools in answering to these complex issues. Most mathematical models for the spread of disease employ differential equations based on uniform mixing assumptions or ad-hoc models for the contact process (see among others [1,2,3,4,5]).

Spatial diffusion of epidemics has been studied by means of partial differential equations or by the equivalent discretization represented by cellular automata on a regular lattice. The cellular automata method has the advantage of allowing arbitrary transitions among states, thus making easier the modeling of a plausible disease evolution, and to make possible the inclusion of quenched disorder, e.g., geographic constraints.

On the other hand, highly transmissible epidemics like flu are better modelled on the social network of contacts, which is more important than the geographic distance between any two people. However, the determination of the social network from available data is far from being trivial. Another possibility is that of resorting to agent-based modeling, following each agent during its displacements. This is expensive in computational terms, while the additional level of accuracy in the description is rarely justified by available experimental data. Eubank et al. have employed TRANSIMS [6] for modeling transport infrastructure, and built EPISIMS[7] for simulating disease spread. Other examples can be found in Refs. [8,9,10].

We employ here a cellular automata model of epidemics on an effective social network that is constructed by considering the actual interactions as the time

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 354–361, 2008.

average of the persistence of individuals being in the same location. Our model utilises large-scale demographic and mobility data based on actual census and land-use. Notably, Brownstein [11] showed that influenza spread firstly in children aged 3-4 years old. Following these findings and data from HPA [12], our model considers that flu season begins with preschoolers.

## 2 Methodology

We consider a generalized cellular automata model in which the cells or nodes are the individuals, and the connections are defined by an adjacency matrix $J_{ij} \in \{0, 1\}$. In a regular lattice, $J_{ij}$ is translationally invariant.

The individual variable $x_i$ corresponds to the disease states: susceptible ($S$), exposed ($E$), infectious ($I$), recovered ($R$). Each individual has moreover a set of properties such as age, residential place, work place, etc. This can be considered equivalent to a quenched disorder.

Susceptible individuals may become exposed to the disease with probability $P_{E|S}$ if they come into "significant" contact with infected individuals through the contact networks. This transition probability also depends on some external parameters such as the virulence $V$ of the flu strain and a seasonal factor $T$, that may be used to tune the simulations. Exposed individual may become infectious after an incubation period (given by the inverse of the incubation rate $P_{I|E}$); the recovery rate given by the inverse of the probability $P_{R|I}$ governs the average length of the infectious period. Vaccinated individuals are also considered recovered. The model thus belongs to the class of SEIR model [13], in which the crucial parameter is the asymptotic fraction of exposed people after the epidemic has become extinct.

One of the main points of our study is to construct the people-people social contact network $J_{ij}$ using locations, ages, residential and work places, and considering the time that two people spend together, in average, in such places. In principle, $J_{ij}$ should be considered a weighted matrix. Due to the lack of precise data, we set a threshold and classify contacts in two classes, "important ones" ($J_{ij} = 1$) and "influent" ones ($J_{ij} = 0$). Moreover, we distinguish the contact network according to the location $k$, so $J_{ij}(k) = 1$ if, in average, people $i$ and $j$ are expected to spend daily a significant fraction of time in location $k$.

### 2.1 Generating the Contact Network

In generating the set of people in the simulation, we utilise actual statistics as well as certain assumptions. For instance, people under 21 years old are more inclined to attend school or college rather than going to work. The age groups of the population are split into the following: 0-4, 5-14, 15-21, 22-44, 45-64, 65-90 (and above). The simulated age distribution of the population corresponds to actual statistics from the UK Census 2001[1], as depicted in Figure 1 (left). The probability of each age group is then calculated accordingly.

[1] http://www.statistics.gov.uk/

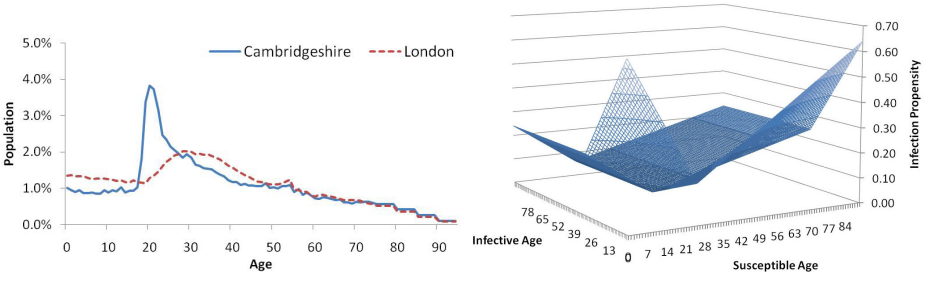



**Fig. 1.** Population Structure of Cambridgeshire and London (*left*). Pairwise infection propensity varies with age (*right*).

If person $i$ goes to location $k$, an edge $T_{ik}$ is drawn. We could also extend this definition by weighing the link with the time spent in the location, and considering time coincidence. However, due to the lack of precise data, we just consider the average one-day window, and set $T_{ik} = 1$ is the person is expected to spend more than one hour in that location. The effective social contact network $J_{ij}$ is then obtained as

$$J_{ij}(k) = T_{ik}T_{jk}. \tag{1}$$

The major characteristic of our system is the division of locations into three distinct groups - public places, workplaces/schools and families. Intuitively, individuals are restricted to travel to certain locations according to their age. Four workplace/school types, one for each of the first four age groups, are explicitly represented in our model. Most individuals of age 0-64 are assumed to attend one school or to work in one workplace. Public place degree and work place degree are assumed to be normally distributed with means 3 and 1, respectively, both with variance 1. Elderly of age 65 or above is assumed retired and can only go to public places.

As suggested by EPISIMS [7], the degree distribution of locations obeys a power law distribution with an exponent, $\gamma$, of roughly 2.8. Since the number of edges is equal to the sum of all individuals' degrees (which can be calculated by the population data and the above assumptions), the degree distributions of public places and workplaces are estimated as follows:

$$n_i = c|L|i^{-\gamma}; \tag{2}$$

$$c = \left(\sum_{i=d_0}^{d_1} i^{-\gamma}\right)^{-1}, \qquad \text{since} \qquad \sum_{i=d_0}^{d_1} n_i = |L|; \tag{3}$$

$$|L| = \frac{\gamma - 2}{\gamma - 1} \cdot \frac{\sum_{p \in P} \mathrm{Deg}(p)}{d_0}, \tag{4}$$

where $n_i$ is the number of locations with degree $i$, $L$ and $P$ are the set of locations and people, $\mathrm{Deg}(p)$ is the degree of the person $p \in P$, $d_0$ and $d_1$ are the minimum and maximum degree of any location. Effectively, each person on

average is involved in total five contact networks from the three respective types of locations. Apart from the family contact network which is generated based on household statistics, the other networks are generated according to the above estimated degree distributions. Note that $d_1$ can be estimated by $d_0 \cdot |L|^{\frac{1}{\beta}}$. After the set $L$ and $P$ are generated with their respective degrees, the probability of the existence of an edge between any $p \in P$ and $l \in L$ is $\mathrm{Deg}(p) \cdot \mathrm{Deg}(l)/\sigma$ where $\sigma$ is the total number of edges.

### 2.2 Disease Spreading and Intervention Mechanisms

The disease starts to spread through the social contact networks within the outbreak area after parameter initialisations (initial infective, virulence, incubation rate and recovery rate), followed by diffusions to other areas. In estimating the probability of a susceptible becoming exposed, we first define the notion of pairwise propensity of infection $Q(i, s)$ between a susceptible of age $s$ and infective of age $i$:

$$Q(i, s) = \mathrm{Inf}(i) \cdot \mathrm{Sus}(s) \cdot V \cdot T, \tag{5}$$

where $\mathrm{Inf}(i)$ is the typical infectivity at age $i$ and $\mathrm{Sus}(s)$ is the typical susceptibility at age $s$. An example of how pairwise infection propensity varies with infectives and susceptibles of different age is depicted in Figure 1 (right). In this case they are estimated by some linear functions and can be easily further refined. Since certain viruses are known to be more persistent and pathogenic than others [14], we attempt to capture this concept by incorporating $V$, the virulence of the virus. Time of the year, described by the seasonal factor $T$, is also known to contribute largely to the prevalence of influenza[15]. For our purpose, $T$ are $V$ are assumed to be constant. To estimate the overall probability $P_{E|S}(s, l)$ of susceptible of age $s$ catching the disease in the location $l$, we take the normalised summation of the above pairwise infection propensity:

$$P_{E|S}(s, l) = \frac{\sum_i J_{is}(l) \cdot Q(i, s)}{\mathrm{Deg}(l)}. \tag{6}$$

Noticeably, the probability of catching the disease in each neighbourhood should be proportional to the time of stay. An individual has more chance to contact with family members and colleagues than a stranger in say the same shopping center due to the spatial size of the location. Since the degree of a location somewhat reflects the spatial size and therefore the chance for a close contact with an infective, we see how the above formulation is more sensitive to the presence of infectives in a smaller place than a larger one.

The granularity of an area is crucial to the feasibility of the model. According to the UK Census 2001, Output Areas(OAs) are based on postcodes normally with size larger than 100 households. All the data used in simulation are based on the OAs area level.In modeling the disease spread across areas, we look at both the population density data and the UK national travel survey [16] which contains traffic information on an area level. We then estimate the number of individuals that are likely to travel to different areas every day. Needless to say, they may either take out or bring back the flu with them.

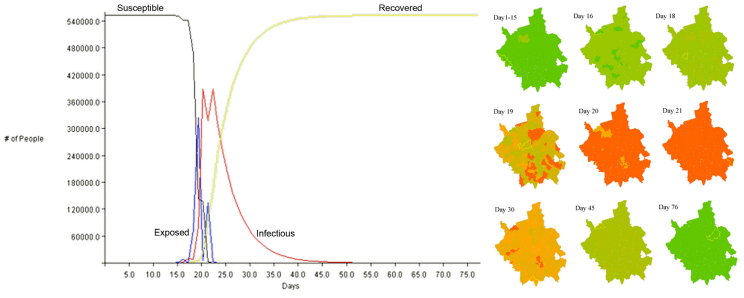



**Fig. 2.** A typical infective-time graph (*left*) and the corresponding epidemic spread in areas (*right*)

Three types of vaccination programs are defined in the model: age-prioritised, targeted and randomised regional vaccination. The first one prioritises on vaccinating a particular age group; the second attempts to vaccinate those who have co-located with an infective whilst the last one is completely random. It may well be argued that none of these schemes is realistic. In real world scenarios, an epidemic would often have started before it was discovered by authorities. Tracing for say those who have contacted the infectives may not be as easily done as on a computer model. Vaccine of particular type may also not be effective against the prevalent type of virus. Nonetheless, we think this will help in understanding different containment strategies and thereby preventing epidemics from developing into a worldwide pandemic.

## 3 Results and Discussion

We have focused on the spread of influenza in England. The main simulation is carried out in the county Cambridgeshire.

Figure 2 depicts a typical spread as modelled in our system. The peak of the exposed people curve is followed by the peak of the curve for number of infectious people. To ease the comparisons, we define the impact of an epidemic to be the sum of infected and exposed individuals per day across the whole period (the sum of the areas under the infective and exposed population curves). Two key assumptions made in the following discussions are that exposed individuals are allowed to recover directly, without being infectious; and vaccination is only effective to susceptible individuals.

With the recovery rate fixed at 0.3, we first look at the effect on the impact of varying the virulence and incubation rate as shown on the left part of Figure 3. The impact surface is plotted by 10,000 data points to smooth out the randomness. It is clear that the total impact of the epidemic increases with both the incubation rate and the virulence of the virus. Based on the assumptions, we see that the total impact increases with both incubation rate and virulence from certain critical combinations with a gradually decreasing steepness (a very different distribution is seen if less or none exposed individuals are allowed to

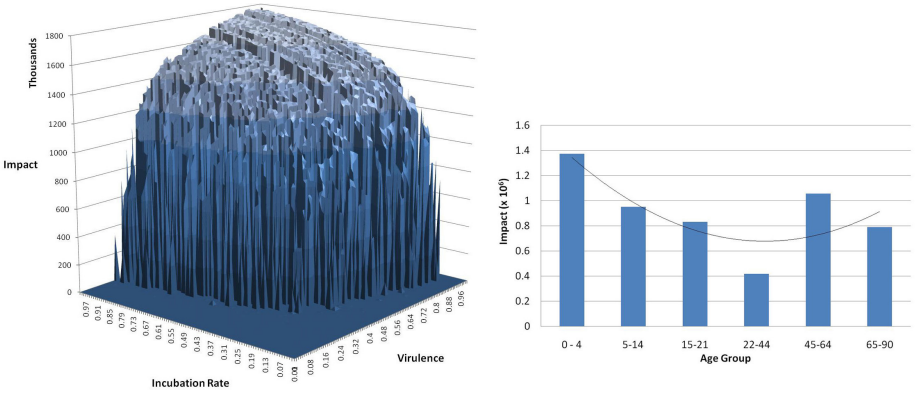



**Fig. 3.** Effect of virulence, incubation rate (*left*) and initial infective age (*right*) on Impact

recover directly, without being infectious). All 10000 runs are assumed to have only one random initial infective. Randomness described by the large number of dents on the surface accounts for majority cases of self-contained spreading of virus in real life scenarios, even when the virulence is high.

The right part of Figure 3 depicts the impact of an epidemic caused by initial infectives of different ages over 10 repetitions with $V = 1$, incubation rate $= 0.5$ and recovery rate $= 0.3$. The reader may wonder why the discrepancies occur when, as we have just seen, the total impact distribution should be independent of the initial infective's age. The reason is that this average is dominated by the number of "successful" epidemics over the 10 repetitions. In simpler terms, a potential epidemic is more likely to be "successful" if the initial infective is of age 0-4 given the current settings.

Lastly, we compare the three vaccination strategies. An *area*-wide threshold percentage of infections(delay threshold) is used to trigger a *region*-wide vaccination program to simulate the delay caused by imperfect monitoring. Vaccination supply is constrained by giving each area per day a number of vaccinations proportional to its population (vaccination percentage). For instance, 50% allows the area to be completely vaccinated in 2 days. We again assume that $V = 1$, incubation rate $= 0.5$, recovery rate $= 0.3$ and the vaccinations are 100% effective on individuals of all age. Suppose we know that the epidemic is started off by a 3 year old child, we plot the respective impacts for the three strategies against the vaccination percentage and the delay threshold. Predictably, as shown in the top part of Figure 4, a smaller delay threshold and faster rate of vaccination always resulted in the least impacts on all three cases. To compare the three strategies, we look at the three pairwise difference charts between them in the bottom part of the figure. A colour corresponding to the top diagrams is used to show the strategy that performed better (resulted in *less* impact) in that specific setting. In general, targeted vaccination outperformed the other two when both the delay threshold and vaccination percentage are low. The supremacy of targeted

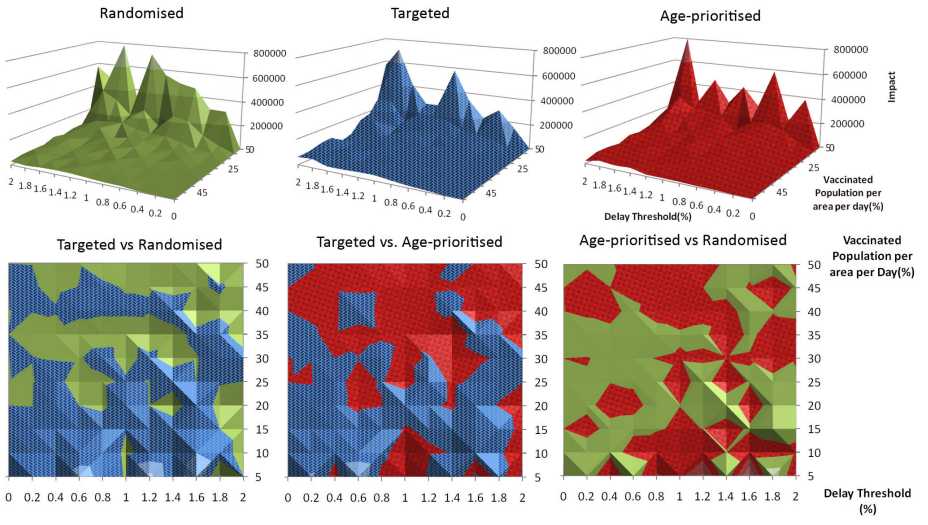



**Fig. 4.** Performance Comparison between three vaccination strategies

vaccination became less eminent as the delay lengthened. This is understandable in that given the delay, the targeted individuals may have already outnumbered the limited vaccination supply and many of them may have already been exposed. As the constraint on allowed vaccinations loosens, i.e. the vaccination percentage increases, age-prioritised vaccination starts to outperform the other two. This indicates that blindly prioritising the vaccination on one age group may not be beneficial if the daily vaccine provision is limited. Surely this allows ample time for other potential age groups to propagate the disease unless the vaccination is carried out as swiftly as possible.

## 4 Evaluation and Future Work

In this paper, we described a contact network model for modeling epidemics with emphasis on demographic information. We discussed the parameters of the model and tested three vaccination strategies. We found that targeted vaccination, albeit unrealistic, is a better strategy under normal circumstances but given less constraint on the provision of vaccines, age prioritised vaccination prevails.

An ideal time dependent bipartite graph requires every individual in the system to be in only one place at a reasonably small time step and evolves according to some mobility models. Various analyses of actual human social dynamics exist [17]. When a social mobility model is employed and thereby a more realistic contact network structure is known, the current people-people infection propensity can be refined. To save the computational cost, the time step is one day and once

the people-location bipartite graphs are built they remain unchanged throughout the simulation. This static projection graph is more highly connected than the time dependent version and is thus a safe simplification to make.

Certain virus is known to be able to survive in normal environment for hours and even days. This can be incorporated into our model by imagining a contaminated location as an infective itself, but we think this unnecessarily increases the degrees of freedom in the model and have therefore omitted this phenomenon. Birth rate and death rate (the MSEIR model [5]) are also not considered.

To conclude, both epidemic spreading and vaccination strategies are highly intricate and stochastic. Our model has shown how one infected individual is sufficient in causing a small scale infection that involved only a few people in a few days to a half a year region-wide epidemic. The ultimate aim of any kind of epidemic modeling is to bolster the development of better efficient countermeasure strategies and this will require further insights into social networks, vaccination and virus-specific pathogenesis.

**Acknowledgements.** This project is supported by EC IST SOCIALNETS project—Grant agreement number 217141.

## References

1. Rvachev, L., Longini, I.: Math. Biosci. 75, 3 (1985)
2. Grenfell, B.T., et al.: Nature 414, 716 (2001)
3. Longini, I., et al.: Am. J. Epidemiol. 123, 383 (1986)
4. Meyers, L.A., et al.: J. Theor. Biol. 240, 400 (2006)
5. Hethcote, H.W.: SIAM review 42, 599 (2000)
6. Nagel, K., et al.: Int. J. Complex Systems 244 (1998)
7. Eubank, S.S., et al.: Nature 429, 180 (2004)
8. Ferguson, N.M., et al.: Nature 437, 209 (2005)
9. Epstein, J.M., et al.: Brookings Institute Center on Social and Economic Dynamics Working Paper 31 (2002)
10. Bian, L.: Environment and Planning B: Planning and Design 31, 381 (2004)
11. Brownstein, S.J., et al.: Am. J. Epi. 162, 686 (2005)
12. Hpa national influenza season summary (2007), http://www.hpa.gov.uk
13. Bailey, N.T.J.: The Mathematical Theory of Infectious Diseases. Murray, J.D. 2nd edn. Mathematical Biology. Springer, Berlin (1993)
14. Rabenau, H.F., et al.: Med. Microbiol. Immunol. 194, 1 (2005)
15. Lowen, A.C., et al.: PLoS Pathog. 19, 1470 (2007)
16. Bureau of Transportation Statistics Highlights of the 2001 National Household Travel Survey (2003), http://www.bts.gov/publications/
17. Eagle, N., Pentland, A.: Personal and ubiquitous computing 10, 255 (2006)

# A Slight Delay in the Onset of Conservation May Bring about an Abrupt Increase of Extinction Risk: Perturbation Experiments in an Ecological Lattice Model

Hiroyasu Nagata, Jin Yoshimura, and Kei-ichi Tainaka

Graduate School of Science and Technology, Shizuoka University, Hamamatsu 432-8561, Japan
f5645024@ipc.shizuoka.ac.jp
jin@sys.eng.shizuoka.ac.jp
tainaka@sys.eng.shizuoka.ac.jp

**Abstract.** We carry out perturbation experiments for both extinction and recovering processes by Monte-Carlo simulations on a finite square lattice, Simulations are tried out by two different methods: local and global interactions. We explore fluctuation enhancement which means the degree of population uncertainty. We find the following results. When an endangered species is recovered, FE strongly emerges. By a slight delay of conservation policy, FE becomes high and the recovering process often fails. These results are remarkable for local interaction. We discuss the issue from the assumption that the protection case may be help towards populations against an endangered species.

**Keywords:** fluctuation enhancement(FE); perturbation experiment (PE); local interaction(LI); global interaction(GI); contact process(CP).

## 1 Introduction

Today, a great number of species continue to go extinct. Almost all extinction is caused by human perturbations. Study of PE is served for the prediction of ecological managements [1]. The most familiar approach is the press perturbation, where one or more condition parameter are altered and held at higher or lower levels. It is well known that the response of an ecosystem contains uncertainty [2]. New stationary states after perturbations are very difficult to predict.

In recent years, lattice models are widely applied in the field of ecology [3]. We use CP which has been extensively studied from mathematical [4], physical [5] and ecological [6] aspects. The CP is a lattice version of the logistic equation. Considerable data on CP have been accumulated, but they are mainly related to stationary state. New data for dynamic process of CP are very rare [7]. Those studies press or pulse perturbations are investigated, but not focus on the recovering process of endangered (nearly extinct) species.

Ecosystems suffer a variety of fluctuation or noise [8]. One of causes of fluctuation is finiteness of population size. When the abundance of a species becomes

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 362–367, 2008.

small, the risk of extinction increases. So far, several authors have estimated the risk of extinction. A typical approach is to apply stochastic differential equations [9] which contain a noise term. Although these equations play an essential role for risk estimation, they have some ambiguity. Another approach to explore the extinction probability is to use the discreteness of the birth-death process. In this case, the intensity of noise can be automatically generated [10]. Our study uses a spatially explicit model (lattice model), is more general because it includes LI. Lattice models are useful to deal with the so-called the FE [11]. We describe the uncertainty in the dynamic process.

## 2 Model and Method

### 2.1 Contact Process (CP)

We deal with CP which is composed of a single species, extensively studied by various fields [4]. The size of lattice is assumed to be finite (L), but the total number of lattice sites ($L \times L$) is much larger than unity. Each lattice site is labeled vacant site (O) or the occupied site (X) by the species. Interactions are defined by

$$\mathrm{X} + \mathrm{O} \longrightarrow 2\mathrm{X} \quad (\text{rate r}), \tag{1a}$$

$$\mathrm{X} \longrightarrow \mathrm{O} \quad (\text{rate m}). \tag{1b}$$

The reaction(1a) means the reproduction of the species, r is the reproduction rate of the species, always set r =1. The reaction(1b) is death process, m denote the mortality rate. When interaction occurs locally, the system(1) corresponds to CP. The reaction(1) is carried out in two different methods: Local interactions(LI) and Global interactions(GI). We first describe the simulation method for LI:

**1)** Initially, we distribute individuals X on a square lattice. The initial distribution is not so important, since the system evolves into a stationary state.
**2)** Spatial pattern is updated in the following two steps:
  **(i)** We perform two-body reaction (1a): Choose one lattice site randomly, and then randomly specify one of four neighboring sites. If the pair of chosen sites are (X,O), then the latter site is changed into X. If other pairs are chosen, we skip to the next step.
  **(ii)** We perform one-body reaction (1b). Choose one lattice point randomly; if the site is occupied by X, the site will become the vacant site (O) with probability m.
**3)** Repeat step 2) $L \times L$=100 × 100 times, called the Monte Carlo step [12].
**4)** The step 3) is repeated. If an extinction event occurs, we stop the simulation run. Here we have employed periodic boundary conditions, replacing any index that exceeds the boundary L, by using the modulo operator on the index.

Next, we describe the method for GI (mean-field simulation) in which long-ranged interactions are allowed, almost the same as LI, but step (i) in 2) LI is replaced with (i)' Two lattice sites are randomly and independently chosen. If these sites are (X,O), then the latter site is changed into X.

GI has a large mathematical merit, the population dynamics can be represented by the mean-field theory. If the lattice size L is infinitely large, we have dx/dt=rx(1-x)-mx. The first term in the right hand comes from the birth process; the factor (1-x) means the density of empty site. The second term is originated in death process. The equation is just the same as logistic equation, so that the population evolves into a stationary state. The steady-state density K can be obtained by setting the time derivatives to be zero, K=1-m/r.Extinction thus occurs, so long as m is larger than a critical value $m_c$. From above equation, we have $m_c = 1$ for GI because r=1.

### 2.2 Fluctuation Enhancement (FE)

FE has been originally studied in the field of physics[12]. Here FE indicates the uncertainty in the dynamical process of phase transitions. An idea is applicable to population ecology, because an extinction of species can be regarded as a kind of phase transition. Our laboratory[7] studied the stochastic process for CP, and found FE. They altered the mortality rate m of a target species to higher or lower level. The same PE was repeated M times (M ensembles) with different random numbers. From the $i-th$ ensemble, the density $x_i$(t) of the species was recorded. We define the ensemble average A(t) and its variance V(t) by

$$A(t) = \frac{1}{M}\sum_{i=1}^{M} x_i(t), \tag{2}$$

$$V(t) = \frac{1}{M}\sum_{i=1}^{M}[x_i(t) - A(t)]^2. \tag{3}$$

The emphasized quantities A(t) and V(t) are averaged over M kinds of ensembles. The FE constitutes an extreme increase in the value of V(t). The variance rapidly increases, or FE occurs and so on, there are a variety of dynamic processes. Here we briefly report the previous results of FE for CP, extinction occurs, so long as m is larger than a critical value $m_c$, known that $m_c$ =1 for GI, $m_c$ =0.618 for LI. When m is increased up to a value near $m_c$, FE was observed, especially, it is clear for local interaction (Fig. 1). Results of decrease experiment of m, we notice that FE is not observed for both LI and GI.

### 2.3 Perturbation Experiments (PE)

We explain the way of PE. Initially t=0, m is set $m_1$, where the species is assumed to survive. The population reaches a steady state for $t > 50$. We adjust $m_1$, as the steady-state density K takes the same value between GI and LI. In order to get K=0.7, we put $m_1$=0.3 for GI, $m_1$=0.27 for LI. The first perturbation is

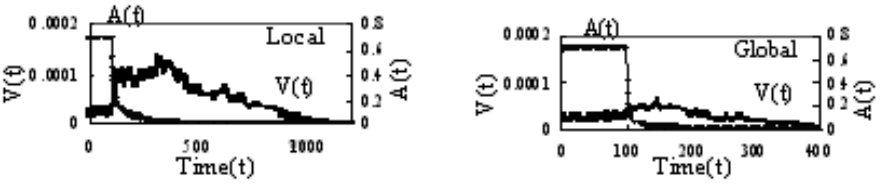



**Fig. 1.** The previous results of PE, $L \times L$=100 × 100. The time dependencies of both A(t) and V(t). Left: LI ($m_1$=0.27, $m_c$=0.618), clearly emerges FE. Right: GI ($m_1$=0.3, $m_c$=1). m was increased from $m_1$ to the critical value $m_c$.

applied at time t =100, m is increased from $m_1$ to $m_c$, and is held at this value, $m_c$ =1 for GI, $m_c$ =0.618 for LI. The species eventually goes extinct, if there are no conservation efforts. Then the second perturbation is applied at t = (100+T), m is recovered to $m_1$. If the period T is very short, the species always recovers its population size. If T is too long, then the species may go extinct. We repeat such an experiment M times (M ensembles) with different random numbers, and obtain of A(t) and V(t) defined by equations (2) and (3).

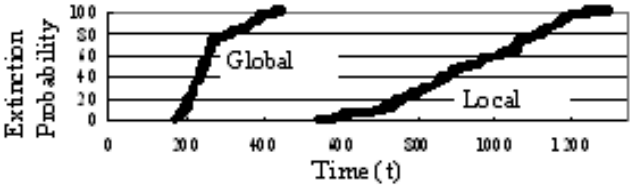



**Fig. 2.** Results for the first PE. Time dependence of extinction probability, defined by the cumulative frequency of extinct ensembles is illustrated.

## 3 Results

### 3.1 Results of the Extinction Process(First PE)

Since the m is increased from $m_1$ to $m_c$, the species eventually goes extinct. FE was clearly observed for LI. We obtaingextinction probabilityh defined by the cumulative number (frequency) of ensembles that extinction occurs. In Fig.2, the time dependence of extinction probability is illustrated for both GI and LI. It is found from this figure that in the case of GI, the extinction probability much quickly increases compared to LI. Hence, we find for GI that slight delay of conservation policy may bring about the abrupt increase of extinction risk.

### 3.2 Results of the Recovering Process(Second PE)

At time $t = (100 + T)$, the m is decreased from $m_c$ to $m_1$. The population recovers its density, so long as the species survives. Fig. 4 displays a typical time dependence of spatial patterns (T=440). (a) represent an initial spatial

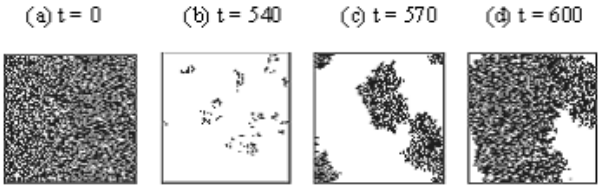



**Fig. 3.** Snapshots of typical spatial patterns. The second perturbation starts from pattern (b), for LI with T=440.

pattern (random distribution), and (b) is the pattern just before the second PE. Recovering process is thus displayed (c) and (d). Note that the spatial distribution for GI is always random.

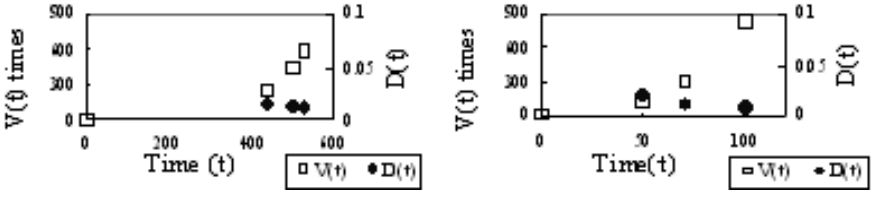



**Fig. 4.** Left is the time dependences of V(t) and D(t) for LI. Right: for GI.

The time dependences of species density are depicted. We find in the case of LI that the population sizes are slowly recovered, and there are a variety of recovering paths. Fig.4 Left displays the time dependences of V(t) and D(t)for LI. FE is extremely amplified in the recovering process. V(t) at the peaks are at least 100 times over larger than those in Fig. 1. Hence, uncertainty in population dynamics is strongly enhanced, when conservation efforts are started. Since there are a lot of recovering processes, it is difficult to predict the conservation success. Moreover, Fig.4 reveals that the value at the peak increases, when T becomes large.

Right Figure is for GI, FE can be observed too. The value of variance eventually reaches a constant value. Such a stationary value means that all ensembles reach steady states; extinction occurs in some ensembles, or the density has 0.7 in other cases. The final stationary value of A(t) and V(t) can be obtained. Let the frequency of survival ensembles be f, then we have A*=0.7f and V*=(0.7)2f(1-f). The final value of V(t) has a maximum, when half ensembles go extinct (f=1/2).

## 4 Discussion and Conclusion

Note that the second perturbation is applied in the case that the species is endangered. We find the following results,

**1)** When an endangered species is recovered, FE strongly emerges.
**2)** By a slight delay of conservation practice, FE becomes extraordinary high. extinction events also increase rapidly with a short delay.

In second perturbation, the magnitude of FE is extremely enhanced. Therefore, a short delay in conservation practice introduces significant uncertainty in the courses of recovery. Extinction events also increase with such a delay. Thus the delay in conservation practice increases uncertainty in the population dynamics and the risks of extinction.

FE in recovering process is much larger than extinction, and it is relatively slow. After the conservation efforts, FE is strongly enhanced; there are a variety of processes after conservation practices. The recovering process proceeds very slowly, so that the species stays in endangered state for a long period. Thus a short delay in conservation practice also increases the risk of extinction. Here slow delays may mean slow and uncertain recoveries. Thus the extinction possibility increases with a delay in the conservation practice. We only focus on the randomness originated in the finiteness of population size. If we take into account various cause of randomness,then the extinction risk may increase.

## References

1. May, R.M.: Stability and complexity in model ecosystems. Princeton Univ. Press, Princeton (1973)
2. Schmitz., O.J.: Press perturbations and the predictability of ecological interactions in a food web. Ecology 78, 55–69 (1997)
3. Durrett, R., Levin, S.A.: The importance of being discrete and spatial. Theor. Popul. Biol. 46, 363–394 (1994)
4. Harris, T.E.: Contact interaction on a lattice. Ann. Prob. 2, 969–988 (1974)
5. Marro, J., Dickman, R.: Nonequilibrium phase transition in lattice models. Cambridge Univ. Press, Cambridge (1999)
6. Itoh, Y., Tainaka, K., Sakata, T., Tao, T., Nakagiri, N.: Spatial enhancement of population uncertainty near the extinction threshold. Ecol. Model. 174, 191–201 (2004)
7. Tainaka, K., Hoshiyama, M., Takeuchi, Y.: Dynamic process and variation in the contact process. Phys. Lett. A 272, 416–420 (2000)
8. Halley, J.M., Inchausti, P.: The increasing importance of 1/f-noises as models of ecological variability. Fluctuation and Noise Letters 4, R1–R26 (2004)
9. Lande, R.: Mutation and conservation. Conservation Biology 9, 782–791 (1995)
10. Renshaw, E.: Modelling biological populations in space and time. Cambridge Univ. Press, Cambridge (1991)
11. Kubo, R., Matsuo, K., Kitahara, K.: Fluctuation and relaxation of macrovariables. J. Stat. Phys. 9, 51–96 (1973)
12. Tainaka, K.: Lattice model for the Lotka-Volterra system. J. Phys. Soc. Jpn. 57, 2588–2590 (1988)

# Lattice Population and Optimality of Sex Ratio: Effect of Sterile Male

Tokiya Nitta[1], Kei-ichi Tainaka[2], Yukio Sakisaka[3], Bungo Saito[2], Tatsuya Togashi[4], and Jin Yoshimura[2]

[1] Department of Nautical Engineering, Tokai University, Shizuoka 424-8610, Japan
[2] Department of Systems Engineering, Shizuoka University, Hamamatsu 432-8561, Japan
[3] Ibaraki Prefectural University of Health Sciences, Ami 300-0394, Japan
[4] Marine Biosystems Research Center, Chiba University, Kamogawa, 299-5502, Japan

**Abstract.** To know the optimality under given conditions is one of most important problems in various fields. In the present paper, we focus on the optimality of sex ratio $\alpha$ in animals. So far, the optimal sex ratio has been obtained by evolutionarily stable strategy (ESS) in most cases. Recently, however, our coauthors have presented a lattice model of mating population to explain the optimality not by ESS but by the sustainability. They mainly studied the symmetrical case between male and female. In the present paper, we deal with asymmetrical cases: there are sterile male. Our results relatively well explain the evolution of animal sex ratio.

## 1 Introduction

In the field of evolutionary ecology, lattice models applying asynchronous processing have growing interest[1,2]. The present paper focus on the optimality of sex ratio in animals. In most cases the optimal sex ratio at birth has been obtained by evolutionarily stable strategy (ESS)[3,4,5,6]. However, ESS neglects the sustainability of population. Recently, our coauthors have presented a lattice model of mating population to obtain the optimality not by ESS but by the sustainability [7]. They mainly dealt with the symmetrical case between male and female, and explained the reason why the sex ratio of many animal is 1/2. In the present paper, we extend the model to deal with an asymmetrical property; for example, the system contains sterile males. We discuss the effect of sterile male on the optimality of sex ratio.

A typical example of sterile male is observed in some animals that form harems. When few strong males of harems monopolize almost females, then a majority of males cannot reproduce offspring. If sex ratio at birth is 1/2, then actual sex ratio in the system is much less than 1/2 (female biased). Nevertheless, real harem animals usually have 1:1 sex ratio at birth even in such an asymmetrical case.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 368–373, 2008.

## 2 Model and Method

### 2.1 Model

Consider a mating population on a two-dimensional lattice. Our model is an extension of the contact process. The population on a square lattice is composed from usual (reproductive) male ($X$), sterile male ($Y$) and female ($Z$). Each lattice site takes one of four states: $X, Y, Z$ or empty ($O$). The sterile male cannot participate to actual mating. Birth and death processes update the lattice. The death process is defined by

$$j \longrightarrow O \quad (\text{rate: } m_j) \tag{1a}$$

where $X, Y, Z$, and $m_j$ is the mortality rate of $j$. On the other hand, birth process is expressed by

$$O \longrightarrow X \quad (\text{rate: } B\alpha\beta), \tag{1b}$$

$$O \longrightarrow Y \quad (\text{rate: } B\alpha(1-\beta)), \tag{1c}$$

$$O \longrightarrow Z \quad (\text{rate: } B(1-\alpha)), \tag{1d}$$

where the parameter $B$ is the birth rate. The parameter $\alpha$ denotes the sex ratio (male ratio per total population), and $\beta$ is the frequency of reproductive males per total males. The birth rate $B$ depends on simulation method; it takes different values between local and global interactions.

### 2.2 Simulation Methods

Simulations are carried out by two methods; local and global interactions. First, we explain the method for local interaction.

1) Initially, we distribute $X, Y$ and $Z$, where each lattice point is either $X, Y, Z$ or $O$.

2) To update, we choose a target site randomly.

i) If the site is occupied by $j$ ($j = X, Y, Z$), then the death process (1a) is performed. Namely, the site $j$ is changed into $O$ with the rate $m_j$.

ii) If the site is empty ($O$), we perform the birth process (1b), (1c) or (1d). The site $O$ becomes $X, Y$ or $Z$ by the rate $B\alpha\beta$, $B\alpha(1-\beta)$ or $B(1-\alpha)$, respectively. Here birth rate $B$ is proportional to both local densities of $X$ and $Z$, but it is independent on the density of $Y$, since the sterile male never participates in the mating.

We repeat step 2) by $L^2$ times, where $L^2$ is the total number of lattice sites. In this paper, we put $L = 200$. This step is called a Monte Carlo step . We further continue the updates, until the system reaches stationary state.

Next, we explain the method for global interaction in which long-ranged interactions are allowed. The global simulation is very similar to the local simulation, but the birth rate $B$ is proportional to both overall densities of $X$ and

$Z$: $B = rxz$. Here $r$ is a positive constant (reproduction rate), and $x$ and $z$ are overall densities of $X$ and $Z$, respectively.

## 3 Mean-Field Theory

In the limiting case that the lattice size is infinite, the population dynamics for global interaction can be represented by the mean-field theory:

$$\frac{dx}{dt} = -m_X x + rxz(1-x-y-z)\alpha\beta \tag{2a}$$

$$\frac{dy}{dt} = -m_Y y + rxz(1-x-y-z)\alpha(1-\beta) \tag{2b}$$

$$\frac{dz}{dt} = -m_Z z + rxz(1-x-y-z)(1-\alpha) \tag{2c}$$

where $x, y$ and $z$ are the densities of male, sterile male and female, respectively, and the factor $(1-x-y-z)$ means the density of empty site. The first (second) term in the right hand side of equation (2) represents the death (birth) process. If $\beta = 1$, equation (2) becomes the system introduced by Tainaka et al [7]. However, equation (2) has the merit that the final density in stationary state can be analytically obtained. In the present paper, we consider the case of equal mortality rates: $m_X = m_Y = m_Z = m$. We can find that equation (2) is roughly classified into fast and slow dynamics.

Hence the mean-field theory predicts three densities $(x, y, z)$ at final survival state as follows ($D > 0$):

$$x = \alpha\beta(1+\sqrt{D})/2 \tag{3a}$$

$$y = \alpha(1-\beta)(1+\sqrt{D})/2 \tag{3b}$$

$$z = (1-\alpha)(1+\sqrt{D})/2 \tag{3c}$$

The total population size $(x+y+z)$ at final equilibrium is thus obtained as follows:

$$x + y + z = (1+\sqrt{D})/2, \tag{4}$$

$$\text{where} \quad D = 1 - \frac{4m}{r\beta}\left(\frac{1}{1-\alpha} + \frac{1}{\alpha}\right).$$

Equation (4) predicts that the optimal sex ratio is 1/2.

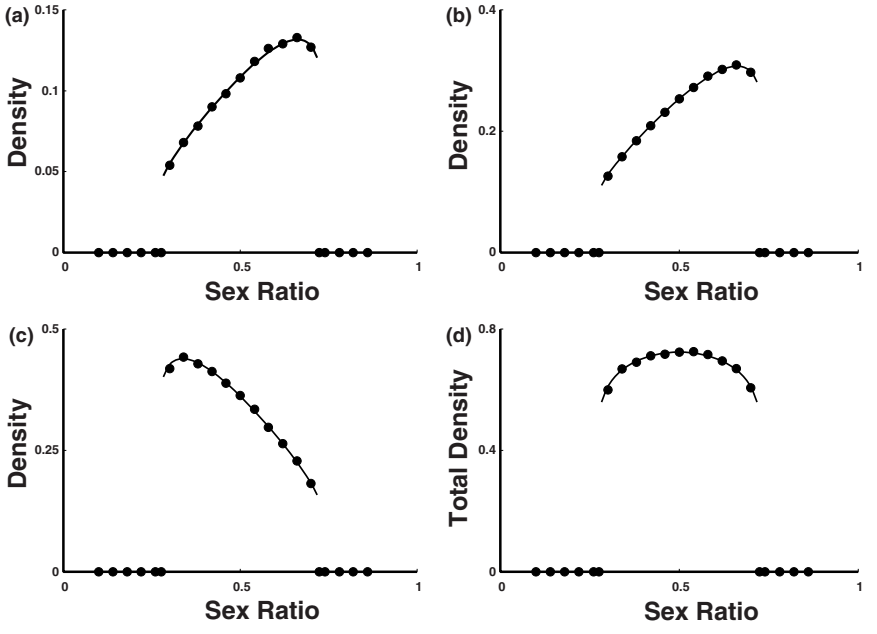



**Fig. 1.** Comparison between global simulation and mean-field theory. Densities of (a) $x$, (b) $y$, (c) $z$, and (d) $x + y + z$ are displayed against the sex ratio. Plots of circles in each figure represent the simulation results of global interaction, while the curves are the predictions of mean-field theory.

## 4 Simulation Results

### 4.1 Results of Global Simulation

In Fig. 1, three densities $(x, y, z)$ and total population size $(x + y + z)$ at final equilibrium are displayed against the sex ratio $\alpha$. In the simulation, each plot is obtained by averaging densities over the period $2000 < t \leq 3000$. The agreement between global simulation (plots) and theory [curve: equations (3)s and (4)] is found to be complete. We find from Fig. 1 that the optimal sex ratio is $\alpha = 1/2$ as predicted by the theory. The survival density is stable, when $D > 0$ . Moreover, sustainable range of sex ratio is not so wide.

### 4.2 Results of Local Simulation

Results for local model entirely differs from global case. In Fig. 2, densities $x, y, z$ and $(x + y + z)$ at final equilibrium are displayed against the sex ratio $(m_X = m_Y = m_Z = m = 0.0010)$. Each plot is obtained by averaging over the period $48000 < t \leq 50000$, since the dynamics of local interaction is slower than that of global interaction. We find from the last figure in Fig. 2 that the optimal sex ratio is $\alpha \geq 1/2$ . The sustainable range of sex ratio is very narrow, compared to the global interaction (Fig. 1).

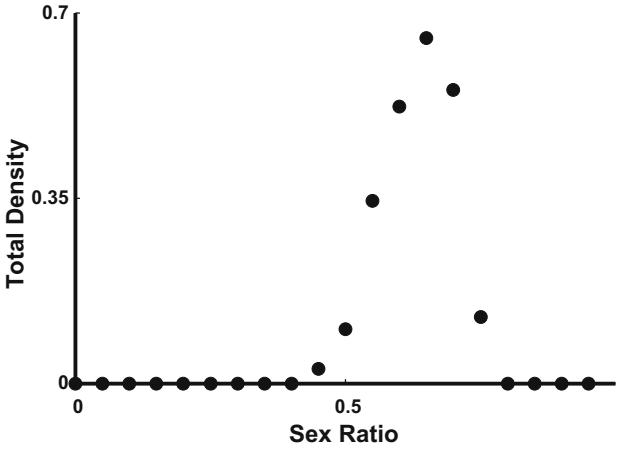



**Fig. 2.** Simulation result of local interaction. The total density $(x + y + z)$ is plotted against the male ratio.

## 5 Conclusion and Discussions

Monte Carlo simulations are carried out in the mating system that contains sterile male $(Y)$. In the case of global simulations, the optimal sex ratio is given by 1/2 (see Fig. 1). This result well explains real sex ratio for harem animals.

In contrast, the local simulations indicate that the optimal sex ratio is male biased $(\alpha \geq 1/2)$. This seems contradicts against actual data. However, the contradiction can be solved by the existence of sterile females. In real animals, females also have sterile properties: Pregnant and immature females effectively denote sterile females. Moreover, females have a long pregnant period; they need a longer matured period than males, because eggs are energetically more costly than sperms [8,9,10]. Both male and female more or less have sterile properties. Probably, these properties compensate, so that almost all animals have the sex ratio of 1/2.

## References

1. Tainaka, K.: Lattice model for the Lotka-Volterra system. J. Phys. Soc. Jpn. 57, 2588–2590 (1988)
2. Nagata, H., Morita, S., Yoshimura, J., Nitta, T., Tainaka, K.: Perturbation experiments and fluctuation enhancement in finite size of lattice ecosystems: Uncertainty in top-predator conservation. Ecological Informatics (in press)
3. Fisher, R.A.: The Genetical Theory of Natural Selection. Oxford University Press, Oxford (1930)
4. Charnov, E.L.: The Theory of Sex Allocation. Princeton University Press, Princeton (1982)
5. Hardy, I.C.W.: Sex Ratios: Concepts and Research Methods. Cambridge University Press, Cambridge (2002)

6. Hamilton, W.D.: Extraordinary sex ratios. Science 156, 477–488 (1967)
7. Tainaka, K., Hayashi, T., Yoshimura, J.: Sustainable sex ratio in lattice populations. Europhysics Letters 74, 554–559 (2006)
8. Sakisaka, Y., Yahara, T., Miura, I., Kasuya, E.: Maternal control of sex ratio in Rana rugosa: evidence from DNA sexing. Molecular Ecology 9, 1711–1715 (2000)
9. Togashi, T., Bartelt, J.L., Cox, P.A.: Simulation of gamete behaviors and the evolution of anisogamy: reproductive strategies of marine green algae. Ecological Research 19, 563–569 (2004)
10. Togashi, T., Nagisa, M., Miyazaki, T., Yoshimura, J., Bartelt, J.L., Cox, P.A.: Gamete behaviors and the evolution of "marked anisogamy": reproductive strategies and sexual dimorphism in Bryopsidales marine green algae. Evolutionary Ecology Research 8, 617–628 (2006)

# Real Option Approach to Quoting Queueing System

K. Ohstuka[1,2] and K. Nishinari[1,2]

[1] Dep. Aeronautics and Astronautics Engineering, University of Tokyo, Japan
[2] PRESTO, Japan Science and Technology Corporation, Japan
tukacyf@mail.ecc.u-tokyo.ac.jp

**Abstract.** The new method of estimating net present values of a queue is proposed. We complementary employ real option theory and the asymmetric exclusion process (ASEP) as the valuating procedure. By pricing the dynamics of the ASEP, we succeed in identifying an ambiguous word, human congestion, quantitatively. An analogy between a clogged queue and a cash flow of an active firm is formulated.

## 1 Introduction

In the area of mathematical science, there have been increased interest in collective dynamics of self-driven particles (SDP) because they can apply to practical problems such as traffic flow, congestion in metropolitan areas and disaster evacuation procedures. This is a novel research field so-called "Jammology". Although a number of studies are carried out on the dynamics of the SDP, little attention has been paid to estimating worth of their dynamics itself. We introduce, in this paper, the method that identifies net present values of a queue by using a real option approach. The queueing model which we employ here is the one dimensional ASEP with open boundaries. This model shows various phenomena even though their interaction rules are simple. In particular, the ASEP forms a shock front (domain wall) under restricted boundary conditions. Motion of the domain wall follows a biased random walk [2]. Hence we apply the concepts of a real option approach to estimate length change of a queue. Our approach make it possible to give a quantitative worth of present congestions.

This paper organized as follows: In Section 2, the models which we will consider are determined. Then, in Section 3, we devoted to show both the estimating method of human queue and the analytical solution of real option approach. In Section 4, nonlinear property of real options are examined. Finally, Section 5, conclusions are mentioned.

## 2 Analogy between the ASEP and Real Option Concept

### 2.1 The ASEP and Domain Wall Dynamics

We consider, in this article, the evaluation of congestion. For simplicity, we employ the one-dimensional ASEP as the model of a queuing system and apply this

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 374–378, 2008.

one to pricing of a queue. At the boundaries of this system, particles are supplied at the left end with $\alpha$, and removed at the right end with $\beta$. In the bulk particles can jump to the neighbouring right-hand site with constant rate 1 provided that this site is empty (had-core exclusion). The ASEP exhibits nonequilibrium phase transitions depending both on $\alpha$ and $\beta$. The steady states in this system have been determined exactly[1]. As shown in Fig. 1, the phase diagram shows the three steady regions: high density, low density and maximal current phases. Except in a maximal current phase, there exists an interesting phenomenon, the domain wall[2,3] . Motion of the domain wall to the left at rate $\alpha$ or to the right at rate $\beta$, so that the domain wall does a biased random walk with drift velocity $V = \beta - \alpha$ and diffusion coefficient $D = (\alpha + \beta)/2$. Positive (negative) values of velocity are defined that the domain wall moves to the left (right) boundary.

We should identify, as a next step, situations which are needs to judge by our new method using the real option approach. When both $\alpha$ and $\beta$ are smaller than $1/2$, this phase is neutral situation. Especially in stochastic system, our intuition cannot regards whether the system is close to congestion or not. That is why this condition is our target to be evaluated. We want to exploit underlying signals of a clogging risk.

In the financial engineering problems, a price of stocks or of exchange rates are assumed following an geometric Brownian motion [4,5,6] bacause of interesting their instantaneous percentage changes. Then their derivatives are verified under no-arbitrage opportunity. Here we focus on the domain wall dynamics always governed by an biased random walk except for in the maximal current region. We will obtain net present values of a queue by estimating this dynamics.

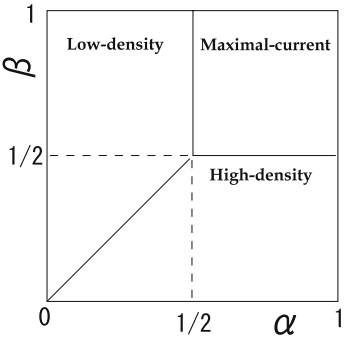



**Fig. 1.** Phase diagram of the ASEP. The solid lines separate the three steady states.

### 2.2 Valuing a Queue

Assume that the stochastic process $X(t)$, the length of a queue, follows an arithmetic Brownian motion:

$$dX = \mu dt + \sigma dz, \tag{1}$$

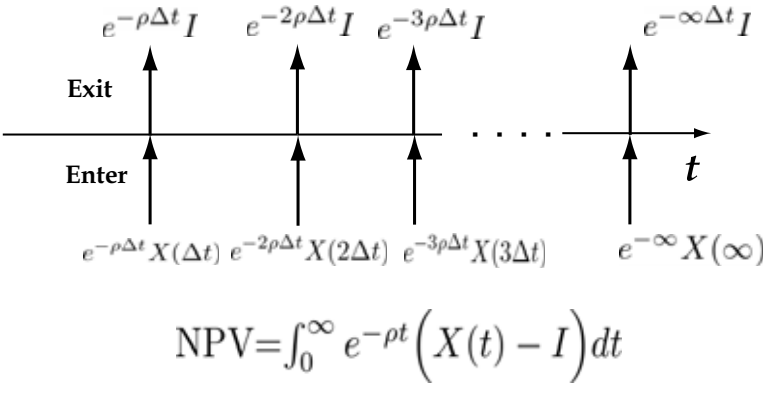



**Fig. 2.** Clogging degree of the 1D ASEP

where $\mu$ is the drift of the domain wall per unit time, $\sigma$ is the instantaneous standard deviation per unit time, and $dz$ is the increment of a standard Wiener process. In the domain wall framework, $\mu$ and $\sigma$ corresponds to $V = \beta - \alpha$ and $D = (\alpha+\beta)/2$ respectively. It means that values of a queue tail primarily depend both on entering and exit rates at boundaries. Note that our main interest is on changes of a queue length itsself, not on percentage changes (not on an geometric Brownian motion).

We define degree of human clogging as $\pi = X(t)-I$. The value $\pi$ is a indicator which shows congestion (free flow) when its values are positive (negative). The variable $I$ fixed at 0.5 which means a queue length equal to half of the system size. In other word, we defined a queue is crowded when $\pi$ is positive. As the phase diagram shows, in high-density region, $X(t)$ always rises above $I$. If in low-density region, $X(t)$ falls or stays below $I$, and we have no-risk to pile up. Therefore, instant clogging from a queue, which we consider, is given by

$$\pi(X) = \max[X - I, 0]. \tag{2}$$

The value of a queue at time $t$ can be expressed as the sum of the instantaneous clogging over the interval ($t$, $dt$) and the continuation value beyond $t + dt$. The value of queue is

$$J(x) = \pi(X)dt + e^{-\rho dt}E[J(X + dX)]. \tag{3}$$

To express weight of time, $J$ is discounted at the rate $\rho$ (see Fig. 2). Expanding the right-hand side using Ito's Lemma, we have

$$J(x) = \pi(X)dt + (1 - \rho dt + o(dt))[J(x) + J'(x)\mu dt + \frac{\sigma^2}{2}J''(x)dt + o(dt)], \tag{4}$$

where $o(dt)$ collects terms that go to zero faster than $dt$. Simplifying, dividing by $dt$, and $dt \to 0$, we get the differential equation

$$\frac{1}{2}\sigma J(X)'' + \mu J(X)' - \rho J(X) + \pi(X) = 0. \tag{5}$$

The solution of Eq. (5) can be written as

$$J(X) = \begin{cases} \frac{C_3 I}{\rho} \exp\Big( -\frac{\mu}{\sigma^2}(X-I)\Big) \sinh\Big(C_2 X\Big) & X < I \\ \Big[\frac{C_3 I}{\rho} \sinh\Big(C_2 I\Big) - \frac{\mu}{\rho^2}\Big] \exp\Big[ -(C_1 + C_2)(X-I)\Big] + \frac{X-I}{\rho} + \frac{\mu}{\rho^2} & X \geq I \end{cases}$$

where

$$C_1 = \frac{\mu}{\sigma^2}, C_2 = \frac{\sqrt{\mu^2 + 2\rho\sigma^2}}{\sigma^2},$$

$$C_3 = \Big\{ \frac{\sqrt{\mu^2 + 2\rho\sigma^2}}{\sigma^2} \exp\Big( \frac{\sqrt{\mu^2 + 2\rho\sigma^2}}{\sigma^2} I\Big) + \Big( \frac{\sqrt{\mu^2 + 2\rho\sigma^2} + \mu}{\sigma^2}\Big) \frac{\mu}{\rho^2} \exp\Big(\frac{\mu I}{\sigma^2}\Big)\Big\}^{-1}.$$

Note that the term $[(X_0 - I)/\rho] + \mu/\rho$ in the solution is the expected value of stream $\pi$ when initial level is $X_0$. This is the net present value (NPV) of a queue obtained by

$$\text{NPV} = \int_0^\infty e^{-\rho t}\Big(X(t) - I\Big) dt = \frac{X_0 - I}{\rho} + \frac{\mu}{\rho}. \tag{6}$$

## 3 Nonlinear Property

To examine the relationship between $J$ and the parameters, $\alpha$ and $\beta$ dependences of $J$ are plotted in Figs. 3(a) and 3(b). Figures show $J$ does not have linear dependence on both parameters. For example, in Fig. 3(b), $J$ does not monotonically decreases with $\beta$ even if an exit rate of the system are increased. It means that a clogging of human flow (cumulative jamming signals) has intrinsically nonlinear properties.

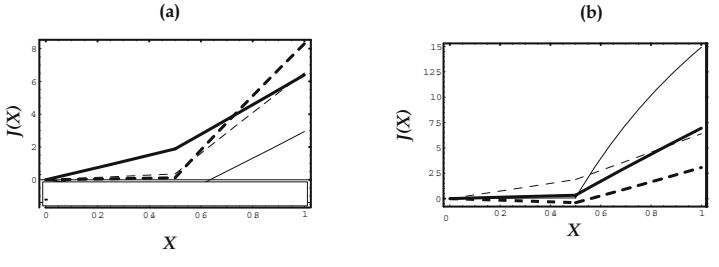



**Fig. 3.** (a)Dependence of $J(X)$ on $\alpha$ when ($\rho$, $\beta$, $I$)=(0.1, 0.25, 0.5). Thin solid line, thick solid line, thin dotted line and thick dotted line correspond to $\alpha = 0.2$, 0.25, 0.3 and 0.4. (b)Dependence of $J(X)$ on $\beta$ when ($\rho$, $\alpha$, $I$)=(0.1, 0.25, 0.5). Thin solid line, thick solid line, thin dotted line and thick dotted line correspond to $\beta = 0.1$, 0.2, 0.25 and 0.35.

## 4 Conclusion

In this paper we formulate the option value that indicates a clogging alert and identify an analogy between the domain wall dynamics of the ASEP and the real option approach. This allow us to use the tools of the option pricing theory to estimate degree of congestion under uncertainty. Several extensions of this paper would prove interesting. However some questions such as

1. What is an actual meaning of discounting human flow ?
2. We should study nonlinear structures of the real option values as shown Fig. 3.

These investigations are left for our future works.

## References

1. Derrida, B., Evans, M.R., Mukamel, D.: Exact diffusion constant for one-dimensional asymmetric exclusion models. J. Phys. A: Math. Gen. 26, 1493–1517 (1993)
2. Kolmeisky, A.B., Schütz, G.M., Kolmeisky, E.B., Straley, J.P.: Phase diagram pf one-dimensional driven lattice gases with open boundaries. J. Phys. A: Math. Gen. 31, 6911–6919 (1998)
3. Rajewsky, N., Santen, L., Schadschneider, A., Schreckenberg, M.: The assymetric exclusion process: Comparision of update procedure. J. Stat. Phys. 92, 151–194 (1998)
4. Hull, J.: Options, Futures and Other Derivatives, 6th edn. Prentice-Hall, Englewood Cliffs (2005)
5. Dixit, A.K., Pindyck, R.S.: Investment under uncertainty. Princeton Univ. Press, Princeton (1994)
6. Øksendal, B.: Stochastic Differential Equations: An Introduction with Applications, 6th edn. Springer, Heidelberg (2003)
7. Harison, J.M.: Brownian Motion and Stochastic flow system. John Wiley & Sons, Chichester (1985)

# Modeling of Environmental Effects on Bridge Components: Possibilities of Cellular Programming

Jan Podroužek, Drahomír Novák, Břetislav Teplý, and Dita Vořechovská

Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic
{podrouzek.j,novak.d,teply.b,vorechovska.d}@fce.vutbr.cz

**Abstract.** The early deterioration of concrete structures due to the effects of external aggressive environment is well known. This paper presents cellular automata approach to the problem of lifetime assessment of concrete structures, particularly bridges, under the diffusive attack of external aggressive agents. The diffusion process is modeled by cellular automata technique as a 2D task describing more realistically the spatial variability of e.g. the chloride ingress within dynamic environment. The effectiveness of the proposed methodology arises from the transparent implementation yet complex behavior of two selected numerical case studies.

**Keywords:** diffusion process, concrete structures, service life, simulation, cellular automata, aggressive environment, bridges.

## 1 Introduction

Concrete is generally effective in protecting embedded steel from corrosion. This protective capability may be reduced significantly due to external stressors affecting the reliability of the structure during the course of time. Current standards require minimum concrete cover thickness, which calculation is, among others, based on the given level of environmental exposure. The amount of concrete used as a protective layer significantly affects the cost, reliability and service life of the structure. Such approach does not directly allow a design focused on a specific service life and/or a specific level of reliability – this would require the inherent uncertainties in relevant characteristics to be dealt with while assessing the service life of a structure. Such tasks necessarily require the utilization of stochastic approaches, analytical models of degradation effects and also simulation techniques, all based on the experimental evidence and relevant observations of structures under real conditions.

This work focuses on nontraditional numerical representation of selected processes, relevant to the assessment of environmental exposure, using cellular automata (CA) technique, a very simple yet effective method, to solve the diffusion equation according to a given set of boundary conditions within dynamic environment. In this way the initiation time (time to reinforcement corrosion onset) is assessed more effectively than by traditional 1D models.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 379–384, 2008.

## 2 Spatial Simulation of Chloride Penetration Based on Cellular Automaton

### 2.1 Field of Application

A bridge construction during its life cycle undergoes a number of events of which few have indispensable effect on the structural health. Some of the most suitable phenomena for cellular automata simulation are presented in this section. Seasonal de-icing salt application may be considered in certain areas noting that the diffusion takes place inside the concrete structure also when there is no external chloride feed, e.g. in summer. During the event of rain the exposed surface regions may have their chloride concentration reduced due to the wash-out effect. Different diffusion components may also be considered, such as carbon dioxide or water. Another example could be the effect of UV rays or temperature. All of the mentioned above would have instant effect on the local material and diffusive properties of the structure. It is clear that such system would be relatively easy to create in terms of cellular programming but would require a large number of field data together with theoretical knowledge of individual phenomena and their interactions. Note that the presented case studies are based on the single-parameter cell system.

### 2.2 Diffusion Equation

The simplest model to describe the process of diffusion of chemical components is based on Fick´s law. For the case of single component diffusion in isotropic, homogeneous and time-invariant media it leads to the second order partial differential equation [1,2]:

$$\frac{\partial C}{\partial t} = D\nabla^2 C \tag{1}$$

where C is concentration of diffusion component (at particular point and in particular time) and D is diffusivity coefficient of the medium. The diffusivity coefficient for concrete depends on many parameters like relative humidity, temperature and others; generally it is time dependant.

The equation (1) can be effectively solved by using cellular automata; the solution was adopted from Biondini et al. [1,2] and further developed as briefly described in the following text.

### 2.3 Cellular Automata

The structure is for our purposes represented by a 2D grid of cells. Each cell has its state value representing the component concentration (e.g. the chloride ions). The process of chloride ingress is governed by a local rule in which the evolutionary coefficients $\Phi_i$ (i = 1,2,3,4,5) assign the level of chloride concentration redistribution within the cell's neighbourhood (von Neumann, radius 1):

$$X_{(t+1)} = \Phi_1 X_t + \Phi_2 N_t + \Phi_3 E_t + \Phi_4 S_t + \Phi_5 W_t \tag{2}$$

where the discrete variables $\{X_t , N_t , E_t , S_t , W_t\}$ represent the concentration of the component in the given cell at time t. The values of the evolutionary coefficients $\Phi_i$ must verify the following normality rule:

$$\sum \Phi_i = 1 \tag{3}$$

as required by the mass conservation law. Clearly, for isotropic media the symmetry condition $\Phi_2 = \Phi_3 = \Phi_4 = \Phi_5$ must be adopted in order to avoid the directionality effects. It can be proven that the values $\Phi_1 = 0.5$ and $\Phi_{2,3,4,5} = 0.125$ lead to a very good accuracy of the automaton. The relationship between the cell size $\Delta x$, time step $\Delta t$, diffusion coefficient D and chloride evolutionary coefficient $\Phi_1$ (governing) is mandatory for the whole grid of cells within a time step:

$$D = \Phi_1 \Delta x^2 \Delta t^{-1} \tag{4}$$

Stochastic effects may be treated as well, modifying the procedure by assuming the evolutionary coefficients $\Phi_i$ to be random values with a given probability density function. Generator of pseudorandom numbers based on the ziggurat method was adopted [3]. One of the essential parts of the CA configuration is the boundary rule setting, or in other words the definition of the system's behaviour in the areas where a contact is with the outer medium, i.e. with a place beyond the simulation region. Several types of boundary rule have been implemented. The best rule suitable for the comparison with conventional 1D analytical models (for chloride ions penetration) is the mirror neighbour rule of hemisphere action, where the state values of the adjacent cells are governed according to the boundary region condition in such a way, that the sum of all state values within a defined cell's neighborhood (von Neumann, radius 1) is equal to the same sum in the successive time step. Note that this approach to the boundary problem is suitable only for the case, where it is supposed that the transported quantity does not leave the simulation region once it enters. This might be the case of chloride ions although it is known, that in reality there is some kind of back-propagation. It is an attractive topic to appropriately define the boundary system interaction for different transport phenomena; however for the purpose of this work it was rational to use some sense of simplification when dealing with the task of boundary interaction.

### 2.4 Local Degradation Modeling and Probabilistic Simulation

In the case of the comparison to the 1D model the aim is to configure such a CA that will provide similar results as the 1D material point model. To achieve this, it was useful to create such a configuration of CA, that the component concentration at certain point at certain time was affected only by the propagation from one direction without considering the gain/loss of the component from the other directions. This demand would satisfy an infinite large simulation area (cross section) or a special modification of the boundary rules, which would simulate the coverage of the infinite simulation area. The analytical 1D model for comparison was adopted from Papadakis using FREET-D tool [4]. The results were almost identical.

# 3 Case Studies

## 3.1 An Application to an Existing Bridge

The bridge was designed in 1969 as a highway crossover in South Tyrol. The inspections showed that there is a serious degradation of the piles and beams in the junctions of the prestressed V beams. These weak zones are mainly caused by an insufficient sealing of the pavement in the region of the expansion joints over the piles. As the bridge was demolished in spring 2008, concrete samples were possible to drill and chemical analysis was performed. Chloride concentrations were measured and localized (based on heuristic expert judgment). Due to the short time period before the paper finalizing, this data were not utilized as input concentration for CA approach to calibrate the present stage of analysis and to make realistic prognosis in time. Figure 1 shows the estimated initial distribution of de-icing salts concentrations (half of the bridge), the maximal surface chloride concentration being 0.2 % Cl- per cement content. Note that distribution of initial concentration along bottom part of the bridge was estimated from the location of stiffeners and supports (leakage through joints), where higher "localized" degradation was detected.

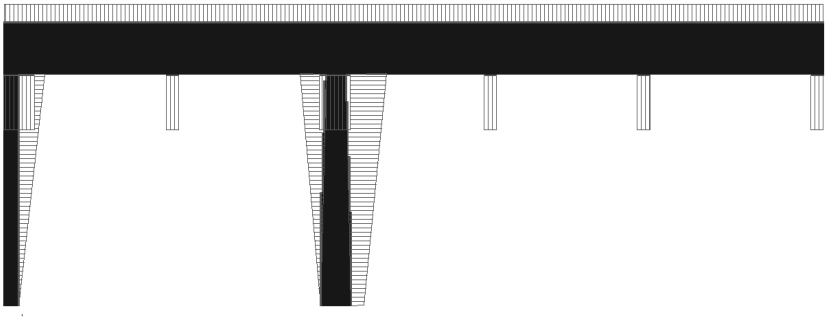


**Fig. 1.** Distribution of initial concentrations of aggressive agents as considered in 2D model

**Fig. 2.** Predictions of chloride concentrations development for 30 years – detail around the supporting pier with critical concentration region (black)

Prediction of chloride concentrations for 30 years is shown in Figure 2. The light tone represents the undamaged state of a cell, while the darker tone represents the degraded state. Black regions represent parts where chloride concentrations were greater than critical chloride concentrations (here 0.4 % Cl- per cement content). The utilization of such approach will enable the engineer to focus on modeling or prognosis of reinforcement corrosion in most jeopardized locations, namely to assess the initiation time, i.e. the durability limit state; and thus support the decision making.

### 3.2 Reinforcement Corrosion

In the presented example a reinforced concrete rectangular cross section is exposed to chloride ingress. To illustrate the versatility of CA four different cases are assumed that differ by the surface areas exposed to chloride action as illustrated in Fig. 3. Surfaces bounded with the dashed line were supplied by chlorides in the way that the surface concentration remained constant over the whole time period except for the case IV where the more realistic seasonal (winter) de-icing salt application was applied. Both reflecting and absorbing boundary conditions (BC) are suitable depending on expected behaviour. For example using the combination of absorbing BC and seasonal chloride feed an interesting task would be to find the equilibrium state when the chloride concentration in the structural core stabilizes on a certain level while the fluctuations remain in the peripheral regions, i.e. after certain time the same amount of chlorides from winter salt application is washed out in the summer having no effect on the total annual $Cl^-$ increment.

Consequently the development of the chloride concentration in the vicinity of reinforcing bars is monitored to demonstrate various performances of different boundary conditions.

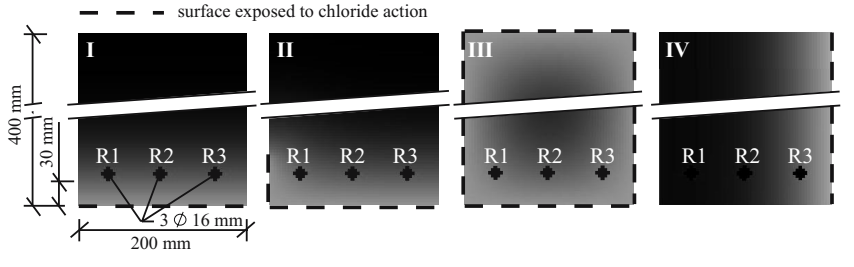



**Fig. 3.** Chloride ingress for different boundary conditions (I, II, III and IV) after 30 years of exposure. Higher concentration values are in light tone, lower are black. Steel reinforcements are R1, R2 and R3.

## 4 Conclusions

The paper exploits possibilities of cellular programming as transparent and general approach to applied degradation modeling of concrete bridges. The authors believe that the approach is promising, especially in the context of lifetime reliability assessment. The South Tyrol Bridge serves as testing application example of the first part of the approach generally applicable for any degrading concrete structure.

**Acknowledgement.** This research was conducted with the financial support of the Ministry of Education of the Czech Republic, project No. MSM0021630519. Authors acknowledge also the grant No.103/07/0760 from the Grant Agency of Czech Republic.

## References

1. Biondini, F., Bontempi, F., Frangopol, D.M., Malerba, P.G.: Cellular Automata Approach to Durability Analysis of Concrete Structures in Agressive Environments. Journal of Structural Engineering ASCE 130(11), 1724–1737 (2004)
2. Biondini, F., Frangopol, D.M., Malerba, P.G.: Time-Variant Structural Performance of the Certosa Cable-stayed Bridge. Strucutral Engineering International, No. 3 (2006)
3. Marsaglia, G., Tsang, W.W.: The Ziggurat Method for Generating Random Variables. Journal of Statistical Software 05(08) (2000)
4. Teplý, B., Chromá, M., Matesová, D., Rovnaník, M.: FREET- D Deterioration Module Program Documentation, BrnoČervenka Consulting (2007), http://www.freet.cz
5. Wolfram, S.: Cellular Automata And Complexity – Collected papers. WolfrAddison-Wesley (1994)
6. Xi, Y., Bažant, Z.P.: Modeling chloride penetration in saturated concrete. Journal of Materials in Civil Engineering 11(1), 58–65 (1999)

# A CA Model of Spontaneous Formation of Concentration Gradients

Roberto Serra[1,2] and Marco Villani[1]

[1] Dipartimento di scienze sociali, cognitive e quantitative Universit di Modena e Reggio Emilia, Via Allegri 9, 42100 Reggio Emilia, Italia
[2] European Centre for Living Technology, Ca' Minich, Venezia, Italia
{rserra,mvillani}@unimore.it

**Abstract.** It is shown that a two-compartment isolated fluid system, where a chemical reaction takes place close to the surfaces of the semipermeable separating membrane, can spontaneously develop a transient concentration difference across the membrane. If the system is open to the flow of chemicals, the difference can persist in the steady state. This allows concentrating chemicals in a single compartment, which may be useful for chemical engineering purposes, and which is particular interesting in the study of the dynamics of vesicles and protocells. The phenomenon is investigated and demonstrated here with a CA model: it is also shown that, in the limiting case of infinitely fast diffusion, the results are coherent with those of a homogeneous model.

**Keywords:** self-organization, entropy, dissipative structure, cellular automaton.

## 1 Introduction

It is generally believed that the approach to thermodynamic equilibrium in isolated systems leads to a monotonic decrease of the inhomogeneities which may be present in the initial conditions of a system, at least when gravitational effects can be neglected. However, it has recently been shown, both by theoretical analysis and experimental test, that a transient onset or increase of concentration differences can be achieved in a two-compartment system when the pores of the separating membrane are asymmetric [1].

We will describe below a different kind of two-compartment isolated system where concentration differences may transiently increase in time without violating the second law of thermodynamics. Moreover, when this system is allowed to exchange matter with the environment, some chemicals are concentrated in the smaller compartment. The effect is related to the presence of chemical reactions which can take place at a significant rate only in a region very close to the membrane surface [2] and to the different permeabilites of the various chemicals involved, as it will be described below.

Applications of this effect to chemical reactors are quite straightforward, but we will briefly comment here on a possible non obvious application to the problem

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 385–392, 2008.

of the origin of life and to the development of a new technology. Vesicles in water are composed by a lipid bilayer which surrounds an aqueous interior: they display interesting physical phenomena [3,4] and, since they can spontaneously fission, they are believed to be important for the study of the origin of life as well as for possible applications of protocells [3,5,6,7].

It would be important to be able to concentrate chemicals in vesicles with respect to the external phase: indeed, the presence of a concentration difference across the membrane corresponds to a high energy state, which may be coupled to other chemical reactions (like it happens e.g. in photosynthesis [8]). If the concentration of some chemicals is higher within the vesicle, then one may achieve very effective chemical processing (synthesis, degradation, removal). Moreover, a more speculative possibility is that the high inner concentration of chemicals might allow the formation of large autocatalytic reaction sets. There are indeed theoretical arguments in favour of the spontaneous development of such cycles [9], but they are very difficult to achieve experimentally: a possible explanation for this difference between theory and experiments is that the concentrations of some key components in the bulk are too low, but this problem might be mitigated in a vesicle using the approach introduced in this work.

In order to describe the phenomenon of interest, let us consider a system composed by two compartments with different volumes separated by a semipermeable membrane (Fig. 1). There are chemicals on both sides, and the permeabilities of the membrane to the various chemicals may be very different. We will suppose that the volumes of the two compartments do not change, and that chemical reactions may take place in both compartments, in a region very close to the membrane surface: for example, the reactions may be catalyzed by some molecules which are bound to the membrane [2]. In section 2 we present a specific CA model of a simple system of this kind, which will be used to investigate its possible behaviours. However, as it will be apparent from the analysis, the phenomenon described below is robust and largely independent from the details of the model. The main results of the analysis are the following.

Let us consider first a closed system and an initial state with equal densities of each chemical on both sides of the membrane. If the densities are exactly those corresponding to the values at chemical equilibrium, they of course remain unchanged. But if they differ from those values (while being equal for each chemical on both sides) and if the volumes of the two compartments are different, then an interesting phenomenon takes place, i.e. a transient density difference is established across the membrane. According to the second law, this difference eventually vanishes but, depending upon the values of the parameters, it may last for a relatively long time.

Moreover, if the larger compartment is not isolated, but there is a mass flow, then the density difference does not vanish but it persists in the steady state. The density of non-permeating chemicals is higher within the smaller compartment, therefore providing a mean to steadily concentrate them, which, as observed above, may lead to relevant applications.

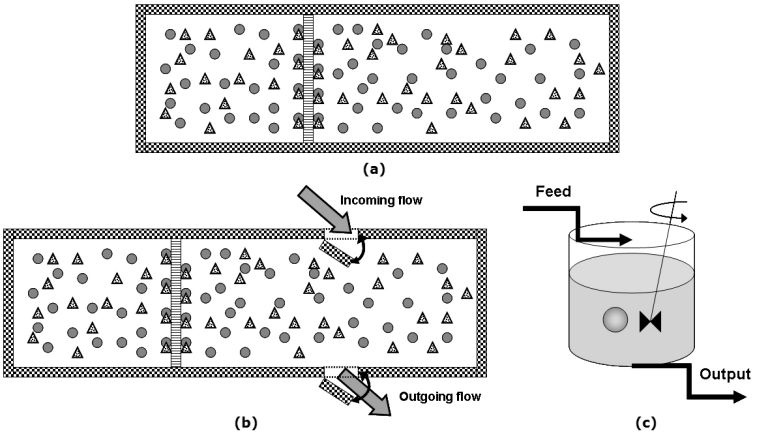



**Fig. 1.** (a) A two-compartment isolated system, with two different chemicals interacting near the semipermeable membrane; (b) The case of an open system, where one of the chemicals enters the reactor and both can leave it; (c) A particularly interesting two-compartment open system: a vesicle inside a continuously stirred tank reactor

In section 2 we introduce the model; in section 3 we present a simplified theoretical treatment of the system, and in the final section 4 we comment on the generality and relevance of the phenomenon.

## 2 The Model and Its Behaviour

In order to study the system described in Fig. 1 let us introduce a 2 dimensional CA model, with a square grid and von Neumann neighbourhood (Fig. 2). The model belongs to the class of macroscopic CA [10] so the state of each cell in each of the two compartments is defined by a vector of real numbers, i.e. the concentrations of the various chemical species. This class of models should not be confused with the more common lattice gases, and it has already proven able to deal effectively with several real world phenomena, including soil bioremediation, landslides and lava flows (for a discussion, see [10]). As it is often the case in macroscopic CAs, also this model is not homogeneous, in that there are some special cells which model the boundaries and the membrane; in the case of an open system, "source" and "sink" cells are used to model drive the flow. The updating is synchronous for all the cells.

Referring to Fig. 2a, the diffusion of chemical $C_j$ among neighbouring cells is modelled as a flow ruled by Fick's law, resulting in the transition rule $\Delta C_0^j = \sum_{i=1}^{4} w_i^j (C_0^j - C_i^j)$, where the weight $w_i^j$ is proportional to the transport coefficient between the two interested cells. In such a way the presence of a membrane among two cells (whose permeability can be different for different chemicals) can be modelled by using lower transport coefficients.

For the sake of simplicity, we consider a simple unimolecular reaction $A \longleftrightarrow X$ which takes place only in the cells which are neighbours of the separating

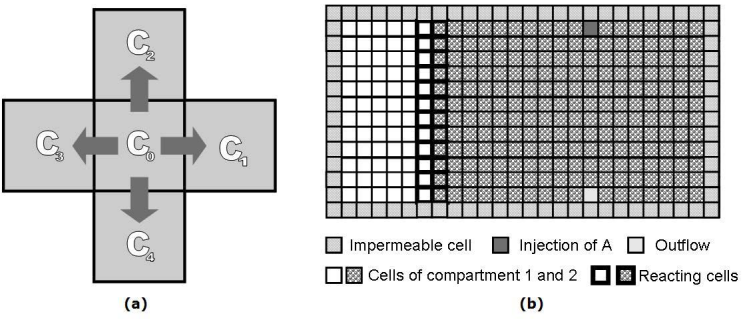



**Fig. 2.** Schematic representation of the CA model of the two-compartment system

membrane (on both sides, represented with thick edges in Fig. 2b). The reaction is supposed to be reversible, the equilibrium being shifted in the region which favours the formation of X, which will therefore sometimes be called the product (and A the reactant). The corresponding transition rules are therefore $\Delta A_0 = -p_f A_0 + p_r X_0$ and $\Delta X_0 = p_f A_0 - p_r X_0$, where $p_f$ and $p_r$ are proportional to the reaction kinetic constants. Oher more complicated reaction schemes have been investigated, leading however to similar phenomena, so we will limit here to describe the simplest case. The system boundaries are modelled as not

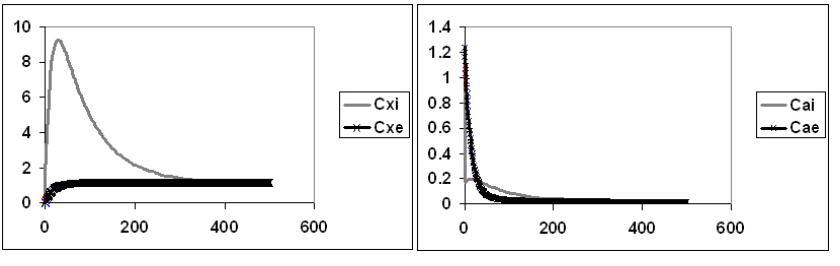



**Fig. 3.** CA model of an isolated system. Average concentration vs. time of X (left) and A (right) in the smaller ( thin line) and larger (thick line) compartment.

permeable walls ($w_i^j = 0$). When A reaches the semipermeable membrane which divides the two compartments, it can cross it while X cannot cross the membrane.

We consider two cases, namely that of a closed system and that of an open system. In this latter case, a source and sink cell are introduced: in the source cell the value of the concentration of reactant A is kept fixed (simulating inflow), while in the sink the value of both A and X are instantaneously set to 0 (simulating outflow).

The behaviour of the isolated system is simulated starting from a condition where the concentrations of both A and X are equal on both sides. If they are equal to each other, but different from their equilibrium values, a transient concentration gradient sets in (see Fig. 3). Of course, in agreement with the

second law, the difference eventually dies out, but this may require a long lasting transient. On the other hand, if the system is open the concentration difference can be steady, as shown in Fig. 4.

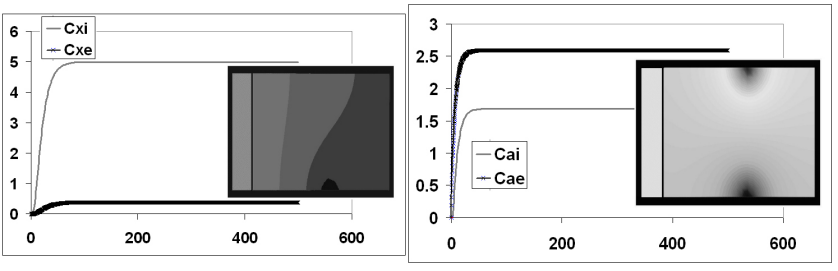



**Fig. 4.** CA model of an open system. Average concentration vs. time of X (left) and A (right) in the smaller (thin line) and larger (thick line) compartment. Also the instantaneous values of the concentration are shown.

## 3 The Limiting Case of Two Homogeneous Compartments

In order to interpret the results of the above simulations we will compare them with those of a simpler model of the same process, where we suppose that the concentrations are homogeneous in each compartment (this corresponds to assume that diffusion in the bulk is very fast).

Let us suppose that a chemical reaction takes place on both sides of the membrane surface, both inside and outside: therefore, all the reactions take place in a small "effective volume" $V_r \approx S\delta$ near the surface (S being its area). The direct and reverse kinetic constants are such that formation of X is favoured. Reactant A can cross the membrane, $\varphi$ being its flow rate and D its diffusion coefficient across the membrane, while X cannot (it would be easy to introduce a small diffusion coefficient of X as well). In order to ease the exposition we will often refer to the smaller and larger volumes as the internal and external one respectively (this is reminiscent of the vesicle case but, at an abstract level, "internal" just means "smaller").

Let $\rho$ and Q denote respectively the densities (moles/litre) and the quantities (moles) of A and X, using a superscript to indicate the chemical species and a subscript the compartment ("i" denoting the smaller one, with volume V', and "e" the larger one, with volume V), so for example $Q_i^A$ is the number of moles of A in the smaller compartment. Diffusion time both in V and V' is supposed to be negligible, so the densities simply equal the ratios of the quantities to the corresponding volumes.

As in section 2, we may let an inflow of A (not of X) in the external compartment: if F denotes a constant volume flow, the quantity of A which enters the external compartment is $F\rho_{ext}^A$, while the quantities of A and X which leave it are respectively $F\rho_e^A$ and $F\rho_e^X$. If the system is closed of course F=0.

Using the above notations, the flow of A across the separating membrane is:

$$\varphi = DS(\rho_i^A - \rho_e^A) \tag{1}$$

where D is the diffusion coefficient of chemical A across the membrane.

Assuming that the kinetics of the reaction is first order, by imposing mass conservation one obtains the following rate equations for the internal and external quantities of A and X:

$$\begin{cases} \frac{dQ_e^A}{dt} = -(kV_r + F)\rho_e^A + k'V_r\rho_e^X + \varphi + F\rho_{ext}^A \\ \frac{dQ_e^X}{dt} = kV_r\rho_e^A - (k'V_r + F)\rho_e^X \\ \frac{dQ_i^A}{dt} = -kV_r\rho_i^A + k'V_r\rho_i^X - \varphi \\ \frac{dQ_i^X}{dt} = kV_r\rho_i^A - k'V_r\rho_i^X \end{cases} \tag{2}$$

In the case of a closed system (F=0) one can compute the equilibrium values and, if the initial conditions coincide with those values, no change is of course observed. Suppose now that the initial densities of A and X are the same in the two compartments but that they do not coincide with their equilibrium values.

In this case one observes (Fig.5) an increase in the inner density of X which may be, depending upon the parameters, very high, and which may last long (although, according to the second law of thermodynamics, it asymptotically vanishes). Note that this behaviour is similar to the one shown in Fig.3 of section 2.

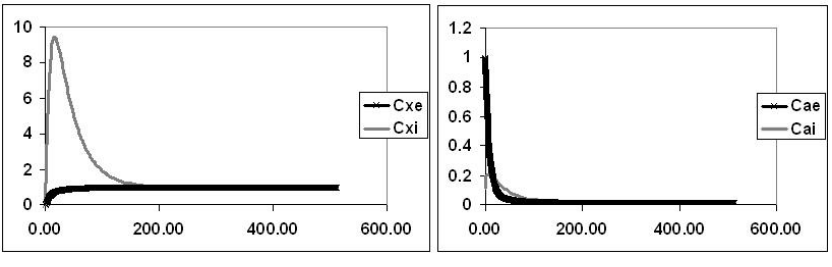



**Fig. 5.** Numerical integration of equation (2) (isolated system); average concentration of X (left) and A (right) in the smaller (thin line) and larger (thick line) compartment

Simulations show that the value of the ratio $\frac{\rho_i^X}{\rho_e^X}$ at the peak of $\rho_i^X$ is inversely proportional to $V'/V$: if the "small" compartment were a vesicle, this would allow a possible independent estimate of its volume. If we assume spherical vesicles also the surface S is thus determined. It has also been observed in simulations that the duration of the transient (defined e.g. as the time elapsed from the beginning of the simulation till the moment when the two curves showing the values of $\rho_i^X$ and $\rho_e^X$ vs. time embrace 95% of the total area between them) depends upon the value of DS. This remark, combined with the previous one, provides a way to estimate the value of the diffusion coefficient D.

Let us now consider the case of a non isolated system, where $F \neq 0$. In this case it can be analytically proven that at steady state the internal concentration of X is larger than the external one, and in particular that:

$$\begin{gathered} \bar{\rho}_i^A = \bar{\rho}_e^A \\ \bar{\rho}_e^X = \frac{kV_r}{(k'V_r+F)}\bar{\rho}_e^A < \bar{\rho}_i^X = \frac{k}{k'}\bar{\rho}_e^A \end{gathered} \tag{3}$$

where a bar denotes asymptotic (steady-state) values. This estimate is in agreement with the results of the CA simulation.

## 4 Conclusions

The fact that a membrane is permeable to one chemical but not to the other might be due to different physical or chemical properties. Since it might be questionable to consider only a unimolecular reaction, we have also analyzed (in the limiting case of fast diffusion) the case of a bimolecular reaction $A+A \longleftrightarrow X$ [11]. The behaviours observed in this case are closely similar to those of the previous one, described in Sections 2 and 3.

We have also analyzed in a similar way the case of a reaction $A + B \longleftrightarrow X + Y$, considering different hypotheses about the permeability of reactants and products, again confirming the concentration effects observed in the simpler model. It is worth to note that the concentration effect at steady state ($F \neq 0$) of non-permeable products may be indeed very large.

A perhaps surprising remark is that the model described here is formally linear: therefore the onset of a concentration difference cannot be considered a nonlinear effect (although it is observed also when nonlinear kinetic equations are considered).One might also observe that the inhomogeneity of the system provides itself a kind of nonlinearity, even if this is not apparent from the transition functions of section 2 nor from the equations of section 3.

From a methodological viewpoint it is worth emphasizing that the comparison of the results of the CA model with the simpler mathematical model of section 3 provides interesting cues to the interpretation of the results of the former, as well as indications concerning the robustness of the major results with respect to changes in the modelling level and approach.

While the phenomenon has been discussed here in a model, it is likely to be real and experimentally verifiable. Indeed, the mechanism which leads to the breaking of the initial equality of internal and external concentrations is that, since reactions take place on the vesicle surface, the same quantities of chemicals per unit time react on both sides of the membrane. But the internal and external volumes are different, and therefore the internal and external concentrations become different. This is essentially independent from the details of the specific kinetic model used.

In a multiphase system, increased concentration of a chemical in a particular phase according to its partition coefficient is of course a well-known phenomenon, but here it arises between two phases with the same physico-chemical properties. It should also be noticed that it is not due to active transport. Therefore the

appearance of concentrations gradients seems surprising, although, as we have seen, in a closed system it is a transient phenomenon which does not violate the second law. In these systems concentration gradients may be coupled to chemical reactions and therefore lead to (transient) interesting effects, while in a flow reactor the gradients are stable and therefore they can provide an energy source driving chemical reactions which might have been otherwise impossible.

## Acknowledgments

Partial support from the EU FET-PACE (Programmable Artificial Cell Evolution) project within the 6th Framework Program and from the Fondazione di Venezia is gratefully acknowledged. The authors also benefited from helpful discussion with Irene Poli, Norman Packard, Timoteo Carletti, Rob Shaw and Ruedi Fuechslin. We recently became aware that John McCaskill observed similar behaviours in a DPD model of vesicle dynamics (unpublished communication).

## References

1. Shaw, R.S., Packard, N., Schroeter, M., Swinney, H.L.: Geometry-induced asymmetric diffusion. PNAS 104, 9580–9584 (2007)
2. Rasmussen, S., Chen, L., Nilsson, M., Abe, S.: Bridging nonliving and living matter. Artificial Life 9, 269 (2003)
3. Luisi, P.L.: The Emergence of Life. Cambridge University Press, Cambridge (2006)
4. McCaskill, J.S., Packard, N., Rasmussen, S., Bedau, M.A.: Evolutionary self-organization in complex fluids. Philosophical Transactions: Biological Sciences, 2007/05/22 (2007, in press) doi:10.1098/rstb.2007.2069
5. Szostak, D., Bartel, P.B., Luisi, P.L.: Synthesizing life. Nature 409, 387 (2001)
6. Rasmussen, S., et al.: Transitions from non living to living matter. Science 303, 963–965 (2004)
7. Serra, R., Carletti, T., Poli, I.: Synchronization phenomena in surface reaction models of protocells. Artificial Life 13, 1–16 (2007)
8. Mitchell, P.: Coupling of phosphorylation to electron and hydrogen transfer by a chemiosmotic type of mechanism. Nature 191, 144–148 (1961)
9. Kauffman, S.A.: At home in the universe. Oxford University Press, Oxford (1995)
10. Di Gregorio, S., Serra, R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. Future Generation Computer Systems 16, 259–271 (1999)
11. Serra, R., Villani, M.: Spontaneous formation of density gradients., European Centre for Living Technology working paper (2008)

# Applying a Cellular Automata Method for the Study of Transport and Deposition of Volcanic Particles

Kae Tsunematsu[1], Jean-Luc Falcone[2], Costanza Bonadonna[1], and Bastien Chopard[2]

[1] Section de Sciences de la Terra, University of Geneva, Switzerland
[2] Computer Science Departement, University of Geneva, Switzerland

**Abstract.** Forecast of volcanic particle dispersal is important for studies of risk assessment. We are interested in simulating a full tephra transport system taking into account turbulent flows and particle aggregation using cellular automata (CA). In this preliminary research, we apply a probabilistic transport CA to our problem. Results show good agreement with field data, indicating that CA are adequate to simulate tephra transport.

## 1 Introduction

Volcanic eruptions are one of the most dramatic natural phenomena and can significantly affect the life of people living close to active volcanoes. When a volcano erupts explosively, fragments of magma are emitted from the crater within a mixture of gas. Fragments of magma that travel through the atmosphere and eventually deposit on the ground are called *tephra*, irrespective to size, shape and composition. Numerical models are important for forecasting tephra transport and deposition in order to compile probabilistic hazard assessments.

Tephra and gas form an eruption column above the crater, called *plume*. Volcanic plumes can be described as multiphase (gas and solid particles) turbulent flows [1], whose physics have been discussed in [2]. A well-developed eruption column typically consists of a gas thrust region, a convective region and an umbrella region (Fig. 1). The injection momentum dominates the gas thrust region, thus, the eruption column rises against gravity. The convective region is where buoyancy forces dominates.

The uppermost region (i.e. umbrella cloud) is characterized by lateral intrusion into the atmosphere. At the level of neutral buoyancy, the ascending plume has the same density as the surrounding atmosphere ($H_b$). Tephra particles are transported first in the eruption column, some of them falling during the ascent when terminal velocity exceeds rising speed of the column. Particles which can reach the top of the plume are carried in the spreading current and will fall out at the base of the umbrella cloud ($H_{cb}$)(Fig. 1). Tephra is transported in the atmosphere by wind and finally deposit on the ground. Observed tephra deposits show some diffusion in grain size distribution [1,3,4].

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 393–400, 2008.

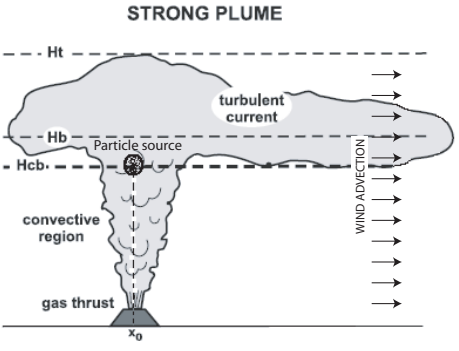



**Fig. 1.** Sketch of plume features. The eruption column is classified into gas thrust region, convective region and umbrella region, from bottom to top. $H_t$ is the total height of the eruption column. $H_b$ is the neutral buoyancy level. $H_{cb}$ is the base of spreading current. At time $t = 0$, we disperse all particles from the point $(x_0, H_{cb})$. Relatively large particles fall out from the plume during the ascent.

Generally, the thickness of tephra deposits decreases with the distance from the eruptive crater [5,6]. However, sometimes, anomalous thickening occurs: double maxima of tephra thickness appear at certain distances from the crater. Observations of anomalous thickening suggest that this phenomenon is related to fine particle aggregation [6], which is believed to happen within any eruption plumes characterized by fine particles. As a result, due to their larger size, aggregates will fall faster than the finer particles they are made of.

Study of tephra fallout progressed in the last two decades and many models have been developed showing agreement with field data [4,5,6,7,8,9]. However, there are still fundamental processes that need to be parameterized and numerically described in order to provide comprehensive and reliable forecasting of tephra deposition. Considering the diffusion of tephra particles and microscopic aggregation, it is useful to use a CA method which can implement both microscopic and macroscopic aspects of particle transport.

This paper is a first step in applying CA for tephra transport. We perform numerical simulations in the atmosphere, from the height $H_{cb}$ (Fig. 1) down to the ground. We show the good agreement between the CA model and field data in the case of sedimentation with no aggregation (e.g. 1875 eruption of Askja volcano, Iceland). On the other hand, we observe significant discrepancies when aggregation processes play an important role in particle sedimentation (e.g. 18 May 1980 eruption of Mt. St. Helens, USA).

## 2 Tephra Transport Model

Processes of tephra transport can be investigated in two areas: umbrella cloud (turbulent current) and the atmosphere under the umbrella cloud (Fig. 1).

The simplest way to compute tephra transport is ballistic calculation (BA). In BA, particles with the same size and the same density reach the same point on the ground. On the other hand, when some diffusion happens, particles disperse in the atmosphere and deposit at different places. A multiparticle CA approach [10] can be devised to account for such a dispersion. We evaluate the difference between CA and BA by comparing the result of the two methods. In our numerical experiments, all tephra begin to disperse from the point $(x_0, H_{cb})$ at time $t = 0$, in a two-dimensional domain (Fig. 1).

Tephra is transported horizontally in the atmosphere by the wind which we assume to be of velocity $u_x$ for the whole simulation domain. The vertical velocity $u_y$ is set to the terminal velocity $u_t$ of each particle (i.e. the speed which the particle reaches due to the balance of gravity, buoyancy and friction forces). $u_t$ depends on particle and atmospheric characteristics. We compute the terminal velocity as a function of the particle Reynolds number, as described in [8].

### 2.1 Cellular Automata Tephra Transport Model

To simulate the transport of tephra particles according to the velocity field described above, we use the CA model defined in [10] which was also successfully applied to snow and sediment transport [11,12]. This model is easy to implement and allows for tracking of individual particles. This property is important to implement the aggregation rules. At each lattice site lies a population of particles subject to a velocity vector $\mathbf{u}$. Ideally, the particles would follow this velocity in the classical way $\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{u}\Delta t$, where $\Delta t$ is the CA discrete time step. However, except for velocity fields parallel to lattice directions with an intensity equals to $\frac{\Delta x}{\Delta t}$ (where $\Delta x$ is the distance between two lattice sites), this would lead the particles off-lattice. In order to constrain the particles on the lattice, we use a probabilistic scheme.

We associate to each lattice direction $i$ a probability $p_i$ proportional to the projection of $\mathbf{u}$ onto the velocity vectors $\mathbf{v}_i$ of norm $v = \frac{\Delta x}{\Delta t}$. In a 2D case, the four directions are $i = \pm x$ and $i = \pm y$, as illustrated in Fig. 2 and the $p_i$ are obtained as

$$\begin{aligned} p_x &= \max\left(0, \frac{\mathbf{v}_x \cdot \mathbf{u}}{v^2}\right) \quad p_{-x} = \max\left(0, \frac{-\mathbf{v}_x \cdot \mathbf{u}}{v^2}\right) \\ p_y &= \max\left(0, \frac{\mathbf{v}_y \cdot \mathbf{u}}{v^2}\right) \quad p_{-y} = \max\left(0, \frac{-\mathbf{v}_y \cdot \mathbf{u}}{v^2}\right) \end{aligned} \tag{1}$$

To have $p_i \leq 1$, the lattice must be chosen so that $u < v$ at every site.

Note that if $p_x > 0$ then $p_{-x} = 0$ and vice versa. The same holds for $p_y$ and $p_{-y}$. Thus, from now on, we assume that $p_x$ and $p_y$ are positive.

To update the system, two independent random numbers $q_x, q_y \in [0, 1]$ are drawn per particle at each time step. If $q_x < p_x$ then the particle moves along the $\mathbf{v}_x$ direction. Similarly, if $q_y < p_y$, the particle moves along $\mathbf{v}_y$. Thus the particle will move to $\mathbf{r} + (\Delta x, 0)$ with probability $p = q_x(1 - q_y)$, to $\mathbf{r} + (0, \Delta x)$

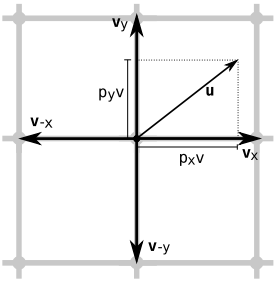



**Fig. 2.** Probabilistic rule of particle transport. The gray lines represents the lattice. The four possible velocity vectors $\mathbf{v}_i$ and the true velocity $\mathbf{u}$ are indicated with black arrows.

with $p = (1 - q_x)q_y$, to $\mathbf{r} + (\Delta x, \Delta x)$ with $p = q_x q_y$ and stay at rest with $p = (1 - q_x)(1 - q_y)$.

It has been shown in [10] that, when the velocity field is constant across the lattice, the average velocity $\langle \mathbf{v} \rangle$ following from the above rules is equal to $\mathbf{u}$. Here we evaluate the dispersion of the trajectory.

For $n$ time steps, the particle trajectory is:

$$\mathbf{r}(n\Delta t) = \mathbf{r}_0 + \mathbf{v}(t_1)\Delta t + \mathbf{v}(t_2)\Delta t + \cdots + \mathbf{v}(t_n)\Delta t = \mathbf{r}_0 + \Delta t \sum_i \mathbf{v}(t_i) \tag{2}$$

where $\mathbf{v}(t_i) \in \{0, \mathbf{v}_{\pm x}, \mathbf{v}_{\pm y}\}$ is the random velocity variable at step $t_i$. On average the above eq. reads

$$\langle \mathbf{r}(n\Delta t) \rangle = \mathbf{r}_0 + \Delta t \sum_i \langle \mathbf{v}(t_i) \rangle = r_0 + n\mathbf{u}\Delta t \tag{3}$$

To compute the standard deviation $\sigma_{\mathbf{r}(n\Delta t)}$, we first compute the average squared trajectory (for a constant $\mathbf{u}$ and for $\mathbf{v}(t_i)$ independent on $\mathbf{v}(t_j)$):

$$\begin{aligned} \langle \mathbf{r}(n\Delta t)^2 \rangle &= \left\langle \mathbf{r}_0^2 + 2\mathbf{r}_0 \Delta t \sum_i \mathbf{v}(t_i) + \Delta t^2 \left( \sum_i \mathbf{v}(t_i) \right)^2 \right\rangle \\ &= \mathbf{r}_0^2 + 2\mathbf{r}_0 n \Delta t \mathbf{u} + n(n-1)\Delta t^2 \mathbf{u}^2 + n\Delta t^2 \langle \mathbf{v}^2 \rangle \end{aligned} \tag{4}$$

Similarly we can compute the squared average trajectory:

$$\langle \mathbf{r}(n\Delta t) \rangle^2 = \mathbf{r}_0^2 + 2\mathbf{r}_0 n \Delta t \mathbf{u} + n^2 \Delta t^2 \mathbf{u}^2 \tag{5}$$

and we get the standard deviation as

$$\sigma^2_{\mathbf{r}(n\Delta t)} = \left[ \langle \mathbf{r}(n\Delta t)^2 \rangle - \langle \mathbf{r}(n\Delta t) \rangle^2 \right] = -n\Delta t^2 \mathbf{u}^2 + n\Delta t^2 \langle \mathbf{v}^2 \rangle \tag{6}$$

Let us define the dimensionless velocity $\bar{\mathbf{v}} = (\Delta t/\Delta x)\mathbf{v}$. With $\mathbf{u}^2 = \langle \mathbf{v} \rangle^2$ and $t = n\Delta t$, eq. 6 is equivalent to a diffusion process

$$\sigma_{\mathbf{r}(n\Delta t)} = \sqrt{[\langle \mathbf{v}^2 \rangle - \langle \mathbf{v} \rangle^2]\, t\Delta t} = \sqrt{\frac{\Delta x^2}{\Delta t} \bar{D}\, t} \tag{7}$$

where $\bar{D} = \langle \bar{\mathbf{v}}^2 \rangle - \langle \bar{\mathbf{v}} \rangle^2$ is the resulting numerical diffusion coefficient. Note that since $(\Delta x/\Delta t) = \text{const} = v$, it is interesting to note that $D \to 0$, when $\Delta x \to 0$. This means that the finer the grid, the smaller the diffusion.

Using eq. 1, it is possible to evaluate the dispersion in function of $\mathbf{u}$:

$$\langle \mathbf{v}^2 \rangle - \langle \mathbf{v} \rangle^2 = u_x \left( \frac{\Delta x}{\Delta t} - u_x \right) + u_y \left( \frac{\Delta x}{\Delta t} - u_y \right) \tag{8}$$

This shows that the dispersion is minimal when $\mathbf{u}$ would lead the particle close to a lattice site and maximal when it would lead the particle between sites. Therefore, although $D$ is not constant when $\mathbf{u}$ is not constant, the lattice can be adjusted to control the maximum dispersion.

### 2.2 Eruptions and Simulation Parameters

We apply our model to sedimentation with and without aggregates. Tab. 1 shows the simulation parameters used to describe the two case studies considered in this work (i.e. 1875 eruption of Askja volcano, Iceland; 18 May 1980 eruption of Mt. St. Helens, U.S.A). The eruption of Askja volcano is without aggregation. On the other hand, Mt. St. Helens eruption includes aggregate and the tephra deposits show a secondary maximum of thickness.

**Table 1.** Case studies. Askja: Askja 1875. Mt. St. Helens: 18 May 1980 eruption. The particle size $\Phi$ is given in granulometry units: $\Phi = -\log_2 d$, where $d$ is the particle diameter in mm. CA parameters are presented below the separation line.Wind velocity is maximum wind velocity at the tropopause.

| | **Askja** | **Mt. St. Helens** |
|---|---|---|
| Type of sedimentation | No aggregation | With aggregation |
| $H_t$ [km] | 26 | 16 |
| $H_{cb}$ [km] | 14.6 | 9 |
| Wind velocity [m/s] | 25 | 32 |
| Particle size ($\Phi$) | -3 to 6 | -3 to 9 |
| Median of grain size | 1.5 | 3 |
| References | [3] | [1] |
| $\Delta x$ [m] | 500 | 500 |
| $\Delta t$ [s] | 16.6 | 12.5 |
| Duration [s] | $10^5$ | $10^5$ |

## 3 Results and Discussion

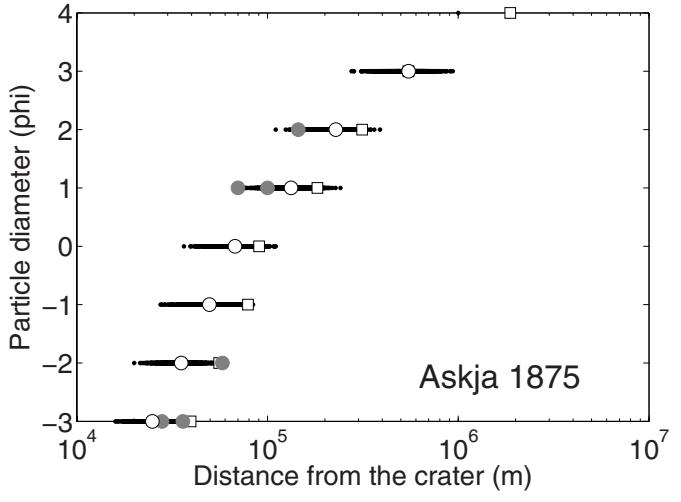
We first compare the CA to the classical BA (Fig. 3). The results of both approaches are in the same range but the average of CA for each class of particle sizes deviates from the BA predictions. The CA probabilistic model behaves in average like BA only when the velocity field is constant. Numerical experiments with a constant terminal velocity (data not shown) confirm this point. In our case, however, the terminal velocity $u_y$ varies with the vertical position of particles within the atmosphere.

Up to the above discrepancy CA results show a good agreement between observed and calculated distances of particle deposition for Askja volcano (no aggregation). In contrast, computed deposition for Mt. St. Helens significantly differ from observations. In fact, the eruption of the Mt. St. Helens has produced a secondary maximum of accumulation at about 300km ENE from the volcano due to aggregation processes that made fine particles fall closer than expected. Fig. 4 shows the grain size distribution of a sample collected at the distance of 182km from the crater as an example. In fact, tephra deposits affected by aggregation processes are typically characterized by bimodal distributions where the coarse particle mode represents the population of particles that fell individually, whereas the fine particle mode represents the population of particles that fell as aggregates. The discrepancy between observed and calculated distances of Mt. St. Helens is therefore due to the fact that the fine particles fell as larger aggregates and thus deposited closer to the vent than expected. According to [8], most particles of diameter $d < 125 \times 10^{-3}$ mm (i.e. $\Phi = -\log_2 d > 3$) are likely to fall as dry or wet aggregates. Eventually most aggregates break when impacting the ground, therefore releasing their fine particles and producing a bimodal grain size distribution as in Fig. 4.

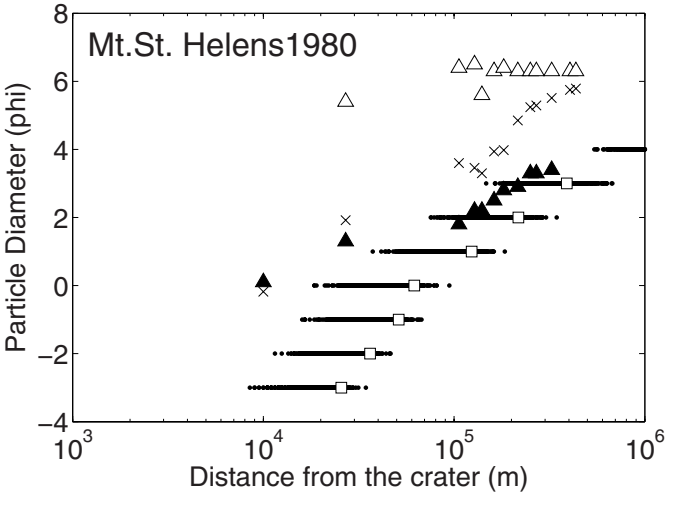
Based on our preliminary results, future work for implementing a multiparticle CA model describing tephra dispersal will focus on the aggregation processes which are believed to happen within any eruption plumes characterized by fine particles. Also we plan to consider the fundamental role of wind advection in the dynamics of volcanic plumes and particle sedimentation, and to couple our transport model to a Lattice Boltzmann solver using a multiscale modeling approach developed in [13].

In our model, the particle diffusion is strongly affected by its velocity in relation to the space discretization (see eq. 7). Since the terminal velocity depends on the particle diameter which varies from $10^{-4}$ to $10^{2}$ mm, the particle diffusion varies considerably. Therefore, future work will also involve grid refinement to adapt the grid to the expected diffusion coefficient.

In conclusion, we have shown that a multiparticle CA model is sufficient for describing simple tephra transport from volcanic plumes without aggregation (e.g. Askja eruption). However, in order to describe sedimentation controlled by aggregation processes (e.g. Mt. St Helens eruption), it is necessary to continue our study and improve our model.




(a)




(b)

**Fig. 3.** Observed and computed deposition distances for different particle sizes for (a) Askja 1875 and (b)Mt.St.Helens 1980. For Askja volcano (a), CA results are shown as black dots (which make black lines as there are many points), BA results as white squares, the average of CA for each class as white circles and observed of grain size modes of individual tephra sample as gray circles. For Mt. St. Helens (b), CA results are shown as black dots (which make black lines as there are many points) and BA results as white squares. There are also observed coarse modes shown as black triangles, fine modes as white triangles and grain size in mean modes are as crosses.

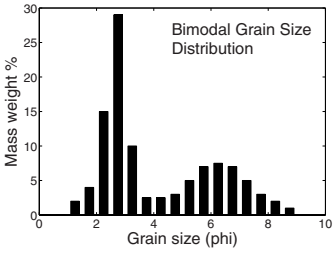



**Fig. 4.** Example of bimodal grain size distribution of a tephra sample produced by the Mt. St. Helens eruption and collected 182km from the volcano [4].

**Acknowledgements.** This work is supported by the Swiss National Science Foundation and the European Commission (COAST project EUFP6- IST-FET Contract 033664).

## References

1. Bonadonna, C., Phillips, J., Houghton, B.F.: Modeling tephra sedimentation from a Ruapehu weak plume eruption. J. Geophys. Res. 110(B8), 1–22 (2005)
2. Woods, A.W.: The fluid dynamics and thermodynamics of plinian eruption columns. Bull Volcanol. 50, 169–193 (1988)
3. Sparks, R.S.J., Wilson, L., Sigurdsson, H.: The pyroclastic deposites of the 1875 eruption of Askja, iceland. Philos. Trans. R. Soc. London 229, 241–273 (1981)
4. Carey, S.N., Sigurdsson, H.: Influence of particle aggregation on deposition of distal tephra from the may 18, 1980, eruption of mount st. helens volcano. J. Geophys. Res. 87(B8), 7061–7072 (1982)
5. Bursik, M.I., Sparks, R.S.J., Gilbert, J.S., Carey, S.N.: Sedimentation of tephra by volcanic plumes: I. theory and its comparison with a study of the fogo a plinian deposit, sao miguel (azores). Bull Volcanol. 54, 329–344 (1992)
6. Bonadonna, C., Ernst, G.J., Sparks, R.S.J.: Thickness variations and volume estimates of tephra fall deposits: the importance of particle reynolds number. J. Volcanol Geotherm Res. 81, 173–187 (1998)
7. Suzuki, T.: A theoretical model for dispersion of tephra. In: Arc Volcanism, Physics and Tectonics. Terra Sci. Tokyo, pp. 95–113 (1983)
8. Bonadonna, C., Phillips, J.: Sedimentation from strong volcanic plumes. J. Geophys. Res. 108(B7), 2340–2368 (2003)
9. Bonadonna, C., Connor, C.B., Houghton, B.F., Connor, L., Byrne, M., Laing, A., Hincks, T.K.: Probabilistic modeling of tephra dispersal: Hazard assessment of a multiphase rhyolitic eruption at tarawera, new zealand. J. Geophys. Res. 110(B03203) (2005)
10. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press, Cambridge (1998)
11. Masselot, A., Chopard, B.: A lattice boltzmann model for particle transport and deposition. Europhys. Lett. 42, 259–264 (1998)
12. Dupuis, A., Chopard, B.: Lattice gas modeling of scour formation under submarine pipelines. J. Comput. Phys. 178(1), 161–174 (2002)
13. Hoekstra, A., Lorenz, E., Falcone, J.L., Chopard, B.: Towards a complex automata formalism for multuscale modeling. Int. J. Multiscale Computational Engineering 5(6), 491–502 (2008)

# Global and Local Processes in a Model of Innovation

Marco Villani, Roberto Serra, Luca Ansaloni, and David Lane

DSSC, University of Modena and Reggio Emilia,
via Allegri 9, I-42100 Reggio Emilia
{marco.villani,roberto.serra,luca.ansaloni,david.lane}@unimore.com

**Abstract.** In this work we present the introduction of spatial constraints in a model of generation and diffusion of innovations. The presence of spatial limitations introduces several feedbacks, whose main effects are the decrease of global diversity in favour of a higher robustness, despite the apparent minor success of the individual agents. All these features hold contemporarily, but the individuated feedbacks are able to explain their only apparently contradictory nature. None of these results is obvious, nor can it be simply deduced from the qualitative theory. Moreover, the simulations could make possible comparisons between the model behaviours and the theory claims, indicating new ways of improvement and development

**Keywords:** Innovation, multi agent based systems, spatial constraints.

## 1 Introduction

During last decades, innovation has become a hot topic in a variety of social contexts, including technology, commerce, social systems, economic development, and policy construction. There are therefore a wide range of approaches to conceptualising innovation in the literature [5]. A consistent theme may be identified: innovation is typically understood as the successful introduction of something new and useful, for example introducing new methods, techniques, or practices or new or altered products and services.

Modelling such processes is a hard challenge. Some authors choose the simplifying assumption that only artifacts are important (technological trajectories) whereas others claim that at least in social issues the human creativity is the key, and that the recombination of existing current capabilities and local search are important [3] [12] [9]. But in general both agents (whatever their nature) and artifacts could be important [7], the agents typically maintaining the role of main source of novelties and the artifacts assuring and shaping their diffusion.

For these reasons in last years agent-based models (ABMs) are attracting considerable attention, as they allow one to escape from the constraints of the "representative agent" and to account for agent heterogeneity (see e.g. [4] [6] [11] [10] [1]). ABMs are well suited to bridge the gap between hypotheses concerning the microscopic behaviour of individual agents and the emergence of collective

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 401–408, 2008.

phenomena in a population composed of many interacting agents. Agent-based models are very attractive, but they still face formidable problems. In particular they allow the modeller perhaps an excess of freedom with respect to the design of agents and their interactions: ABMs often have too many variables and parameters. In these cases, what is needed to get meaningful results is a principle which limits the degrees of freedom of the modeller. While Ockhams razor is of the outmost importance, it does not seem to be sufficient by itself to limit the *embarasse de richesse* of ABMs.

In our work we consider another approach, namely that of relying upon an existing qualitative theory of the phenomenon in order to constrain our modelling options. The theory provides a basic ontology for the model, its entities and their modes of interaction [7]. The model represents therefore a simplified universe inhabited by (some of) the entities posited by the theory. By means of this model it is possible to realise simulations, which in this context have the role that experiments play in the "hard" sciences, as for example physic or chemistry [2]. This approach has already led us to some interesting results highlighting the interrelations among the agents innovation modalities and the resulting structures emerging in the agent-artifact space [8] [13]; nevertheless, the model we used ($I_2M$ in the following) was based upon the hypothesis that each of its entities could potentially interact with all the others. This simplification is in most interesting cases unrealistic. Moreover, a wide literature exists concerning the influence of spatial proximity on economics and innovations (e.g. in so-called industrial districts). Therefore in this paper we consider the effect of introducing a spatial topology in the agent space of our model, and to allow only local interactions.

## 2 Theory and Model

Let us now examine the major constraints the Lane and Maxfield theory imposes on the model. The first direct claim of the theory is that agents and artifacts are both important for innovation. This notion underlies the concepts of an "agent-artifact space" and in particular that of "reciprocality", which essentially claims that artifacts mediate interactions between agents, and vice versa, and that both agents and artifacts are necessary for a proper understanding of the behaviour of market systems and of innovation. One straightforward consequence of this claim is that it excludes the possibility to project agent-artifact space onto one of the two constitutive subspaces.

In the theory, artifacts are given meanings by the agents that interact with them, and different agents take different roles with respect to transformations involving artifacts. The meaning of artifacts cannot be understood without taking into account the roles which different agents can play in time. In this context, innovation is not just novelty, but rather a modification in the structure of agent-artifact space, which unleashes a cascade of further changes. Therefore the basic requirements for a model aiming at a dialogue with the theory are the following:

1. both agents and artifacts must be represented
2. the meanings of the artifacts must be generated within the model itself: it is inappropriate here to resort to an external oracle to decide a priori which meanings are better than others
3. the roles of agents must also be generated within the model:
4. agents must interact with artifacts and with other agents
5. an agent should be able to choose the other agents with whom to start a relationship
6. agents must have intentionality: they may be interested in certain types of artifacts or in entering a relationship with particular other agents

## 3 A Brief Description of the Basic Model

Let us now briefly introduce the main features of the model that has been developed on the basis of the constraints described above. We will limit ourselves here to a very concise overview of the model, and we refer the interested reader to [8] [13] for a more complete and detailed account. Formally, the basic structure of our ABM is:

**Time.** We model a system evolving in discrete time steps, i.e. $t = 1, 2, \ldots$

**Artifacts.** In present model artifacts are numbers, each number representing a type (e.g. cars, chairs, wheels, shoes, etc.)

**Agents.** The system is populated by a set of agents $I_t = \{1, 2, , N_t\}$

**Recipes.** Each agent $i \in I_t$ owns a set of $R_t$ recipes $\boldsymbol{r}_{i,t} = (r^1_{i,t}, \ldots, r^R_{i,t})$, that is, a set of input and mathematical operators. The recipes are the structures the agents use to build artifacts, by processing the already existing artifacts

**States.** Each agent $i \in I_t$ is characterized by a vector of $L$ states $\boldsymbol{x}_{i,t} = (x^1_{i,t}, \ldots, x^L_{i,t})$. These variables are fast ones, and can be endogenously modified by agents decisions (e.g. goal, reciprocal trust, etc.)

**Micro-Parameters.** Each agent $i \in I_t$ is also characterized by a vector of $H$ parameters $\boldsymbol{\theta}_i = (\theta^1_i, \ldots, \theta^H_i)$. Micro-parameters are slow-variables, i.e. quantities that cannot be endogenously modified by the agents within the time-scale of the dynamic process. Therefore, $\boldsymbol{\theta}_i$ typically contains information about behavioural characteristics of agent $i$ (e.g. agents propensity to innovate, etc.)

**Macro-Parameters.** The system as a whole is characterized by a vector of $M$ time-independent macro-parameters $\boldsymbol{\Theta} = (\Theta^1, \ldots, \Theta^M)$ governing the overall setup. Examples of $\boldsymbol{\Theta}$ parameters are some invariable characteristic of recipes and artifacts, the presence of raw materials, etc.

**Interaction Structures.** At each $t$, the way in which information is channelled among agents and artifacts is governed by the agents list of acquaintances, which determines (and is determined by) the agents actions.

**Decision Rules.** Each agent is endowed with a set of decision rules $\boldsymbol{D}_{i,t} = \{D^b_{i,t}(\cdot|\cdot), b = 1, \ldots, B\}$, mapping the actual situation into the future one. Examples of such decision rules are: innovation rules, useless recipes deleting rules, etc.

**Aggregate variables.** By aggregation (e.g. average, sum, etc.) of micro-variables, one obtains a vector of K macro-variables $\boldsymbol{X}_t = (X_t^1, \ldots, X_t^K)$ which contain all macro information relevant to the analysis of the system. Examples are: system diameter, average production level, etc.

In $I_2M$ agents can "produce" artifacts, each artifact being assembled by means of a recipe; the artifacts in turn can be used by other agents to build their own artifacts. Agents can try to widen the set of their recipes by applying genetic operators to their own recipes, or they can try to collaborate with another agent: in this case, in order to build the new recipe the agent can manipulate also the partners ones, realising in such a way a joint project. With respect $I_2M$, the model we are presenting in this work has the additional characteristic of dealing also with the quantity aspects, and take explicitly into account the number of items produced by each recipe. Therefore, each agent has a stockpile where its products are put, and from which its customers can obtain them. Each agent tends to an optimal level of artifacts items present within its stock: if the stock level is lower than the desired one the agent increases the production of the corresponding recipe; if the stock level is higher the agent decreases the production of the recipe. Each agent has to produce each recipe at least once, and cannot produce a recipe more than a given number of times each step. Production is assumed to be fast, i.e. it does not take multiple time steps.

The meaning of artifacts is just what agents do with them, while the role of agents is defined by which artifacts they produce, with whom, and for whom. The role of agents is also partly defined by the social networks in which they are embedded. In order to better determine with which acquaintances to interact, the agents can give an evaluation (a vote) to each existing relationship: the higher is the vote, the higher is the probability of choosing the partner to realise a joint project.

A key point is the structure of artifact space. What is required is that the space has an algebraic structure, and that suitable constructors can be defined to build new artifacts by combining existing ones. We have adopted a numerical representation for artifacts and the use of mathematical operators, therefore our agents are "producers" of numbers (indicated as "names" in the following), by means of the manipulation of other numbers.

If the recipes output is not used by other recipes, or if one of the needed inputs is not present within that part of the world known to the agent, a counter is incremented; otherwise the counter is set to its default. When this counter exceeds a given threshold the corresponding recipe is discarded, because it is useless or not realisable. As far as innovation is concerned, an agent can invent new recipes or manipulate the existing ones. In the present version of the model no new agents are generated, while agents can die because of lack of inputs or of customers.

The model is asynchronous: at each time step an agent is selected for update, and it tries to produce what its recipes permit. So, for each recipe, it looks for the input artifacts and, if they are present in the stocks of its suppliers, it produces the output artifact and puts it into its stock (the stocks of the supplier are of

course reduced). This sequence is iterated until the desired level of production is reached.

Besides performing the usual buy-and-sell dynamics, an agent can also decide to innovate. Innovation is a two-step process: the first defines a goal, i.e. an artifact which the agent wishes add to the list of its product, while in the second step the agent attempts to produce the desired artifact. In the goal-setting phase, an agent chooses one of the known types of artifacts (which, recall, is a number M) and then either tries to imitate it (i.e. its goal is equal to M) or to modify it by a jump (i.e. by multiplying M times a random number in a given range). It has been verified that imitation alone can lead to a sustainable production economy, in which however innovation eventually halts [12]. In the goal-realising phase (after setting its goal), an agent tries to reach its target by combining recipes to generate a new one via genetic algorithms.

## 4 The Topology

In $I_2M$ each agent has its own set of acquaintances, which depends on the agents relationships (client-provider relationships or simple acquaintance rapports) and the agents past history, but potentially it could know its whole environment. This is not always the case in the real world: at least in the past, the environment of economic and social agents was spatially limited. In effect also the list of an agents acquaintances within the actual world, despite the presence of long-range communication, could be described in first approximation as a cluster having high correlation within the spatial domain, on which subsequently one can introduce additional relations. Consequently, one of the first actions aiming to introduce more realistic characteristics in $I_2M$ is that of introducing spatial relationships and releasing the "global knowledge" hypothesis; the regular topology of cellular automata could be one of these first steps.

In order to perform such a modification it is enough to fix the agents list of acquaintances to an a priori given subset: in this work we adopt a 2D regular lattice, each position having a Moore neighbourhood with a fixed radius the opposite sides of the lattice are assumed communicating (the so-called toroidal configuration) in order to assure the same neighbourhood size to each agent. All the already presented characteristics of $I_2M$ hold unchanged, but the actions of artifact use, goal selecting and relationship evaluation are limited to the agents neighbourhood. In the following, we call this model $I_2M - CA$.

## 5 The Model Behaviour: A Preliminary Analysis

In order to better evaluate the impact of local relationships, in this first set of experiments we dont use the agents capabilities of evaluating their relationships and choosing their partners: during the experiments describer here each agent builds recipes by itself, without help from its neighbours.

We use a CA having 11x11 cells, each cell being a complete $I_2M$ agent; the agents list of acquaintances is defined by the Moore neighbourhood of radius R, the radius being fixed for each series of experiments; the automaton stops after 10000 steps, a number of iteration sufficient to reach a stable configuration; for each radius we report statistics made on 10 runs. The variables observed in this work belong to two main classes: extensive variables (the number of agents, the number of artifacts, the system diversity that is, the number of different artifact types contemporarily present within the system – and the system diameter – the difference between the highest and the lowest name contemporarily present within the system) and intensive variables (the global diversity divided by the number of agents, the median number of recipes owned by each agent, and the median production of the set of recipes owned by each agent).

In order to analyze the role of locality, the behaviour of the variables should be compared with the case where each agent has a complete knowledge of the whole system, i.e. the case in which the radius of the neighbourhood is equal to 5.

From the data shown in Fig. 1 one can see that the higher is the neighbourhood radius,

- the higher is the system diameter
- the higher is the artifact diversity per agent
- the lower is the number of recipes replicating low names
- the higher is the average production level of the recipes

but:

- the lower is the number of surviving agents
- the lower is the number of recipes owned by each agent

Notice the behaviour of the number of surviving agents. In the standard $I_2M$, when there is no collaboration in innovation, not all agents can survive [12]. In $I_2M - CA$ the fraction of agents that die vanishes as the interaction radius decreases, until the case of radius equal to 1 and 2, for which no agent dies. This despite the apparently conflicting facts that with radius 5 the agents can sustain higher diameter, higher diversity per agent and own recipes having higher production level. In other words, a locally organised world seems more robust to loss of agents than a globalised one, despite its lesser artifact diversity.

In effect, this apparent contradiction is resolved by noting the presence of some interesting features of the "local" world of $I_2M - CA$. The most interesting is that with low radius the world is highly fragmented, and each fragment has high probability of rebuilding the same basic (i.e., with low numerical value) artifacts; on the other hand, these basic artifacts are otherwise unavailable. As consequence, agents that in a "globalised" world wouldnt have a chance of building recipes that produce basic artifacts now can generate such recipes, sell the corresponding outputs, and survive. So, it is possible to observe that the lower is the radius of the neighbourhood, the higher is the number of recipes and in particular the number of recipes that build the must basic artifact (see Fig. 1d). Another interesting feature arises from the innovation processes: with less knowledge of the world, agents take their inspiration for new goals by imitating a

reduced set of artifacts, diminishing in such a way the diversity of the whole system. On the other hand, the relatively higher diversity of systems with high interaction radius allows them to sustain high diameters for their artifact space and high artifact diversity per agent.

Eventually, the relatively high diversity of these latter systems makes each artifact more essential (by increasing its probability of being unique or at least rare) increasing in such a way the average production level of the realised recipes. This third feature allows the interpretation of the behaviour shown in Fig. 1.

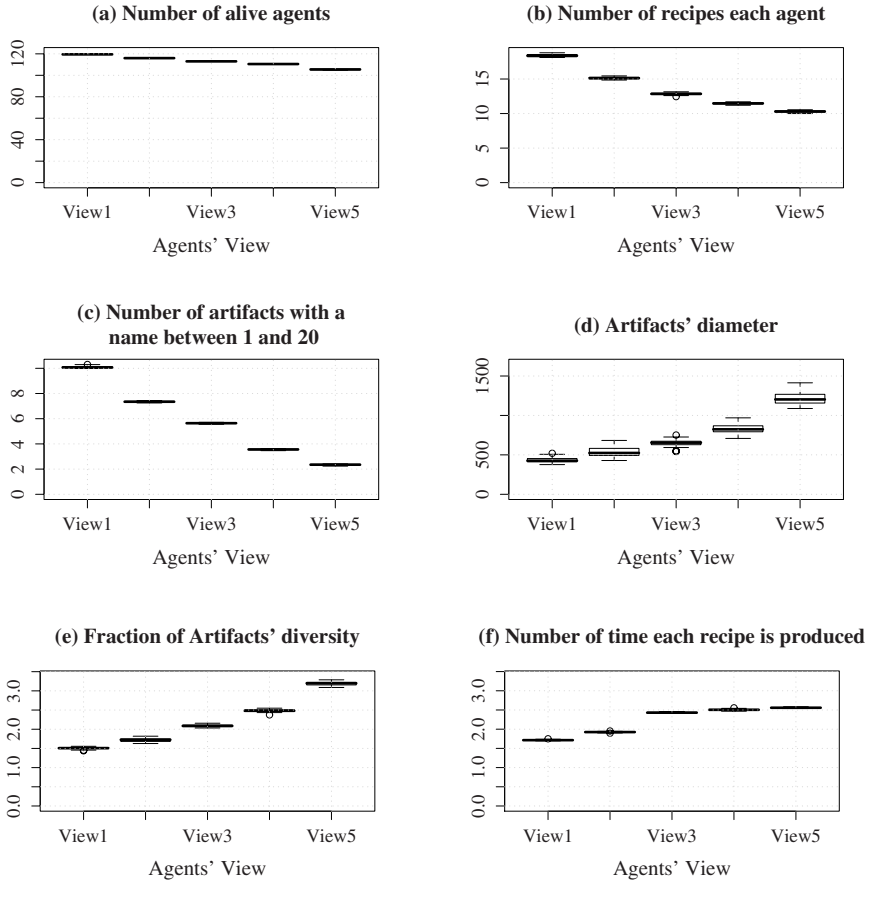



**Fig. 1.** Statistical behaviour of $I_2M - CA$ model: averages over 10 runs at step 10000. The thick line is the median value, the box extremes indicate the second and third quartiles, the upper and lower line indicates the first and fourth quartiles, and the dots are the outliers. See the text for a detailed description of the variables and of its behaviour.

## 6 Conclusions

In this work we present the introduction of spatial constraints in a model of creation and diffusion of innovations. Also in this simple version, the presence of strong spatial limitations introduces several features, whose main effects are the decrease of global diversity and of the system diameter, in favour of a higher robustness, despite the lower production levels per recipe. The identified features explain why the simultaneous appearance of the above-listed effects is only apparently, not actually, contradictory. None of these results is obvious, or can be simply deduced from the qualitative theory. Moreover, the simulations allow comparisons between the model behaviours and the theory claims, indicating new ways of improvement and development. While the present version of spatial constraints represents a highly simplified model, the results obtained so far appear to encourage the further development of $I_2M - CA$. Future research improvements to the model should be aimed at exploring the effect of topologies different from that of regular lattices, and simulating more complex interactions among agents and artifacts.

## References

1. Axelrod, R., Tesfatsion, L.: A guide for newcomers to agent-based modeling in the social science, vol. 2. North-Holland, Amsterdam (2006)
2. Cangelosi, A., Parisi, D.: Simulating the Evolution of Language. In: Computer simulation: A new scientific approach to the study of language evolution. Springer, London (2002)
3. Dosi, G.: Technological paradigms and technological trajectories. Research Policy, 147–162 (1982)
4. Epstein, J.M., Axtell, R.: Growing Artificial Societies: Social Science from the Bottom up. MIT Press, Cambridge (1996)
5. Fagerberg, J., David, C., Nelson, R.R.: Innovation: A Guide to the literature Fagerberg and January (2004)
6. Gilbert, N., Terna, P.: How to build and use agent-based models in social science. Mind and Society (2000)
7. Lane, D., Maxfield, R.: Ontological uncertainty and innovation. Journal of Evolutionary Economics 15(1), 3–50 (2004)
8. Lane, D., Serra, R., Villani, M., Ansaloni, L.: A theory-based dynamical model of innovation processes. Complexus 2(3-4), 177–194 (2005)
9. March, J.G.: Exploration and exploitation in organizational learning. Organization Science (1991)
10. Ormerod, P., Rosewall, B.: On the methodology of assessing agent-based evolutionary models in the social sciences. In: Second Brisbane workshop on complex adaptive systems (2002)
11. Cioffi Revilla, C.: Invariance and universality in social agent-based simulations. PNAS (2002)
12. Schumpeter, J.: The Theory of Economic Development. Harvard University Press (1934)
13. Serra, R., Villani, M., Lane, D.: Complexity perspectives on innovation and social change. In: Modelling Innovation. Springer, Berlin

# GP Generation of Pedestrian Behavioral Rules in an Evacuation Model Based on SCA

Stefania Bandini, Sara Manzoni, Giancarlo Mauri, Stefano Redaelli, and Leonardo Vanneschi

Complex Systems and Artificial Intelligence Research Center
University of Milan–Bicocca, Italy
Tel.: +39-02-64487808
{bandini,manzoni,redaelli,vanneschi}@disco.unimib.it

**Abstract.** This paper presents a research in the context of pedestrian dynamics according to Situated Cellular Agent (SCA), a Multi-Agent Systems approach whose roots are on Cellular Automata (CA). The aim of this work is to apply Genetic Programming (GP) approach, a well known Machine Learning method belonging to the family of Evolutionary Algorithms, to generate suitable behavioral rules for pedestrians in an evacuation scenario. The main contribution of this work is in the design of a testset of GP generated behaviors to represent basic behavioral models of evacuees populating a only locally known environment, a typical scenario for CA-based models.

## 1 Introduction

Within pedestrian dynamics Cellular Automata approaches have found an interesting and fruitful application context (for instance in evacuation studies and public spaces' design [1,2]). Since relatively recent first proposals [11,12], CA approach has rapidly grown and shown interesting results concerning the study of potentially complex behaviors that can result from local interactions among pedestrians within a shared, limited, and only partially known spatial environment [3,4]. According to CA peculiarities the spatial environment can be represented as a regular grid of cells, whose state can include the representation of the presence of individuals (or other environmental obstacles). Pedestrian movement is represented by CA state transition rules and the dynamics of the system result from local state change of CA cells.

When adopting CA for modelling purposes with the aim of study pedestrian behavior and their interaction, CA models suffer, like traditional analytical approaches, the limitation of considering individuals as homogeneous entities whose behavior is implicitly represented in CA cell state and state transition function. SCA4CROWDS [5] is an ongoing research aiming at developing formal and computational tools to support the design, execution and analysis of models to study potentially complex dynamics that can emerge in crowds (e.g. pedestrian dynamics as effect of physical, social and emotional interactions) [6,23]. SCA4CROWDS formal model has been developed as an extension of Cellular Automata (CA)

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 409–416, 2008.

[7] exploiting MAS [8] advantages in modeling heterogeneous systems [9,10]. Despite models based on CA, SCA4CROWDS models systems of reactive situated agents (i.e. pedestrians) that move on structured spatial environments, and that can interact at-a-distance through the emission-diffusion-perception of signals and locally according to local transition rules. Theoretical experimentations are being developed with SCA4CROWDS formal and computational tools in order to provide methodological guidelines for sounding computational models of psychological and anthropological theories on crowds (e.g.[13]).

The work presented in this paper concerns the integration into SCA4CROWDS framework of formal and related computational tools to effectively combine Genetic Programming (GP) [14] with SCA approach to study the behavior of pedestrians (a similar approach has already been proposed for the calibration of a CA–based model of costumers in shopping areas [18]). In particular, we will present here a model based on SCA principles where pedestrian behaviors are based on GP, a well known evolutionary approach which extends the genetic model of learning to the space of programs. The best set of behavioral rules in an evacuation context drives a system of pedestrians evacuating a structured and unknown environment. Evacuation scenarios are traditionally considered realistic when they assume that pedestrians have a limited and very local information about the environment and a basic intelligent behavior that can be originated by instinctive or learning processes. To support the development and experimentation of this type of scenario, we developed a behavioral model based on a set of rules generated by Artificial Ant on the Santa Fe trail, a benchmark problem in GP [14]. The latter aims at finding navigation strategy for an ant moving on a regular grid (where some of the cells contain food pellets) that maximizes its food intake. This problem specification has been used as a methodological example useful for the SCA–based model calibration.

After an overview of SCA approach for pedestrian and crowds modeling, Genetic Programming approach will be introduced in Section 3, while Section 4 describes experiments with SCA4CROWDS framework where pedestrian behavioral rules of a set of evacuees are generated by GP. Currently we are developing an analytical analysis with references and benchmarks in pedestrian dynamics.

## 2 SCA Approach to Pedestrian Dynamics

According to SCA modelling approach, human crowds are described as system of autonomous, situated agents that act and interact in a spatially structured environment. Situated agents are defined as reactive agents that, as effect of the perception of environmental signals and local interaction with neighboring agents, can change either their internal state or their position on the structured environment. Agent autonomy is preserved by an action–selection mechanism that characterizes each agent, and heterogeneous MAS can be represented through the specification of agents with several behavioral types through L*MASS formal language [20] (an execution environment for SCA-based models is also available, i.e. SCA platform [19]). Interaction between agents can occur either locally,

causing the synchronous change of state of a set of adjacent agents, and at–a–distance, when a signal emitted by an agent propagates throughout the spatial structure of the environment and is perceived by other situated agents (heterogeneous perception abilities can be specified for SCA agents). Interaction primitives are defined by L*MASS language.

SCA model is rooted on basic principles of CA: it intrinsically includes the notions of state and explicitly represents the spatial structure of agents' environment; it takes into account the heterogeneity of modelled entities and provides original extensions to CA (e.g. at–a–distance interaction). According to SCA framework, the spatial abstraction in which the simulated entities are situated (i.e. *Space*) is an undirected graph of *sites* (i.e. $p \in P$), where graph nodes represent available space locations for pedestrians and graph edges define the adjacency relations among them (and agents' suitable movement directions). Each $p \in P$ is defined by $\langle a_p, F_p, P_p \rangle$, where $a_p \in A \cup \{\perp\}$ is the agent situated in $p$ , $F_p \subset F$ is the set of fields active in $p$ and $P_p \subset P$ is the set of sites adjacent to $p$. Pedestrians and relevant elements of their environment that may interact with them and influence their movement (i.e. *active elements of the environment*) are represented by different types of SCA agents that can change their internal state ($s \in \Sigma_\tau$), move into an adjacent site or interact with other agents. An agent *type* $\tau = \langle \Sigma_\tau, Perception_\tau, Action_\tau \rangle$ is defined by:

- $\Sigma_\tau$: the set of states that agents of type $\tau$ can assume;
- $Perception_\tau : \Sigma_\tau \rightarrow W_F \times W_F$ function for agents of type $\tau$: it associates each agent state to a pair (i.e. *receptiveness coefficient* and *sensitivity threshold*) for each field in $F$;
- $Action_\tau$: the behavioral specification for agents of type $\tau$ in terms of L*MASS language [20].

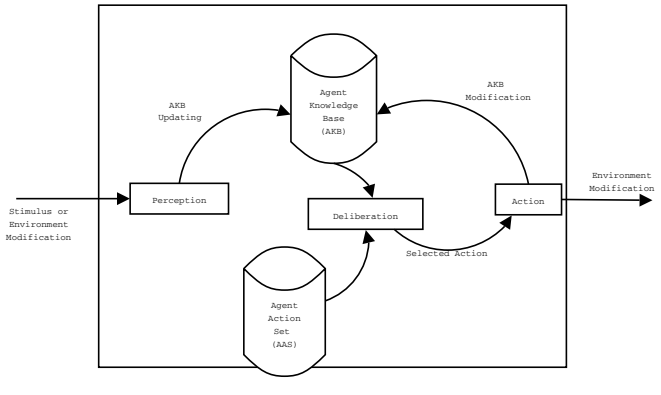



**Fig. 1.** A schematic representation of the internal architecture of SCA agents

Agent architecture (see figure 1) is composed by three tasks that define the agent actual behavior (i.e. *Perception*, *Deliberation*, and *Action*) and two knowledge containers:

- ***Agent Knowledge Base (AKB)*** is the internal representation of agent state and of its local perceptions (e.g. set of fields active in its site, set of empty sites in its surrounding). The AKB updating can be the effect of agent actions or of a change in the agent environment perceived by the agent (e.g. an adjacent site becomes empty, a new field reaches the agent site or the agent moves to another site).
- ***Agent Action Set (AAS)*** collects the set of actions that are allowed to the agent in terms of L*MASS language. AAS is defined according to the agent type and cannot change during agent execution. In the model presented in Section 4, we will experiment GP to evolve AAS towards an optimal strategy for evacuation.

SCA approach does not specify a standard way to define agents' perception, deliberation and action. SCA platform (the execution environment for SCA-based models) has been designed in order to be incrementally extended to several execution strategies. In our experiments we adopted a synchronous–parallel execution method for the system (i.e. at each timestep each agent update their AKB perceiving their local environment and selects the action to be performed). The phase between perception and execution is *deliberation* that is, the component of an agent responsible of conflict resolution between actions, when multiple actions are possible.

## 3 Genetic Programming

Genetic Programming (GP) [14] is a Machine Learning method that belongs to the family of Evolutionary Algorithms [15,16,17]. Its peculiar characteristic is that potential solutions (often called individuals) to be evolved are not fixed length strings of characters, as for Genetic Algorithms (GAs) or other evolutionary methods, but, generally speaking, *computer programs*. The fitness of a program is usually calculated by running it one or more times with a variety of inputs and seeing how close the program outputs are to a desired target. Programs can be represented as trees, lines of code, expressions in prefix or postfix notations, strings of variable length, etc. For tree-based GP, which is the original [14] and more popular version of GP and the one we use in this paper, the set of all the possible structures that can be generated is the set of all the possible trees that can be built recursively from a set of function symbols $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ (used to label internal tree nodes) and a set of terminal symbols $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ (used to label tree leaves). Each function in the fuction set $\mathcal{F}$ takes a fixed number of arguments, specifying its *arity*. Functions may include arithmetic operations ($+$, $-$, $*$, etc.), mathematical functions (such as `sin`, `cos`, `log`, `exp`), boolean operations (such as `AND`, `OR`, `NOT`), conditional operations (such as `If-Then-Else`), iterative operations (such as `While-Do`) and other domain-specific functions that may be defined. Each terminal is typically either a variable or a constant, defined on the problem domain. In the work presented in this paper the generated structure describes the behavior of a pedestrian agent. The terminal actions are atomic movements as: move towards, turn

to the left or turn to the right; while functions define nodes with two and three childs (see a deep descripion in section 4). Once such a formal language to code programs has been defined, the GP paradigm breeds those programs to solve problems by executing the following steps:

1. Generate an initial population of computer programs (or individuals).
2. Iteratively perform the following steps until the termination criterion has been satisfied:
   (a) Execute each program in the population and assign it a fitness value according to how well it solves the problem.
   (b) Create a new population by iteratively applying the following operations:
      i. Probabilistically select a set of computer programs to be reproduced, on the basis of their fitness (*selection*).
      ii. Copy some of the selected individuals, without modifying them, into the new population (*reproduction*).
      iii. Create new computer programs by genetically recombining randomly chosen parts of two selected individuals (*crossover*).
      iv. Create new computer programs by substituting randomly chosen parts of some selected individuals with new randomly generated ones (*mutation*).
3. The best computer program appeared in any generation is designated as the result of the GP process at that generation. This result may be a solution (or an approximate solution) to the problem.

Most commonly used termination criteria are: (1) at least one individual in the current population has a satisfactory fitness value, or (2) a prefixed number of generations has been executed.

## 4 Modeling Pedestrian Behavior for Evacuation with GP

In this work, we present a GP configuration inspired by the Artificial Ant on the Santa Fe trail benchmark to simulate the trajectory of a set of agents in a square space, with the aim of automaticallty generating suitable pedestrian paths for evacuation. The Artificial Ant on the Santa Fe trail is a problem where an artificial ant is placed on a regular toroidal grid, where some of the cells contain food pellets. The problem goal is to find a navigation strategy for the ant that maximizes its food intake. The ant starts in the upper left cell of the grid, identified by the coordinates (0, 0), facing east. It has a very limited view of its world. In particular, it has a sensor that can see only a single immediately adjacent cell in the direction the ant is currently facing.

Inspirating to this problem we defined a pool of $N$ pedestrian agents, with $N > 1$ placed in a regular SCA space with Von Neumann neighborhood. The space is defined as a regular not toroidal square grid representing a room to be evacuated. In this scenario agents have a random starting postition and they have a direction that points to one of the adjacent sites; they have no knowledge

of the space (they can perceive only the adjacent place in the direction the agent is currently facing) and also the initial direction is randomly setup for each agent. No food is contained in the grid, and the only special place on the space is the given location identified as the exit.

Agent's behavior is built by GP and the set of possible structure that can be generated is the set of all possible trees that can be generated from $\mathcal{T} = \{Right, Left, Move\}$ and $\mathcal{F} = \{Progn2, Progn3\}$. The set of terminals $\mathcal{T}$ corresponds to the actions the agent can perform: turn right by 90°, turn left by 90° and move forward in the currently facing direction. In the set $\mathcal{F}$ function $Progn2$ takes two arguments and causes the agent to unconditionally execute the first argument followed by the second one, while $Progn3$ is analogous, but it takes three arguments, that are executed in an ordered sequence. An individual built with these sets $\mathcal{F}$ and $\mathcal{T}$ can be considered as a "*navigation program*" that allows the agent to navigate the grid. When the number of agents $N$ is $N > 1$ the set of $\mathcal{F}$ and $\mathcal{T}$ builted by GP is applied to all the agents.

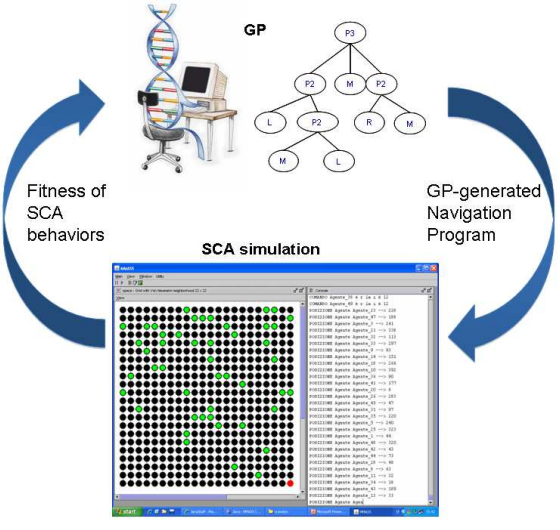



**Fig. 2.** The fitness of all Navigation Programs (the tree in the upper side of the figure) generated by the GP are obtained through a SCA simulation (a screenshot is shown in the lower side of the figure)

As fitness function, the number of agents which reach the exit and the total number of steps that they perform in order to reach the exit are considered (any action is considered as taking one time step). This turns the problem into a minimization one, with the globally optimal solution having a fitness value equal to 0. This problem specification represents a methodological example that can be very useful for SCA–based model calibration when data on real crowded situations have been collected. In these cases a suitable fitness function have to be specificated properly.

SCA simulation are used to obtain the fitness value of each individual behaving according to the *navigation program* (see figure 2). The experiments are performed in the synchronous mode that SCA organizes into two tasks: *deliberation* where each agent selects the next position according to its perception of the local environment, and *action* where agents try to perform what they have deliberated. All agents complete the phase *deliberation* before the *action* phase starts, and the order in which agents perform these phases are random at each simulation cycle. Conflics between the agents are managed by the SCA platform, and during the execution phase on the SCA–based simulation, if an agent cannot perform the action expressed by its behavior (i.e. the *navigation program*), it changes randomly its direction turning to the right or to the left (in figure 2 a screenshot of a SCA simulation). The possible causes that do not allow an agent to move according to its *navigation program* are the presence of obstacles: static obstacles are the borders of the space, or other structure that can be added to the space map, and dynamic obstacles are the other agents.

Agent behavior is limited to 600 time steps. This time-out limit is sufficiently small to prevent a random walk of the agent to cover all the 1024 squares (in case of a $32 \times 32$ space) before timing out.

## 5 Conclusions and Future Works

This paper presents an application of GP and CA-based approach to pedestrian dynamics. An analysis of the presented model is still ongoing. Future experiments on more structured spaces as buildings with a given number of rooms are in progress too. SCA approach allows specifying and simulating heterogeneous systems of agents with different behavioral rules. Therefore more complex studies on pedestrian dynamics considering different types of behaviors will be experimented and compared.

Other examples of current ongoing works concerns the study of aggregation phenomenon in *Open Crowds* [22] and, a SCA-based specification of *Affectons* formal framework proposed to study complex crowds' dynamics emerging from emotional interaction [23] and its adoption to study *emotional crowds* (e.g. at concerts or sports events). A further application for CA and GP in pedestrian dynamics context will concern the position optimization of obstacle and mobile structures in public spaces' design (see [1,24]).

## References

1. Klupfel, H.: A Cellular Automaton Model for Crowd Movement and Egress Simulation. PhD thesis, Universität Duisburg-Essen (2003), http://www.ub.uni-duisburg.de/ETD-db/theses/
2. Dijkstra, J., Timmermans, H., Jessurun, A.: A multi-agent cellular automata system for visualising simulated pedestrian activity. In: Theoretical and Practical Issues on Cellular Automata. LNCS, pp. 29–36. Springer, Heidelberg (2001)

3. Schadschneider, A., Kirchner, A., Nishinari, K.: CA approach to collective phenomena in pedestrian dynamics. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493, pp. 239–248. Springer, Heidelberg (2002)
4. Proceedings of 4th Conference on Pedestrian Dynamics, Wuppertal, February 2008. Springer, Heidelberg (forthcoming, 2008)
5. Bandini, S., Federici, M.L., Manzoni, S., Vizzari, G.: Towards a methodology for situated cellular agent based crowd simulations. In: Dikenelli, O., Gleizes, M.-P., Ricci, A. (eds.) ESAW 2005. LNCS (LNAI), vol. 3963. Springer, Heidelberg (2006)
6. Still, G.K.: Crowd Dynamics. PhD thesis, University of Warwick, Warwick (2000), http://www.crowddynamics.com/
7. Bandini, S., Manzoni, S., Simone, C.: Enhancing cellular spaces by multilayered multi agent situated systems. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493, pp. 156–167. Springer, Heidelberg (2002)
8. Ferber, J.: Multi-Agent Systems. Addison-Wesley, Harlow (1999)
9. Bandini, S., Manzoni, S., Simone, C.: Modelling heterogeneity in multi agent systems. In: Shafazand, H., Tjoa, A.M. (eds.) EurAsia-ICT 2002. LNCS, vol. 2510, pp. 685–692. Springer, Heidelberg (2002)
10. Bandini, S., Manzoni, S., Simone, C.: Heterogeneous agents situated in heterogeneous spaces. Applied Artificial Intelligence 16, 831–852 (2002)
11. Schadschneider, A.: Cellular Automaton Approach to Pedestrian Dynamics - Theory. In: Pedestrian Dynamics, pp. 75–86. Springer, Heidelberg (2002)
12. Schadschneider, A.: Cellular Automaton Approach to Pedestrian Dynamics - Applications. In: Pedestrian Dynamics Applications, pp. 87–98. Springer, Heidelberg (2002)
13. Bandini, S., Federici, M.L., Manzoni, S., Redaelli, S.: A SCA–based model for Open Crowd aggregation. In: Pedestrian Dynamics. Springer, Heidelberg (2008)
14. Koza, J.: Genetic Programming. MIT Press, Cambridge (1992)
15. Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
16. Fogel, L., Owens, A., Walsh, M.: Artificial Intelligence Through Simulated Evolution. John Wiley and Sons, New York (1966)
17. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
18. Kitazawa, K., Batty, M.: Pedestrian behaviour modelling: An application to retail movements using a genetic algorithm. In: 7th International Conference on Design and Decision Support Systems in Architecture and Urban Planning (2004)
19. Bandini, S., Manzoni, S., Vizzari, G.: Towards a platform for mmass–based simulations: focusing on field diffusion. Applied Artificial Intelligence 20, 327–351 (2006)
20. Bandini, S., Manzoni, S., Pavesi, G., Simone, C.: L*MASS: A language for situated multi-agent systems. In: Esposito, F. (ed.) AI*IA 2001. LNCS (LNAI), vol. 2175, pp. 249–254. Springer, Heidelberg (2001)
21. Schreckenberg, M., Sharma, S.: Pedestrian and Evacuation Dynamics. Springer, Berlin (2002)
22. Canetti, E.: Crowds and Power. The Noonday Press/Farrar, Straus and Giroux (1984)
23. Adamatzky, A.: Dynamics of Crowd–Minds. World Scientific, Singapore (2005)
24. Kretz, T., Schreckenberg, M.: The f.a.s.t.-model. In: Yacoubi, S.E., Chopard, B., Bandini, S. (eds.). LNCS, pp. 712–715. Springer, Heidelberg (2006)

# A Three-Dimensional Pedestrian-Flow Simulation for High-Rising Buildings

Tomoyuki Hamada[1], Takayuki Hagiwara[1], Takashi Teramoto[1], Shin Morishita[2], Michiyuki Umetsu[3], and Michiyo Ohgama[4]

[1] Mechanical Engineering Research Laboratory, Hitachi, Ltd., 832-2 Horiguchi, Hitachinaka, Ibaraki 312-0034, Japan
{tomoyuki.hamada.ua,takayuki.hagiwara.cf,takashi.teramoto.sh}@hitachi.com
[2] Yokohama National University, 79-7 Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa 240-8501, Japan
mshin@ynu.ac.jp
[3] East Japan Marketing & Communications, Inc., 1-5-5 Ebisu-Minami, Shibuya-ku, Tokyo 150-8508, Japan
Umetsu.Michiyuki@jeki.co.jp
[4] MOSAIC Co. Ltd., 2-18-19 Omote-cho, Maebashi, Gunma 371-0024, Japan
ohgama@mosaic.co.jp

**Abstract.** A three-dimensional pedestrian-flow simulation system for optimally designing a floor plan and elevator configuration of a high-rising building was developed. With this simulation system, the pedestrian flow is simulated by synchronizing horizontal traffic on each building floor and vertical traffic in elevators to generate three-dimensional movements of pedestrians in the building. The simulation system comprises an interaction model, in which the horizontal traffic affects the vertical traffic through elevator-calling actions of people, and the vertical traffic affects the horizontal traffic by massive injection of people onto floors from elevator cages. The effectiveness of the developed system was verified by comparing the simulated pedestrian flows with actual flows in two real buildings.

**Keywords:** Pedestrian flow, Simulation, Elevator traffic, High-rising building.

## 1 Introduction

The functions of high-rising buildings have recently been increasing in variety. In some cases, a single building contains office space, shopping space, hotel space, and a train station. In these buildings, there are a variety of pedestrian flows, such as fast flows toward the station, excursive flows in the shopping area, and crowd flows in event spaces. In addition to their functional variation, high-rising buildings comprise complicated elevator configurations, including sky-lobby systems and double-decker elevators, for increasing the efficiency of floor usage. Consequently, an optimal floor plan and elevator configuration suitable for these different pedestrian flows is necessary for these high-rising buildings. A system

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 417–424, 2008.

for simulating the complicated pedestrian flow in a high-rising building is thus required.

There are several research works on pedestrian-flow simulation in buildings. Okazaki et al. proposed a simulation method using a magnetic field [1]. Burstedde et al. added the concept of a "floor field" to a conventional cellular-automaton model [2]. These pedestrian-flow simulations, including these two research works, are mostly applied to evaluating emergency evacuation from a building, where pedestrian flows that do not use vertical transportation (e.g., elevators) are mainly considered. On the other hand, elevator manufacturers study elevator traffic for improving efficiency of vertical transportation [3][4]. In these studies, solving the "cage-allocation problem" to shorten waiting times in the elevator hall has mostly been focused on.

The pedestrian flows on a building's floors and the elevator traffic, however, affect each other. That is, a pedestrian's arrival time at the elevator hall and entering motion to the elevators determine the elevator up and down motions, and a massive injection of people onto a floor from an elevator cage determines the pedestrian flow of that floor. To evaluate this interaction between floor traffic and vertical elevator traffic, we developed a "three-dimensional pedestrian-flow simulation system." In the following, the configuration of the simulation system is described, and the results of experiments to verify the effectiveness of the system are presented.

## 2 System Configuration

Figure 1 shows a general block diagram of the three-dimensional pedestrian-flow simulation system. The input data consists of map data and inflow setting data. The map data determines the floor layout of each story of the building. The inflow setting data determines the number of incoming pedestrians at every entrance over a certain time period. The inflow setting data also determines the destination of the incoming pedestrians.

The simulation body consists of modules calculating floor flow, elevator traffic, and their interaction. The floor-flow calculation module uses a cellular automaton model for implementing pedestrian behavior. The elevator-traffic calculation

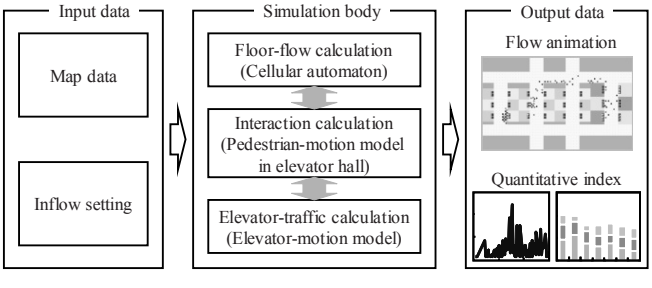



**Fig. 1.** General block diagram of 3-D pedestrian-flow simulation system

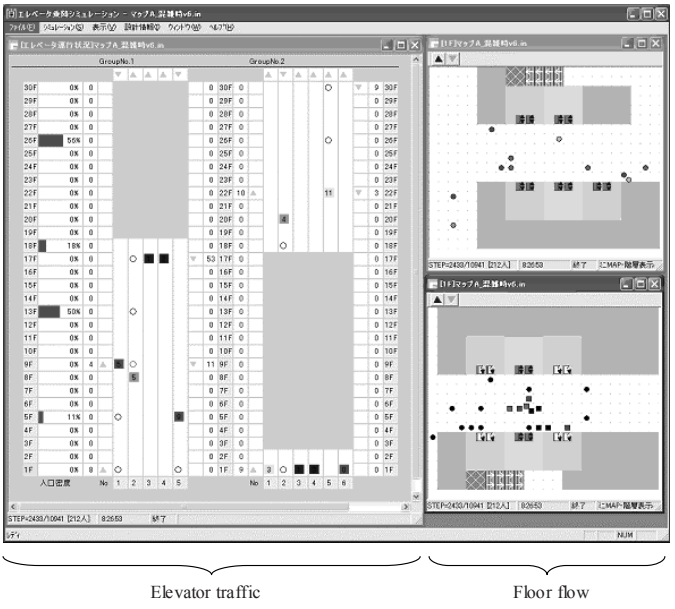



**Fig. 2.** Example of animated simulation result

module uses an elevator-motion model that simulates the cage motion of elevators by a cage-allocation algorithm. The interaction-calculation module simulates the pedestrian motion in the elevator hall, including elevator calling. Through this interaction calculation, the floor flow and vertical traffic are synchronized.

The simulation results are displayed as animations of pedestrian motions and elevator motions. The simulation results are analyzed to indicate quantitative indexes such as congestion degree and average trip time of pedestrians. Figure 2 shows an example scene of the animated simulation results.

## 3 Floor-Flow Model

There are several approaches to modeling a pedestrian flow on a floor [1][2]. The cellular-automaton (CA) model is popularly used for pedestrian-flow simulation, because of its advantage that the complex system of crowd dynamics is represented by simple local-neighbor rules [5]. Our simulation system thus applies a cellular-automaton model to simulate several thousands of pedestrians in a high-rising building in a practical time.

In our CA model, the simulation space is divided into square cells, and state variables are defined according to "pathway," "wall," "entrance," "exit," and "pedestrian." The pedestrian cells also have a state variable for walking direction. A pedestrian has one of eight walking directions: four orthogonal and four diagonal. A pedestrian may move in the walking direction or two neighboring

**Fig. 3.** Walking direction (solid-line) and possible directions of motion (doted-line)

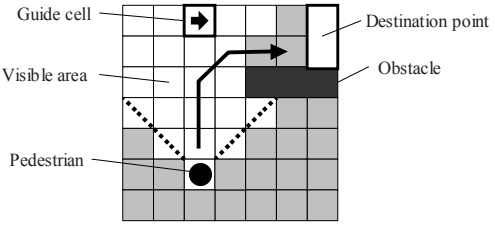



**Fig. 4.** Visible area set by field of view and guide cell

directions, as shown in Fig. 3. The actual moving direction will be determined by the status of the surrounding cells [6].

A real pedestrian walks while observing the surrounding environment to find their destination. With that in mind, we added a field-of-view model to the CA model in order to simulate acquisition of long-range information. In the field-of-view model, an area visible to the pedestrian (hereafter, "visible area") is defined in front of the pedestrian according to location and walking direction, as shown in Fig. 4. The pedestrian walks toward a target seen in the visible area. The target may be an exit, an elevator, an escalator, a flight of stairs or a room. As in the real world, the target is not visible if it is behind an obstacle. A guide cell is therefore introduced to lead pedestrians to the correct place. When a pedestrian finds a guide cell in their visible area, they walk in the direction indicated by the guide cell.

## 4 Elevator-Traffic Model

Elevators are usually arranged in several groups, and an appropriate elevator is selected from a group and allocated to a calling order from the elevator hall (hall call). This cage-allocation control, called "group control," avoids concentrating cages to a single hall call. Our simulation system applies such a group-control algorithm in its elevator-traffic model.

Several advanced algorithms for selecting an elevator cage from the elevator group have been developed; however, we adopted a simple algorithm. By means of this algorithm, the cage nearest to the floor of the hall call is selected. The distance between the cage and the floor is calculated in consideration of direction of cage motion and the location of the hall call.

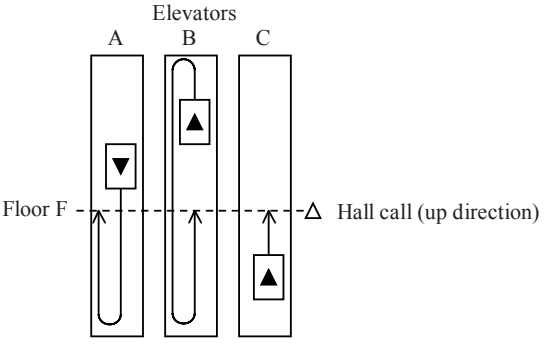



**Fig. 5.** Logical distances between cages and floor of hall call

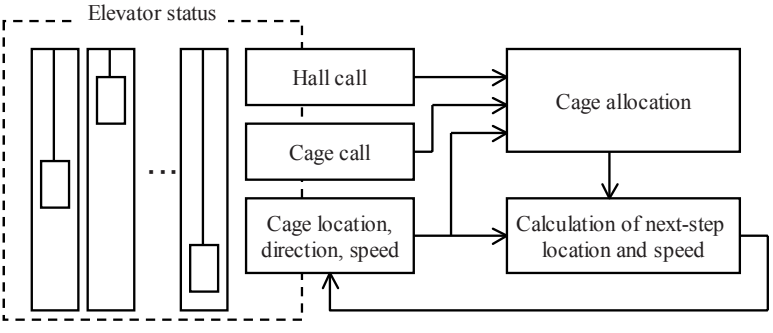



**Fig. 6.** Block diagram of elevator-traffic calculation

For example, in the case shown in Fig. 5, elevator cage A is nearest to the floor of the hall call in terms of physical distance; however, an additional "turn-around distance" is taken into account in the logical distance, because the direction of cage A's current motion is different from the direction requested by the hall call (hereafter, hall-call direction). As for cage B, the turn-around distances in the upper part and the lower part of the hoistway are taken into account. As a result, cage C is selected as the nearest cage. These turn-around distances are taken into account in order to avoid frequent changes of cage direction (which may confuse the passengers inside a cage).

Figure 6 shows the block diagram of elevator-traffic calculation. The status of each elevator group is defined by status variables, namely, "hall call," "cage call" (destination floors set by pushing floor buttons inside the cage), and "location and speed of each cage". The locations and speeds of the cages are calculated in the "next-step calculation" from current locations and speeds at certain time intervals synchronized with the CA calculation of floor flows. If there is a hall call, a cage is selected by the cage-allocation algorithm, as mentioned above, and assigned to the hall call. When the cage is approaching the assigned floor of the hall call or the cage call, the cage decreases its speed and stops at that floor. The

hall call is set when a pedestrian whose destination is different from the current floor destination enters the elevator hall.

## 5 Pedestrian-Motion Model for an Elevator Hall

It takes a certain amount of time for pedestrians to get on or off an elevator. Accumulation of this "loading/unloading time" on every floor becomes considerable. To take account of the time loss due to loading/unloading, we implemented a model of pedestrian motion in an elevator hall.

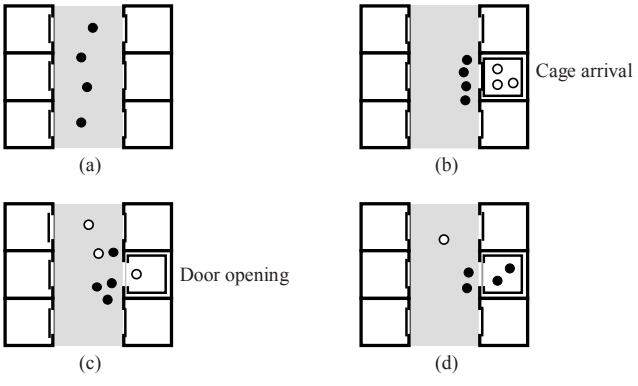



**Fig. 7.** Four steps of a pedestrian-motion sequence in an elevator hall

Pedestrian motion in an elevator hall consists of the sequence of four steps shown in Fig. 7. When pedestrians are waiting for the elevator, they spread apart in the elevator hall. When an elevator cage arrives or the arrival lamp lights up, the pedestrians gather in front of the arriving elevator. After the elevator doors open, the pedestrians inside the cage get off the elevator, and the waiting pedestrians in the hall get on.

The locations of the pedestrians in the elevator hall are controlled by an attractive force. By changing the force parameter, a spread distribution in the waiting phase and a condensed distribution in the cage-arrival phase are generated. In regards to the getting on/off actions, to avoid unusually quick getting on/off actions, the minimum time interval between a pedestrian and the following one is regulated by a delay parameter.

## 6 Verification

To verify the accuracy of the pedestrian-flow simulation system, we compared the simulation results to measured flows in real buildings. That is, we investigated pedestrian flows around the start of office hours at two office buildings. General

**Table 1.** Overviews of investigated buildings

| Building | A | B |
|---|---|---|
| Location | Tokyo | Osaka |
| Stories | 20 | 14 |
| Estimated capacity (persons) | 3,000 | 1,500 |
| Number of elevators | 12 (2 banks) | 5 (1 bank) |

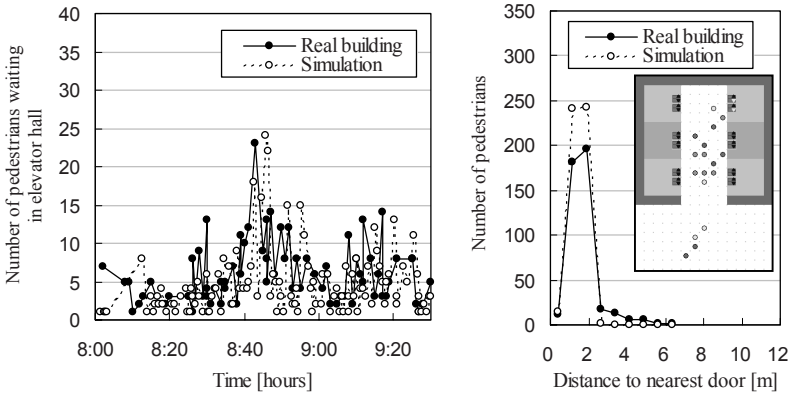



**Fig. 8.** Verification result on building A

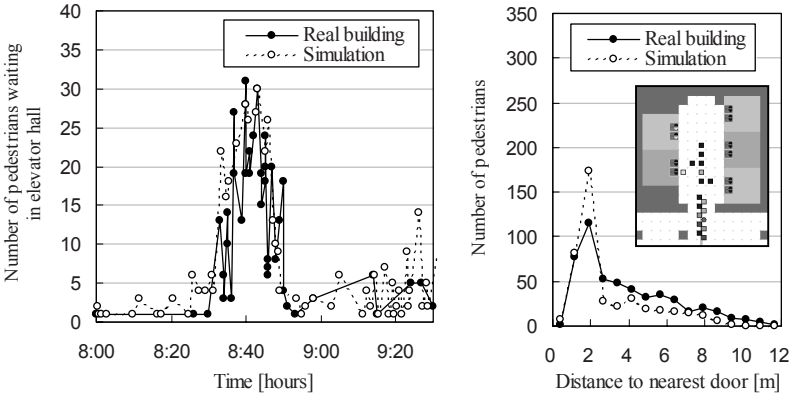



**Fig. 9.** Verification result on building B

details regarding the buildings are given in Table 1. In the simulations, the inflow data was set equivalent to the real inflow of pedestrians physically counted at the entrance of the buildings. Since the interaction effect between the floor flow

and the elevator traffic appears directly in the number and the distribution of pedestrians waiting at the elevator hall, we compared these two factors determined by the simulation and by measurements taken in the buildings.

Figure 8 compares the simulation results and real-building data on building A. The number of pedestrians waiting in the elevator hall on the ground floor, shown in Fig. 8(a), in both sets of data agree well in terms of general trend and peak number. The simulated and real-building distributions of pedestrians in the elevator hall, shown in Fig. 8(b), also agree quite well.

Figure 9 compares the simulation results and real-building data on building B. The number and distribution of pedestrians waiting in the elevator hall also agree well in case of this building. In this case, pedestrians formed a line in the elevator hall during the peak-inflow period; the line formation appears as the long tail to the graph in Fig. 9(b). This "line formation" phenomenon was similarly generated by the simulation.

## 7 Summary

A three-dimensional pedestrian-flow simulation system for evaluating the interaction between floor traffic and vertical elevator traffic in a high-rising building was developed. With this simulation system, a pedestrian-flow simulation based on a CA model is synchronized with an elevator-traffic model. The simulated pedestrian flows were compared with measured flows in the building in question, and the good agreement between the simulated and real pedestrian flows verify the effectiveness of the simulation system.

## References

1. Okazaki, S., Matsushita, S.: A Study of Simulation Model for Pedestrian Movement with Evacuation and Queuing. In: Proceeding of the International Conference on Engineering for Crowd Safety, pp. 271–280 (1993)
2. Burstedde, C., et al.: Simulation of Pedestrian Dynamics Using a Two-dimensional Cellular Automaton. Physica A 295, 507–525 (2001)
3. Siikonen, M.-L.: Elevator Traffic Simulation. Simulation 61(4), 257–267 (1993)
4. Fujino, A., et al.: An Elevator Group Control System with Floor-attribute Control Method and System Optimization Using Genetic Algorithms. Trans. on Industrial Electronics 44(3), 546–552 (1997)
5. Morishita, S., Nakatsuka, N.: Simulation of Emergency Evacuation by Cellular Automata. In: Proceedings of 6th International Conference on Complex Systems, pp. 92–97 (2002)
6. Morishita, S., Shiraishi, T.: Evaluation of Billboards Based on Pedestrian Flow in the Concourse of the Station. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 716–719. Springer, Heidelberg (2006)

# Compartment Lines Forming Emergent Alternative Configurations of Vehicles on Weaving Sections

R. Nishi[1], H. Miki[2], A. Tomoeda[1], and K. Nishinari[1,3]

[1] Dep. Aeronautics and Astronautics Engineering, University of Tokyo, Japan
{tt076379,tt67055,tknishi}@mail.ecc.u-tokyo.ac.jp
[2] SAKURA ACADEMIA Co., Japan
[3] PRESTO, Japan Science and Technology Corporation, Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656, Japan

**Abstract.** Drawing Compartment lines on weaving sections is planed for easing the traffic congestion through forming emergent alternative configurations of vehicles before weaving. The degree of the alternative configurations was quantified in weaving sections partitioned into identical cells. A new stochastic Cellular Automaton (CA) model was ruled for the dynamics along the line with the interactions between vehicles neighboring each other. The longitudinal formation of the zigzag pattern along the line was obtained for the first time through simulations and the four cluster approximation. Furthermore, both results fit to each other.

## 1 Introduction

These days, traffic dynamics has been studied more and more diligently [1] [2]. In particular, scientists have been interested in interactions among vehicles during merging and weaving. H.Kita modeled merging interactions with game theory [3]. P.Hidas investigated vehicle interactions in merging and weaving by using agent based simulations [4]. L. C. Davis introduced the cooperation in merging by adding interactions between pairs of vehicles in opposite lanes [5]. He showed that velocity of vehicles in cooperative merging was higher than that in no cooperation.

However, these previous works did not propose concrete plans to realize communications among vehicles in merging and weaving. One of the studies regarding plans to communications is the study of internal-vehicle communication device (IVC). Y.Ikemoto *et al.* proposed a self-control method of cooperative behavior in junctions with IVC [6]. His method used a local communication with a close-by vehicle without infrastructures. But IVC technology has serious problems in holding communication networks when a vehicle in a network crosses vehicles in other networks.

The purpose of this paper is to propose a concrete plan easily applicable to real weaving sections to ease the traffic congestion through communications among vehicles. This plan is drawing a compartment line forbidding changing lanes on

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 425–432, 2008.

the center of merging area of weaving sections. The line lets vehicles to communicate with the neighboring ones and arranges emergence of the alternative configurations of vehicles. These configurations lead to the zipper-like merging and lessen the disturbance caused by the lane-change.

This paper focuses on the emergent alternative configurations along the line before lane-changes by Cellular Automaton (CA). We made a new traffic CA model named Multi Lanes Stochastic Optimal Velocity (MLSOV) model. This is extended from Stochastic Optimal Velocity (SOV) model [7] with the interactions between a vehicle and the neighboring one. We quantified the degree of the zigzag pattern named *Geminity* (*Ge*) and obtained for the first time the longitudinal increase of *Ge* along the line by simulation and the four cluster approximation.

This paper is organized as follows. In Sec. Ⅱ we explain our plan of the compartment line in detail. In Sec. Ⅲ we make models of weaving sections along the line for simulations and the four cluster approximation. Sec. Ⅳ is the results of the simulations and the four cluster approximation and Sec. V is the conclusive discussion.

## 2 Compartment Lines on Weaving Sections

Weaving Sections are composed of two roads connected with each other as figure 1 (a). They have a merging area and a bifurcation area. They are used for *Kosuge Junction* of *Metropolitan Expressway Co., Ltd.* in Japan and other roads. Vehicles intending to move from one road to the other need to change lanes across the center line of them. Today, there is heavy traffic congestion in them. For instance, the average speed is less than 20 km/h at the upstream area of *Kosuge Junction* from 10 am to 12 am on weekdays in November, 2005 while it is more than 30 km/h at the downstream area. This congestion is caused by lane-changes of vehicles across the center line of it. Vehicles changing lanes disturb the movement of the following vehicles.

We propose a concrete plan to solve this problem. Our plan is drawing a compartment line forbidding lane-change between the center two lanes as figure 1 (b). The key factor of this plan is the emergent zigzag patterns of vehicles along the line. Vehicles on a lane and vehicles on the opposite lane see each other along the line. And they move away from each other to change lanes smoothly. This moving away of each vehicle accumulates and forms a zigzag pattern of vehicles. Vehicles in a zigzag pattern do not disturb following vehicles during changing lanes.

## 3 Modeling for Simulations and the Four Cluster Approximation

We prepared for simulations and the four cluster approximation to analyze the emergent alternative configurations along the compartment line as follows. (1) Modeling the center two lanes of weaving sections along a compartment line with CA. (2) Defining MLSOV model as the dynamics of vehicles along the line.

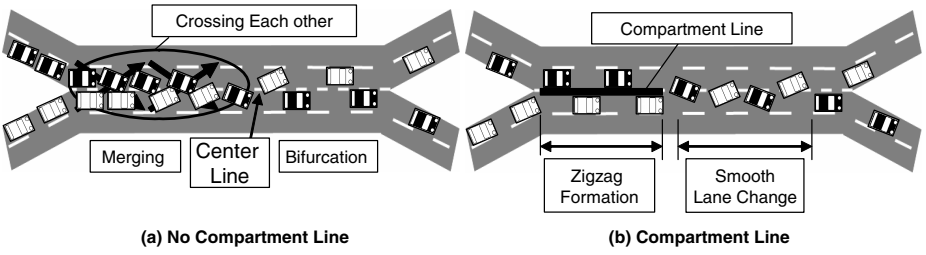



**Fig. 1.** (a) A weaving section without the compartment line. Vehicles disturb the movement of other vehicles by changing lanes across the center line. (b) A weaving section with the compartment line. The emergent zigzag pattern along the line leads to the smooth zipper-like lane-change.

(3) Defining the four cluster approximation. (4) Quantifying the degree of the alternative configurations in CA.

### 3.1 A CA Model of Weaving Sections along a Compartment Line

The center two lanes of weaving sections along a compartment line are partitioned into identical cells as figure 2. The cell size is fixed and each vehicle occupies a cell. The boundary condition of it is open and each vehicle is updated in parallel. Each Vehicle flows in the leftmost cell on lane 1 or lane 2, moves straight ahead along the line and flow out of the rightmost cell.

We set parameters of this model as figure 2. The length of lane 1, lane 2 and the line is $d$. The space coordinate $x$ is set as figure 2. The leftmost cells are at $x = 0$ and the rightmost cells are at $x = d - 1$. Each vehicle flows in $x = 0$ side by side with the neighboring vehicle with the probability $\alpha$ as long as both cells at $x = 0$ are empty. This artificial flow-in condition helps to evaluate clearly the forming zigzag pattern of vehicles at $x \geq 0$. The probability of flowing out from the rightmost cell of lane $i$ is $\beta_i$ $(i = 1, 2)$.

### 3.2 MLSOV Model as the Dynamics along a Compartment Line

We made a multi-lanes stochastic CA model with interactions of vehicles along the line. We named this model as Multi Lanes Stochastic Optimal Velocity (MLSOV) model. It is extended from Stochastic Optimal Velocity (SOV) model [7] which is a kind of single-lane stochastic models. We chose to extend SOV model because the fundamental diagrams of SOV model have the meta-stationary state seen in real traffic data. In both SOV model and MLSOV model, $i$-th vehicle at time $t$ moves straight forward one cell in one time step with probability $v_i^t$ as long as the next cell is empty. This movement is described as

$$x_i^{t+1} = \begin{cases} x_i^t + 1, & \text{with probability } v_i^t \\ x_i^t, & \text{with probability } 1 - v_i^t, \end{cases} \tag{1}$$

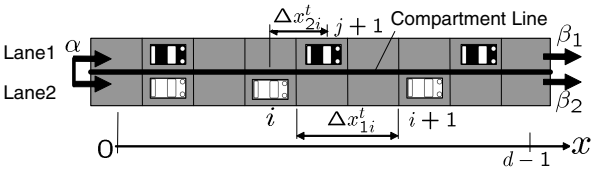



**Fig. 2.** CA model of a weaving section along the compartment line. Each vehicle flows in the cell on lane $i$ $(i = 1, 2)$ at $x = 0$ side by side with the neighboring vehicle with the probability $\alpha$, moves straight ahead and flow out of the cell at $x = d - 1$ with the probability $\beta_i$. $i+1$-th vehicle is the closest vehicle $\Delta x^t_{1i}$ cells ahead of $i$-th vehicle on the same lane. $j+1$-th vehicle is the closest vehicle $\Delta x^t_{2i}$ cells ahead on the neighboring lane.

where $x^t_i$ is the position of $i$-th vehicle at time $t$ as figure 2. $i + 1$-th vehicle is the closest vehicle $\Delta x^t_{1i}$ cells ahead of $i$-th vehicle on the same lane. $v^t_i$ is called intension and normalized as $0 \leq v^t_i \leq 1$.

In MLSOV model, the time evolution of $v^t_i$ is determined by not only $i+1$-th vehicle but also $j+1$-th vehicle which is the closest vehicle $\Delta x^t_{2i}$ cells ahead on the neighboring lane as figure 2. $v^t_i$ is given as

$$v^{t+1}_i - v^t_i = a\left\{V(\Delta x^t_{1i}, \Delta x^t_{2i}) - v^t_i\right\}, \tag{2}$$

where $V(\Delta x^t_{1i}, \Delta x^t_{2i})$ is the two-lanes Optimal Velocity (OV) function [8]. $v^t_i$ approaches $V(\Delta x^t_{1i}, \Delta x^t_{2i})$ with the response parameter $a$ $(0 \leq a \leq 1)$. Now, we set a simple OV function as

$$V(\Delta x^t_{1i}, \Delta x^t_{2i}) = \begin{cases} 0, & \Delta x^t_{1i} = 0 \\ r, & \Delta x^t_{1i} \geq 1 \text{ and } \Delta x^t_{2i} = 0 \\ q, & \Delta x^t_{1i} \geq 1 \text{ and } \Delta x^t_{2i} = 1 \\ p, & \Delta x^t_{1i} \geq 1 \text{ and } \Delta x^t_{2i} \geq 2. \end{cases} \tag{3}$$

This $V$ set each vehicle see only one cell ahead. We pick up typical two cases of the time evolution of $v^t_i$ with the initial condition of $v^0_i = p$; (a) $a = 0$ and (b) $a = 1$. In case (a), MLSOV model corresponds to the single-lane Asymmetric Simple Exclusion Process (ASEP). Each vehicle moves irrespective of neighboring ones and $v^t_i$ is given as

$$v^t_i = \begin{cases} 0, & \Delta x^t_{1i} = 0 \\ p, & \Delta x^t_{1i} \geq 1. \end{cases} \tag{4}$$

In case (b), MLSOV model corresponds to the two-lanes Zero Range Process (ZRP) and $v^t_i$ is given as

$$v^t_i = \begin{cases} 0, & \Delta x^t_{1i} = 0 \\ r, & \Delta x^t_{1i} \geq 1 \text{ and } \Delta x^t_{2i} = 0 \\ q, & \Delta x^t_{1i} \geq 1 \text{ and } \Delta x^t_{2i} = 1 \\ p, & \Delta x^t_{1i} \geq 1 \text{ and } \Delta x^t_{2i} \geq 2. \end{cases} \tag{5}$$

### 3.3 The Four Cluster Approximation

We prepared for analyzing the zigzag patterns along the line with MLSOV model through not only simulations but also the four cluster approximation. There are 10 kinds of state numbered $S(n)$ $(n = 1, 2, ...10)$ in the four cells at $x = \{k, k+1\}$ $(i \geq 0)$ as figure 3. The symmetry between lane 1 and lane 2 is considered. We defined $\Pi(n)_{kt}$ $(n = 1, 2, ...10)$ as the probability of the state $S(n)$ at time $t$. $\Pi(n)_{kt}$ is normalized as $\sum_1^{10} \Pi(n)_{kt} = 1$. The time evolution of $\Pi(n)_{k_t}$ is described as

$$\Pi_{kt+1} = P_k \Pi_{kt}, \tag{6}$$

where $\Pi_{kt} = \{\Pi(1)_{kt}, \Pi(2)_{kt}, \ldots, \Pi(10)_{kt}\}$ and $P_k$ is the state transition matrix at $x = \{k, k+1\}$. $\Pi_{k\infty}$ is given as the solution of $\Pi_{k\infty} = P_k \Pi_{k\infty}$ with the normalized condition of $\sum_1^{10} \Pi(n)_{k\infty} = 1$.

The hopping probability of MLSOV model is updated in the positive $x$ direction in making $P_k$. We assume the flow of vehicles along the line to be free flow and replace the time update of MLSOV model with the spatial update. We defined $v_i^k$ as the hopping probability of vehicle $i$ on the four cells at $x = (k, k+1)$. $v_i^k$ is given as

$$\begin{aligned} v_i^k &= (1-a)\tilde{v}^k + aV(\Delta x_{1i}, \Delta x_{2i}) \\ &= \begin{cases} (1-a)\tilde{v}^k, & \Delta x_{1i} = 0 \\ (1-a)\tilde{v}^k + ar, & \Delta x_{1i} \geq 1 \text{ and } \Delta x_{2i} = 0 \\ (1-a)\tilde{v}^k + aq, & \Delta x_{1i} \geq 1 \text{ and } \Delta x_{2i} = 1 \\ (1-a)\tilde{v}^k + ap, & \Delta x_{1i} \geq 1 \text{ and } \Delta x_{2i} \geq 2, \end{cases} \end{aligned} \tag{7}$$

where $\tilde{v}^k$ is the intension common to the four cells at $x = (k, k+1)$. $\tilde{v}^k$ is updated in the positive $x$ direction as

$$\tilde{v}^{k+1} = (1-a)\tilde{v}^k + a\bar{V}^k, \tag{8}$$

where $\bar{V}^k$ is the mean OV function at $x = k$ determined by the stationary state $\Pi_{k\infty}$. $\bar{V}^k$ is given as

$$\bar{V}^k = \frac{p\Pi(3)_{k\infty} + q\Pi(6)_{k\infty} + 2r\Pi(7)_{k\infty} + r\Pi(9)_{k\infty}}{\Pi(3)_{k\infty} + \Pi(6)_{k\infty} + 2\Pi(7)_{k\infty} + \Pi(9)_{k\infty}}. \tag{9}$$

$P_k$ is determined by the left boundary condition at $x = \{k-1, k\}$ and the right boundary condition at $x = \{k+1, k+2\}$. The left boundary condition in determining $P_k$ is given strictly. The left boundary condition in determining $P_0$ is given as a pair of vehicles flowing in both cells at $x = 0$ with the initial hopping probability $\tilde{v}^0$. They flow in with the probability $\alpha$ as long as both cells are empty. The left boundary condition in determining $P_k$ $(k \geq 1)$ is the stationary

state at $x = (k-1, k)$. This state is given strictly by $\mathit{\Pi}_{k-1\infty}$. However, the right boundary condition in determining $P_k$ needs approximation. The right boundary condition in determining $P_0$ is approximated as a pair of vehicles existing on both cells at $x = 2$ with the probability $\alpha/(1+\alpha)$. The right boundary condition in determining $P_k$ $(k \geq 1)$ is the stationary state at $x = (k+1, k+2)$. This state is not yet calculated and is approximated by $\mathit{\Pi}_{k-1\infty}$ which is the stationary state at $x = (k-1, k)$.

### 3.4 Quantifying the Degree of the Alternative Configurations in CA

We quantified the degree of the alternative configurations of vehicles named *Geminity* (*Ge*). *Ge* is a function of $x$ and $Ge(k)$ denotes the degree of the zigzag pattern of vehicles at $x = k$.

In simulations, $Ge(k)$ is calculated by counting the states of the four cells at $x = \{k, k+1\}$ through $M$ times of simulations with time steps of $0 \leq t \leq T$. We defined $c(n)_i$ $(n = 1, 2, ...10)$ as the total counted times of $S(n)$ at $x = \{k, k+1\}$. When there is at least one vehicle at $x = k$, the state of the four cells at $x = \{k, k+1\}$ can be $S(n)$ $(n = 3, 5, 6, 7, 8, 9, 10)$. Only $S(3)$ among them represents the perfect alternative state of vehicles obeying (1)-(3). Thus, $Ge(k)$ is given by $c(n)_k$ as

$$Ge(k) = c(3)_k / \big(c(3)_k + c(5)_k + c(6)_k + c(7)_k + c(8)_k + c(9)_k + c(10)_k\big). \quad (10)$$

In the four cluster approximations, $Ge(k)$ is also described as

$$\begin{aligned} Ge(k) = \mathit{\Pi}(3)_{k\infty} / (&\mathit{\Pi}(3)_{k\infty} + \mathit{\Pi}(5)_{k\infty} + \mathit{\Pi}(6)_{k\infty} \\ &+ \mathit{\Pi}(7)_{k\infty} + \mathit{\Pi}(8)_{k\infty} + \mathit{\Pi}(9)_{k\infty} + \mathit{\Pi}(10)_{k\infty}). \end{aligned} \quad (11)$$

*Ge* ranges from 0 to 1. The large value of $Ge(k)$ denotes that the zigzag pattern of vehicles at $x = k$ is highly achieved.

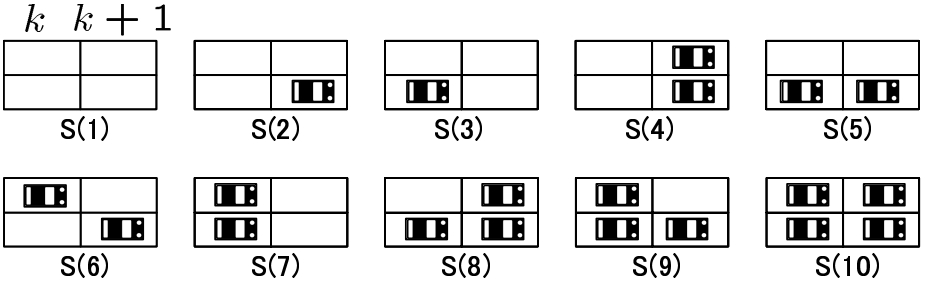



**Fig. 3.** 10 kinds of the state in the four cells at $x = \{k, k+1\}$. The symmetry between lane 1 and lane 2 is considered.

## 4 The Results of the Simulations and the Four Cluster Approximation

We measured the relationships between $Ge(x)$ and $x$ with simulations of MLSOV model and with the four cluster approximation.

The conditions common to the simulations and the four cluster approximation are shown as follows. The length of the road is given as $d = 100$. The parameters of MLSOV model are given as $p = 0.999$, $q = 0.8$ and $r = 0.8$. Five kinds of the response parameter $a$ are given as $a = \{0, 0.001, 0.01, 0.1, 1\}$. The probability of flowing in the cells at $x = 0$ is given as $\alpha = 0.05$. Vehicles flow in with the initial intension $p$. The conditions of the simulation are shown as follows. $\beta_i$ $(i = 1, 2)$ is given as the $v_i^t$ of each vehicle on the cell at $x = d - 1$. The times of simulations in the same condition is given as $M = 10$. The total time of a simulation is given as $T = 100000$ steps.

The results of the simulations and the four cluster approximation are shown as figure (4). $Ge(x)$ increased monotonically with $x$ in all cases of simulation and the four cluster approximation. This relationships between $Ge(x)$ and $x$ showed for the first time the achievement of zigzag pattern toward the space axis.

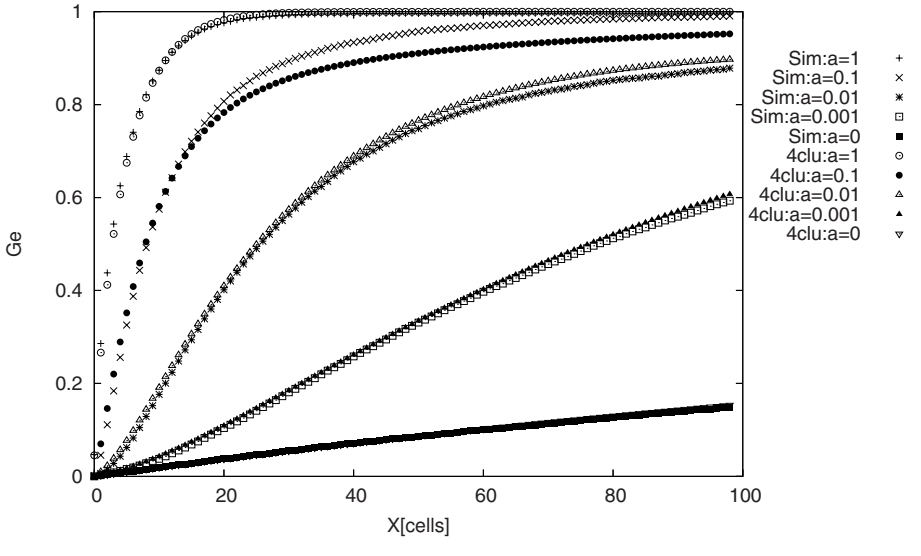



**Fig. 4.** $Ge(x)$ *vs.* $x$ of the simulations and the four cluster approximation. The length of the road is given as $d = 100$. The probability of flowing in the cells at $x = 0$ is given as $\alpha = 0.05$. Vehicles flow in with the initial intension $p$. The parameters of MLSOV model are given as $p = 0.999$, $q = 0.8$ and $r = 0.8$. $a$ is given as $a = \{0, 0.001, 0.01, 0.1, 1\}$. In simulations, $\beta_i$ $(i = 1, 2)$ is given as the $v_i^t$ of each vehicle at $x = d - 1$. The times of simulations in the same condition is given as $M = 10$. The total time of a simulation is given as $T = 100000$ steps.

The sharpness of the increase of $Ge(x)$ was positively correlated with $a$. $a$ is the strength of the response of each vehicle toward the forward vehicle on the opposite lane. This positive correlation suggests that the more strong the response of each vehicle toward the forward vehicle on the opposite lane, the more strong the achievement of the alternative configurations becomes.

The increase of $Ge(x)$ in the four cluster approximation fit to the increase of $Ge(x)$ in the simulation. This correspondence suggests that the four cluster approximation is a good theoretical approximation to evaluate the forming of zigzag pattern toward the space axis.

## 5 Conclusive Discussions

We proposed a concrete plan to relieve the traffic congestion on weaving sections by drawing a compartment line. The key point of our plan is the emergence of alternative configurations of vehicles. We focused on the alternative configurations of vehicles along the line before lane-change. We make a CA model of a weaving section along the line and analyzed these configurations with MLSOV model through simulations and the four cluster approximation. We quantified the degree of the zigzag pattern at a cell named *Geminity* ($Ge$). We obtained for the first time the achievement of alternative configurations toward the space axis through the simulations and the four cluster approximation. And it is found that the sharpness of the increase of $Ge(x)$ became strong with the increase of $a$. The results of the four cluster approximation approximated closely to the results of simulation of MLSOV model.

## References

1. Helbing, D.: Rev. Mod. Phys. 73 1067 (2001)
2. Chowdhury, D., Santen, L., Schadschneider, A.: Phys. Rep., 199–329 (2000)
3. Kita, H.: Transportation Research Part A 33(3), 305–312 (1999)
4. Hidas, P.: Transportation Research Part C 13, 37–62 (2005)
5. Davis, L.C.: Physica A: Statistical Mechanics and its Applications. 361(2), 606–618 (2006)
6. Ikemoto, Y., Hasegawa, Y., Fukuda, T., Matsuda, K.: In: Proceedings of the 2004 IEEE International Conference on Robotics 8 Automation, ew Orleans, LA (April 2004)
7. Kanai, M., Nishinari, K., Tokihiro, T.: Phys. Rev. E. 72, 035102(R) (2005)
8. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Phys. Rev. E 51, 1035 (1995)

# Real-Time Railway Network Simulator "KUTTY"

A. Tomoeda[1], M. Komatsu[1], I.Y. Yoo[1], M. Uchida[2],
R. Takayama[3], and K. Nishinari[1,4]

[1] Dep. Aeronautics and Astronautics Engineering, University of Tokyo, Japan
[2] Dep. Environmental and Ocean Engineering, University of Tokyo, Japan
[3] Dep. Mechanical Engineering, University of Tokyo, Japan
[4] PRESTO, Japan Science and Technology Corporation
tt67055@mail.ecc.u-tokyo.ac.jp

**Abstract.** In the railway transportation service, on-time operation of trains is quite important for customers. However, once an accident on a route occurs, on-time operation becomes difficult due to a change in the flow of passengers which is caused by the interruption. Thus, the real-time railway simulation tool plays a significant role to estimate the changing flow of passengers. In this paper, we propose real-time Tokyo Metro Railway Network simulation tool "KUTTY" and by using this simulator we can immediately estimate a change in the flow of passengers even if an accident occurs at the busiest area. Moreover, we have found that our homogenization system eases congestion in crowded area more than the conventional method.

## 1 Introduction

A traffic disturbance of trains is caused nearly everyday and gives us a lot of stress. This traffic disturbance is attributed to several causes, e.g. the slight delay of trains and traffic accidents. In these troubles, how does the train company adjust and operate trains? The conventional method of rescheduling the timetable is that the specialist, who is called "Sujiya", reschedules the timetable by his own empirical rule. This conventional method is not performed in units of network, but rather units of line. Furthermore, this adjustment focuses on the distribution that trains are equally-spaced without the flow of passengers. Recently, many and various kinds of simulations and models on the train network as complex systems have been studied [1,2,3,4], but few simulation models take into account the behaviors of passengers. In order to create models with these specific behaviors a cellular automaton (CA) model [5,6,7] is widely implemented and very useful in computer simulations. Since the CA model is described by rule-based system and has discrete values, it easily represents the complicated behaviors and do computer simulations in comparison with the models described by a partial differential equation.

In Ref. [7], we proposed the public conveyance model (PCM) which represented the bus clustering and estimated the efficiency of the bus system on

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 433–440, 2008.

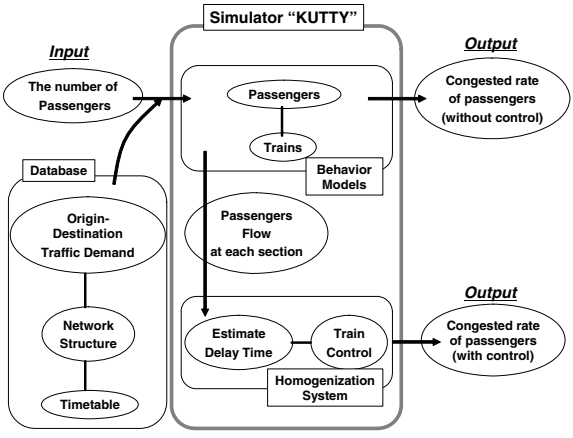



**Fig. 1.** The design flow diagram of our simulator "KUTTY"

one-dimensional route. In this PCM, passengers arrive at a stop randomly and their destinations does not consider explicitly. Furthermore, since buses run on a one-dimensional route, passengers are not allowed to choose the route. Thus, in this paper, we have expanded this PCM by introducing realistic passengers' behaviors explicitly and proposed the real-time railway network simulation tool "KUTTY" which is applied to the *Tokyo Metro Railway Network*. Moreover, we have studied the congested rate of passengers in each segment by incorporating the homogenization system which is based on the distribution of passengers.

This paper is organized as follows: in Sec. 2 the handling of the network structure in Tokyo Metro Railway Network is proposed and the Origin-Destination (OD) traffic demand is estimated. In Sec. 3 we explain the models of passengers and trains which are implemented in our simulator and several simulation results is shown in Sec. 4. Sec. 5 is devoted to conclusions and discussions.

## 2 Database

Our real-time simulator "KUTTY" is operated by extracting the data from a database besides the input data (Fig. 1). At first in this section we present the assembly of a database and explain about its components: *network structure* and *OD traffic demand*.

### 2.1 Network Structure

Basically, the static structure of a train network is composed of three elements:*station*, *segment* (connecting couples of stations on a route) and *link* (connecting couples of transfer stations). In order to deal with the Tokyo Metro Railway Network as a complex system, we use the graph theory [8] with $N$ nodes

and $K$ edges. In the graph theory, a *station* is "node", a *segment* and a *link* are "edge", and the travel time of one segment is expressed as "weighted edge". The Tokyo Metro Railway Network consists of 9 lines and 138 stations. This network is considered as a graph with $N = 169$ nodes and $K(= K_1 + K_2) = 220$ edges ($K_1 = 170$ segments and $K_2 = 50$ links). We define that the each node corresponds one-to-one with the ID number of each station of every line. That is, even if the names of two stations are the same, we consider these nodes are different (e.g. "Otemachi" of Tozai Line (T09) and "Otemachi" of Marunouchi Line (M18) are different nodes). Thus, the changing lines of passengers is also expressed as one edge between a couple of transfer stations even as the same named stations. The connection of every $N$ is represented by adjacency matrix $A = \{a_{ij}\}$, that is, if an element $a_{ij}$ of the $N \times N$ square matrix equals to 1, there is an edge between node $i$ and node $j$, whereas if $a_{ij} = 0$, the two nodes $i$ and $j$ do not connect.

As of today, passengers can travel from any stations in the Tokyo Metro Railway Network to all the other stations in this network by changing trains at most twice. Therefore, by restricting the changing trains at most twice, we decrease the calculation amount in searching for a possible route. Additionally, we have made the database of all paths by using *Dijkstra's Algorithm*[9].

### 2.2 Origin-Destination Estimation

The real data of OD traffic demand is indispensable for building a model of passengers. In general, it is very difficult to obtain this sort of data. However, the Tokyo Metro Company posts one-day ridership of all stations on its website. Moreover, at the mutual entry stations, the ridership includes not only the number of embarking and disembarking passengers but also the influx and efflux of passengers which corresponds to the boundary condition for simulations. Thus, we easily obtained the ridership data of each station of a day.

Now, in order to divide the ridership data into two significant data, which correspond to the number of embarking passengers and disembarking passengers, and to estimate the OD traffic demand from this ridership data, we set the following three assumptions:

1. The number of embarking passengers is assumed as half the number of the ridership at every station.
2. If one arrives at the system, one has a pair of stations of origin and destination which is prorated by the ridership of each station.
3. Passengers can not get off at the station where they got on.

Note that, the first assumption is motivated by the common expectation that most of the people go back to the original station where they left, since there are their own houses. Under these three assumptions, we have calculated the number of disembarking passengers proportionally by using the ridership of each station.

From this OD estimation, we have found that the margin of error between the assumed number of embarking passengers and the calculated number of

disembarking passengers is considerably less than plus or minus 5 percent. This result provide conclusive evidence that we can use this estimation data for constructing the travel model of passengers.

## 3 Models and Homogenization System

In this section, we first explain the passengers' model based on the OD traffic demand, and secondly, we present the model of trains which is similar to the model of buses in PCM [7]. And then, we also present the homogenization system based on the passengers flow.

### 3.1 Model of Passengers

Now, let us explain the model of passengers in detail. When a passenger $p$ arrives at the network, one has a pair of stations, namely origin and destination, which is determined proportionally by the data of OD estimation. Then, one can easily obtain the set of all paths $S(p)$ to travel one's own destination and calculate all the costs of each path $s \in S(p)$ at a time $t$ by following cost function $E(p, s, t)$,

$$E\Big(T(p,s), C(p,s), D(p,s,t)\Big) = aT(p,s)^{\alpha} + bC(p,s)^{\beta} - cD(p,s,t)^{-\gamma}, \qquad (1)$$

where $a, b, c, \alpha, \beta, \gamma$ are parameters of positive value. In this formula, $T(p, s)$ and $C(p, s)$ are the total travel time and the total number of changing lines of the path $s$ respectively. $D(p, s)$ is the maximum density of trains and platforms through the path $s$. For example, the case of $a, \alpha > b, \beta$ represents that a passenger think the travel time is more important than the changing lines. Note that, it is not essential that the coefficient and the power in the third term of (1) are negative value. This is because in $D(p, s, t) \in [0, 1]$ we reflect that the third term increases as parameter $\gamma$ increases like other terms. The number of states $M(p, s, t)$ is defined as a function of $E(p, s, t)$.

$$M(p,s,t) = \exp\Big[-E(p,s,t)\Big] \qquad (2)$$

This formula is motivated by the fact that the number of states increases as the cost becomes low. Moreover, by normalizing the number of states, the probability $P(p, s, t)$, which is the probability distribution for a passenger $p$ to select a route $s$, is described as follows,

$$P(p,s,t) = \frac{M(p,s,t)}{\sum_{s_i \in S(p)} M(p,s_i,t)}. \qquad (3)$$

After one chose a path $s \in S(p)$ with this probability $P(p, s, t)$, one travels to the destination.

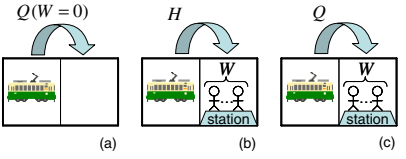



**Fig. 2.** All cases of the hopping probability of a train. (a) at a free cell (b) at the front cell of a station (first time) (c) at the front cell of a station (after one stop in the case of (b))

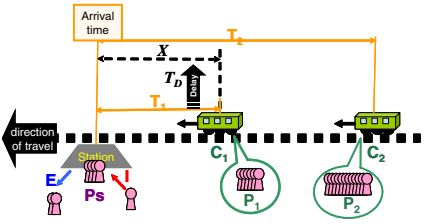



**Fig. 3.** Schematic view of our homogenization system

### 3.2 Model of Trains

Let us imagine that the each *segment* of all lines is partitioned into 4 identical cells and the each *station* is designated as 1 cell on the route such that each cell can accommodate at most one train at a time. The symbol $H$ is used to denote the hopping probability of a train entering into next cell and $W$ denotes the number of passengers waiting at a station at the instant of time when a train arrives there. The maximum carrying capacity of trains is represented by $W_{\max}$. Furthermore $Q$ is the hopping probability of a train to the next cells that are not designated as a station. Thus, we assume the form $H$ is

$$H = \min\Big(Q, \frac{1}{a\min(W, W_{\max}) + 1}\Big), \tag{4}$$

where $a$ is a parameter and in this paper we set $Q = 1.0, a = 0.2$ respectively. In contrast to the symbol $H$ which was used in the sense of the hopping probability itself in Ref. [7], this form $H$ is used in the sense of the probability concerning the delay, that is, a train can stay the same cell only once with probability $1-H$. After the stay, the train moves with probability $Q$ in the same way at a free cell. This is because differing from the bus system, trains are controlled in units of very small segment by the railway signaling system. Since it is not realistic that trains stop stochastically again and again, the delay of trains is expressed as the stochastic stop only once. Fig. 2 depicts the hopping probabilities in the train model schematically.

### 3.3 Homogenization System

In this section, we explain the homogenization system based on the passengers flow. To ameliorate the railway transportation, many train companies control the distance of successive trains as a common way. Specifically, if some trains are late, first of all, train companies try to fair the distance between every successive trains. However, in this adjustment, the distribution of passengers does not considered. In this study we have proposed another adjusting system based

on the passengers model which is previously stated. The feature of this system is that the number of passengers in each train is homogeneously-distributed by adjusting the distribution of trains. Moreover, we restrict that trains adjust their following distance by simply decelerating, since it is realistically difficult that trains accelerate for adjusting the distance of previous train due to their maximum velocity.

FIG. 3 shows the schematic view of our homogenization system. In this figure, $P_1$ and $P_2$ correspond to the number of boarding passengers on a train $C_1$ and $C_2$ respectively. $P_s$ and $E$ show the number of passengers who wait at a station and exit from a train. Symbol $I$ corresponds to the number of passengers who arrive at the station per unit time. Furthermore, the arrival time of $C_1$ and $C_2$ at the next station is $T_1$ and $T_2$. Our aim is that the number of passengers on $C_1$ and $C_2$ is homogenized by $T_1$ extended to $X$. We have obtained the extended amount of time $X$ from following equation,

$$P_1 + (P_s + IX) - E = P_2 + I(T_2 - X) - E. \tag{5}$$

Left-hand side (Right-hand side) of (5) is the number of passengers on $C_1$ ($C_2$) after departing from the next station. Using $X$ obtained from (5), the delay time $T_D$ of $C_1$ is decided by

$$T_D = f(X - T_1). \tag{6}$$

In this study, we simply consider that the train remains in the same place until $X - T_1 = 0$.

## 4 Simulations and Results

Fig. 4 shows two snapshots of our simulator "KUTTY" which displays the flow rate of the passengers of each segment. By simulating the flow of passengers

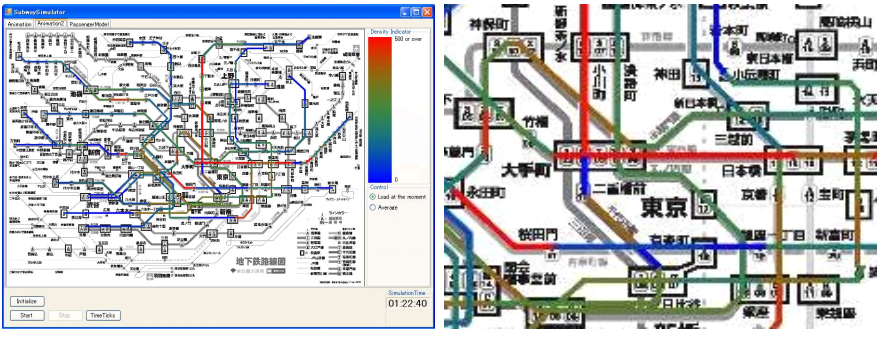
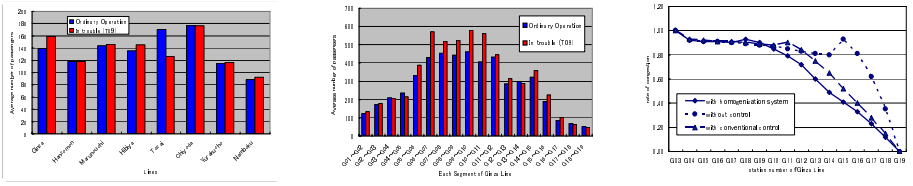



**Fig. 4.** Snapshots of our simulator "KUTTY". Left one is entire picture and right one is closeup picture around Otemachi station. In KUTTY, the high flow (low-flow) is painted in red (blue).




(a)Each line (b)Each segment of Ginza Line (c)Rate of congestion

**Fig. 5.** (a)The comparison plot of the number of passengers in each line between in ordinary operation and in trouble. (b)The comparison plot of the number of passengers in each segment of the Ginza Line between in ordinary operation and in trouble. (c)The comparison plot of the congestion rate among not controlled system, ordinary controlled system, and our homogenization system.

quantitatively in all segments of all over the network, we have found that the most congested area in Tokyo Metro Railway Network is Otemachi station of the Tozai Line (T09). Thus, in this paper, we have simulated the case where the accident occurs at Otemachi station of the Tozai Line. Under this situation, the flow of passengers will be changed significantly, since passengers can not pass the much-used area. This simulation is very important clues for adjusting trains, since the potential needs of users for the alternative transportation in troubles must be grasped.

Fig. 5 is the quantitative results of several simulations. The left figure shows that the number of passengers who take the Tozai Line decreases by about 25 percent from ordinary operation due to an accident at T09. By contraries, the number of passengers who take the Ginza Line and Hibiya Line increases by about 15 percent and 10 percent respectively. We have presumed that 25 percent of the ridership of Tozai Line mainly change trains to the Ginza Line or Hibiya Line and travel a long way round the Otemachi station, since the Otemachi station is interrupted. Next, we have simulated the number of passengers at all segments of Ginza Line. As shown the middle figure, in the area from G05 to G11, which correspond to the area from the Akasaka-mitsuke station to the Nihombashi station, the number of passengers discernibly increases. This result indicates that if the accident occurs at Otemachi station of the Tozai Line, passengers use the Ginza Line as an alternate route. Thus, we have obtained the changing flow of passengers in trouble. Moreover, if the flow of passengers increases suddenly like Ginza line noted above, the trains may not be operated on-time. In this situation, the trains should be controlled by some kind or another controlled system. In this paper, when passengers at G03 station of Ginza Line increases suddenly, we have compared the congestion rate, which is derived from (the number of passengers in trains)/(the capacity of trains), in three systems: not controlled system, ordinary controlled system, and our homogenization system. In the right figure, two controlled systems narrow the region in which the congestion rate of passengers is high. Moreover, we have found that our homogenization system

is superior than the ordinary controlled system in terms of the congestion rate. In ordinary controlled system the adjustment is made several times, whereas in our homogenization system the adjustment is made only once.

## 5 Conclusion

In this paper, we have proposed a network simulator "KUTTY", which is based on the behaviors of passengers. "KUTTY" could simulate the complicated network of the Tokyo Metro Railway Network, and use the real passenger distribution at different stations and lines provided by Tokyo Metro company. "KUTTY" can provide an estimation of the changed passenger flow soon once the accident occurs in the Tokyo Metro Railway Network.

From the simulation, where the Otemachi station of the Tozai Line is interrupted, we have found that in trouble the passengers choose their own optimum path and we have quantitatively estimated the flow of passengers. We have also simulated the effect of our homogenization system, in which the number of passengers in each train is homogeneously-distributed by considering the flow of passengers in each segment. As a result of this system, our homogenization system most reduced the congestion rate and surprisingly our system is made only once to adjust the distribution of trains

In future works, we would like to adjust trains simultaneously all over the network and reschedule the time-table. Finally we hope this simulator is applied any other network transportation system and will help in optimizing the operation.

## References

1. Pursula, M.: Simulation of traffic systems - an overview. Journal of Geographic Information and Decision Analysis 3, 1 (1999)
2. Latora, V., Marchiori, M.: Physica A 314, 109 (2002)
3. Watts, D.J., Strogatz, S.H.: Nature 393, 440 (1998)
4. Meignan, D., Simonin, O., Koukam, A.: Simulation Modeling Practice and Theory. 15, 659 (2007)
5. Helbing, D.: Rev. Mod. Phys. 73, 1067 (2001)
6. Chowdhury, D., Santen, L., Schadschneider, A.: Phys. Rep. 329, 199 (2000)
7. Tomoeda, A., Nishinari, K., Chowdhury, D., Schadschneider, A.: Physica A 384, 600 (2007)
8. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs (1993)
9. Dijkstra, E.W.: Numerische Mathematik 1, 269 (1959)

# Cellular Automata Simulation of Traffic Jam in Sag Section

Yukiko Wakita, Takuya Kuroda, and Eisuke Kita

Nagoya University, Graduate School of Information Sciences, Nagoya 464-8601, Japan

**Abstract.** A traffic congestion often occurs near a sag section on a freeway. Since a road gradient changes gradually in a sag section, a driver dose not notice it and therefore, a traffic congestion occurs.

This paper describes the cellular automata simulation of the traffic flow through a sag section. The simulation is performed by the stochastic velocity model[4]. The results show that the effect of a sag zone to the traffic congestion becomes strong according to the increase of car density and that average velocity tends to decrease not at the sag section but at a little ahead point of sag section.

## 1 Introduction

Since a traffic congestion strongly affects not only a traffic system but also environment, many researchers have been studying for solving the problems[1,2]. A traffic congestion is occurred by several reasons. A sag section, which is a bottleneck of road structures, is also one of the obvious and important reasons[3,5,6]. A road gradient changes slightly and gradually in a sag section. Since drivers often do not notice the slight change, car velocity decreases gradually and therefore, a traffic congestion occurs.

The effect of a sag section for the traffic congestion is discussed in this paper. For this purpose, one uses the cellular automata simulation. The simulation model is based on the stochastic velocity model[4]. In this model, the car velocity and the movement is defined with a stochastic variable. Since the maximum movable distance of vehicles is restricted to one cell alone, the local rules to control the car behavior can be simplified. In the numerical examples, one discusses the dynamic property of the traffic congestion near the sag section.

## 2 Simulation Model

### 2.1 Object Under Consideration

A road including a sag zone is illustrated in Fig.1. The cell representation of the road is shown in Fig.2. Since the cell size is $3m \times 3m$, the road length is 3333 cells ($9999m$). The sag zone is specified from a 2668th cell from left end to a rightmost cell. A open condition is specified at the right end. Cars go through the domain from the left end and go out from the right end. When cars go through

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 441–446, 2008.

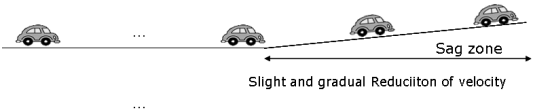



**Fig. 1.** Sag zone

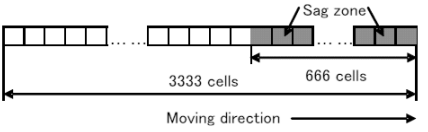



**Fig. 2.** One-lane road with sag zone

the sag zone, negative acceleration is applied to them. The negative acceleration $a_{sag} = 0.3(m/s^2)$ is defined as the average value of the them calculated from the gradients of the sag zones in the Tomei highway in Japan.

### 2.2 Local Rules

Cars are controlled according to the local rules. Cars go straight ahead or change a driving lane and therefore, the car behaviour local rules are only for going straight ahead and changing lanes. Addition to them, the velocity local rule is employed.

**Safety Car Distance.** Assume that cars move so that their distance from a forehand car is kept to be safety. According to Japanese automobile inspection certification manual, a safety car distance $G_{safe}$ is defined as the function of the car velocity $v$ as

$$G_{safe} = 0.15 \times v + 0.0097 \times v^2. \tag{1}$$

**Velocity Local Rule.** The velocity local rule change a car velocity according to a car distance. The parameter $v, v_{\max}, G_{safe}$ and $a$ denote the present car velocity, the feasible maximum velocity, the safety car distance and the acceleration of a car, respectively.

| | |
|---|---|
| Acceleration | If $v < v_{\max}$ and $G > G_{safe}$ then, $v \leftarrow v + a$. |
| Deceleration | If $v > 0$ and $G < G_{safe}$ then, $v \leftarrow v - b$ |

where the acceleration rate $a$ and the deceleration rate $b$ are estimated from the actual traffic data as $0.6 < a < 2.4$ and $1.2 < b < 3.0$, respectively.

**Local rule for going straight ahead**

1. A distance from a forehand car $G$ is estimated.
2. If $G < G_{safe}$, a car moves to a forehand cell at the next time-step.
3. If not so, a car stops at the present cell.

**Local rule for changing lanes**

1. A distance from a forehand car is estimated at the present lane; $G_1$.
2. A distance from a forehand car is estimated at the other lane; $G_2$.
3. If $G_1 < G_2$, a car changes to the other lane.

## 3 Numerical Results

### 3.1 Traffic Jam in Sag Zone

It is pointed out that the traffic congestion starts not just from a sag zone but from the slightly far point from the sag zone. When a cluster of cars reaches at a sag zone, a forehand car of the cluster is decelerated firstly. The velocity reduction propagates from the forehand car to the following cars, just like a shock wave. If the car density of the cluster is high enough, the traffic congestion occurs there. The frequency of the traffic congestion depends on the car density in the cluster.

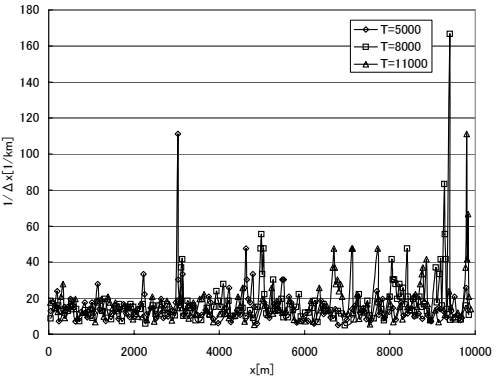
First, the effect of the car density $\alpha$ to a car position every five minutes is discussed in the numerical example for observing the development of a traffic congestion. The value $\alpha$ is defined as the rate of cars going into the object domain from the left end. The car density is defined as the inverse number of the car distance $\Delta x$. The traffic flows at $T = 5000, T = 8000$ and $T = 11000$ for different car density $\alpha$ are compared in Fig.3. The abscissa and the ordinate denote the car position and the car density, respectively.

We notice from the results at $T = 5000$ and $T = 8000$ that there is high car density between $8001m$ and $9999m$ from the left end. Next, we notice from results from $T = 8000$ to $T = 11000$ that there is also high car density between $6001m$ and $8000m$. These mean a traffic congestion occurs in the sag zone first and then, develops from the sag zone to the upstream of the sag zone. When a cluster of cars reaches a sag zone, a forehand car is decelerated and therefore, the car density is increased. If the density is higher, the traffic congestion occurs there.

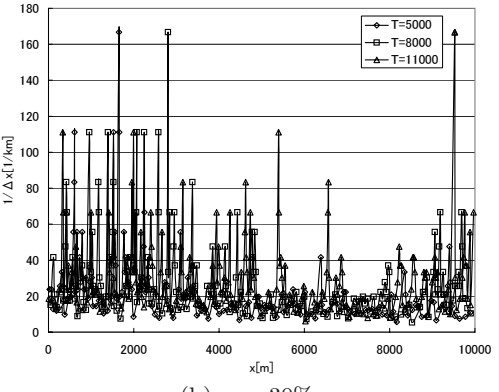
We will compare the traffic flows at difference values of $\alpha$. When focusing on the traffic congestion occurred between $6001m$ and $8000m$ at $T = 8000$ to $T = 11000$, we notice that car density increases according to the increase of the car density $\alpha$. This means that the traffic congestion strongly depends on the car density $\alpha$.

### 3.2 Behavior of a Car

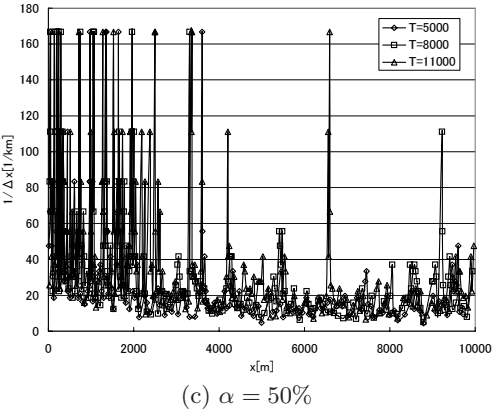
We will focus on the behavior of a 300th car going from the left end. The reason why the 300th car is taken is that it goes through all traffic congestions after traffic congestions develop enough. The rate of cars going from the left end is fixed as $\alpha = 10\%$.




(a) $\alpha = 10\%$




(b) $\alpha = 30\%$




(c) $\alpha = 50\%$

**Fig. 3.** Development of traffic congestion

The results are shown in Fig.4. The abscissa and the ordinate denote the car-head distance from the left end $\Delta x$ and the velocity $v$, respectively. We notice from Fig.4 that the curves are so-called hysteresis loops and that cars move between the

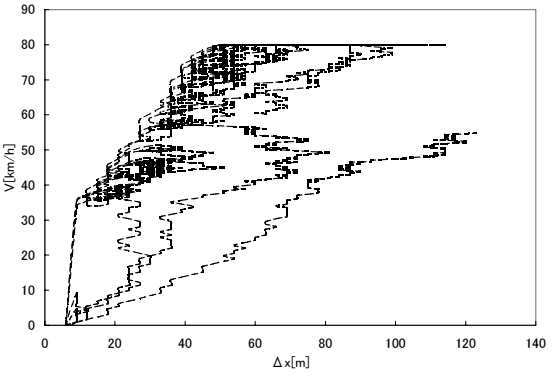



**Fig. 4.** Behavior of a car

high car density at the low velocity and the low car density at the high velocity. Besides, the profile of the hysteresis loop dose not depend on the value $\alpha$.

Next, we estimate the variation of the hysteresis loop at different traffic zones. The zones are taken as follows:

- Zone A: $0m$ to $4000m$ (at upstream which is far from sag zone),
- Zone B: $4001m$ to $6000m$ (at upstream which is near sag zone),
- Zone C: $6001m$ to $8000m$ (at upstream which is adjacent to sag zone), and
- Zone D: $8001m$ to $9999m$ (in sag zone).

The results are shown in Tables 1 to 3. The average velocity and the average absolute deviation of a car velocity $v$ are referred to as $v_{ave}$ and $v_{avedef}$, respectively. Besides, the average value and the average absolute deviation of a car-head distance $\Delta x$ are as $\Delta x_{ave}$ and $\Delta x_{avedef}$, respectively.

We notice the following points from Tables 1 to 3. In cases of $\alpha = 10\%$ and 30%, the values $v_{ave}$, $v_{avedef}$, $\Delta x_{ave}$, and $\Delta x_{avedef}$ in zone A are smaller than them in zone B. This means that the hysteresis loop in zone A is bigger than that in zone B. Although the traffic flow in zone A is instable, the flow is getting stable to the zone B and C if the value of $\alpha$ is small. However, also in zone B and C, the value $\Delta x_{avedef}$ in case of $\alpha = 50\%$ is bigger than that in the lower value of $\alpha$. This means that the traffic flow is instable during whole zone if the value of $\alpha$ is big enough.

**Table 1.** Variation of car velocity and car-head distance ($\alpha = 10\%$)

| Position | $v_{ave}$ | $v_{avedef}$ | $\Delta x_{ave}$ | $\Delta x_{avedef}$ |
|---|---|---|---|---|
| 0-4000$m$ | 76.27 | 5.61 | 70.954 | 15.40 |
| 4001-6000$m$ | 77.78 | 3.11 | 68.52 | 14.80 |
| 6001-8000$m$ | 78.48 | 0.00 | 74.70 | 13.97 |
| 8001-9999$m$ | 49.30 | 15.07 | 51.30 | 26.84 |

**Table 2.** Variation of car velocity and car-head distance ($\alpha = 30\%$)

| Position | $v_{ave}$ | $v_{avedef}$ | $head_{ave}$ | $head_{avedef}$ |
|---|---|---|---|---|
| 0-4000$m$ | 50.12 | 15.80 | 40.32 | 16.77 |
| 4001-6000$m$ | 54.82 | 11.53 | 42.96 | 14.73 |
| 6001-8000$m$ | 55.75 | 12.79 | 42.27 | 16.74 |
| 8001-9999$m$ | 40.98 | 9.02 | 34.19 | 14.82 |

**Table 3.** Variation of car velocity and car-head distance ($\alpha = 50\%$)

| Position | $v_{ave}$ | $v_{avedef}$ | $head_{ave}$ | $head_{avedef}$ |
|---|---|---|---|---|
| 0-4000$m$ | 36.88 | 16.35 | 29.20 | 13.09 |
| 4001-6000$m$ | 53.54 | 14.26 | 45.17 | 15.79 |
| 6001-8000$m$ | 51.39 | 16.41 | 51.21 | 31.13 |
| 8001-9999$m$ | 51.95 | 18.81 | 54.83 | 26.77 |

## 4 Conclusions

In this study, the traffic flow simulation is performed in order to estimate the effect of the sag zone to the development of a traffic congestion.

When there exists a sag zone in the road, a traffic congestion occurs in the sag zone and the congestion develops to the upstream of the sag zone. The sag zone affects the traffic congestion more strongly according to the increase of cars. The curve of the car-head distance and the velocity is hysteresis loop. In case of low traffic density, the traffic flow is instable at the far upstream point from the sag zone and then, become stable to the point near the sag zone. However, in case of high traffic density, the traffic flow is instable during whole road.

In the future, we would like to estimate the effect of the sag zone to traffic flows on the cruising and the passing lanes of multi-lane road.

## References

1. Kerner, B.S.: Phase transitions in traffic flow. In: Helbing, D., Herrmann, H.J., Schrekenberg, M., Wolf, D.E. (eds.) Traffic and Granular Flow 1999 – Social, Traffic and Granular Dynamics, pp. 253–284. Springer, Heidelberg (1999)
2. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal of Physics I France 2, 2221–2229 (1992)
3. Tadaki, S., Nishinari, K., Kikuchi, M., Sugiyama, Y., Yukawa, S.: Observation of congested two-lane traffic caused by a tunnel. Journal of the Physical Society of Japan 71(9), 2326–2334 (2002)
4. Tamaki, T., Yasue, S., Kita, E.: Traffic flow simulation using cellular automata. In: Proceedings of Civi and Structural Engineering Computing (2003)
5. Yamamoto, S., Hieda, Y., Tadaki, S.: Effects of bottlenecks on vehicle traffic. Journal of the Physical Society of Japan 75(11), 114601–114605 (2006)
6. Yukawa, S., Kikuchi, M., Tadaki, S.: Dynamical phase transition in one-dimensional traffic flow model with blockage. Journal of the Physical Society of Japan 63(10), 3609–3618 (1994)

# Simulation of Fire Evacuation by Real-Coded Cellular Automata (RCA)

Kazuhiro Yamamoto[1,*], Satoshi Kokubo[1], Hiroshi Yamashita[1], and Katsuhiro Nishinari[2]

[1] Dep. Mechanical Science and Engineering, Nagoya University, Japan
[2] Dep. Aerospace Engineering, University of Tokyo, Japan
kazuhiro@mech.nagoya-u.ac.jp

**Abstract.** We present simulation of fire evacuation by real-coded Cellular Automata (RCA), which is a new approach for pedestrian dynamics. Here, we consider the evacuation from a relatively large room with one or two exits. To describe the flame spread in fire, a percolation model is applied, where the flame position is determined stochastically. In the simulation, we focus on several parameters including the number of people in room, the distance from the flame, and the location and size of the exit.

## 1 Introduction

Fire is one of the most serious disasters. The damage in fires is mainly caused by high heat fluxes from the flame, accidental explosions, and toxic species in smoke generated by combustion reaction, which causes fatalities, destruction of houses and buildings, and air pollution [1]. In order to mitigate these losses, it is important to design the room size and exit location in the building for the fire evacuation so as to set the evacuation route and provide effective instruments including fire extinguishers and alarms. Additionally, an appropriate management for safety such as dairy training for fire evacuation is needed. In planning individual actions for safety and in evaluating the effectiveness of facilities and instruments, it is plausible to understand the phenomena in fires and validate the fire evacuation plan in advance. However, it is difficult to conduct experiments inside room or building in fire, because the costs are expectedly huge, and the evacuating people are exposed to danger. Therefore, the simulation of fire evacuation is needed.

For this purpose, we need to describe the pedestrian flow in fire evacuation. Since its dynamics is caused by collective crowd behavior, we have difficulties to handle directly each pedestrian by solving coupled differential equations, although the social force model has been proposed [2]. As one of the key approaches, the model of Cellular Automata (CA) has been developed to describe pedestrian dynamics, where a physical system of time and space are all discrete. So far, CA have been applied in a variety of scientific researches including traffic models and biological fields. In the original CA model for pedestrian dynamics

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 447–454, 2008.

[3-5], the von Neumann neighborhood is adopted. In this case, the pedestrian is moved to the nearest cells at next time step, but his movement is limited only in four directions: forward, backward, left, and right. Even if Moore neighborhood cells are used, the pedestrian is forced to move much faster in oblique directions. This might be a problem if we discuss the evacuation time.

Then, we have proposed a new model, which is called the real coded-Cellular Automata (RCA) [6,7]. In this model, it is possible to consider any direction and any velocity of pedestrian movement. Some examples have been already presented, including the lane formation in the street and the bottleneck in the room evacuation. It is confirmed that RCA can be a good tool for simulating the pedestrian dynamics.

In this study, we attempt to simulate fire evacuation by RCA. To describe the flame spread in fire, a percolation model is applied, where the flame position is determined stochastically. Here, we consider the evacuation from a relatively large room with one or two exits. In the simulation, we consider several parameters including the number of people in room, the moving velocity of the evacuee, and the location and size of the exit. The distance from the flame is also changed to evaluate the safety route, which is the unique parameter in fire evacuation. The evacuation route and number of fatalities involved in fire are discussed.

## 2 Numerical Method

### 2.1 RCA Model for Pedestrian Dynamics

Here, we explain our approach for arbitrary velocity and directions for pedestrian dynamics. The update rule of RCA consists of 4 steps to determine the position of the evacuee. These rules are applied to each pedestrian randomly. The velocity of all evacuees is the constant of $v_i$, which is changed in simulation. The update rules are almost the same as the original RCA rule [6,7], except that the evacuee is moved based on the floor field. Examples of the floor field are shown in Fig. 1, where there is one exit in the room. It is the static floor field describing the shortest distance to the exit. In fire evacuation, it could be possible that, along the path of the shortest distance to the exit, the evacuee may cross the burning area or encounter the flame. Then, we assume that he takes the route whose distance from the flame is $L$ in order to keep away from the burning area, which is shown in Fig. 2. The rule for the flame spread is explained in the next section.

### 2.2 Percolation Model for Flame Spread in Fire

In general, a fire becomes more serious as the burning area increases by flame spread. The mechanism of flame spread is very complex. It is explained with heat transport from the flame, pyrolysis reaction of combustible material, and the mixing of gaseous fuel with ambient oxygen in the air [1,8,9]. In this study, we use simplified model of the flame spread. It is assumed that the combustion occurs under the homogeneous atmosphere. Here, the flame spread rate is the

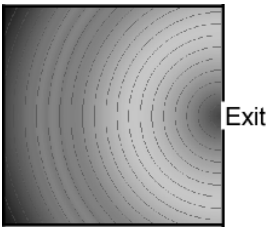



**Fig. 1.** Floor field in room with one exit

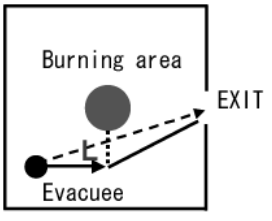



**Fig. 2.** Route of fire evacuation

only parameter to describe the degree of burning intensity. To consider the flame spread in CA code, the percolation model is used, which can be applied in the discrete time and space.

Figure 3 shows the percolation model for flame spread. The open circle is the combustible spot, which can be burned, and solid circle is the burning area. Fig. 3(a) is the example of an initial condition where the ignition point is shown at the center. Fig. 3(b) shows the burning area caused by the flame spread. In percolation model, the burning area is connected with each other. That is, the flame can not propagate if there are no combustible spots shown by open circles. As the time step goes, the burning area is developed in the neighborhood. At each time step, the flame spread occurs on the unburned combustible spots next to the burning area. There are 8 directions including forward, backward, left and right as well as four oblique directions. To control the flame spread rate, whether the flame can propagate or not is determined by the stochastic process. That is, the unburned region next to the burned region is ignited by some possibility. To consider the uniform flame spread, the probability in oblique direction is smaller than those in other four directions.

### 2.3 Calculation Domain

Figure 4 shows the calculation domain in the case of room fire. We consider three cases where only the number and location of the exit are different. There is one exit in case A. There are two exits in case B, which are located at both sides. In case C, there are two exits, but these are located at the same side. Each room size is 16m $\times$ 16m, and the exit width, $W$, is 1.2 or 2.4 m. The time step of $\Delta_t$

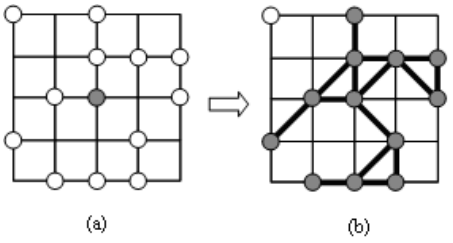



**Fig. 3.** Percolation model for flame spread

is 0.5 s, and the spatial grid of $\Delta$ is 0.4 m [7]. In this simulation, all grids are combustible spots.

As already explained in the previous section, the burning area is determined by the percolation model. For three cases, the initial ignition point is located at the center of the room. As the flame spread occurs, the burning area in the room increases. The possibility of forward, backward, left or right flame spread is set to be unity. In this case, the flame spread rate is 0.8 m/s ($= \Delta/\Delta_t = 0.4$m/ 0.5s). To consider the uniform flame spread, the possibility of flame spread in other four oblique directions is set to be 0.3.

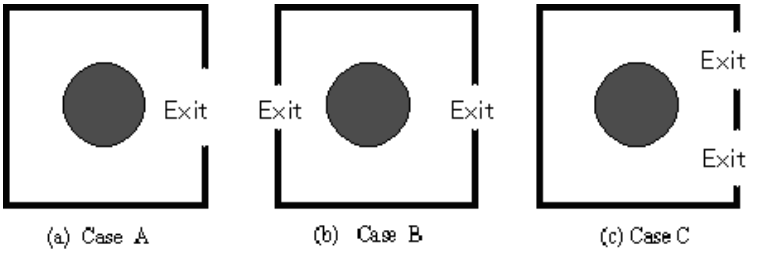



**Fig. 4.** Calculation domain for room fire; three cases are considered

## 3 Results and Discussion

### 3.1 Evacuee Velocity and Distance from Spreading Flame

Figure 5 shows the dynamics of fire evacuation. The calculation domain is case A, and the number of people in the room, $N$, is 100. The exit width is 1.2 m, and the evacuee velocity is 2.2 m/s. The distance between the evacuee and the flame is set to be 1.6 m. Three results at t = 0.5, 5.0, and 9.0 s are shown. The evacuees who can move directly toward the exit are not affected by the burning area. However, those passing near the burning area must keep away from the flame and take the longer route. It is found that more than 10 people are involved in fire and could not evacuate. It should be noted that, depending on the exit width, the bottleneck is observed at the exit [7]. Therefore, there are two circumstances the evacuee may

be possibly involved in fire: one is the place where he passes close to the burning region, and the other is the exit of the room.

Next, we change the distance from the spreading flame, $L$. Figure 6 shows the examples of fire evacuation for $L = 0.4$ and 2.8 m, respectively. The calculation domain is also case A, and there are 50 people in the room with the exit width of 1.2 m. For each cases, the evacuee velocity is 3.0 m/s, and the time is at $t = 2.5$ s. It is found that, as the distance from the spreading flame is shorter, the evacuee is forced to take the route close to the burning area. In this case, there are more possibilities to be involved in fire before the evacuee reaches the exit.

Needless to say, whether one can evacuate or not largely depends on his evacuee velocity, $v_i$. Therefore, we change the velocity of evacuees in the room. Three velocities of 2.2, 3.0, and 5.0 m/s are considered. We examine the number of fatalities involved in fire, $N_D$, by changing $L$ and $v_i$. The results are shown in Fig. 7. It is found that more people can evacuate if they can move faster, resulting in the smaller $N_D$. Interestingly, as we increase the distance from the flame, $N_D$ becomes smaller at the beginning, but $N_D$ increases again. That is, the minimum $N_D$ is observed around $L = 1.2$ to 1.6 m. Since the flame spread rate is 0.8 m/s, it is expected that the evacuee can easily keep away from the flame at higher $v_i$. Then, the number of fatalities becomes smaller as $v_i$ is higher. However, it is not safe if he passes near the burning area, because the flame is coming toward the evacuee. When he must take more distance from the burning region, his evacuation route is longer. It takes more time to arrive at the exit, and there are more chances to be involved in fire. Hence, there is the minimum $N_D$ in Fig. 7.

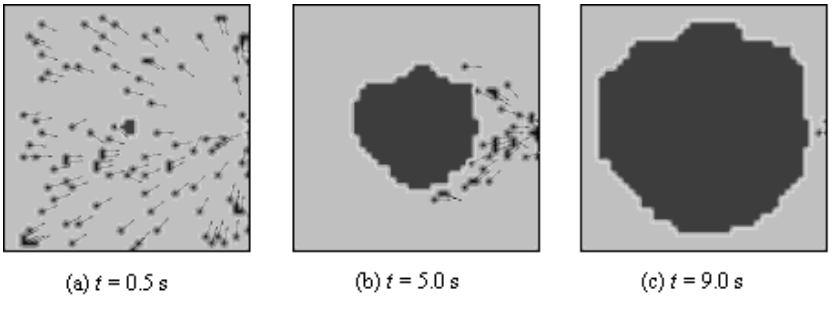



**Fig. 5.** The position of evacuee and burning area in the room in case A; $L = 1.6$ m, $N = 100$, $W = 1.2$ m, $v_i = 2.2$ m/s

### 3.2 Effect of Exit Location

Here, we change the number and location of the exit. In case A, the exit width is 2.4 m. On the other hand, we set $W = 1.2$ m in cases B and C, ensuring that the total exit width is 2.4 m. By changing number of people in room, we examine the number of fatalities involved in fire in three cases. Results are shown in Fig. 8. Expectedly, as more people in the room, the number of fatalities, $N_D$, increases.

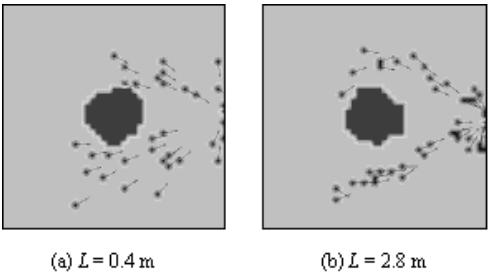



**Fig. 6.** The position of evacuee and burning area in the room in case A; $t = 2.5$ s, $N = 50$, $W = 1.2$ m, $v_i = 3.0$ m/s

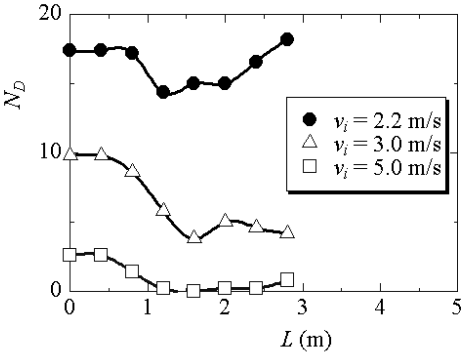



**Fig. 7.** Number of fatalities involved in fire by changing distance from spreading flame; three evacuee velocities of $v_i = 2.2$, 3.0, and 5.0 m/s are considered

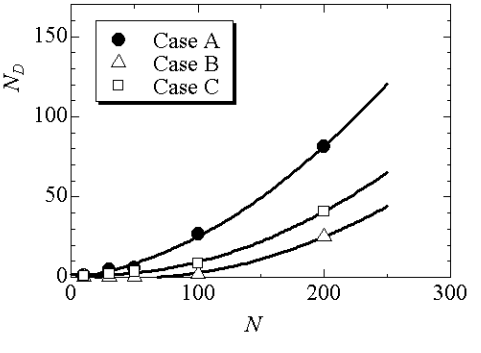



**Fig. 8.** Number of fatalities involved in fire by changing number of people in room; $v_i = 2.2$ m/s, and $L = 1.6$ m

In case A, even though the bottleneck is less observed at wider exit, the evacuee must take longer route. There are more chances to be involved in fire. In case C, although the situation is similar to case A, $N_D$ is smaller. By comparing three

cases, $N_D$ in case B is the smallest. This is because the evacuee can move in the same direction of flame spread, and he has less chances to involved in fire. Moreover, the distance of the evacuation route is shortest, which is almost half of other two cases. Thus, as for the location of the exit, case B is the safest, where there are two exits at both sides.

## 4 Conclusions

We have simulated fire evacuation by real-coded Cellular Automata (RCA). To describe the flame spread in fire, a percolation model is applied. In the simulation, we have changed the number of people in the room ($N$), distance from the flame ($L$), the location and size of the exit, in order to discuss the evacuation route and number of fatalities involved in fire ($N_D$). The following results are obtained.

1. The evacuees who can move directly toward the exit are not affected by the burning area. However, those passing near the burning area must keep away from the flame and take the longer route.
2. Depending on the exit width, the bottleneck is observed at the exit. Therefore, there are two circumstances the evacuee may be possibly involved in fire: one is the place where he passes close to the burning region, and the other is the exit of the room.
3. Whether one can evacuate or not largely depends on his evacuee velocity. Three velocities of 2.2, 3.0, and 5.0 m/s are considered. It is found that more people can evacuate if they can move faster, resulting in the smaller $N_D$.
4. As the distance from the flame is increased, $N_D$ becomes smaller at the beginning, but increases again. That is, the minimum $N_D$ is observed around $L = 1.2$ to 1.6 m. This means that it may not be safe to keep away too much, because it takes more time to arrive at the exit due to longer evacuation route, resulting in more chances to be involved in fire.
5. To examine the effect of exit location, three cases are considered: one exit in case A, two half size exits at both side in case B, and two half size exits at one side in case C. It is found that $N_D$ is the smallest in case B, which is explained by the fact that the evacuee moves in the same direction of flame spread, with the shorter evacuation route.

These are useful information for planning the safety guideline to mitigate the losses in fire.

## Acknowledgement

This work was partially supported by Toyota Physical and Chemical Research Institute in Japan.

## References

1. Hirano, T.: Combustion Science for Safety. Proc. Combust. Inst. 29, 167–180 (2002)
2. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407, 487–490 (2000)
3. Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J.: Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A 295, 507–525 (2001)
4. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionics-inspired cellular automata model for pedestrian dynamics. Physica A 312, 260–276 (2002)
5. Nishinari, K., Kirchner, A., Namazi, A., Schadschneider, A.: Extended Floor CA Model for Evacuation Dynamics. IEICE TRANS. INF. &SYST. E87-D, 726–732 (2004)
6. Yamamoto, K., Kokubo, S., Nishinari, K.: New Approach for Pedestrian Dynamics by Real-Coded Cellular Automata (RCA). In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 728–731. Springer, Heidelberg (2006)
7. Yamamoto, K., Kokubo, S., Nishinari, K.: Simulation for Pedestrian Dynamics by Real-Coded Cellular Automata (RCA). Physica A 379, 654–660 (2007)
8. Yamamoto, K., Mori, K., Onuma, Y.: Trans., Effects of Ambient on Flame Spread over Solid Fuel. Japan Society for Mechanical Engineers 68-672, 2437–2443 (2002)
9. Yamamoto, K.: Trans., A Study for Model of Flame Spread over Solid Fuel. Japan Society for Mechanical Engineers 69-680, 954–961 (2003)

# Walking-Distance Introduced Queueing Theory

Daichi Yanagisawa[1,2], Akiyasu Tomoeda[1],
Ayako Kimura[1], and Katsuhiro Nishinari[1,3]

[1] Department of Aeronautics and Astronautics, School of Engineering,
The University of Tokyo
[2] JSPS Research Fellow
[3] PRESTO, Japan Science and Technology Corporation
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{tt087068,tt67055,tt076360,tknishi}@mail.ecc.u-tokyo.ac.jp

**Abstract.** We introduce the effect of delay in walking from the head of the queue to the service windows in the queueing theory, and obtain the suitable type of queueing system under various conditions. When there are plural service windows, the queueing theory indicates that a fork-type queue, which collects people into a single queue, is more efficient than a parallel-type queue, i.e., queues for each service windows. However, in the walking-distance introduced queueing theory, we find that the parallel-type queue is more efficient when sufficiently many people are waiting in queues, and service time is shorter than walking time. We also consider the situation where there are two kinds of people, whose service time is short and long. The analytical result says that we can decrease people's waiting time and their stress by setting up queues for each kind of people separately.

## 1 Introduction

Queueing theory has been considerably studied since Erlang started designing telephone exchanging system in 1909 [1], and many important theories have been developed [2,3,4,5]. Nowadays, it is applied to study many social systems such as the internet [6,7], a resource management system [8], a vehicular traffic system [9] and a pedestrian traffic system [10].

Traffic flow and pedestrian dynamics have been studied actively by using the theory of particle systems, fluid dynamics and cellular automaton [11,12]. Especially, we simulate the dynamics of cars and pedestrians efficiently by using cellular automaton models since its time and space are both discrete. Pedestrians' movement is treated as a stochastic process in the floor field model [13]. Then we have succeeded in calculating the total evacuation time analytically by using the floor field model, and studied the way of smooth evacuation [14].

Calculating the mean waiting time in the queueing system, whose bottleneck is a service window is similar to calculating the total evacuation time from a room, whose bottleneck is an exit. However, the former does not take into account of the effect of spatial structures, while the latter does.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 455–462, 2008.

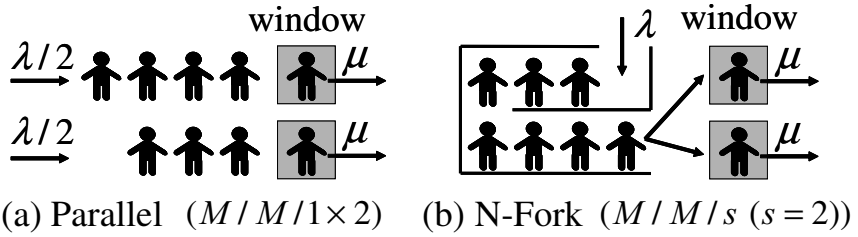



**Fig. 1.** Schematic views of queueing systems. (a) Parallel-type queueing system ($s = 2$) ($M/M/1 \times 2$). (b) N-Fork-type queueing system ($s = 2$) ($M/M/2$). $s(\in \mathbf{N})$, $\lambda \in [0, 1]$, and $\mu \in [0, 1]$ represent the number of service windows, the arrival rate, and the service rate, respectively.

According to the queueing theory, the waiting time of a fork-type queueing system is shorter than that of a parallel-type queueing system (Parallel) (Fig. 1 (a)). However, fork-type queueing system considered in the normal queueing theory (N-Fork) (Fig. 1 (b)) does not reflect the effect of the walking distances from the head of the queue to the service windows. The effect of the distances cannot be ignored in a system such as a large immigration inspection floor in the international airport since walking distances become very long. Therefore, we extended the queueing theory by introducing the effect of distances by using cellular automaton. We will show that the mean waiting time becomes shorter in the parallel-type queue than in the fork-type queue by introducing the effect of walking distances (D-Fork) (Fig. 2 (a)) when sufficiently many people are waiting in the queue and the rate of walking time against service time is large. We also consider the way to shorten the mean waiting time and people's stress involved, when two kinds of people, whose service time are short and long, come into the system at the same time.

## 2 Walking-Distance Introduced Queueing Theory

### 2.1 Walking Distances

In reality, people take some times to walk from the head of the queue to the serviece windows. However, the walking time is not taken into account in the N-Fork (Fig. 1 (b)). Therefore, we consider the D-Fork as in Fig. 2 (a) by representing the distances using cellular automaton. The cells, which are indicated as window, are service windows, and the other cells are passage cells. The place that people are waiting, which is not divided into cells, is a queue. For simplicity, we study the system, whose head of the queue is at the end of it, i.e., the service window 1 is in front of the head of the queue, and a service window which is the largest number is farthest from the head. $\lambda$, $\mu$, and $s$ are the arrival rate, the service rate, and the number of service windows, respectively. $a$ and $b$ represent the length of the passage, and $k$ is the length between two service windows. The distance from the head of the queue to the service window $n$ is described as

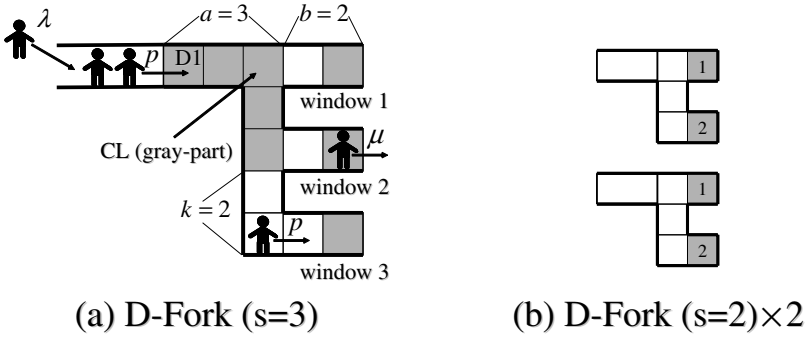



**Fig. 2.** Schematic views of queueing systems. (a) D-Fork-type queueing system ($s = 3$). $p \in [0, 1]$ is the hopping rate of the people. Several people walk through the area CL at the same time. Therefore, people sometimes cannot go forward since other people stand in front of them. (b) Double D-Fork-type queueing system ($s = 2$) $\times$ 2. The numbers in the service windows represent the window number.

$d_n = a + b + k(n - 1)$. Fig. 2 (a) represents the case $s = 3$, $a = 3$, $b = 2$, and $k = 2$. Service windows have two states: vacant and occupied. When a person at the head of the queue decides to move to the vacant service window $n$, it changes into occupied state. The person proceed to the service window by one cell with the probability $p$ in one time step as asymmetric simple exclusion process. A service starts when the person arrives at the service window and after it finishes the state of the service window changes into vacant state.

### 2.2 Update Rules

The simulation of walking-distance introduced queueing theory consists of the following five steps per unit time step.

1. If there is at least one vacant service window and one person in the queue, and the first cell of the passage D1 (Fig. 2 (a)) is vacant, then the person decide to proceed to a vacant service window which is the nearest to the head of the queue, and the state of the service window become occupied.
2. Add one person to the queue with the probability $\lambda$.
3. Proceed each person in the passage cells to his/her service windows with probability $p$ if there is not other person at their proceeding cell.
4. Remove people at the service windows and change their states into vacant state with the probability $\mu$.
5. If 1. takes into practice, proceed the person at the head of the queue to D1 with probability $p$.

### 2.3 Mean Field Analysis

We define the sum of the walking time and the service time at service window $n$ as a throughput time $\tau_n$ and its reciprocal as a throughput rate $\mu_n$. Here, we

calculate the mean throughput rate $\hat{\mu}_n$ when $n$ service windows are occupied, and obtain stationary equations of the D-Fork. The mean value of the throughput time $E(\tau_n)$ is described as follows in the case we suppose that all passage cells are vacant by mean field approximation.

$$E(\tau_n) = \frac{1}{\mu} + \frac{a+b+k(n-1)}{p}. \tag{1}$$

Then the throughput rate $\mu_n$ is obtained as

$$\mu_n = \frac{1}{E(\tau_n)} = \frac{1}{\frac{1}{\mu} + \frac{a+b}{p} + \frac{k(n-1)}{p}} = \frac{\tilde{\mu}}{1+\alpha\tilde{\mu}(n-1)} = \frac{\tilde{\mu}}{1+2\beta(n-1)}, \tag{2}$$

where $\tilde{\mu}$, $\alpha$, and $\beta$ are

$$\frac{1}{\tilde{\mu}} = \frac{1}{\mu} + \frac{a+b}{p}, \quad \alpha = \frac{k}{p}, \quad \beta = \frac{1}{2}\frac{\frac{k}{p}}{\frac{1}{\tilde{\mu}}}. \tag{3}$$

In the case $2\beta(n-1) \ll 1$, we calculate the mean throughput rate $\hat{\mu}_n$ as

$$\hat{\mu}_n = \frac{1}{n}\sum_{l=1}^{n} \mu_l \approx \frac{\tilde{\mu}}{1+\beta(n-1)}. \tag{4}$$

By using (4) the stationary equations are described as follows:

$$\begin{aligned} \lambda P_0 &= \hat{\mu}_1 P_1 \\ \lambda P_{n-1} + (n+1)\hat{\mu}_{n+1} P_{n+1} &= (\lambda + n\hat{\mu}_n) P_n \qquad (1 \le n \le s-1) \\ \lambda P_{n-1} + s\hat{\mu}_s P_{n+1} &= (\lambda + s\hat{\mu}_s) P_n \qquad (n \ge s). \end{aligned} \tag{5}$$

We obtain the mean waiting time $W_q$ by solving (5) analytically. In the case $\beta = 0$, we have the stationary equations of $M/M/s$ [15] from (5), thus $\beta$ represents the effect of walking time.

In our simulation the distribution of the throughput time is gamma distribution. We approximate it as exponential distribution in this calculation, however, when $\beta$ is small the results from the exponential distribution approximated well to those from gamma distribution. According to Pollaczek-Khintchine formula [15], $W_q$ is large when the variance of the service time distribution is large. Since the variance of exponential distribution is larger than that of gamma distribution, theoretical calculation estimate $W_q$ larger than that of simulation.

## 3 Comparison of a Parallel Queue and a Fork Queue

We compare the mean waiting time $W_q$ of the Parallel (Fig. 1 (a)), the N-Fork (Fig. 1 (b)), and the D-Fork (Fig. 2 (a)). In the following, the service rate includes the effect of $a+b$, that is to say $\tilde{\mu} = \mu$ since the main object is to study the difference between a parallel-type queueing system and a fork-type

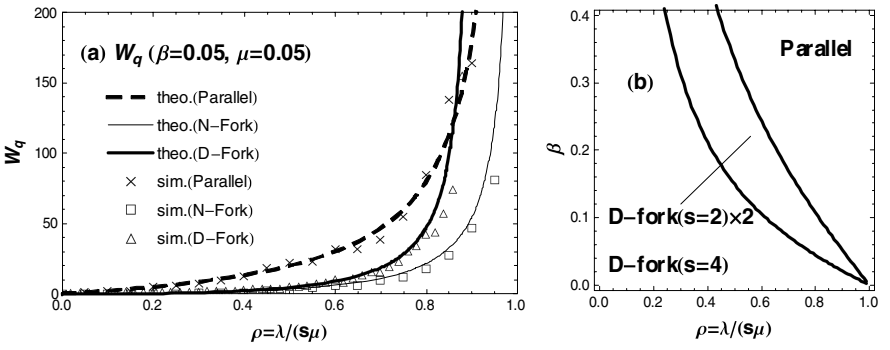



**Fig. 3.** (a) Comparison of $W_q$ of the Parallel, the N-Fork, and the D-Fork in the case $a = 1,\ b = 1,\ p = 1,\ \beta = 0.05,\ \mu = 0.05$. (b) A queueing system which makes the mean waiting time $W_q$ minimum in the case $s = 4,\ \mu = 0.05$.

queueing system. $a+b$, which represents the distance from the head of the queue to the nearest service window, influences a parallel-type queueing system and a fork-type queueing system in the same way.

Figure 3 (a) show $W_q$ against the utilization $\rho(= \lambda/(s\mu))$. The results of analysis agree with those of the simulation very well. We see that $W_q$ of the N-Fork is smaller than that of the Parallel and the D-Fork in $0 \leq \rho < 1$. There is a possibility that more than one person is waiting in one queue and no one is in the other queue in the Parallel ($s \geq 2$), however there is no vacant service window in the N-Fork when people are waiting in the system. This is the reason why $W_q$ of the N-Fork is always smaller than that of the Parallel. Since the N-Fork does not take into account of the effect of the walking distances, i.e. $\beta = 0$, it is obvious that $W_q$ of the N-Fork is smaller than that of the D-Fork. The N-Fork is the most efficient of the three; however, it is an ideal system and does not exist in reality. By focusing on the curves of the Parallel and the D-Fork, we can clearly observe the crossing of them. This means that when the utilization $\rho$ is small, i.e., there are not sufficiently many people in the system; we should form the D-Fork to decrease the waiting time. On the contrary, when the utilization $\rho$ is large, i.e., there are many people in the system, we should form the Parallel. When $\beta$ become large, the crossing point move to the left. The strong effect of the walking distances extend the suitable $\rho$ region for the Parallel. This agrees with our intuition, since the D-Fork is influenced by the distances but the Parallel does not. The reversal phenomenon of $W_q$ is obtained for the first time by introducing the effect of distances.

Figure 3 (b) shows the type of queueing system, which minimize $W_q$ against $\rho$ and $\beta$ in the case $s = 4$. This figure is useful for designing queueing systems. The curves divide the $\rho - \beta$ plane into three regions. In the lower left region $W_q$ of the D-Fork ($s = 4$) is the smallest, and in the upper right region $W_q$ of the Parallel is the smallest. Surprisingly, $W_q$ of the D-Fork ($s = 2$) $\times 2$ (Fig. 2 (b)) is the smallest in the middle region. This indicates that the choice of the type of queueing systems is not only the Parallel and the D-Fork, but also a combination

of them. According to (3), $\beta$ represents the ratio of walking time and service time. Therefore, the D-Fork is suitable when service time is much longer than walking time. The value of $\beta$ is small in most D-Fork in reality, however, in large queueing system such as an immigration inspection floor in the international airport, we should divide the large D-Fork into the several small D-Forks to decrease the effect of the walking distances.

## 4 Method for Decreasing Waiting Time and Stress

In the former sections, we discuss the queueing systems, whose distribution of interarrival time is exponential distribution, i.e. random arrival. Here, we consider the situation that many people arrive at the same time, such as arrivals of people who alight from trains and airplanes. The total throughput time $T_{total}$ that all people finish leaving the system is calculated and studied to decrease it. We consider the system, which has four service windows. There are two kinds of people whose service time are short (SP) and long (LP). The service rate of SP and the number of service windows only for SP are denoted as $\mu_S$ and $s_S$, respectively. Similarly, those of LP are described as $\mu_L$ and $s_L$. We have three strategies as follows:

**(S-a)** Mix ($s = 4$) (Fig. 4 (a))
Both SPs and LPs use the same windows.
**(S-b)** Separate ($s_S = 1,\ s_L = 3$) (Fig. 4 (b))
SPs use the window only for SP and LPs use the windows only for LP.
**(S-c)** Separate ($s_S = 1,\ s_L = 3$) $\rightarrow$ Long ($s_L = 4$) (Fig. 4 (c))
First same as (b), but after all SPs have left, the window, which was only for SP, is open for LP.

The distribution of the $T_{total}$ when 50 SPs and 50 LPs arrive at the same time is described in Fig. 5 (a). We see that the mean of $T_{total}$ of S-c is the smallest and that of S-b is the largest. When S-b or S-c is adopted, the queue of the SP is not affected by the distances. Thus SPs leave the system quickly. After all SPs have left, all four service windows are used efficiently in S-c, however, one window is not used in S-b. This makes S-c the best and S-b the worst.

We also discuss the stress of waiting people by Fig. 5 (b), which describes the number of people waiting in the queue against the time step $t$. People suffer from a stress when they are waiting in the queue and do not when they leave the system. Therefore, an area surrounded by the axes and the curves represents the sum of all people's stress until they left. Clearly, we observe that (Area of S-c) $<$ (Area of S-b) $<$ (Area of S-a). It is interesting that when we compare $T_{total}$, S-a is better than S-b, however, comparing the stress, the result is opposite. We also find that S-c is the best strategy from the both point of views: $T_{total}$ and the stress.

If the arrival of people is random, it is difficult to adopt S-c since we cannot find out when to change the type of the queueing system. However, if we have information about people's arrival, we can decrease both waiting time and stress of people by adapting the type of the queueing system into proper one.

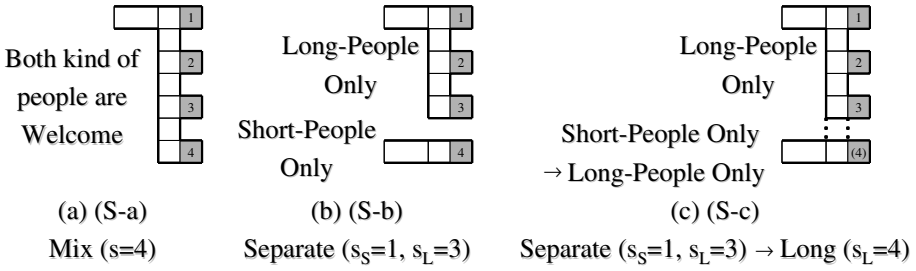



**Fig. 4.** Schematic views of queueing systems. (a) (S-a) Mix queueing system ($s = 4$). (b) (S-b) Separate queueing system ($s_S = 1,\ s_L = 3$). (c) (S-c) Separate queueing system ($s_S = 1,\ s_L = 3$) → Queueing system for LP ($s_L = 4$).

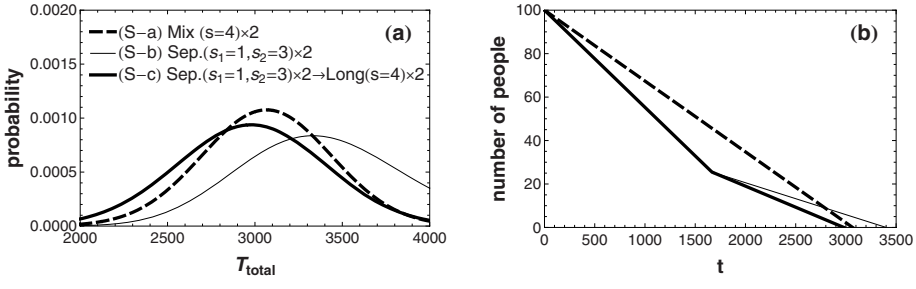



**Fig. 5.** (a) The probability distribution of the total throughput time $T_{total}$. (b) Number of people in the queueing system against time step $t$. The value of plot S-b and S-c decrease dramatically, when people start to be given services. This is because SPs get out from the system quickly by avoiding the disturbance by LPs.

When we apply this study to the real system, the service rates are estimated by the data from the measurement.

## 5 Conclusion

We have introduced the effect of walking distances from the head of the queue to the service windows and shown that the performance of a parallel-type queueing system is better than that of a fork-type queueing system when there are sufficiently many people in the system and walking distances are long enough. It also turns out that dynamical changing of a queueing system decrease both waiting time and stress of people when two kinds of people, whose service time are short and long, come into the system at the same time. These results are obtained analytically for the first time in this paper. It is an important future work to study the way to make people form a queue which neither occupy much space nor cause congestion.

## Acknowledgement

This work is supported by Japan Society for the Promotion of Science.

## References

1. Erlang, A.K.: The theory of probabilities and telephone conversations. Nyt. Tidsskr. Mat. Ser. B 20, 33–39 (1909)
2. Kendall, D.G.: Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of the Imbedded Markov Chain. Ann. Math. Statist. 24(3), 338–354 (1953)
3. Burke, P.J.: The Output of a Queueing System. Operations Research 4(6), 699–704 (1956)
4. Jackson, J.K.: Networks of waiting lines. Operations Research 5(4), 518–521 (1957)
5. Little, J.C.D.: A proof for the queueing formula $L = \lambda W$. Operations Research 9(3), 383–387 (1961)
6. Kasahara, S.: Towards Queueing Theory for the Internet Design. Technical Report of IEICE 101(649), 25–30 (2002)
7. Mukherjee, G., Manna, S.S.: Phase transition in a directed traffic flow network. Phys. Rev. E 71, 066108 (2005)
8. Barabasi, A.-L.: The origin of bursts and heavy tails in human dynamics. Nature 435, 207–211 (2005)
9. Helbing, D., Treiber, M., Kesting, A.: Understanding interarrival and interdeparture time statistics from interactions in queueing systems. Physica A 363, 62–72 (2006)
10. Helbing, D., Jiang, R., Treiber, M.: Analytical investigation of oscillations in intersecting flows of pedestrian and vehicle traffic. Phys. Rev. E 72, 046130 (2005)
11. Chowdhury, D., Santen, L., Schadschneider, A.: Statistical physics of vehicular traffic and some related systems. Phys. Rep. 329, 199–329 (2000)
12. Helbing, D.: Traffic and related self-driven many-particle systems. Rev. Mod. Phys. 73, 1067–1141 (2001)
13. Nishinari, K., Kirchner, A., Namazi, A., Schadschneider, A.: Nonmembers: Extended Floor Field CA Model for Evacuation Dynamics. IEICE Transactions on Information and Systems E87-D(3), 726–732 (2004)
14. Yanagisawa, D., Nishinari, K.: Mean Field Theory for Pedestrian Outflow through an Exit. Phys. Rev. E 76, 061117 (2007)
15. Bolch, G., Greiner, S., de Meer, H., Trivedi, K.S.: Queueing Networks and Markov Chains. A Wiley-Interscience Publication, U.S.A (1998)

# An Improved Double Byte Error Correcting Code Using Cellular Automata

Jaydeb Bhaumik, Dipanwita Roy Chowdhury, and Indrajit Chakrabarti

Indian Institute of Technology,
Kharagpur 721302, India
{jaydeb.gssst,drc.cse,indrajit.ece}@iitkgp.ernet.in

**Abstract.** Cellular Automata(CA) based VLSI implementation of $t$-byte errors correcting code has been established by previous research to be superior to the other existing techniques employed for realizing Reed-Solomon(RS) code. However, the scheme suffers from the limitation that it can correct $t-$byte errors ($t \geq 2$) provided errors are confined either wholly to the information bytes or entirely to the check bytes. The work reported in the present paper overcomes this limitation and corrects the errors likely in both information and check bytes. Moreover one weakness found in an earlier similar work has been identified and rectified using a modified check symbol expressions.

**Keywords:** Byte Error Correcting Code, Cellular Automata, RS Code and VLSI.

## 1 Introduction

RS code has found many applications in storage devices (CD, DVD), wireless communications, high speed modems and satellite communications. The complexity of RS encoder and decoder increases with the error correcting capability of the code. Hence many researchers have put their effort to minimize the complexity of RS encoder/decoder for communication applications. A number of general encoding and decoding schemes of the RS code is available in the literature [7, 8].

But VLSI system designer always prefers to have simple, regular, modular and cascadable structure with local interconnection for reliable high speed operations of the circuit. It has been found that these parameters are supported by local neighborhood Cellular Automata (CA). In [1] CA based byte error correcting code has been proposed. The proposed design in [1] requires less hardware compared to the existing techniques used for RS code. A new scheme has been proposed for pipeline implementation of CA based tbEC - tbED codes that are analogous to the conventional RS code in [2]. Another design scheme has been reported for parallel implementation of CA based SbEC/DbED and DbEC/DbED code that is also analogous to the conventional RS code in [3]. A new high speed VLSI architecture for decoding RS codes with Berlekamp-Massey (BM) algorithm has been published in [4]. In this scheme, the speed bottleneck in BM

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 463–470, 2008.

algorithm is eliminated by using fully systolic architecture. A new degree computationless modified Euclid (DCME) algorithm and its dedicated architecture for RS decoder has been reported in [5]. The architecture has low hardware complexity compared with conventional modified Euclid architecture.

However, the scheme in [1] can correct t-byte errors ($t \geq 2$) provided errors are totally confined to information or check byte only. The scheme[1] fails if one error is in information byte and another in check byte. Another weakness of the scheme in [1] is that single byte error in $k^{th}$ information byte and double byte errors one in $k^{th}$ information byte and another in the last information byte correspond to the same equation for error location identification.

In this paper, an improved double byte error correcting code using CA has been proposed. The new scheme over comes the limitation of [1] and can correct errors even if one error occurs in the information byte and another is in the check byte. Also the scheme can unambiguously determine error locations. CA-based VLSI design is attractive because of its simplicity, regularity and higher throughput.

The rest of this paper is organized as follows. In the next section, a brief overview of existing CA-based double byte error correcting code is described. Then, we describe the weakness and limitation of the existing CA-based double byte error correcting code in section 3. Section 4 discusses the proposed new scheme and finally the paper is concluded in section 5.

## 2 CA-Based Double Byte Error Correcting Code

In this section, we discuss the preliminaries of CA and CA based double byte error correcting code which has been proposed in [1].

### 2.1 Cellular Automata Preliminaries

A cellular automata (CA) consists of a number of cells arranged in a regular manner, where the state transitions of each cell depends on the state of its neighbors and each cell consists of a storage element (D flip-flop) and a combinational logic implementing the next-state function. The next-state function for a three-neighborhood CA cell can be expressed as follows.

$$q_i(t+1) = f[q_{i-1}(t), q_i(t), q_{i+1}(t)]$$

where $q_i(t)$ is the output state of the $i^{th}$ cell at time $t$ and $f$ is the next state function also called the rule of the automata[6]. An $n$-cell CA can be characterized by an $n \times n$ characteristic matrix $T$. $S_{t+1}$ can be computed by multiplying $[S_t]$ with $T$, where $S_t$ and $S_{t+1}$ represent the states of the CA at $t^{th}$ and $(t+1)^{th}$ time instant respectively. This matrix has additional properties of being tridiagonal and having a primitive characteristic polynomial. This guarantees that linear recurrence has maximum period $2^n - 1$. Detailed information on CA may be found in [6]. Only linearity property of CA is used in byte error correcting code.

### 2.2 Overview of the CA-Based Double Byte Error Correcting Code

CA-based byte error correcting code [1] is similar to extended RS code. But compared to RS code, the proposed code is much simpler and requires much less hardware. In two byte error correcting code, encoder generates four check bytes from a block of N-byte information. After that the check symbols are appended to information bytes to form the code word. Now four check bytes should be generated by running the CA for $N$ cycles, while sequentially feeding the $N$ information bytes ($D_k$), where $0 \leq k \leq (N-1)$. The four check bytes are as follows:

$$C_0 = D_{N-1} \oplus D_{N-2} \oplus ... \oplus D_0 \tag{1}$$

$$C_1 = D_{N-1} \oplus T[D_{N-2}] \oplus ... \oplus T^{N-1}[D_0] \tag{2}$$

$$C_2 = D_{N-1} \oplus T^2[D_{N-2}] \oplus ... \oplus T^{2(N-1)}[D_0] \tag{3}$$

$$C_3 = D_{N-1} \oplus T^3[D_{N-2}] \oplus ... \oplus T^{3(N-1)}[D_0] \tag{4}$$

The primary objective of decoding is to retrieve the correct information byte. Decoding is done by employing the properties of maximum length group CA. The syndrome corresponding to the $q^{th}$ check byte, $S_q$ is defined as

$$S_q = C_q \oplus C_q^{'} \quad ; \quad 0 \leq q \leq 3. \tag{5}$$

where $C_q$ is the $q^{th}$ received check byte and $C_q^{'}$ is the $q^{th}$ check byte recomputed from the received information bytes (with possibility of error present).

***Decoding Algorithm***

**step 1:** If all the syndrome bytes $S_0, S_1, S_2, S_3$ are zeros, then there is no error in the received information block.

**step 2:** If any one or two of the syndrome byte(s) is/are nonzero but the other are zeros, then the check byte(s) is/are in error.

**step 3:** If more than two syndrome bytes are nonzero then any one of the following three cases may occur.

1) One error is in the information byte

2) Double byte errors are in the information byte.

3) One error is in the information byte and other error is in the check byte.

**step 4:** In case none of the above conditions regarding the syndrome bytes hold, then more than two errors have occurred.

Suppose two byte errors have occurred in two information bytes. If $E_k$ and $E_l$ are the errors in $k^{th}$ and $l^{th}$ information bytes, then the corresponding syndrome equations are

$$S_0 = E_k \oplus E_l \tag{6}$$

$$S_1 = T^i[E_k] \oplus T^j[E_l] \tag{7}$$

$$S_2 = T^{2i}[E_k] \oplus T^{2j}[E_l] \tag{8}$$

$$S_3 = T^{3i}[E_k] \oplus T^{3j}[E_l] \tag{9}$$

Also $i + k = N - 1$ and $j + l = N - 1$, where $0 \leq k, l \leq N - 1$. If at least three syndromes are non-zero and there exist two integers $i$ and $j$ $(0 \leq i, j \leq (N-1))$ such that

$$T^i[S_2] \oplus S_3 = T^{2j}(T^i[S_0] \oplus S_1) \tag{10}$$

$$T^{2i}[S_1] \oplus S_3 = T^j(T^{2i}[S_0] \oplus S_2) \tag{11}$$

then the $k^{th}$ and the $l^{th}$information bytes where $k = N-1-i$ and $l = N-1-j$ are erroneous. Let $T^i \oplus T^j = T^y$. Again, if $L = 2^n - 1$ is the cycle length of a $n$-cell maximum length group CA, then $(T^y)^{-1} = T^{-y} = T^{(L-y)} = T^p$. The error magnitudes are determined using the two equations given below.

$$E_l = T^p(T^i[S_0] \oplus S_1) \tag{12}$$

$$E_k = S_0 \oplus E_l \tag{13}$$

If $D_m^{'}$ , $E_m$ are the received $m^{th}$ information byte and the calculated $m^{th}$ error byte respectively, then the correct information byte can be obtained as

$$D_m = D_m^{'} \oplus E_m; \qquad where\ 0 \leq m \leq (N-1) \tag{14}$$

The next section reports the weakness and limitation of scheme [1].

## 3 Weakness and Limitation of the CA-Based Double Byte Error Correcting Code

CA based byte error correcting code proposed in [1] is very good from VLSI implementation point of view. But the scheme has a weakness and one limitation. This section explains the weakness and limitation of the scheme proposed in [1].

### 3.1 Weakness of the CA-Based Double Byte Error Correcting Code

***Case 1 : Single byte error in the information byte***
Assume that a single byte error has occurred in the $k^{th}$ information byte. If $E_k$ is the error in $k^{th}$ byte, then according to [1] corresponding syndrome equations are

$$S_0 = E_k; \quad S_1 = T^i[E_k]; \quad S_2 = T^{2i}[E_k]; \quad S_3 = T^{3i}[E_k] \tag{15}$$

where $S_0, S_1, S_2, S_3$ are the four syndrome bytes. From (15) we get

$$S_1 \oplus S_3 = T^i(S_0 \oplus S_2) \tag{16}$$

$$S_2 \oplus S_3 = T^{2i}(S_0 \oplus S_1) \tag{17}$$

Equations (16) and (17) may be used to determine the single byte error location in information byte. But we can't use the two equations if the error is in the last information byte. For an error in the last information byte the syndromes are all equal i.e. $S_0 = S_1 = S_2 = S_3 = E_{N-1}$, where $E_{N-1}$ is the error in the last information byte. Therefore, $S_2 \oplus S_0 = 0;\ S_1 \oplus S_3 = 0;\ S_1 \oplus S_0 = 0;\ S_3 \oplus S_2 = 0$

and equations (16) and (17) will be satisfied for all $i$ and we can not determine error location uniquely.

***Case 2 : One error in the*** $(N-1)^{th}$ ***i.e. the last information byte and the other in the*** $k^{th}$***information byte***
The syndrome equations according to [1] are

$$S_0 = E_k \oplus E_{N-1} \tag{18}$$

$$S_1 = T^i E_k \oplus E_{N-1} \tag{19}$$

$$S_2 = T^{2i} E_k \oplus E_{N-1} \tag{20}$$

$$S_3 = T^{3i} E_k \oplus E_{N-1} \tag{21}$$

Using above four equations we get

$$S_1 \oplus S_3 = T^i (S_0 \oplus S_2) \tag{22}$$

$$S_2 \oplus S_3 = T^{2i} (S_0 \oplus S_1) \tag{23}$$

It is observed that the equations (16), (17) in case1 are same as equations (22), (23) in case2. So it is impossible to compute error location uniquely. But we can separate the case1 and case2 by checking any one of the additional conditions.

$$T^i S_0 \neq S_1;\ T^i S_1 \neq S_2;\ T^i S_2 \neq S_3 \tag{24}$$

When any one of the above three conditions and equations (16), (17) are satisfied then one error in the $k^{th}$ information byte and another in the $(N-1)^{th}$ information byte is identified. So it requires only one extra checking.

### 3.2 Limitation of the CA-Based Double Byte Error Correcting Code

The scheme [1] suffers from the limitation that it can correct errors provided errors are totally confined to information or check bytes only. If one error is in the information byte and another is in the check byte, then the error is located in maximum of three information byte position. In the next section, we modify the scheme [1] to overcome the above weakness and limitation.

## 4 Our Improved Scheme

In our improved double byte error correcting code, check bytes are generated by running the CA for N cycles while sequentially feeding the N information bytes.

***Algorithm to compute Check byte*** $C_i$
**begin**
$X := 0$; ($X$ denotes the CA state)
**for** $k = 0$ *to* $N-1$ **do**

**begin**
$X := X \oplus D_k$;
Run the CA for one cycle;(CA with characteristic matrix $T^i$)
**end**;
$C_i := X$;
**end**;

For double byte error correcting code, the expression for check symbols $C_0, C_1, C_2$ and $C_3$ are given below.

$$C_0 = D_{N-1} \oplus D_{N-2} \oplus ... \oplus D_0 \tag{25}$$

$$C_1 = TD_{N-1} \oplus T^2[D_{N-2}] \oplus ... \oplus T^N[D_0] \tag{26}$$

$$C_2 = T^2 D_{N-1} \oplus T^{2(2)}[D_{N-2}] \oplus ... \oplus T^{2(N)}[D_0] \tag{27}$$

$$C_3 = T^3 D_{N-1} \oplus T^{3(2)}[D_{N-2}] \oplus ... \oplus T^{3(N)}[D_0] \tag{28}$$

Next we will derive the equations to identify the error locations from the received information and check bytes. Suppose two byte errors have occurred in $k^{th}$ and $l^{th}$ information bytes with $k \neq l$. The corresponding syndrome equations are

$$S_0 = E_k \oplus E_l \tag{29}$$

$$S_1 = T^i[E_k] \oplus T^j[E_l] \tag{30}$$

$$S_2 = T^{2i}[E_k] \oplus T^{2j}[E_l] \tag{31}$$

$$S_3 = T^{3i}[E_k] \oplus T^{3j}[E_l] \tag{32}$$

where $S_0, S_1, S_2, S_3$ are the four syndrome bytes and $E_k$ and $E_l$ are the corresponding errors in the $k^{th}$ and $l^{th}$ bytes. Also $i + k = N$ and $j + l = N$, where $1 \leq i, j \leq N$. From the above four syndrome equations we get

$$T^i[S_2] \oplus S_3 = T^{2j}(T^i[S_0] \oplus S_1) \tag{33}$$

$$T^{2i}[S_1] \oplus S_3 = T^j(T^{2i}[S_0] \oplus S_2) \tag{34}$$

So using the above two equations we can determine the error locations if both the errors occur in the information byte.

**Theorem 1.** The scheme identifies the error locations uniquely if single/double byte(s) error occurred and it is independent of error position.

**Proof** We establish the result for the following cases where indistinguishability of error location identification equations can happen.

***Case 1 : One error in the*** $(N-1)^{th}$ ***i.e. the last information byte and the other in the*** $k^{th}$ ***information byte***

According to proposed scheme the equations for error location identification are

$$T[S_2] \oplus S_3 = T^{2i}(T[S_0] \oplus S_1) \tag{35}$$

$$T^2[S_1] \oplus S_3 = T^i(T^2[S_0] \oplus S_2) \tag{36}$$

***Case 2 : Single byte error in the $k^{th}$ information byte***
The equations for error location identification are

$$T^i(S_2 \oplus S_0) = S_1 \oplus S_3 \tag{37}$$

$$T^{2i}(S_1 \oplus S_0) = S_3 \oplus S_2 \tag{38}$$

If single byte error occurs in the last information byte then also we can identify the error location using equations (37) and (38). According to our scheme $S_0 \neq S_1 \neq S_2 \neq S_3$, so above two equations will be satisfied for a particular $i$.

It is observed that equations (35), (36) in case1 are different from equations (37), (38) in case 2. Thus, it overcomes the weakness of scheme [1] given in section 3.1. Note that we can also derive the equations in case1 and case2 from equations (33) and (34). Next we describe the situation where one error is in the information byte and other is in the check byte.

***Two Byte Error location identification when one information byte and one check byte are erroneous***

**Theorem 2.** The scheme decodes correctly if one error is in the information byte and another is in check byte.

**Proof.** Assume $e_0$, $e_1$, $e_2$ and $e_3$ are the errors in the 1st, 2nd, 3rd and 4th check byte respectively. If one error is in $k^{th}$ information byte and another is in any one of the four check bytes, then any one of the four different cases may occur.

**1.** If 1st check byte and $k^{th}$ information byte are erroneous, then the syndrome equations are

$$S_0 = E_k \oplus e_0; \quad S_1 = T^i E_k; \quad S_2 = T^{2i} E_k; \quad S_3 = T^{3i} E_k \tag{39}$$

$$S_3 = T^i S_2; \quad S_3 = T^{2i} S_1; \quad S_1 \neq T^i S_0 \tag{40}$$

$$E_k = T^{L-i} S_1 \tag{41}$$

Equation (40) is used to determine the error location $k$, where $k + i = N$ and equation (41) is used to determine error magnitude.

**2.** If 2nd check byte and $k^{th}$ information byte are erroneous, then the syndrome equations are

$$S_0 = E_k; \quad S_1 = T^i E_k \oplus e_1; \quad S_2 = T^{2i} E_k; \quad S_3 = T^{3i} E_k \tag{42}$$

$$S_3 = T^i S_2; \quad S_2 = T^{2i} S_0; \quad S_1 \neq T^i S_0 \tag{43}$$

$$E_k = S_0 \tag{44}$$

Equation (43) is used to determine the error location $k$ and equation (44) is used to determine error magnitude.

**3.** If 3rd check byte and $k^{th}$ information byte are erroneous, then the syndrome equations are

$$S_0 = E_k; \quad S_1 = T^i E_k; \quad S_2 = T^{2i} E_k \oplus e_2; \quad S_3 = T^{3i} E_k \tag{45}$$

$$S_1 = T^i S_0; \quad S_3 = T^{2i} S_1; \quad S_3 \neq T^i S_2 \tag{46}$$

$$E_k = S_0 \tag{47}$$

Equations (46) are used to determine the error location $k$ and equation (47) is used to determine error magnitude.

**4.** If 4th check byte and $k^{th}$ information byte are erroneous, then the syndrome equations are

$$S_0 = E_k; \quad S_1 = T^i E_k; \quad S_2 = T^{2i} E_k; \quad S_3 = T^{3i} E_k \oplus e_3 \tag{48}$$

$$S_1 = T^i S_0; \quad S_2 = T^{2i} S_0; \quad S_3 \neq T^i S_2 \tag{49}$$

$$E_k = S_0 \tag{50}$$

Equation (49) is used to determine the error location $k$ and equation (50) is used to determine error magnitude.

So, the proposed scheme can identify error locations unambiguously if one error is in information byte and the other is in check byte. Thus it overcomes the limitation of scheme [1] given in section 3.2.

## 5 Conclusion

The paper presents an improved scheme for the double byte error correcting code using CA which overcomes the weakness and limitation of the existing scheme. The proposed scheme can determine error locations which is independent of erroneous byte position and number of errors provided number of errors are less than or equal to error correcting capability of the code. The proposed code is much simpler and requires much less hardware for decoding compared with conventional RS code having two-byte error correcting capability.

## References

1. Roy Chowdhury, D., Sen Gupta, I., Chaudhuri, P.P.: CA-Based Byte Error- Correcting code. IEEE Transaction on Computers 44(3), 371–382 (1995)
2. Nandi, S., Rambabu, C., Chaudhuri, P.P.: A VLSI Architecture for Cellular Automata Based Reed-Solomon Decoder. In: *4th* International Symposium on Parallel Architecture, Algorithm and Networks, Australia, pp. 158–165 (1999)
3. Sasidhar, K., Chattopadhyay, S., Chaudhuri, P.P.: CAA Decoder for Cellular Automata Based Byte Error Correcting Code. IEEE Transaction on Computers 45(9), 1003–1016 (1996)
4. Sarwate, D.V., Shanbhag, N.R.: High-Speed Architectures for Reed-Solomon Decoder. IEEE Transaction on VLSI systems 9(5), 641–655 (2001)
5. Baek, J.H., Sunwoo, M.H.: New Degree Computationless Modified Euclid Algorithm and Architecture for Reed-Solomon Decoder. IEEE Transaction on VLSI systems 14(8), 915–920 (2006)
6. Chaudhuri, P.P., Roy Chowdhury, D., Nandi, S., Chattopadhyay, S.: Additive Cellular Automata: Theory and Applications, vol. 1. IEEE Computer Socity press, Los Alamitos (1997)
7. Rao, T.R.N., Fujiwara, E.: Error-Control Coding for Computer Systems. Prentice-Hall, Englewood Cliffs (1989)
8. Lin, S., Costello, D.J.: Error Control Coding: Fundamentals and Applications. Prentice-Hall, Inc., Englewood Cliffs (1983)

# Nonlinear Pseudorandom Sequences Based on 90/150 LHGCA*

Un-Sook Choi[1], Sung-Jin Cho[2,**], Han-Doo Kim[3], Yoon-Hee Hwang[4], and SeokTae Kim[5]

[1] Department of Multimedia Engineering, Tongmyong University
Busan 626-847, Korea
choies@tu.ac.kr
[2] Division of Mathematical Sciences, Pukyong National University
Busan 608-737, Korea
sjcho@pknu.ac.kr
[3] School of Computer Aided Science, Institute of Basic Science
Inje University, Gimhae 621-749, Korea
mathkhd@inje.ac.kr
[4] Department of Information Security, Graduate School, Pukyong National University
Busan 608-737, Korea
yhhwang@pknu.ac.kr
[5] Division of Electronic, Computer and Telecommunication Engineering,
Pukyong National University, Busan 608-737, Korea
setakim@pknu.ac.kr

**Abstract.** In a stream cipher the plaintext digits are encrypted one at a time, and the transformation of successive digits varies during the encryption. LFSRs produce sequences having large periods and good statistical properties, and are readily analyzed using algebraic techniques. But the output sequences of LFSRs are also easily predictable, if we know proper successive output sequences in output sequences. In this paper, we give a new method which generates nonlinear sequences using maximum-length cellular automata.

## 1 Introduction

Pseudorandom sequences have many applications in cryptography and communication engineering. The inherent simplicity of LFSRs, the ease and efficiency of implementation, some good statistical properties of the LFSR sequences, and the algebraic theory underlying these devices turn them into natural candidates for use in the construction of pseudorandom generator, targeted to the implementation of efficient stream cipher cryptographic strength. On the other hand, some of the attractive properties listed above are also the reason for the failure

* This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD)(KRF-2006-331-D00458).

** Corresponding author.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 471–477, 2008.

of many of these constructions. The shrinking generator which is one of clock-controlled generators, is well-known pseudorandom sequence generator, proposed by Coppersmith et al.[1]. It is a very simple generator with good cryptographic properties.

Sabater et al.[2] proposed the algorithm to convert the shrinking generator into a 90/150 group CA-based linear model which is simple and can be applied to shrinking generators in a range of practical interest. But they didn't consider that individual cells of CA(Cellular Automata) can generate sequences having the same characteristic polynomial[3] and CA can generate sequences having short period according to seed vectors, even if the period of 90/150 group CA and the period of sequence generated by shrinking generator are same.

In this paper, we propose a new method which generates nonlinear pseudorandom sequences using two maximum-length 90/150 LHGCA obtained by Cho et al.'s algorithm[4]. The generator which generates these sequences is possible to overcome spatial weak points of the interleaved sequence generator proposed by Gong[5]. Unlike the method proposed by Sabater et al.[2], the new sequence generator can generate nonlinear sequences whose cycle lengths are always same for a given initial state. The nonlinear pseudorandom sequence obtained by our method has a larger period and a higher linear complexity than the shrunken sequence generated by LFSRs.

## 2 Preliminaries

CA is an array of cells where each cell is in any one of the permissible states. At each discrete time steps the next state of particular cell is usually assumed to depend only on itself and on its two neighbors (three-neighborhood dependency) for a local neighborhood CA. The state of the $i$th cell at time $(t+1)$ can thus be denoted as:

$$q_i(t+1) = f(q_{i-1}(t), q_i(t), q_{i+1}(t))$$

where $f$ represents the combinatorial logic and it is called next state function.

In this paper we deal with 90/150 linear hybrid group cellular automata (LHGCA).

Characterizations of linear CA based on matrix algebraic tool have been reported in [6]. The matrix algebraic tool employing minimal polynomial and characteristic polynomial of the state transition matrix of CA showed various interesting features of CA behaviour. The most effective application of linear group CA has been proposed in the field of pseudorandom pattern generation, since many researchers[7] showed that maximum-length CA whose all nonzero states lying in a single cycle produce high quality pseudorandom patterns. It has been established that the maximum-length cycle in the state transition diagram of 90/150 CA can be produced only if the characteristic polynomial is primitive([3], [4]). In this paper only one-dimensional maximum-length 90/150 LHGCA are considered. 90/150 LHGCA is completely specified by which cells use rule 90

and 150. A natural form for the specification is an $n$-tuple $< d_1, d_2, \cdots, d_n >$ , called the *rule vector*, where

$$d_i = \begin{cases} 0, & \text{if cell } i \text{ uses rule 90} \\ 1, & \text{if cell } i \text{ uses rule 150} \end{cases}$$

The shrinking generator was introduced by Coppersmith et al.[1]. Nevertheless, due to its simplicity and provable properties, it is a promising candidate for high-speed encryption application. The shrinking generator is a well-known keystream generator composed of two LFSRs. A control LFSR $R_1$ is used to select a portion of output sequence of a second LFSR $R_2$. Therefore the keystream produced is a shrunken version of the output sequence of $R_2$.

According to [1], let $R_1$ and $R_2$ be maximum-length LFSRs whose characteristic polynomials are primitive, of lengths $L_1$ and $L_2$, respectively, and let $\{k_i\}$ be an output sequence of the shrinking generator formed by $R_1$ and $R_2$. If $gcd(L_1, L_2) = 1$, the period of $\{k_i\}$ is $(2^{L_2} - 1) \cdot 2^{L_1 - 1}$ and its linear complexity $LC$ satisfies the following inequality $L_2 \cdot 2^{L_1 - 2} < LC \leq L_2 \cdot 2^{L_1 - 1}$.

## 3 Interleaved Sequences

The interleaved sequences were introduced by Gong[5]. Interleaved sequences are constructed by taking sequences and combining them under control of a shift sequence $\mathbf{e}$. Let $\mathbf{a} = \{a_i\}$ be a binary sequence. If $\mathbf{a}$ is a periodic sequence with period $l$, then we write $\mathbf{a}$ by $[a_0, a_1, \cdots, a_{l-1}]$. The left shift operator $L$ on $\mathbf{a}$ is defined as $L(\mathbf{a}) = \{a_1, a_2, \cdots\}$, i.e. the left shift operator $L$ when applied to a sequence will shift the sequence to the left by one position. For $L^i(\mathbf{a}) = \{a_i, a_{i+1}, a_{i+2}, \cdots\}$, $i$ is said to be *phase shift* of $\mathbf{a}$. Two periodic sequences $\mathbf{a}$ and $\mathbf{b}$ are shift equivalent if there exists an integer $k$ such that $a_i = b_{i+k}$ for all $i = 0, 1, 2, \cdots$. Let $\mathbf{u} = [u_0, u_1, u_2, \cdots, u_{st-1}]$ be a binary sequence with period $st$, where $s$ and $t$ are integers greater than 1. We can arrange the elements of the sequence $\mathbf{u}$ into an $s \times t$ array as follows:

$$A = \begin{pmatrix} u_0 & u_1 & \cdots & u_{t-1} \\ u_t & u_{t+1} & \cdots & u_{2t-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{(s-1)t} & u_{(s-1)t+1} & \cdots & u_{st-1} \end{pmatrix}$$

If each column vector of $A$ is either phase shift of a binary sequence $\mathbf{a}$ of period $s$, or zero sequence, then $\mathbf{u}$ is called an $(s, t)$ *interleaved sequence*. Let $A_j$ be the $j$th column vector of $A$ which is the matrix form of $\mathbf{u}$, then $A = (A_0, A_1, \cdots, A_{t-1})$. $A_j$ is the transpose of either $L^{e_j}(\mathbf{a})$ , or $(0, \cdots, 0)$, where $e_j$ is the phase shift of $\mathbf{a}$. If $A_j = (0, \cdots, 0)^t$, then we denote $e_j = \infty$. $\mathbf{u}$ is called an $(s, t)$ *interleaved sequence associated with* $(\mathbf{a}, \mathbf{e})$, and $\mathbf{e} = (e_0, e_1, \cdots, e_{t-1})$ is called a *shift sequence* of $\mathbf{u}$.

This generator has some troubles that it must be paralleled $t$ LFSRs with period $s$ to generate a $(s, t)$ interleaved sequence. In this paper we propose the

method which is possible to overcome spatial weak points of the interleaved sequence generator. This method employs a maximum-length 90/150 LHGCA whose characteristic polynomial is primitive. High quality pseudorandom sequences can be generated from the CA. This is due to the apparent random phase shift of the output bit sequences from its various stages that are cell positions. Each cell position generates pseudorandom sequences. Unlike LFSRs, the phase shift is generally different between stages of a CA. Schemes finding phase shifts of maximum-length 90/150 LHGCA were proposed in [3].

## 4 SI Sequence Based on 90/150 LHGCA

Sabater et al.[2] considered the linear model of shrinking generator described in [8] in terms of 90/150 LHGCA. They proposed a synthesis algorithm for the 90/150 LHGCA which is equivalent to any shrinking generator. This LHGCA is formed by concatenations of basic maximum-length 90/150 LHGCA and their mirror images, with one or two modification in each LHGCA component. The characteristic polynomial of the 90/150 LHGCA obtained by the algorithm is the same as the one of the original shrinking generator. Since the number of concatenations is $2^{L_1-1}$($L_1$ is the length of $R_1$) and the length of the basic primitive 90/150 LHGCA is $L_2$, the required length of the 90/150 LHGCA is given by $L = L_2 2^{L_1-1}$. For example, consider a shrinking generator with the following component LFSRs: a selector register $R_1$ with length $L_1 = 3$ and the second register $R_2$ with length $L_2 = 4$. Then the period of the shrunken sequence is $(2^4 - 1)2^{(3-1)} = 60$. In order to generate the same sequence as shrunken sequence obtained by the shrinking generator, it needs the 90/150 LHGCA whose characteristic polynomial is $p(x)^N$, where $p(x)$ is a primitive polynomial with degree 4 and $2^{3-2} < N \leq 2^{3-1}$. But all cycles of 90/150 LHGCA synthesized are not equal to the period of the 90/150 LHGCA though the period of the 90/150 LHGCA is equal to the period of the shrunken sequence generated by the shrinking generator. Table 1 shows the configuration and cycle structure of 90/150 LHGCA whose characteristic polynomials are $(x^4+x^3+1)$,$(x^4+x^3+1)^2$ and $(x^4+x^3+1)^4$. In Table 1, $a(b)$ means that the number of cycles with length $b$ is $a$.

**Table 1.** Configuration and cycle structure of the 90/150 LHGCA synthesized

| characteristic polynomial | configuration | cycle structure |
|---|---|---|
| $x^4+x^3+1$ | 1101 | 1(1), 1(15) |
| $(x^4+x^3+1)^2$ | 11000011 | 1(1), 1(15), 8(30) |
| $(x^4+x^3+1)^4$ | 1100001001000011 | 1(1),1(15),8(30),1088(60) |

The period of the sequence generated by 90/150 LHGCA **C** with some initial state whose characteristic polynomial is of the form $p(x)^N$ is not always equal to the period of **C**. It means that the 90/150 LHGCA which is equivalent to any

shrinking generator is not secure. To overcome this problem, we present a method which generates a new nonlinear pseudorandom sequence. Each cell position of 90/150 maximum-length LHGCA generates a pseudorandom sequence. In addition the phase shift is generally different between stages of a CA. The new sequence generator compose of two 90/150 maximum-length LHGCAs: a selector 90/150 maximum-length LHGCA $\mathbf{C_1}$ that produces a sequence used to decimate the sequences generated by the other 90/150 maximum-length LHGCA $\mathbf{C_2}$.

Let $T_1$ (resp. $T_2$) be the state transition matrix for a given $m$-cell (resp. $n$-cell) maximum-length 90/150 LHGCA and let $u_0$ (resp. $v_0$) be the initial state of $T_1$ (resp. $T_2$). Then we obtain a $(2^m-1)\times m$ (resp. $(2^n-1)\times n$) matrix $A$ (resp. $B$) consisting of $m$ (resp. $n$) independent pseudorandom sequences generated by $T_1$ (resp. $T_2$) as its columns. Here $gcd(2^m-1, 2^n-1) = 1$. Define a $(2^n-1)(2^m-1)\times(n+1)$ matrix $S = (s_{ij})$ as follows:

$$S = \left(\begin{array}{c|c} B & A_j \\ \vdots & \vdots \\ B & A_j \end{array}\right),$$

where $A_j$ is the $j$th column of $A$.

Let $S_I^*$ be the $2^{m-1}(2^n-1)\times(n+1)$ matrix obtained from $S$ by discarding the $i$th row of $S$ if $s_{i,m+1} = 0$. Let $S_I$ be the $2^{m-1}(2^n-1)\times n$ matrix obtained by deleting the $(n+1)$th column of $S_I^*$. Then $S_I$ is the following matrix:

$$S_I = \begin{pmatrix} k_0 & k_1 & \cdots & k_{n-1} \\ k_n & k_{n+1} & \cdots & k_{2n-1} \\ \vdots & \vdots & \ddots & \vdots \\ k_{(2^{m-1}(2^n-1)-1)n} & k_{(2^{m-1}(2^n-1)-1)n+1} & \cdots & k_{2^{m-1}(2^n-1)n-1} \end{pmatrix}$$

**Definition 4.1.** Let $\mathbf{K} = [k_0, k_1, \cdots, k_{2^{m-1}(2^n-1)n-1}]$ be a sequence obtained by $S_I$ with period $2^{m-1}(2^n-1)n$. We call $\mathbf{K}$ a $(2^{m-1}(2^n-1), n)$ *shrunken interleaved sequence* (SI sequence).

**Example 4.2.** Consider a SI sequence generator with the following two component maximum-length 90/150 LHGCA $\mathbf{C_1}, \mathbf{C_2}$ :
1. Let $\mathbf{C_1}$ be the maximum-length 90/150 LHGCA with length $m = 2$, rule vector $< 01 >$ and initial state $(0, 1)$. Then

$$T_1 = \begin{pmatrix} 0\,1 \\ 1\,1 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} 0\,1 \\ 1\,1 \\ 1\,0 \end{pmatrix}$$

2. Let $\mathbf{C_2}$ be the maximum-length 90/150 LHGCA with length $n = 3$, rule vector $< 011 >$ and initial state $(0, 0, 1)$. Then

$$T_2 = \begin{pmatrix} 0\,1\,0 \\ 1\,1\,1 \\ 0\,1\,1 \end{pmatrix} \quad \text{and} \quad B^t = \begin{pmatrix} 0\,0\,1\,0\,1\,1\,1 \\ 0\,1\,0\,1\,1\,1\,0 \\ 1\,1\,0\,0\,1\,0\,1 \end{pmatrix}$$

where $B^t$ is the transpose of $B$. Since $A_2^t = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}$, we obtain the following matrix $S$:

$$S^t = \begin{pmatrix} 0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1 \\ 0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0 \\ 1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1 \\ 1\,1\,\underline{0}\,1\,1\,\underline{0}\,1\,1\,\underline{0}\,1\,1\,\underline{0}\,1\,1\,\underline{0}\,1\,1\,\underline{0}\,1\,1\,\underline{0}\,1\,1\,\underline{0} \end{pmatrix}$$

Hence we obtain the $(14, 3)$ SI sequence as the following:

$$\mathbf{K} = [00101101011110100110001011010101110011111 0]$$

**Theorem 4.3.** Let $\mathbf{C_1}$ be a maximum-length 90/150 LHGCA with length $m$ and let $\mathbf{C_2}$ be a maximum-length 90/150 LHGCA with the $n$ degree minimal polynomial $f(x)$, where $gcd(2^m - 1, 2^n - 1) = 1$. Then
(1) The minimal polynomial $m(x)$ of the SI sequence is of the following form:

$$m(x) = [f^*(x^n)]^N,$$

where $2^{m-2} < N \leq 2^{m-1}$ and $f^*(x)$ is the reciprocal of $f(x)$.

(2) The linear complexity $LC$ of the SI sequence satisfies the following

$$2^{m-2}n^2 < LC \leq 2^{m-1}n^2$$

## 5 Conclusion

In this paper, we proposed a new method which generates nonlinear pseudorandom sequences using two maximum-length 90/150 LHGCA. The generator which generates these sequences is possible to overcome spatial weak points of the interleaved sequence generator proposed by Gong. Unlike the method proposed by Sabater et al., the SI sequence generator can generate nonlinear sequences whose cycle lengths are always same for a given initial state. The SI sequence obtained by our method has a larger period and a higher linear complexity than the shrunken sequence generated by LFSRs.

## References

1. Coppersmith, D., Krawczyk, H., Mansour, Y.: The shrinking generator. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
2. Sabater, A.F., Martinez, D.G.: Modelling nonlinear sequence generators in terms of linear cellular automata. Applied Mathematical Modelling 31, 226–235 (2007)
3. Cho, S.J., Choi, U.S., Hwang, Y.H., Pyo, Y.S., Kim, H.D., Kim, K.S., Heo, S.H.: Computing phase shifts of maximum-length 90/150 cellular automata sequences. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 31–39. Springer, Heidelberg (2004)

4. Cho, S.J., Choi, U.S., Kim, H.D., Hwang, Y.H., Kim, J.G., Heo, S.H.: New synthesis of one-dimensional 90/150 linear hybrid group cellular automata. IEEE Trans. Comput-Aided Des. Integr. Circuits Syst. 26-9, 1720–1724 (2007)
5. Gong, G.: Theory and applications of $q$-ary interleaved sequences. IEEE Transaction on Information Theory 41-2, 400–411 (1995)
6. Das, A.K., Chaudhuri, P.P.: Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation. IEEE Trans. Comput. 42, 340–352 (1993)
7. Tsalides, P., York, T.A., Thanailakis, A.: Pseudorandom number generators for systems based on linear cellular automata. IEE Proc(Part E) Computers Digital Techniques 138, 241–249 (1991)
8. Sarkar, P.: Computing Shifts in 90/150 cellular automata sequences. Finite Fields Their Appl. 42, 340–352 (2003)

# Cryptographically Strong S-Boxes Based on Cellular Automata

Miroslaw Szaban[1] and Franciszek Seredynski[2]

[1] Institute of Computer Science, University of Podlasie
3 Maja 54, 08-110 Siedlce, Poland
[2] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
mszaban@ap.siedlce.pl, sered@ipipan.waw.pl

**Abstract.** Substitution boxes (S-boxes) are the main elements of block ciphers, wide used in modern cryptography. In this paper we propose a new method to create S-boxes, which is based on application of Cellular Automata (CA). We present results of testing CA-based S-boxes, which confirm that CA are able to realize efficiently Boolean functions corresponding to classical S-boxes. Proposed CA-based S-boxes offer cryptographic properties comparable or better than classical S-box tables.

**Keywords:** Cellular Automata, S-Boxes, Block Cipher, Cryptography, Boolean Functions.

## 1 Introduction

Cryptography plays an important role in security of data in the modern world. Two main cryptography systems are used today to provide a secure communication: secret and public-key systems. An extensive overview of currently known or emerging cryptography techniques used in both types of systems can be found in [10]. The main concern of this paper are cryptosystems with a secret key. The main interest of this work are Boolean functions used in S-boxes and applied in efficient algorithms in secret key systems. Many known secure standards of symmetric key cryptography, such as e.g. [3], [4], use efficient and secure algorithms working on the base of S-boxes. S-boxes are ones of the most important components of block ciphers.

In the next section the concept of the S-box and its most known applications are presented. Section 3 describes a few cryptographic criteria to examine Boolean functions realized with S-boxes. Section 4 presents the concept of CA. In section 5 the idea of substitution of S-boxes by CA is proposed. Section 6 presents results of examination of CA-based S-boxes, their quality measured by efficient criteria and comparison with earlier proposals. The last section concludes the paper.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 478–485, 2008.

## 2 S-Boxes in Cryptography

S-box is a function $f$, which from each of $n$ Boolean input values of $B^n$ block consisting of $n$ bits $b_i$ $(i \leq n)$ generates some $k$ Boolean output values called $B^k$ block consisting of $k$ bits $b'_j$ $(j \leq k$ and $k \leq n)$ : $f : B^n \rightarrow B^k$, what corresponds to the mapping $(b_0, b_1, ..., b_n) \rightarrow (b'_0, b'_1, ..., b'_k)$. When $n$ is equal to $k$, the function $f$, from $n$ different input values maps $n$ different outputs values, and such a S-box is called bijective [5].

One of well known application of S-boxes is applying them in Data Encryption Standard (DES) as the "heart" of this algorithm [3]. In DES algorithm 64 input bits are transformed into 64 output bits. The main part of DES algorithm is a function $f$ (see, [3]), which uses 8 wide known S-boxes. Each of these eight S-boxes realize functions *S1, ..., S8*, which collectively transform the 48 bit input block into 32 bit output block. These functions are tables composed of 16-columns and 4-rows. Each function takes a 6-bit block as input and yields a 4-bit block as output.

Let us consider the function *S1* represented as a table. Suppose that the input block of this function is the block $B^6$, e.g. *110010*. Two bits from $B^6$, the first and the last one (e.g. *10*) define row *2* of the *S1* block. Four middle bits *1001* define the column *9* of the *S1* block. The intersection of the column *9* and the row *2* points the number *12*, e.g. *1100*, and these bits are considered as the $B^4$ output block.

S-boxes are also used in modern symmetric key cryptography systems, e.g. in the new standard Advanced Encryption Standard (AES) (see, [4]). AES is successor of DES, and provides much better cryptographic quality then DES.

## 3 The Most Important Cryptographic Criteria for Boolean Functions

In our study we propose to use CA as a function which can be characterized by the same properties and realize the same functions as wide known S-boxes. The quality of S-boxes designed with use of CA must be verified by required properties of S-boxes. The most important theorems for this purpose are recalled from cryptographic literature [8], [1], [5], [2].

A Boolean function $f : Z_2^n \rightarrow Z_2$ maps $n$ binary inputs to a single binary output. The number of possible outputs is equal to $2^n$.

The non-linearity $N_f$ of Boolean function $f$ is the minimal distance to the set of affine functions and is calculated as:

$$N_f = \frac{1}{2}(2^n - WH_{max}(f)), \tag{1}$$

where $WH_{max}(f)$ denotes Walsh Hadamard Transform. The more higher is the non-linearity of observed ciphers ($WH_{max}$ is low) the cipher is more difficult to cryptanalysis.

Another important property of stream ciphers is autocorrelation $AC_f$. Autocorrelation is similar to correlation, but polar form $f(x)$ correlates with polar

form $f(x \otimes s)$ its *shifted version*. The autocorrelation of a Boolean function $f$ is absolute maximum value of any Autocorrelation Transforms, and is denoted by the equation:

$$AC_f = max_{s \neq 0} |\sum_x \hat{f}(x)\hat{f}(x \otimes s)|, \quad (2)$$

where $s \in Z_2^n - \{0\}$. Lower is the autocorrelation of observed ciphers the cipher is more difficult to cryptanalysis.

## 4 Cellular Automata

One dimensional (1D) CA is in the simplest case a collection of two-state elementary cells arranged in a lattice of the length $N$, and locally interacting in a discrete time $t$. For each cell $i$ called a central cell, a neighbourhood of a radius $r$ is defined, consisting of $n_i = 2r + 1$ cells, including the cell $i$. When considering a finite size of CA, and a cyclic boundary condition is applied, it results in a circle grid.

It is assumed that a state $q_i^{t+1}$ of a cell $i$ at the time $t + 1$ depends only on states of its neighbourhood at the time $t$, i.e. $q_i^{t+1} = TF(q_i^t, q_{i1}^t, q_{i2}^t, , q_{in}^t)$, and a transition function $TF$, called a rule, which defines a rule of updating a cell $i$. A length $L$ of a rule and a number of neighbourhood states for a binary uniform CA is $L = 2^n$, where $n = n_i$ is a number of cells of a given neighbourhood, and a number of such rules can be expressed as $2^L$. For CA with e.g. $r = 2$ the length of a rule is equal to $L = 32$, and a number of such rules is $2^{32}$ and grows very fast with $L$. CA for systems with a secrete key were first studied by Wolfram [12] and later by many other authors (see, e.g. [11]).

## 5 Cellular Automata and Constructing S-Boxes

A classical S-box is a function expressed as a table composed of natural numbers. A quality of S-boxes are measured with use of different functions which examine its different properties [6], [7], [1], [9]. Cryptographic literature shows us many examples and methods of searching of S-box tables. Each of these methods in the consequence still search for the combinations of numbers in the table. We would like to propose another method without using a table as the base of the S-box. In our approach CA are expected to perform the same tasks as S-boxes. In this proposition creation of specific table is unnecessary, because the CA as a tool equivalent to Universal Turing's Machine (see, [13]) can realize any function, in particular functions related to S-boxes.

The CA-based S-box which we propose can be seen as a vector composed of the following elements:

- a number of CA cells
- an initial state of CA
- rule/rules applied to CA

- a number of CA time steps
- input/output bits of CA-based S-box.

Selected cells of CA (in its initial state) serve as input bits of the S-box, and the same cells, after declared time steps, are considered as the output of the S-box. To construct CA performing the S-box function it is necessary to find appropriate CA rules and verify produced results according to the S-box functions criteria.

## 6 Designing CA-Based S-Boxes and Their Analysis

### 6.1 Searching for CA Rules and Construction of CA-Based S-Boxes

We start from considering uniform CA consisting of the 8 cells, and examining all 256 rules ($r = 1$). As a bijective S-box, CA with a size equal to 8 cells (an initial CA state corresponds to S-box input) is examined in time steps $t$ (a CA state at this moment corresponds to the S-box output) equal to $5, 6, 7, 8, 30, 50, 100$. Non-linearity and autocorrelation values of all examined 256 CA rules were calculated to select the best CA rules. The best found CA rules at this stage of experiments were the following ones: 30, 57, 86, 99, 135, 149. These rules provide non-linearity and autocorrelation values higher then the other examined rules. Scores of the best CA rules for non-linearity are in the range [101, 108] and for autocorrelation in the range [56, 80].

Generally, higher value of $N_f$ means that the S-box provides higher quality related to non-linearity criterion, in opposite to autocorrelation of the S-box in which higher quality of the S-box corresponds to lower value of $AC_f$. The best (worst) theoretical values of non-linearity and autocorrelation corresponding to this 8 bit CA-based S-box are equal to 128 (0) for non-linearity and 0 (256) for autocorrelation, respectively. We will show in the paper that values of non-linearity and autocorrelation obtained in experiments can be successfully compared with results (lower or similar) presented by [6], [7] and [1]. On the base of these preliminary results, we will be able conclude that behavior of CA which implement the S-box provides a good quality.

During all conducted experiments presented in this paper we assume that we have to do with CA-based S-boxes with 8 inputs and 8 outputs, despite the fact that a size of CA can be larger than 8. The next step of the research was a verification of CA rules quality from the point of view of bijectivity. When the CA size is equal to 8 input/output cells then diversity of outputs obtained from each of possible inputs is lower than 20%. Therefore, CA-based S-box was examined with a number of cells ranging from 8 to 500, because larger CA size provides higher diversity. During experiments the problem of allocation of 8 examined input/output bits in larger CA arises. An initial state of large CA was randomly selected, but input/output bits of CA-based S-box need to be determined. These 8 bits (main bits) are always located in one block (first bits in CA - for simplicity) as a part of CA's state. Other bits of large CA forms background, an environment for evolution of the block of the main bits.

Another idea on how to arrange bits in the background is to locate bits separately in CA cells. In the series of tests main bits were located in CA cells in the distance 1, 2, 5 and 10 cells from each other. The remaining CA cells were set randomly. Results of experiments (not presented in this paper) show that in fact the way of arranging main bits in the background has small influence on diversification of outputs - the difference is of the range of 2%.

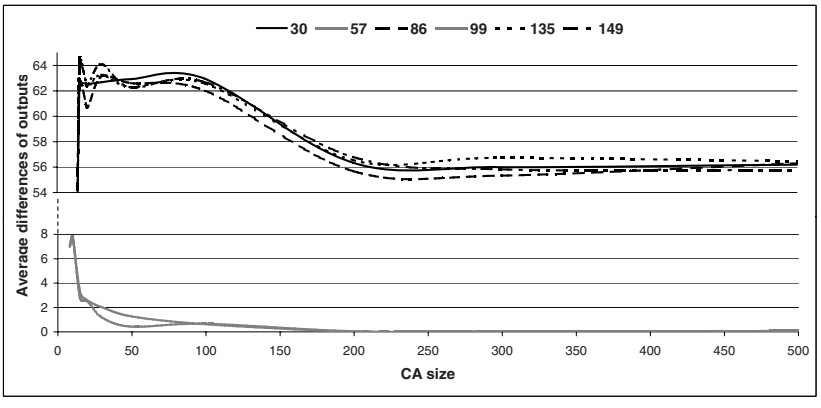



**Fig. 1.** The diversification of outputs (in %) given by CA (after 100 time steps) with rules: 30, 57, 86, 99, 135, 149 for CA's sizes in the range [8, 500] averaged over 1000 CA runs

Figure 1 presents results of the verification in bijectivity condition. One can see that for rules 57 and 99 the maximal number of different outputs of CA-based S-box is equal to only 8% values out of 256 values. It means that these rules are not suitable for the purpose of the S-box. Therefore, in our next experiments these rules will not be taken more into consideration. Figure 1 shows also results for rules 30, 86, 135 and 149. One can see that results for these rules are much better than for the previous ones: the maximal number of different outputs is equal to about 63% and this feature concerns a relatively wide size of CA up to 100 cells. These results seem to be promising and therefore for the next study we focused on CA with 100 cells. Reassuming, in all our next experiments CA size will be equal to 100, but the number of the main bits (input/output of CA-based S-box) will be equal to 8. Large CA (100 cells) will be evaluated in discrete time steps (100 time steps), but we will observe and examine only the main bits (8 input/output bits) in large CA.

The last conducted experiments concerned examination of CA rules which passed both verification procedures. During two previous experiments rules 30, 86, 135 and 149 were selected as the best rules from the set of CA rules with neighborhood radius equal to 1. These rules were next studied to define the final quality with a new CA size, i.e. values of non-linearity and autocorrelation were calculated. Results of the study are presented in the next section.

### 6.2 Comparison of CA-Based S-Boxes with Classical S-Boxes

In the 1000 runs with different random background of bits in CA with 100 cells, the main bits were arranged in one block. In each experiment CA was running 100 time steps. Rules were selected from the set of rules with neighborhood radius $r = 1$. Table 1 presents results of our experiments, and compares them with ones obtained by Clark [1] and Millan [6].

**Table 1.** Comparison of results of non-linearity $N_f$ and autocorrelation $AC_f$ (the best values are in the bold) obtained with use of CA (with CA size equal to 100 cells, in 1000 CA runs, each run of 100 time steps), with ones by Clark and Millan ($*$ - no data).

| Method | Observed boundary values $(N_f, AC_f)$ | The best values in single runs $(N_f, AC_f)$ | The worst values in single runs $(N_f, AC_f)$ |
|---|---|---|---|
| CA with rule 30 | (90, 112) | (105, **44**), (**110**, 48), (108, 48) | (98, 112), (90, 88) |
| CA with rule 86 | (91, 96) | (108, **48**), (106, 48), (**109**, 52) | (104, 96), (91, 76) |
| CA with rule 135 | (91, 104) | (108, **48**), (106, 48), (**109**, 52) | (96, 104), (91, 84) |
| CA with rule 149 | (91, 100) | **(109, 44)**, (108, 48), (106, 48) | (99, 100), (91, 76) |
| Clark et. al [1] | (90, $*$) | (100, $*$), ($*$, 80), ($*$, 102) | (90, $*$) |
| Millan et. al [6] | (80, $*$) | (98, $*$), ($*$, 98), ($*$, 100) | (80, $*$) |

Observed boundary values presented in Table 1 mean that in all CA runs we could not find a value of $N_f$ which is lower than 90, and the higher value for $AC_f$ is not greater than 112. The best and the worst qualities presented in 3-rd and 4-h column of Table 1 correspond to single runs, selected from the set of 1000 runs. Result of these runs are shown in each row. One can see that the best value of non-linearity is equal to 110 and corresponds to the rule 30. Low level of autocorrelation for this rule is also provided. On the other hand, the rule 149 is characterized by the best value of autocorrelation equal to 44, with high level of non-linearity.

In [6] and [1] for the same size (8 input bits, 8 output bits) of the S-box, S-boxes were found with values of $N_f$ in the ranges from [80, 100] and [90, 100], respectively (see, Figure 2). The best autocorrelation values $AC_f$ presented in [6] and [1] are equal to values from the sets $\{98, 100\}$ and $\{80, 102\}$, respectively (see, Table 1). If we compare these results with our results we can conclude that (a) even our boundary values of $(N_f, AC_f)$ are comparable with their results, and (b) our the best results are better than pointed in their study.

Our boundary values of $N_f$ equal to $\{90, 91\}$ for appropriate CA rules are in ranges of values presented in [6] and [1]. Similarly our boundary values of $AC_f$ equal to $\{96, 100, 104, 112\}$ for appropriate CA rules are comparable with their best results.

On the other hand, our the best results presented in the 3-rd column of Table 1 are characterized by much higher values of non-linearity $N_f$ (108, 109 and 110 for appropriate CA rules) and are better than the value equal to 100 found in

[6] and [1]. Also, the best results of autocorrelation $AC_f$ presented in the 3-rd column of Table 1 characterizing by much lower values (44 and 48 for appropriate CA rules) are better than values equal to 98 and 80, presented in [6] and [1], respectively.

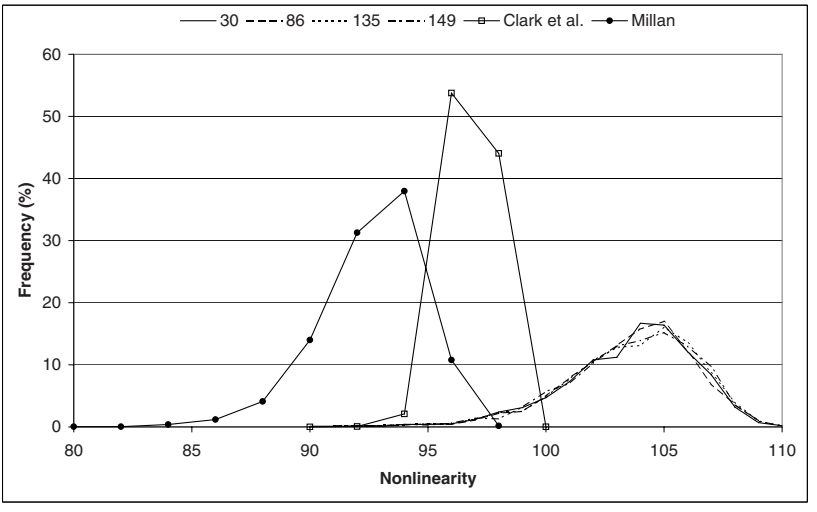



**Fig. 2.** Frequency (in %) of distribution for non-linearity in our CA-based S-boxes with rules: 30, 86, 135, 149, compared with Clark et al. results [1] and Millan results [6]

Frequency of non-linearity distribution in our results and results given by Clark [1] and Millan [6] are presented in Figure 2. In 1000 runs most of CA-based S-boxes for $N_f$ ranging from [98, 108] and the best CA-based S-boxes give values even equal to 110. These ranges of $N_f$ values keep better quality than ranges [80, 100] and [90, 100] for results of non-linearity presented in [6] and [1]. Also, the most of our results concerning autocorrelation $AC_f$ (not presented in this paper) ranging from [52, 84], and are better than results 98 and 80 presented in [6] and [1], respectively.

Lets observe non bijective S-boxes. These $8 \times m$ S-boxes, from 8 inputs provide $m$ outputs ($m = 2, 3, 4, 5, 6, 7$). The quality ($N_f$, $AC_f$) of these S-boxes presented in literature ([6], [7], [1] and [9]), shows that exists a relationship between values of non-linearity and autocorrelation of the best S-boxes and the number of output bits. If the number of output bits grows then the quality of S-boxes goes down (i.e., the value of non-linearity goes down and the value of autocorrelation grows). We can conclude that non-linearity and autocorrelation of our proposed CA, which realize the S-box functions provides for some classes of S-boxes higher values than obtained in [6], [7], [1] and [9], for $8 \times m$ S-boxes. Our the best CA (see, Table 1) keep better quality (higher non-linearity, lower autocorrelation) than result (108, 56) presented in [1] and the result (110, 56) presented in [9] for $8 \times 5$ S-boxes.

## 7 Conclusions and Future Work

The paper presents a new idea of creating S-boxes using CA approach. Applying CA to create S-boxes eliminates inefficient memory consuming tables which are used in the classical approach, and gives a possibility to create S-boxes in compact and elegant way. CA from input block of bits generate output block of bits and this output is evaluated by the same examine criteria like the traditional S-box. Obtained preliminary results are very promising. Conducted experiments have shown that CA-based S-box is characterized by a high non-linearity and low autocorrelation. These values correspond to values related to classical S-boxes or outperform them. The open issue is the question of enlarging the maximal value of the number of possible output values of CA-based S-box. This issue is the subject of a current research.

## References

1. Clark, J.A., Jacob, J.L., Stepney, S.: The Design of S-Boxes by Simulated Annealing. New Generation Computing 23(3), 219–231 (2005)
2. Dowson, E., Millan, W., Simpson, L.: Designing Boolean Functions for Cryptographic Applications. Contributions to General Algebra 12, pp. 1–22. Verlag Johannes Heyn, Klagenfurt (2000)
3. Federal Information Processing Standards Publication, Fips Pub 46-3, DES (1999), http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
4. Federal Information Processing Standards Publications, FIPS PUBS 197, AES (2001), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
5. Fuller, J., Millan, W., Dawson, E.: Multi-objective Optimisation of Bijective S-boxes. In: Proceedings of CEC 2004, Portland, OR (2004)
6. Millan, W.: How to Improve the Non-linearity of Bijective S-boxes. LNCS, vol. 143, pp. 181–192. Springer, Heidelberg (1998)
7. Millan, W., Burnett, L., Carter, G., Clark, A., Dawson, E.: Evolutionary Heuristics for Finding Cryptographically Strong S-Boxes. In: Varadharajan, V., Mu, Y. (eds.) ICICS 1999. LNCS, vol. 1726. Springer, Heidelberg (1999)
8. Millan, W.: New Cryptographic Applications of Boolean Function Equivalence Classes. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 572–583. Springer, Heidelberg (2005)
9. Nedjah, N., de Macedo Mourelle, L.: Designing Substitution Boxes for Secure Ciphers. International Journal Innovative Computing and Application 1(1), 86–91 (2007)
10. Schneier, B.: Applied Cryptography. Wiley, New York (1996)
11. Seredynski, F., Bouvry, P., Zomaya, A.Y.: Cellular Automata Computations and Secret Key Cryptography. Parallel Computing 30, 753–766 (2004)
12. Wolfram, S.: Cryptography with Cellular Automata. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 429–432. Springer, Heidelberg (1986)
13. Wolfram, S.: A New Kind of Science. Wolfram Media Inc., Champaign (2002)

# Computational Hematology in Systems Biology

Toru Horisawa[1], Toshihiko Komatsuzaki[2], and Yutaka Saikawa[1]

[1] Department of Pediatrics, Kanazawa University Graduate School of Medical Science, 13-1 Takaramachi, Kanazawa, Ishikawa, 920-8641 Japan
{horisawa,saikawa}@ped.m.kanazawa-u.ac.jp
[2] Graduate School of Natural Science and Technology, Kanazawa University, Kakuma-machi, Kanazawa, Ishikawa, 920-1192, Japan
toshi@t.kanazawa-u.ac.jp

**Abstract.** One of the key issues in hematology is how blood cell production is conducted and sturdily maintained under physiological and pathological conditions. Due to the difficulty in manipulating hematopoietic stem cells *in vitro*, control of hematopoietic processes remains unclear. Models are required for easy access and are highly expected to show not only reproduction of blood cell behaviors *in vivo*, but also predictability of specific cellular responses under experimental stimulations. We demonstrate herein a granulopoietic model using a 3-dimensional cellular automaton (3D-CA) model. This model incorporates a spatio-temporal concept to describe granulopoiesis that develops in the finite space of the bone marrow cavity. This 3D-CA model is valuable in clinical simulation studies such as determination of a granulocyte growth factor administration for cancer patients with neutropenia. Analyzing the cellular kinetics of hematopoiesis by 3D-CA modeling will provide novel strategies for cancer treatment.

## 1 Introduction

Mature blood cells and their precursors in bone marrow (BM) are ultimately derived from a small population of hematopoietic stem cells (HSCs) [1]. A single HSC displays a high self-renewal capacity [2,3] and can differentiate into any type of blood cell, including lymphocytes, granulocytes, monocytes, erythrocytes, megakaryocytes, eosinophils, basophils and mast cells (Fig.1a). Numbers of each type of cell are maintained within a very narrow range in normal individuals, at approximately 5000 granulocytes, $5 \times 10^6$ erythrocytes and $150,000 - 300,000$ platelets/$\mu$L of whole blood. The mechanisms regulating hematopoiesis are not completely understood. Dynamic regulatory mechanisms for maintaining the number of blood cells presumably involve negative feedback. However, no organs or tissues that detect cell numbers or levels of factors for granulopoiesis have been identified, so the existence of feedback in the systems remains hypothetical. Furthermore, little is known how self-maintenance of the HSC population is achieved. Given the difficulties in manipulating HSCs *in vitro*, mechanisms controlling HSC kinetics remain unclear.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 486–493, 2008.

Computer simulations using cellular automaton (CA) models may provide a useful tool for understanding the systems involved [4]. CA models can produce complex patterns based on simple strategies describing behaviors of elements, analogous to the appearance of complex systems as commonly seen in biological events [5]. We have recently developed a 3D-CA model for granulopoiesis and demonstrated the emergent properties of feedback regulation and characterization of HSC behavior in the granulopoietic system [6,7]. We describe herein our efforts to establish appreciation of the granulopoietic CA model adapted under clinical situations such as chemotherapy and hematopoietic stem cell transplantation (HSCT) for the treatment of cancer patients.

## 2 Methods

### 2.1 Compartmentalization of Granulopoiesis

To describe human granulopoiesis (Fig.1b), we divided the granulopoietic process into 10 compartments in which each cell stage is represented by a model compartment characterized by transit time ($T$), number of mitoses and the fraction of actively proliferating cells (Fig.2). Stages 1 to 15, set arbitrarily, are defined as the state variables in this simulation. Fundamental transit time steps that reflect cell maturation are appointed according to transit times (in hours).

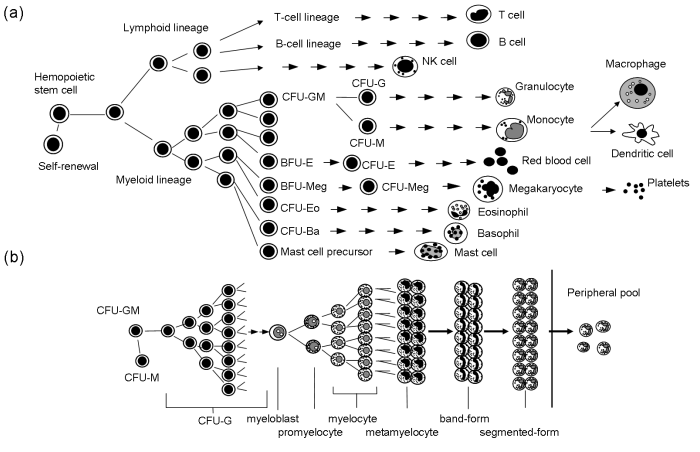



**Fig. 1.** Sequential development of progenitors and mature cells from stem cells [1]. a) A single hematopoietic stem cell differentiates into the progenitors of all blood lineages. Progenitor cells committed to specific lineages are termed *colony-forming unit* (CFU). More primitive progenitors than CFUs were termed *burst-forming units* (BFUs). b) Granulocyte differentiation pathway. Granulopoiesis following CFU-GM consists of the proliferating cell stages (CFU-G, myeloblasts, promyelocytes, and myelocytes) and the postmitotic, non-proliferating, and maturing stages (metamyelocytes, and band and segmented-form granulocytes).

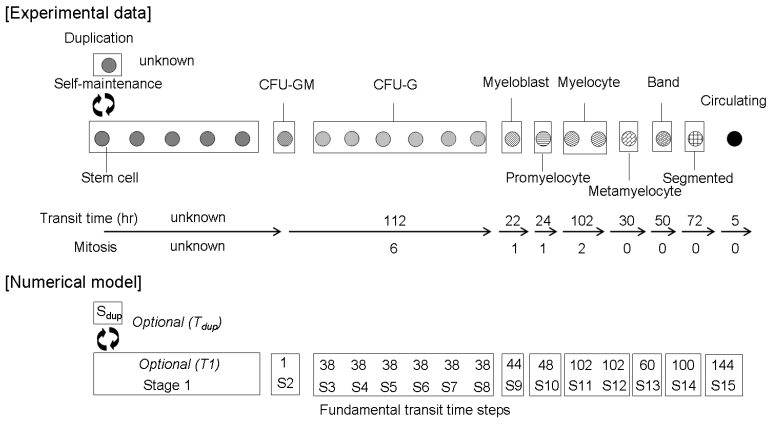



**Fig. 2.** Compartmentalization of granulopoiesis. Granulopoietic processes were divided into 10 compartments characterized by transit times, the number of mitoses, and the fraction of actively proliferating cells.

The compartment of stem cells (stage 1) includes pluripotent stem cells with self-renewal capacities and cells at a very early stage of granulocytic lineage. At this early process of HSC differentiation has not been clarified, the given transit time is referred to as optional ($T_1$). Likewise, all features of HSC self-renewal remain obscure. We define this compartment as duplication, with the transit time ($T_{\mathrm{dup}}$) referred to as optional. The compartment of committed progenitor cells (CFU-GM, stage 2) is fed by the influx of cells originating from the compartment with pluripotent stem cells. The next compartments represent the proliferating cell stages of CFU-G (6 mitoses, stages 3-8), myeloblasts (1 mitosis, stage 9), promyelocytes (1 mitosis, stage 10), myelocytes (2 mitoses, stages 11 and 12) and the postmitotic maturing stages of metamyelocytes (stage 13), band (stage 14) and segmented forms (stage 15). Mature granulocytes leave the bone marrow and enter the circulation compartment and marginal pool in the peripheral blood. Importantly, this model does not presuppose the presence of peripheral negative feedback loops. Model parameters, including transit times and the number of mitoses used here for normal granulopoiesis, were taken directly from the literature or deduced from published experimental data [8].

### 2.2 Stem Cell Division Model

The stochastic model of HSC differentiation used here is shown in Fig. 3. An HSC replicates (self-renewal) or differentiates to a CFU-GM cell. Once committed to a differentiation pathway, the HSC gives rise to a clone that contributes to granulopoiesis. Three patterns of the HSC division process are theoretically proposed for this model system: each stem cell undergoing cell division either generates two, one, or no daughter stem cells until differentiated to CFU-GM in a stochastic situation. Given three probabilities of $p$, $r$, and $q$ (Fig. 3); $p+r+q=1$, an expansion of stem cell numbers is induced under the situation that $p > r+q$,

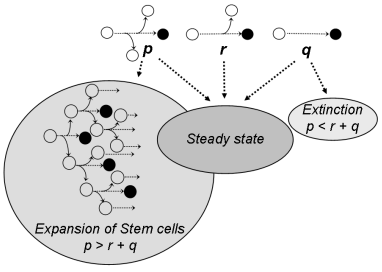



**Fig. 3.** Stem cell division model. Open and closed circles indicate HSC and CFU-GM cells. The probability of each type of cell division is expressed as $p$, $r$, and $q$, respectively.

while an extinction for a finite population occurs if $p < r + q$. A strict steady state is maintained if $r = 1$, while a stationary state is also possible if $r < 1$, and $p = q$. The stochastic concept of single cell growth described here is in line with the classic models proposed by others [9].

### 2.3 Development of a CA Model

The CA model is developed for a 3-dimensional space that assumes bone marrow is where granulocyte-lineage cells are distributed. The space consists of $130 \times 130 \times 130$ unit cubic areas, with the size of each corresponding to a single biological cell (Fig.4a). The analytical space contains structural objects mimicking vessels and trabecular bone as components of the BM cavity. To represent cell distributions in space, two kinds of state variables are assigned to each unit area. First, a set of cell states is prepared so that cells are located at certain positions and distinguishable by proliferation stage. According to the assumption regarding the granulopoietic process, a total of 16 cell states (including a cell absence case) are required (Fig.2). As the second state variable, cell age is defined in addition to the areas in which cells are present. Age is counted at every simulation step to express cell maturation until reaching respective transit times for the next stage (see fundamental transit time steps, Fig.2).

### 2.4 Description of the Local Neighbor Rules

State variables are updated synchronously according to local neighbor rules at every calculation step. These rules include movement of cells, transition to a different stage depending on state of the unit element and neighbors. Essentially, the cell is able to move to any of the nearby elements from 26 directions selected randomly at every time step (Fig. 4b). To avoid collisions with other cells at a certain neighbor site, conflicting directions are evaded during movements by the following implementation: each cell is mapped, and prior to movement, a moving direction is randomly chosen from the available nearby vacant sites. If cells happen to conflict at a destination, an arbitrarily selected cell is then allowed to move while the others are kept in their present locations. Respective transition to the next stage in the cell lineage is fundamentally determined by intrinsic

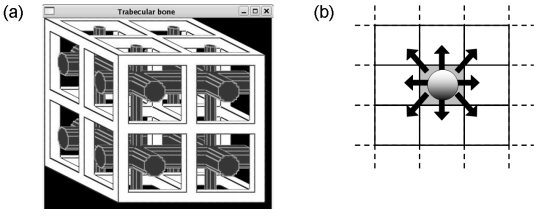



**Fig. 4.** Schematic presentation of a 3D-CA model. a) The CA model is developed for a 3-D space consisting of a $130 \times 130 \times 130$ unit cubic area with structural objects mimicking vessels and trabecular bone forming components of the BM cavity. b) A 2-dimensional view of cell movement. A cell moves to any of the nearby 26 elements, randomly selected at every time step.

properties of each cell, such as transit time compared with maturation (age) counts, although the process that incorporates cell multiplication is influenced by local neighbor conditions. Cells are restricted in proliferation and cell division and are kept in the current state if no vacancy is present in an adjacent space. Consequently, cells may overrun transit time steps.

## 3 Results

### 3.1 Homeostatic Production of Granulocyte-Lineage Cells

The simulation program as developed in this study yields 3D distribution patterns of granulocyte-lineage cells. Since total cell numbers in BM are theoretically maintained at a balance between cellular influx (production) and efflux (leaving the BM), rates of self-renewal and differentiation of HSCs represent critical simulation parameters. Based on the stochastic HSC division model, $T_1/T_{\mathrm{dup}} = 2$ was initially given to avoid exhausting the HSC population. Changing these parameters produced 3 different features of granulopoiesis: extinction; steady state; or oscillation. Fig.5 shows representative steady-state granulopoiesis achieved with a typical set of stimulation parameters, with fundamental transit time steps as shown in Fig.2 and optional $T_1$ and $T_{\mathrm{dup}}$ at $1,600$ and 800 steps, respectively.

### 3.2 A Model for Anticancer Drug-Induced Neutropenia

Incorporated intracellular anticancer drugs inhibit mitosis, resulting in induction of cell death. As a result, both cancer cells and normal mitotic hemopoietic cells are affected. Anticancer drug-induced neutropenia, which often causes life-threatening infections in patients and requires prompt medical intervention, is achieved by eliminating the mitotic compartment of cells (CFU-G-myelocytes) from the model with a constant disappearance rate during drug exposure (24h). As non-mitotic mature granulocytes are spared from drug induced-cell death (although still subject to natural death), the number of granulocytes in peripheral blood was maintained for the first 6 days after drug administration (Fig.6a, b).

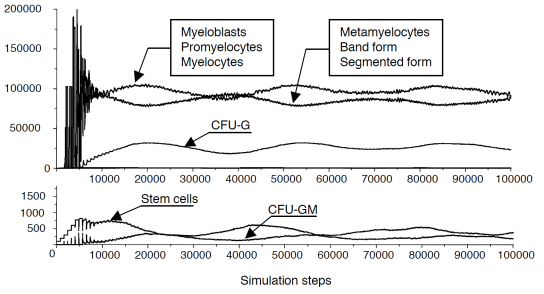



**Fig. 5.** A representative granulopoiesis model. Steady-state granulopoiesis is achieved with fundamental transit time steps as shown in Fig.2 and optional parameters: $T_1 = 1,600$ steps and $T_{\mathrm{dup}} = 800$ steps.

Thereafter, neutropenia, defined as cell numbers below the initial level, appeared and lasted for 7 days in this model. These cellular kinetics fit well with clinical observations and experimental results [10].

### 3.3 Simulation Studies of G-CSF Administration for the Treatment of Neutropenia

G-CSF is a regulatory growth factor for the granulocyte lineage. Administration of G-CSF to humans results in a dose-dependent increase in circulating granulocytes. G-CSF has been demonstrated to be effective in reducing the incidence of febrile neutropenia when administered immediately after chemotherapy [11]. However, G-CSF administration schedules for granulocytic recovery have been poorly examined [10]. G-CSF function was achieved by reducing transit times of CFU-G through mature granulocytes by up to 50% without increasing the frequency of mitosis. Administration of G-CSF was initiated on the indicated days after chemotherapy (Fig.6c, arrow) in the model with 11-day neutropenia. G-CSF clearly reduced the duration of neutropenia (5-7 days) compared to that in control (11 days). Interestingly, mobilization of mature granulocytes was observed when G-CSF was initiated within 5 days after chemotherapy, resulting in a temporary increase in granulocyte number, and paradoxically neutropenia appeared earlier. Effects equivalent to early administration (within 5 days) were observed even if G-CSF was started during neutropenia (7-14 days after chemotherapy). This model strongly indicated appropriate timing for G-CSF administration and an absence of cost-benefit for immediate use after chemotherapy.

### 3.4 Simulation Studies for Hematopoietic Stem Cell Transplantation (HSCT)

HSCT involves the transplantation of HSCs derived from BM (i.e., bone marrow transplantation (BMT)), HSCs mobilized in peripheral blood after chemotherapy with or without G-CSF administration (peripheral blood stem cell transplantation (PBSCT)) or HSCs contained in umbilical-cord blood (cord blood stem cell

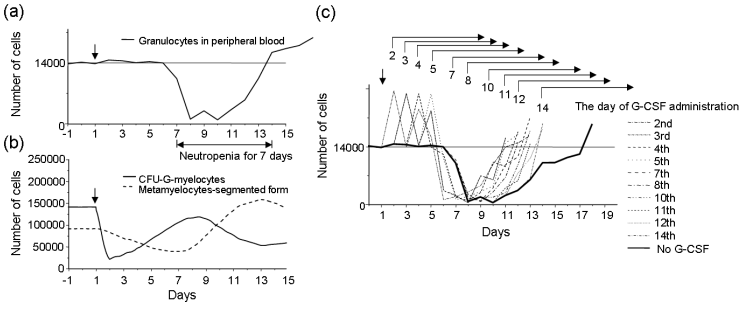



**Fig. 6.** Simulation studies for anticancer drug-induced neutropenia and G-CSF administration schedules. a), b) Cellular kinetics of granulocytes in peripheral blood (a) and differentiating cells in BM (b), after administration of anticancer drug (arrow). c) Simulation studies for G-CSF administration schedule for the treatment of anticancer drug-induced neutropenia.

transplantation (CBSCT)). After several weeks of growth in the recipient BM, expansion of HSCs and their progeny is sufficient to normalize blood cell counts (engraftment). One of the characteristic features of CBSCT is prolonged time to engraftment compared to BMT [12]. Median time to engraftment is 21 days in CBSCT and 14 days in BMT. When the granulopoietic model was initiated with a mixture of stem cells, CFU-GM and CFU-G cell population as a BMT source, continuous production of mature granulocytes appeared in peripheral blood at 13 days, compared to 18 days when started with a single population of stem cells as a CBSCT source (Fig.7). These HSCT simulation models demonstrated that diversity of hematopoietic progenitor cells contained in the grafts was responsible in part for differences in time to engraftment.

## 4 Discussion

The present study demonstrated that 3D-CA granulopoietic modeling was highly valuable in clinical simulation studies, as: 1) the cellular dynamics of drug-induced neutropenia in cancer patients fit well with clinical observations; 2) analyses of cellular kinetics indicated appropriate timings of G-CSF administration and no cost-benefit of immediate use after chemotherapy for the treatment of neutropenia; and 3) diversity of hematopoietic progenitor cells contained in the grafts was responsible for time differences to engraftment in HSCTs.

Several assumptions were required in this model for simplification. First, this model simulated only granulopoiesis, so specific interactions of different lineage cells were not considered. Second, stem cells in the model were deterministically programmed to differentiate toward granulocytes. Third, apoptotic events (programmed cell death) of cells were excluded in this model. Last, the BM cavity consists of a vast network of vascular channels as well as non-hematopoietic cells forming the microenvironment and niches, and is clearly not as simple as portrayed in the model. Despite these simplifications, the results obtained in this model will help to provide fundamental principles for kinetic analyses of the

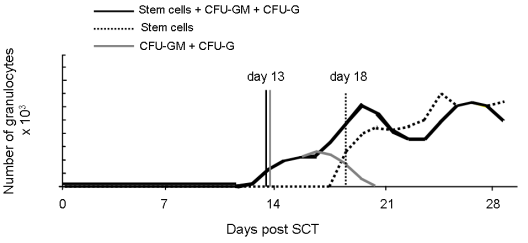



**Fig. 7.** Simulation studies for duration to engraftment after HSCT with different components in grafts

hematopoietic system. Currently, residual leukemic cell kinetics and leukemic stem cell behaviors in patients are under investigation for prediction of relapse.

## References

1. Quesenberry, P.J., Colvin, G.A.: Hematopoietic Stem Cells, Progenitor Cells, and Cytokines. In: Beutler, E., Lichyman, M.A., Coller, B.S., Kipps, T.J., Seligsohn, U. (eds.) Williams Hematology, 6th edn. McGraw-Hill, New York (2001)
2. Till, J.E., McCulloch, E.A.: A direct measurement of radiation sensitivity of normal bone marrow cells. Radiation research 14, 213–222 (1961)
3. Lemishka, I.R., Raulet, D.H., Mulligan, R.C.: Developmental potential and dynamic behavior of hematopoietic stem cells. Cell 45, 917–927 (1986)
4. Wolfram, S.: Cellular automata as models of complexity. Nature 311, 419–424 (1984)
5. Wootton, J.T.: Local interactions predict large-scale pattern in empirically derived cellular automata. Nature 413, 841–844 (2001)
6. Komatsuzaki, T., Saikawa, Y.: Modeling of the inherence of feedback regulation and stem cell behavior in granulopoiesis. In: Proceedings of the 2nd international conference on systems biology, pp. 208–212 (2001)
7. Saikawa, Y., Komatsuzaki, T.: Emergent properties of feedback regulation and stem cell behavior in a granulopoiesis model as a complex system. Complex systems 14, 45–61 (2003)
8. Schmitz, S., Franke, H., Brusis, J., Wichmann, H.E.: Quantification of the cell kinetic effects of G-CSF using a model of human granulopoiesis. Experimental Hematology 21, 755–760 (1993)
9. Wichmann, H.E., Loffler, M., Schmitz, S.: A concept of hematopoietic regulation and its biomathematical realization. Blood cells 14, 411–429 (1988)
10. Koumakis, G., Vassilomanolakis, M., Barbounis, V., Hatzichristou, E., Demiri, S., Palataniotis, G., Pamouktsoglou, F., Efremidis, A.P.: Optimal timing (preemptive versus supportive) of granulocyte colony-stimulating factor administration following high-dose cyclophosphamide. Oncology 56, 28–35 (1999)
11. Clark, O.A.C., Lyman, G.H., Castro, A.A., Clark, L.G.O., Djulbegovic, B.: Colony-stimulating factors for chemotherapy-induced febrile neutropenia: A meta-analysis of randomized controlled trials. Journal of Oncology 23, 4198–4214 (2005)
12. Cohen, Y., Nagler, A.: Umbilical cord blood transplantation - how, when and for whom? Blood reviews 18, 167–179 (2004)

# Dynamically Reorganising Vascular Networks Modelled Using Cellular Automata Approach

Paweł Topa

AGH University of Science and Technology, al. Mickiewicza 30,
30-059 Kraków, Poland
topa@agh.edu.pl

**Abstract.** We present the model of tumour induced angiogenesis that apart from predominating phenomena, as influence of VEGF, includes newly discovered factors such as Dll4/Notch signalling and remodelling processes. The Cellular Automata approach is employed to model cellular and intracellular processes that occur in cancer tissue and surroundings. Vascular system is modelled by using the Graph of Cellular Automata, which combines graph theory with the Cellular Automata paradigm. Additionally, an outline of model verification method which uses graph descriptors is presented and exemplified.

## 1 Introduction

Angiogenesis is the process of blood vessels formation [1]. The cells influenced by the stresses (e.g., low $O_2$, low $pH$), synthesise angiogenic stimulators (most of all VEGF — Vascular Endothelial Growth Factor) [1]. Stimulators diffuse towards the nearest blood vessels and activate the endothelial cells (ECs) that lines the vessel walls. In the response the endothelial cells start to proliferate and migrate being attracted by the gradient of VEGF. The wall of parent blood vessel becomes degraded and a lumen of a new capillary is formed.

The process of angiogenesis has a crucial role in solid tumour growth. Clusters of growing tumour cells are short of oxygen and nutrients. The "starving" tumour cells produce VEGF and other angiogenic stimulators (Tumour Angiogenesis Factors — TAFs) that activate neighbouring vessels.

Tumour induced angiogenesis is a very promising target in anti-cancer therapy [3]. Inhibition of angiogenesis or regression of existing vasculature may suppress tumour development. Treatment targeted on improving vasculature around tumour can be helpful for drug delivery during chemotherapy.

The computer models of angiogenesis employ continuous or discrete approaches [2]. The continuous models base on differential equations and as a result we obtain distributions of endothelial cells in the tissue. The main disadvantage of the continuous approach is the lack of information about the structure of vascular network. In contrast, the discrete models are able to produce vascular network of a given topology [2]. These models mostly base on the Cellular Automata (CA) paradigm and assume that modelled species (endothelial cells, angiogenic factors

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 494–499, 2008.

etc.) are treated individually. In [6], Anderson and Chaplain assumed that growth of a single sprout is governed by move of the endothelial cell located at its tip. It moves across regular, rectangular network of CA according to predefined rules. At each step of simulation the tip cell moves in one out of four directions or waits with a certain probability. The probabilities are calculated by using diffusion equation with terms reflecting VEGF and fibronectin influence [6]. Additional rules that model vessels branching and anastomosing were also defined.

In [8] we proposed a new framework for modelling multiscale phenomena in tumour induced angiogenesis. Our approach combines the Cellular Automata paradigm with a graph structure into a new modelling tool which we call the Graph of Cellular Automata [7]. We assumed that the tissue is represented by a regular mesh of cellular automata and some of the accompanying diffusion processes are modelled by using simple rules of local interaction. The vascular network is represented by the graph structure constructed over the regular mesh. The graph is made of some CA cells, for which we define additional relations of neighbourhood with other cells.

The role of Dll4/Notch in angiogenesis has been investigated experimentally quite recently [9]. This ligand regulates the processes of sprouting and vessels maturation. Inhibition on Dll4/Notch causes creation of dense but dysfunctional (immature) vascular network, while its stimulation results in creation of sparse though functional network. Both the inhibition and stimulation can be considered as anti-cancer therapeutic strategy, The influence of Dll4/Notch on the dynamics of the process of angiogenesis has not been modelled so far.

Unlike in the normal angiogenesis (e.g. during embryogenesis), tumour induced angiogenesis forms mostly pathological vascularisation [4]. Defected vessels and vessels, which do not transport blood decline and are removed from the tissue. Conversely, the vessels with circulating blood survive [5].

Models of any physical processes require verification. In the case of angiogenesis this verification is not trivial. We use the feature vectors of statistical descriptors, hoping that these descriptors, as an ensemble, will have sufficient generalisation power to classify various network structures. The analysis of the networks, i.e., investigation of their similarities and dissimilarities can be made then by using pattern recognition tools, such as classifiers, clustering and feature extraction techniques.

In this paper we present extended version of our previous Cellular Automata model [8]. First, we shortly introduce its foundation and our recent extensions. Then we discuss exemplary results. We propose also the method of quantitative analysis of vascular networks, which employs the statistical graph descriptors [11]. Finally, we discuss the conclusions.

## 2 Tumour-Induced Angiogenesis Model with Vascular Network Reorganisation

The formal definition of the model and the outline of the essential algorithms were presented in [8]. Shortly, the model works as follows:

1. The cells, which form the graph ("vessel" cells) are the sources of nutrients (e.g. $O_2$). Nutrients are distributed to surrounding cells producing a certain gradient of concentration.
2. Tumour cells with nutrients concentration below a certain threshold change their states into "hungry" and start producing TAFs. TAFs are distributed over the mesh of CA with certain gradient of concentration.
3. The growth of a single vessel is governed by the move of its tip. The consecutive tip cell positions follow local TAFs gradient.
4. Initially, a newly formed vessel is not mature enough to be able to supply nutrients. The maturation level of each cell in the graph is incremented at every step of simulation until it reaches the "mature" state.
5. When TAFs concentration in "vessel" cell exceeds a certain threshold and the cell is "mature" enough, a new sprout is generated. The vessels, which glue together produce anastomoses.

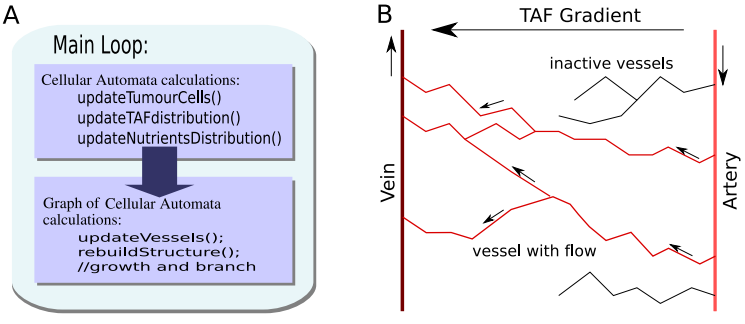



**Fig. 1.** A: The outline of the model of tumour-induced angiogenesis [8]. B: Initial geometry for model of tumour induced angiogenesis enforces the direction of blood flow.

In Fig. 2 we present an outline of the model. At each step of simulation, procedures, which implement the defined CA rules are applied both to the Cellular Automata and the Graph of Cellular Automata. The model is very flexible and new modules can be easily included.

In this paper we investigate the influence pf Dll4/Notch ligand on sprouting and maturation processes. The process of removing of unused and dysfunctional vessels (the vessels without circulating blood) is also taken into account. The new rules are as follows:

1. The "mature" vessels supply oxygen only if they create the closed circuit with blood flowing due to pressure gradient.
2. Dysfunctional vessels, i.e., the vessels which do not transport blood are removed gradually.
3. The "vessel" cells have additional parameters corresponding to Dll4/Notch signalling. It describes sensitivity of the cells to the TAFs and the rate of

maturation. The value of Dll4/Notch, which exceeds a given threshold stimulates sprouting but delays the process of maturation. Conversely, low value of this parameter keeps low level of sprouting but speeds up maturation.

We define the following set of the control parameters:

1. $t_g$ — TAFs(VEGF) gradient coefficient.
2. $o_g$ — Oxygen gradient coefficient.
3. $dll4$ — Dll4/Notch signalling level in "vessel" cells.
4. $tt$ — TAFs threshold that activates "vessel" cells.

We assume also that some previously defined parameters [8] are functions of TAFs and Dll4/Notch concentration, i.e., the probability of branching is directly proportional, while maturation speed is inversely proportional to this value.

Calculating blood flow in such a complex, chaotic and dynamically changing network structure is extremely difficult. We assume, that blood flows only in the vessels that create closed circuits. We detect such the circuits, calculating "pressure" at each node of the network and "blood flow" in every edge. The edges with nonzero pressure gradient between corresponding nodes are treated as "efficient" ones.

To simplify the model, we construct initial conditions of simulations with two vessels: artery and vein (see Fig. 2B). We assume that only the artery has the ability of sprouting. The sprouts attracted by TAF gradient grow towards the vein. This way the blood flow direction from artery to vein is biased (see Fig. 2B). Closed circuits and blood flow in the edges are updated at each step of simulation.

At each step of simulation the cells in vessels network, which do not transport blood, are marked as inactive. After a certain amount of time-steps, these cells are removed from the vascular network.

## 3 Results

The simulations were performed on the mesh of $100 \times 100$ cells. Figure 2A displays vascular network generated without removal of unused vessels while in Fig. 2B we present the results of simulation with this option turned on. As shown in Fig. 2C the "brush border effect" (sudden increase of density of sprouts) appears as a result of existence of TAF threshold, which activates endothelial cells.

The networks displayed in Fig. 2A and Fig. 2B were generated with lower value of $dll4$ parameter (Dll4/Notch signalling) than those shown in Fig. 2C. As a result we obtain considerably sparser vascular network with a better organised structure.

### 3.1 Quantitative Verification

To verify the model, the networks obtained from simulations have to be compared to the networks extracted from images of real vascular systems. We require a parameter space (feature space) that will be able to differentiate their structural properties. To make such the structural analysis possible we use the feature vectors composed of statistical descriptors of complex networks [11]. This method

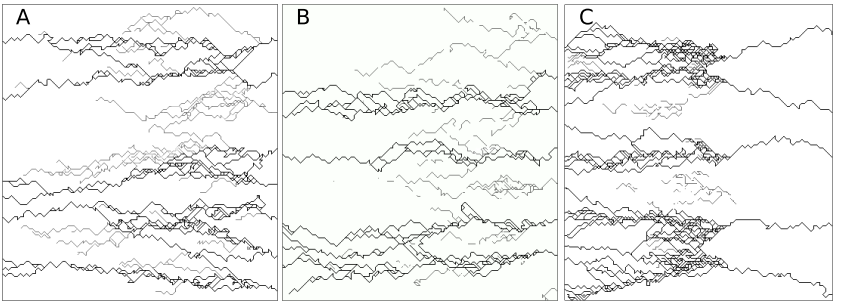



**Fig. 2.** Sample simulation results (see comments in the text). The vessels with blood flowing are depicted in black while those without blood circulation are marked in grey.

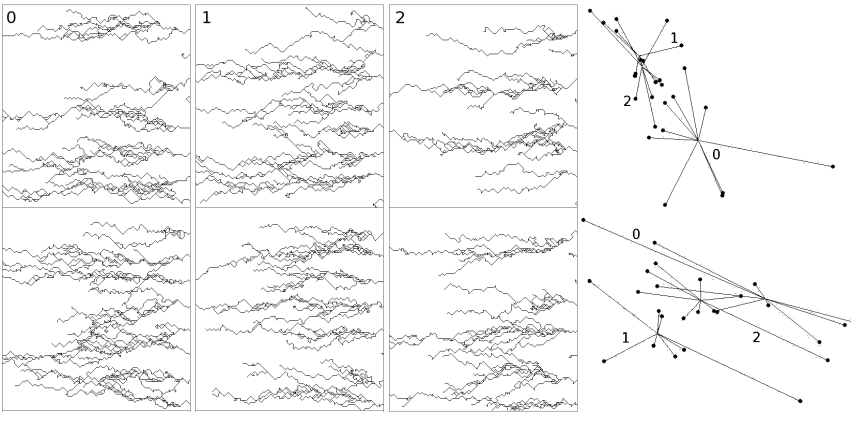



**Fig. 3.** Three sets of simulated network topologies (0, 1, 2) representing 3 different values of parameters. The result of visualisation of descriptor feature space with MDS method is displayed. Two different views of reduced 3-D feature space are shown.

will allow for investigating model parameters, which influence topology of the simulated vascular networks. These parameters can be tuned reflecting various types of drug treatment (e.g. inhibiting of VEGF). Thus our model can serve as a kind of virtual laboratory for testing of various therapy strategies.

We applied a selected set of graph descriptors [11] to calculate feature vectors and compare them using pattern recognition tools such as the multidimensional scaling [10]. Each network is described by 11 component feature vector. MDS method translates 11-D feature space into its 3-D representations. In Fig.3 we show the sample results of simulation generated for three different combinations of parameters (sets marked as $(0, 1, 2)$). The feature vectors have been calculated for each set. We visualise the results using multidimensional scaling procedure. We can see clusters in the feature space corresponding to each class of generated networks (0, 1 and 2). This way we can estimate the influence of various biological

factors on the network topology. There are only preliminary tests but we show that the method can effectively separates network samples generated for various model parameters.

## 4 Conclusions

We show that the general framework of the Graph of Cellular Automata used for modelling tumour-induced angiogenesis [8] can be easily extended by new factors and processes. We have defined additional parameter that reflects influence of Dll4/Notch signalling on sprouting and maturation processes. It controls both the reaction of endothelial cells in the vessels, VEGF concentration and the rate of vessels maturation. We show that by changing the parameter corresponding to the Dll4/Notch activity we can obtain instead of rich but chaotic and dysfunctional vascular network more sparse structure but potentially efficient.

The model has been extended with algorithms that detect vessels with blood flowing. The process of remodelling of vascular network in case of tumour induced angiogenesis is very important. Treatments targeted against tumour induced angiogenesis aim at vessels regression. Also therapies that support vessels network normalisation require rules for removal of unused and defective sprouts. In this way the network system tries to build up more and more efficient structure.

The application of graph descriptors for comparison of the vessel networks, their classification and model validation is a very promising method. In the future we plan to select from tens of known descriptors [11], those which are the most relevant in the detection and generalisation of dominant networks topologies.

**Acknowledgements.** The author thanks A. Dudek (University of Minnesota Medical School), W. Dziwnel and T. Arodź (AGH University of Science and Technology) for valuable discussions. This research is partially supported by the Polish Ministry of Education and Science (grant no. 3 T11F 010 30).

## References

1. Carmeliet, P., Jain, R.K.: Nature, 407 (2000)
2. Mantzaris, N.V., Webb, S., Othmer, H.G.: J. Math. Biol. (2004)
3. Ferrara, N., Kerbel, R.S.: Nature, 438 (2005)
4. Tonini, T., Rossi, F., Claudio, P.P.: 22 (2003)
5. Gee, M.S., et al.: American J. of Patbiology, 162 (2003)
6. Anderson, A.R.A., Chaplain, M.A.J.: Bull. of Math. Biol. 60 (1998)
7. Topa, P., Dzwinel, W., Yuen, D.: Int. J. Mod. Phys. C. 17 (2006)
8. Topa, P.: In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173. Springer, Heidelberg (2006)
9. Hellstrom, M., et al.: Nature, 445 (2007)
10. Dzwinel, W., Błasiak, J.: Future Generation Computer Systems, 15 (1999)
11. Goryczka, S., Arodź, T.: Bio-algorithms and med-systems 2 (2006)

# A Stochastic Multi-agent Model of Stem Cell Proliferation

Richard C. van der Wath and Pietro Liò

The Computer Laboratory, University of Cambridge, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK

**Abstract.** In this paper we describe a Multi-Agent framework to model the dynamics of self-renewing and differentiating stem cells. Our model is extended to an effective stochastic simulator of 5-bromo-2-deoxyuridine (BrdU) uptake and loss on different detection thresholds. Matching the results of an existing BrdU model validates our effort. Additionally our Multi-Agent approach enables much more statistically rich analysis and provides a modelling platform stem cell scientists can easily understand and relate to.

## 1 Introduction

There is a great need for quantitative modelling and simulation of stem cell dynamics to elucidate the intra-cellular and inter-cellular components and interaction networks present in a stem cell system. A Multi-Agent model, where each agent represents a cell, can serve as a very natural conceptualisation of inter-cellular stem cell interactions, and hence be easily grasped by stem cell biologists.

In this paper we present such a multi-agent model to simulate the dynamics of self-renewing and differentiating stem cells based on Hierarchical Stem cell Proliferation (HSP). The hierarchical structure of our model is motivated in Section 2 and in Section 3 we derive the agent-based algorithmic version of the universal stem cell dynamics introduced in the previous section. Finally, in Section 4 we present a case study where our base model is implemented as a stochastic simulator of 5-bromo-2-deoxyuridine (BrdU) uptake and loss on different detection thresholds. The results of our approach matches that of a previous Ordinary Differential Equation (ODE) based model, but since it is stochastic it has the ability to produce much more statistically rich results.

## 2 Hierarchical Stem Cell Proliferation

Stem cells have the unique ability to produce copies of themselves (self-renewal) as well as other types of cells (differentiation) when they divide [2]. Hierarchical Stem Cell Proliferation (HSP) is a conceptualisation that views stem cells as losing their self-renewal potential as they progress down the hierarchy by becoming more and more specialised. The resulting tree-like structure (Figure 1)

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 500–505, 2008.

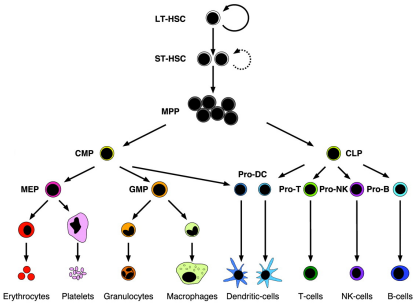



**Fig. 1.** Hematopoietic proliferation hierarchy. Hematopoietic stem cell populations in the mouse range from 12-200 cells/million of total bone marrow cells [1].

determines the possible properties and developmental stages of daughter cells after division. The leaves represent terminally differentiated cells which have lost their ability to self-renew or differentiate.

Advantages of the hierarchical treatment of stem cell proliferation include the fact that it is easily implemented as a multi-agent model (as we will show in the next section), and it also closely correspond to the stem cell biologists' perception of the reality. By the latter we mean that the most common way for biologists to identify stem cells (ASC at least) is by their cell surface markers [1]. Each 'node' in Figure 1 thus corresponds to a population of cells with a unique set of surface markers, and the hierarchy shown has in most cases been determined through a series of transplantation assays (whose description is beyond the scope of this paper). A further advantage of building a model on HSP is that it does not require the implementation of the whole hierarchy but allows one to focus on a small substructure at a time, depending on the objective and the data one wish to model.

## 3 A Multi-agent Model Based on HSP

Since stem cells are rare entities with relatively low numbers compared to other cells, it is quite feasible to simulate large stem cell systems with single cell (agent) based methods in a reasonable amount of time. Each node in the proliferation tree can be represented by a different type of agent whose differentiated progeny are determined by the structure of the tree. Agents' actions would be influenced by their interaction with each other and with their local environment, returning to steady state after stress and injury events.

We have implemented our model as an Object Oriented application in Java using the Repast Agent Simulation Toolkit [3] and an informal specification is given below. In the following section we present a case study where we show how this base model can be extended to test the validity of two opposing hypothesis about chromosome segregation in HSCs.

Each agent is defined by:

- A unique `identification number` so that individual agents' fate can be traced.
- A `type` which will define the type of stem cell this agent represents.
- Some form of `state`. The state is a general interface for properties that change over time.
- A list of possible `actions`. The default actions are: self-renew, differentiate and die. Each action has a certain probability per time unit (tick) associated with it.
- A `control` structure which assigns probabilities to the agent's actions.

The standard HSP simulation would proceed as follows:

1. Fix the number of different cell types, their position in the hierarchy and the structure of the hierarchy.
2. Initialise the initial number of agents of each type, and $T$, the total number of ticks the simulation must run.
3. Initialise the `state` and `control` of each agent.
4. Run the main loop (Algorithm 1).

**Algorithm 1.** Main loop of HSP

**while** $tick < T$ **do**
  **for all** $agent \in agentlist$ (in random order) **do**
    $action \Leftarrow agent.actionlist$ (with equal probability)
    $prob\_action \Leftarrow agent.control(action)$
    execute $agent.action$ with probability $prob\_action$
  **end for**
  $tick \Leftarrow tick + 1$
**end while**

## 4 A Case Study

In this section we will demonstrate the powerful versatility of HSP model by simulating the uptake and loss of BrdU. BrdU is a synthetic nucleotide and an analogue of thymidine which means it can be incorporated in synthesised DNA by substituting for thymidine. Fluorescent-marked antibodies that attaches to BrdU can then be used to detect it, thus providing a mechanism to measure the proportion of cells in a given population that where actively replicating their DNA. BrdU data can be misleading however when interpreted directly. Much more certainty about turnover -and death rates can be gained by using computational and mathematical models to simulate BrdU dynamics. A trivial extension of our model through the implementation of chromosomes in the `state` of each agent transforms our model into a very useful stochastic simulator of BrdU dynamics.

**Algorithm 2.** Asymmetric segregation

**for** $i = 1$ to $num\_chromosomes$ **do**
  $daughter1.chr(i).5prime \Leftarrow mother.chr(i).5prime$
  $daughter1.chr(i).3prime \Leftarrow brdu\_present$
**end for**

**Algorithm 3.** Random segregation

**for** $i = 1$ to $num\_chromosomes$ **do**
  $rnd \Leftarrow Random.uniform(0, 1.0)$
  **if** $rnd > 0.5$ **then**
    $daughter1.chr(i).5prime \Leftarrow mother.chr(i).5prime$
    $daughter1.chr(i).3prime \Leftarrow brdu\_present$
  **else**
    $daughter1.chr(i).5prime \Leftarrow brdu\_present$
    $daughter1.chr(i).3prime \Leftarrow mother.chr(i).3prime$
  **end if**
**end for**

### 4.1 Extending the Base Model: Agents with Chromosomes

Chromosomes are implemented as collections of boolean pairs in the `state` variable of each agent. Each unit of the boolean pair represents one of the DNA strands of a cell's chromosome (3-prime or 5-prime strand) and is set to `true` if the strand has taken up BrdU and `false` otherwise. A global boolean variable `brdu_present` indicates whether BrdU is applied or not, so that both BrdU uptake and loss can be simulated. With our agent-based approach we can count the explicit number of chromosomes that are BrdU positive at each time point and this allows us to also model the sensitivity of BrdU detection.

### 4.2 Simulating BrdU Kinetics in Murine Hematopoietic Stem Cells

A recent study by Kiel et. al. [4] uses BrdU data from murine HSCs to refute the asymmetrical segregation of chromosomes hypothesis in favour of the random segregation hypothesis. The asymmetrical segregation or immortal strand hypothesis was first proposed in 1975 [5] and suggests that adult stem cells retain older DNA strands. We repeated the study of Kiel et. al. to validate and evaluate the usefulness of our BrdU model. Model settings and parameter values used are chosen to match the values reported in Kiel et. al.:

- We used a population of 400 agents, each with 40 chromosomes.
- Turnover rate is 6% per day. We implemented this as 3% self-renew, 3% differentiate and 0% death rates per day.
- BrdU uptake is simulated for 10 days, followed by a chase period of 130 days.

In addition we implemented two versions of the stem cell agent's divide function, one for each hypothesis (Algorithm 2 and 3).

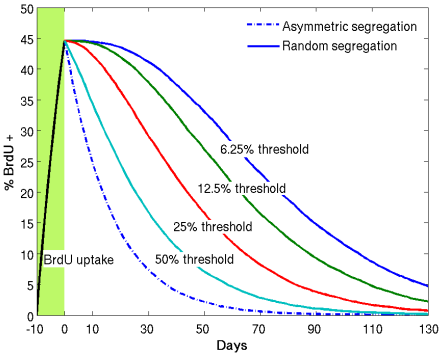



**Fig. 2.** Simulation results of 50 runs, showing the average predicted percentage of 400 cells that are BrdU positive after an uptake period of 10 days (shaded in green). A threshold of 25% means at least a quarter of the chromosomes needs to be BrdU labelled before the agent is regarded as BrdU positive.

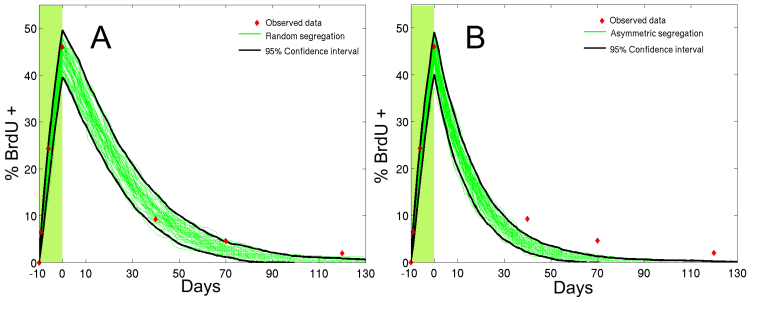



**Fig. 3.** 50 Stochastic simulations of: A - random segregation (50% detection threshold); and B - asymmetric segregation. Also showing 95% confidence intervals and observed experimental data.

### 4.3 Results and Comparison to Existing BrdU Models

The average simulated BrdU measurement based on different detection thresholds of 50 stochastic runs are shown in Figure 2. Note that the detection threshold has no effect during the first 10 days (the time BrdU was applied) and also during the chase period for asymmetric segregation.

The plot in Figure 2 compares favourably with Figure 3a of Kiel et. al. also supporting random segregation above the immortal strand hypothesis. This can be more clearly seen in Figure 3 which compares the individual 50 runs for the random segregation and asymmetric segregation models with the observed data respectively. Note that our model is much more precise in terms of simulating the detection threshold, since the model of Kiel et. al. simulates the number of divisions $N$ before BrdU becomes undetectable rather than the labelled chromosomes itself. But $N$ depends on whether one or both chromosome strands

are labelled, causing the 'overlap' effect between different detection thresholds as apparent from the legend in Figure 3a of Kiel et. al.

The model of Kiel et. al. and that of other well known BrdU models in literature [6] are all ODE-based and thus deterministic in nature. There are several disadvantages in modelling biological systems as deterministic processes, the most obvious being the fact that a deterministic model needs to assume complete knowledge of the biological system under consideration. This is not possible for most biological systems that researchers are interested in (due to the mere complexity of the spatial position, size, velocity, etc. of billions of molecules, not to speak of the atoms that constitute these molecules). Hence deterministic models invariably have to adopt a higher level view, representing actual biomolecular reactions as some form of aggregate. On this higher level the system dynamics behaves in a stochastic manner and needs to be modelled as such.

### 4.4 Conclusion

Stochastic results from our model confirms the conclusion of Kiel et. al. that HSCs do not asymmetrically segregate chromosomes upon division, but further analysis showing the late stage observed values falling outside 95% confidence bounds (Figure 3 A) and a difference in statistical locality (not shown) suggests there might be a more complex process involved than that assumed by Kiel et. al. For example it is quite likely that the HSC population as identified by the $CD150^{+}CD48^{-}CD41^{-}$lineage$^{-}$Sca-$1^{+}$c-$kit^{+}$ markers are not in fact a homogeneous population with a homogeneous turnover rate.

## References

1. Wilson, A., Oser, G.M., Jaworski, M., Blanco-Bose, W.E., Laurenti, E., Adolphe, C., Essers, M.A., Macdonald, H.R., Trumpp, A.: Dormant and self-renewing hematopoietic stem cells and their niches. Ann. NY. Acad. Sci. 1106, 64–75 (2007)
2. Potten, C.S., Loeffler, M.: Stem cells: attributes, cycles, spirals, pitfalls and uncertainties. lessons for and from the crypt. Development 110(4), 1001–1020 (1990)
3. North, M.J., Collier, N.T., Vos, J.R.: Experiences creating three implementations of the repast agent modeling toolkit. ACM Trans. Model. Comput. Simul. 16(1), 1–25 (2006)
4. Kiel, M.J., He, S., Ashkenazi, R., Gentry, S.N., Teta, M., Kushner, J.A., Jackson, T.L., Morrison, S.J.: Haematopoietic stem cells do not asymmetrically segregate chromosomes or retain brdu. Nature 449(7159), 238–242 (2007)
5. Cairns, J.: Mutation selection and the natural history of cancer. Nature 255(5505), 197–200 (1975)
6. Bonhoeffer, S., Mohri, H., Ho, D., Perelson, A.S.: Quantification of cell turnover kinetics using 5-bromo-2'-deoxyuridine. J. Immunol. 164(10), 5049–5054 (2000)

# coreBIST: A Cellular Automata Based Core for Self Testing System-on-Chips

Rupsa Chakraborty and Dipanwita Roy Chowdhury

Dept. of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India-721 302
{rupsac,drc}@cse.iitkgp.ernet.in

**Abstract.** In this paper, a cellular automata based Built-in self-test (BIST) core design for a self testing System-on-Chip (SoC) is proposed. The objective of the core is to generate pseudo-random test patterns that are injected into the various IP cores within an SoC. The corresponding output patterns are compacted and analyzed for correctness, during the test mode of the SoC. The BIST core was tested on some synthetic SoCs built by integrating ISCAS 85 benchmark circuits. Considerable reduction in the total test time and area is noticed, compared to the corresponding non-BISTEDed SoCs.

**Keywords:** Cellular Automata, Built-in self-test, System-on-Chip, Test-Pattern-Generator, Response-Analyzer.

## 1 Introduction

Recent advancement in the IC technology has made it possible to incorporate the various components of a traditional printed-circuit-board on a single silicon substrate. Such systems are called system-on-chips (SoCs) [1]. Each individual circuit component of a SoC is commonly called *core*. After a chip is manufactured, testing the cores deeply embedded into the SoC is a challenge. One of the factors is that there is limited access into the core directly from the primary i/o pins of the SoC. The application of test vectors into the SoC from an external automatic test equipment (ATE) is a slow and expensive process, adding to the test complexity. Built-in self-test (BIST) methodology of testing individual IP cores is a widely accepted strategy [2]. Linear feedback shift register (LFSR) and cellular automata (CA) are primarily used as the hardware generating pseudo-random test patterns in BIST applications. LFSR based test compression techniques in SoCs have been proposed in [3], [4].

In this paper, we propose a design where all the components required for testing a SoC is integrated into a single core. We call this *coreBIST*. The core contains a linear hybrid cellular automata based test generator and a signature analyzer. In a SoC design, a system is built by integrating reusable cores from various core vendors. In many cases the cores themselves may be BISTed. In the method we propose, we are going to show that there is a considerable reduction in the total hardware overhead of the SoC with *coreBIST* compared to the SoC where each core has a dedicated BIST.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 506–511, 2008.

The rest of the paper is organized as follows. Section 2 explains the various techniques of generating pseudo-random test patterns and the CA implementation. Section 3 explains the proposed method of test pattern generation, signature analysis, and the BIST hardware. Section 4 illustrates our approach with experimental results on synthetic SoCs built using ISCAS 85 combinational benchmark circuits . Section 5 concludes the paper.

## 2 State-of-the-Art of Pseudo-random BIST

Two main pseudo-random test pattern generators that are mostly used are linear-feedback-shift-register (LFSR) and cellular automata (CA). Among these LFSR is the most frequently used test pattern generator (TPG). One of the reasons is that LFSR is more area efficient since it needs lesser combinational logic per flip-flop than a CA based implementation. CA based test TPG has the advantage over LFSR in that it has a structure where the feedback interconnection between any two cells are physically closer making it easily extendible. Another advantage is that the test patterns produced by a CA are more random than those produced by a LFSR.

A CA is constructed based on some rules. Rules provide the logical relationship of how flip-flops are to be connected to its two nearest neighbors. In a particular CA implementation, there may be a combination of several rules. A popular combination of rules is Rule 90 and Rule 150 since this combination, in a certain order, gives the exhaustive set of test patterns, i.e., for an n-bit CA it gives all the $2^n - 1$ combination of patterns. All zero pattern is not considered. Such a CA is often called linear hybrid cellular automata (LHCA). The order in which the rules are to be applied in order to generate exhaustive patterns using CA can be found in [5]. In rule 90, the value of a particular cell, at time t, is obtained by XORing the values of its previous and next cells at time t-1, see equation 1. In rule 150, this value is obtained by XORing the values of previous, next as well as the value of the cell itself, at time t-1, see equation 2.

$$Rule\ 90: \ x_i^t = x_{i-1}^{t-1} \oplus x_{i+1}^{t-1}. \tag{1}$$

$$Rule\ 150: \ x_i^t = x_{i-1}^{t-1} \oplus x_i^{t-1} \oplus x_{i+1}^{t-1}. \tag{2}$$

A generic cell for a CA using combination of rule 90/rule150 is shown in Figure 1 [2]. When the enable line is high, the structure acts according to rule 150 else it acts according to rule 90.

## 3 *coreBIST* - Concept and Design

The general structure of *coreBIST* is shown in Figure 2. This core has two main blocks - test pattern generator and test response analyzer. *coreBIST* has a single input pin which is also a primary input of the SoC. This pin is the test enable line. When the signal to this pin is high, implying that the SoC is in the test mode, then the BIST within the core is activated. When the SoC is in the

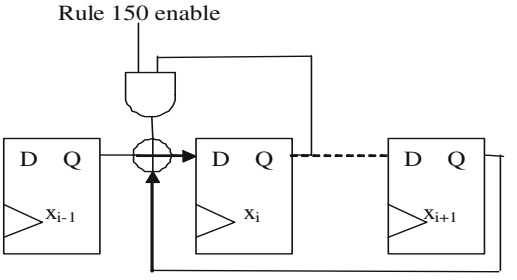



**Fig. 1.** A generic CA cell for rule 90 and rule 150 combination

functional mode, then this line is kept low. The core has a single output pin which is a primary output of the SoC. This pin is high when the test is going on and there is no error, this pin is set low when a fault is identified in the circuit. The concept of the integration of the *coreBIST* into a SoC is illustrated

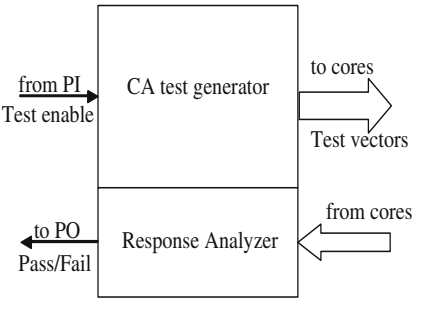



**Fig. 2.** General structure of *coreBIST*

in Figure 3. Among all the cores the one with the largest input pin counts is identified. Let it be $n$. The length of the register of the CA test generator in *coreBIST* is of size $n$. The cores with smaller pin counts can be tested with the same CA. Each line of the n-bit output of *coreBIST* is driving the test input lines of one or more cores. Preprocessing of the test-set of each core is needed to understand which core a *coreBIST* output line should be connected to. There is an advantage of hardware and area reduction, over the traditional BISTed core SoC testing, in the proposed method where in a single core the whole testing hardware for the SoC is implemented. In the traditional method of testing each core has a dedicated BIST, as shown in Figure 4. The improvement can be observed in Table 1.

### 3.1 Pseudo-random Test Pattern Generation

CA implementation of TPG is popular since CA produces patterns that have good randomness properties. Rules 90 and 150 have been combined to produce

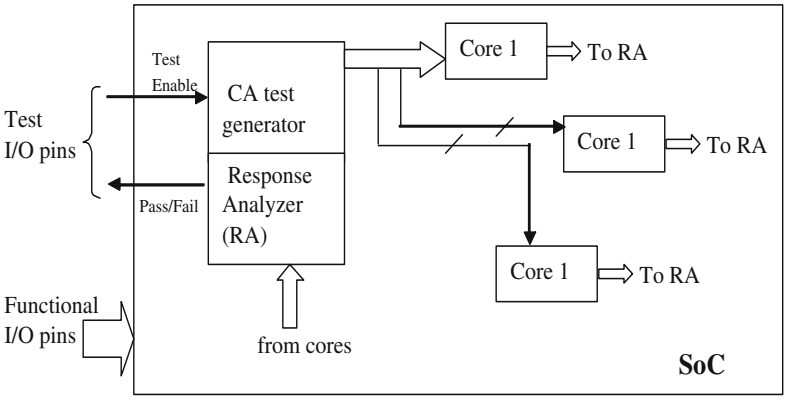



**Fig. 3.** Integration of *coreBIST* into a SoC

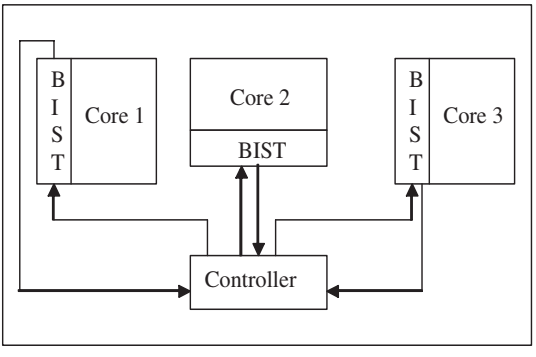



**Fig. 4.** SoC cores with dedicated BIST

exhaustive pseudo-random patterns in this work. In order to determine the CA register size and seed selection, a preprocessing step is required before the CA is built.

*Preprocessing:*

*STEP 1: Selecting CA register size.* The size of the CA register assigned the size of the largest input pin count among all the cores in the SoC.
*STEP 2: Distributing TPG output lines.* Amongst the test set for each core, the cores that can be tested concurrently by sharing test generator output lines of *coreBIST* are identified.
*STEP 3: Test pattern compaction.* Set a signature for testing the response.

### 3.2 Output Response Analyzer

To verify the output response, the technique of data compaction has been used. The expected responses to the test patterns are compacted into a representative

pattern (*repPattern*). During testing the test responses are compacted in the similar manner and finally compared to the *repPattern*. In case of a match, the circuit passes test and when there is is a mismatch the circuit fails test. Pass/Fail is indicated by the output line of the response analyzer which is also a primary output. Here the output pattern for a particular input test data is stored in a register. The input to each element of the register is XORed with the previous value stored. At the end of testing, the value stored will be the final pattern to be compared with the *repPattern*.

## 4 Experimental Results

For the purpose of testing our method, we have built some synthetic SoCs using the ISCAS 85 combinational benchmark circuits, as cores.

**Working flow for timing verification**

*STEP 1.* For each core, the test set for stuck at faults were generated using a commercial automatic test pattern generating tool, Tetramax.
*STEP 2.* An n-bit linear hybrid cellular automata was used, where n is the maximum number of primary input lines among all the cores in the SoC.
*STEP 3.* The initial value to which the registers of the CA needs to be set to, called seed, is then found out, using algorithm ***setSeed()***, to get a good test coverage.
*STEP 4.* Next, the hardware to implement the CA based test pattern generator is designed.
*STEP 5.* After that, the hardware to implement the response analyzer is designed. Both the test generator and the response analyzer is incorporated into a single core. There is only one input line to the core, the BIST enable line, and only one output line which says if the testing has been successful or not.

**Table 1.** Synthetic SoC Details

| Name | ISCAS Cores | CA reg. length | Compactor reg. length | % area improvement |
|---|---|---|---|---|
| SoC1 | c17,c1908,c880,c335 | 60 | 136 | 9.4 |
| SoC2 | c432,c449,c17 | 41 | 41 | 2.7 |
| SoC3 | c335,c17,c6288,c449 | 41 | 98 | 6.0 |
| SoC4 | c17,c1908,c6288 | 33 | 59 | 4.3 |
| SoC5 | c17,c432,c1908 | 36 | 34 | 4.2 |

Experiments were done on synthetic SoCs built by using ISCAS 85 benchmark circuits. Table 1 shows the details of the SoCs. Column 1 provides the names of the synthetic SoCs. Column 2 shows the ISCAS combinational circuits that were used as cores. Column 3 gives the length of the CA register. This corresponds to the input pin count of the core with maximum input count. Column 4 gives the length of the output register where the compacted output is stored. The length

of this register is the total output pin count of the maximum number of cores that can be tested in parallel. A comparison of the SoC having *coreBIST* with a SOC built with the same cores but having a CA based BIST implementation inside each core, with respect to area, was made, and the result is shown in column 5. In each case some improvement was observed.

## 5 Conclusion

This paper proposes the design and analysis of a new core called *coreBIST*. This core represents the BIST for an entire SoC. The CA based BIST inside the core provides a considerable improvement of test coverage and hardware overhead over the conventional test access methodology. Experimental results on synthetic SoCs built using ISCAS 85 benchmark circuits show the elegance of the core design.

## References

1. Bergamaschi, R.A., Cohn, J.: The A to Z of SoCs. In: International Conference on Computer-aided design. IEEE/ACM, San Jose (2002)
2. Stroud, E.C.: A Designer's Guide to Built-In Self-Test. Kluwer Academic Publishers, Dordrecht (2002)
3. Kim, H.S., Kang, S.: Increasing Encoding Efficiency of LFSR Reseeding-Based Test Compression. IEEE Trans. on CAD of Integrated Circuits and Systems 25(5) (May 2006)
4. Wang, Z., Chakrabarty, K., Wang, S.: SoC Testing Using LFSR Reseeding, and Scan-Slice-Based TAM Optimization and Test Scheduling. In: Design, Automation and Test in Europe Conference and Exhibition. IEEE Press, Germany (2007)
5. Chudhury, P.P., Chowdhury, D.R., Nandi, S., Chattopadhyay, S.: Additive Cellular Automata Theory and Applications, vol. 1. IEEE Computer Society Press, Los Alamitos (1997)

# GPU Accelerated Computation and Visualization of Hexagonal Cellular Automata

Stéphane Gobron[1], Hervé Bonafos[2], and Daniel Mestre[1]

[1] Institute of Mouvement Sciences, CNRS, Marseille, France
{stephane.gobron,daniel.mestre}@univmed.fr,
[2] herverv@aol.com

**Abstract.** We propose a graphics processor unit (GPU)-accelerated method for real-time computing and rendering cellular automata (CA) that is applied to hexagonal grids. Based on our previous work [9] –which introduced first and second dimensional cases– this paper presents a model for hexagonal grid algorithms. Proposed method is novel and it encodes and transmits large CA key-codes to the graphics card and consequently, this technique allows to visualize the CA information flow in real-time to easily identify emerging behaviors even for large data sets. To show the efficiency of our model we first present a set of characteristic hexagonal behaviors, and then describe computational statistics for central processing unit (CPU) and GPU on a set of different hardware and operating system (OS) configurations. We show that our model is flexible and very efficient as it permits to compute CA close to a thousand times faster than classical CPU methods. Additionally, free access is provided to our downloadable software for hexagonal grid CA simulations.

**Keywords:** Hexagonal cellular automaton, GPU-accelerated computation, Digital imaging, Real-time rendering, Emerging behavior.

## 1 Introduction

This paper belongs to a series of papers concerning the computation and rendering real-time boolean multi-dimensional cellular automata (CA). Based on our previous

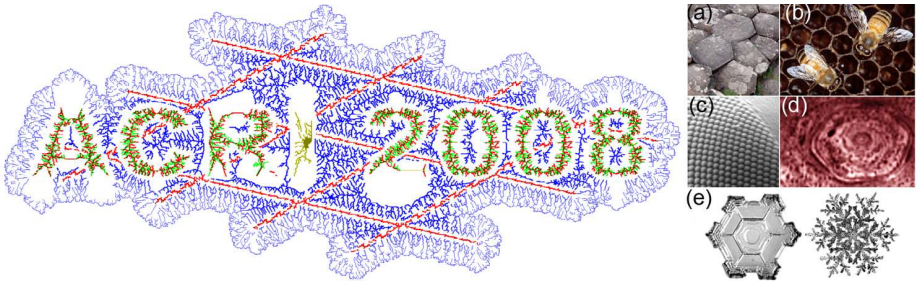



**Fig. 1.** [Left] Example of hexagonal CA ($2D_b^hCA$) applied to the conference logo (101 steps); [right] Example of hexagonal structures found in nature: (a) stones found in the *Giant's Causeway*, Ireland; (b) honeycomb; (c) eye of a fly; (d) hexagon spotted over Saturns surface; (e) snowflakes

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 512–521, 2008.

approach proposing first dimension and second *von Neumann* dimension CA [9], the current work focuses on 2D hexagonal structures ($2D_b^hCA$).

As shown by the literature described in next Subsection 1, CA is a powerful tool that can be used in a wide variety of domains. Unfortunately, in CA, emerging phenomena are often impossible to predict by theoretical approaches [20]. To help researchers compute complex phenomena –such as species competition and evolution [3]– faster, or identifying emerging behaviors almost instantaneously, we propose real-time graphical visualization models. A demonstration of the capabilities of our approach is available as a freeware provided on first author web site. In the following subsections we first present the literature focusing on hexagonal CA and then describe the structure of the paper.

**Background.** This study belongs to the research fields of CA and computer science (CS) specialized in real-time digital imaging applied to visualization of multidimensional massive set of cells. Computer graphics literature dealing directly with CA has become more and more prevalent, especially since GPU programming became popular. There has been a growing interest in the field following the publishing of a fundamental textbook by Stephen Wolfram [20] in 2002, covering all practical aspects of CA. Assuming that every structure with interacting elements is a type of CA, references will actually be too numerous to mention. Moreover this paper is a direct extension of our introduction to multi-dimensional model [7] where the reader can find a non exhaustive list of references in fields relative to formal, classical, and applied CA is provided, as well as CG and GPU programming ([10]). Within the scope of this paper, we will focus on hexagonal CA (hCA) references.

As shown in Figure 1$^{[right]}$, the hexagonal structure is very common in nature. Indeed, due to one of its properties regarding neighborhood cells (central symmetry), it offers a natural structure for simulation. This has inspired researchers to utilize hexagonal CA in a wide range of applications ($hCA$): reaction-diffusion system [1]; modeling forest fire [22,4]; simulation of debris flows [11]; physics-based simulation of material decomposition [12]; biology and repartition of forest studies [14]; gas basic flow simulation [18]; biological model of tuna school formation [17]. A number of research papers involving $hCA$ were presented also in a previous Conference on Cellular Automata for Research and Industry ($ACRI2006$) such as: the study of CA inalterability of topology without memory in [2]; bacteria modeling [21]; fluid (greases) simulation [13]; robot path planning [19]. While there is a wide range of literature on applications of hexagonal CA, we did not find any references to general algorithms for GPU-based 2D hexagonal CA, which indicates that our approach is novel and original.

## 2 Previous Work

In order to increase the performance of CA computations we use a GPU programming based approach. The following section summarizes concepts relative to graphics processor units (GPU) and cellular automaton (CA) which are detailed in our previous work [9] where we proposed a GPU solution for first and second direct neighbors Boolean CA respectively called $1D_bCA$ and $2D_b^{vN}CA$.

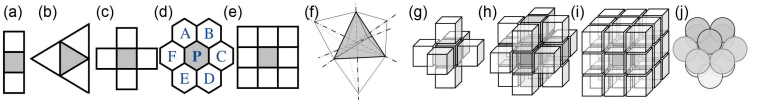



**Fig. 2.** Multidimensional CA geometric representations: (a) $1DCA$; (b) $2D^tCA$; (c) $2D^{vN}CA$; (d) $2D^hCA$; (e) $2D^MCA$; (f) $3D^tCA$; (g) $3D^{vN}CA$; (h) $3D^{M_1}CA$; (i) $3D^{M_2}CA$; (j) $3D^{h_1}CA$

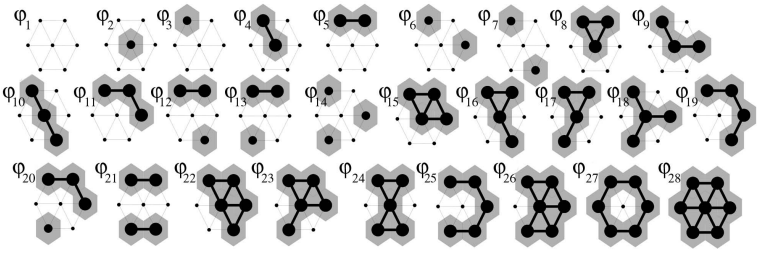



**Fig. 3.** All symmetrical configurations for Boolean $2D_b^hCA$

**Graphical Processor Unit (GPU).** Over the last few years, GPU cluster programming ([5]) has increased CPU computational capacity by a factor of 10 to 30 depending mainly on the graphics card series and the operating system. Our current approach uses a C++ code on CPU to communicate with OpenGL Shading Language (GLSL) programs on GPU. The graphics card uses three algorithms in its pipeline: the *vertex Shaders*, the *Geometric Shaders*, and the *Fragment* or *"Pixel" Shaders*. To use the graphics card as a computational device only the pixel Shaders is useful, however, we must compile and call the vertex Shaders. The main loop which allows cellular buffers to be considered as textures and copied back to matrices, is detailed in [9]. To include a summary here, the following main steps illustrate the aforementioned relationship between CPU and GPU:

1. associate a Vertex and Pixel Shaders pointers to source files;
2. compile Vertex and Pixel Shaders programs => return 2 pointers;
3. if compilation is fine...
4. ...get the Shaders-program pointer...
5. ...attach Vertex and Pixel pointers to the Shaders program;
6. ...link them all together.

**Terminology, Concept and Definitions.** The terminology used in this paper, as in previous papers ([9]), include various notions of CA types, neighborhoods, rules, changes of state, and CA-key codes. Some cellular automata structures are presented in Figure 2 which illustrates most $1^{st}$, $2^{nd}$, and $3^{rd}$ dimensional CA. Considering the multitude of possible CA structure and types, we use the following rule to symbolize them all:

$$[\text{dimension}]D^{[\text{structure}]}_{[\text{state nb}]}CA \equiv dD^{N}_{n}CA \text{ with:} \quad (1)$$

- $d$ as dimension, *e.g.* $d$=3 for space;
- $n$ the number of states that a cell $C$ can have, *e.g.* for Boolean $n \equiv$ '$b$' $= 2$;
- $N$ the structure, *i.e.*
  - for the square/cube structure with direct neighbors $N \equiv$ '$vN$' $= 4$ in 2D, $vN$ stands for **v**on **N**eumann;
  - for the square/cube structure with indirect neighbors $N \equiv$ '$M_i$' $= 8$ in 2D, $M_i$ stands for **M**oore with $i$ being the growing number of possible indirect cases;
  - for the hexagonal grid $N \equiv$ '$h$' $= 6$ in 2D;
  - for the triangle or tetragonal structure $N \equiv$ '$t$' $= 3$ in 2D and $= 4$ in 3D;

Therefore, in the $dD^{N}_{n}CA$ domain, for each cell $C$ with $N$ number of neighbors, let's define $c$ and $\rho$ respectively for the number of configuration structure –see Figure 3 including all possible rotations *i.e.* the CA key-code $\varphi_1$, $\varphi_2$,... , $\varphi_c$)– and the number of possible CA, then:

$$\text{with } c = n^{(N+1)} \text{ and } \rho = n^c \text{ then } \rho = n^{n^{(N+1)}} \quad (2)$$

Let $\tau$ be the truth table made with the neighbor of $C$: $A, B, C...X$. Notice that the "pivot" cell is call "P" in Figure 2.

Now that basic concepts relative to Shaders and multidimensional CA have been defined, we move on in the following section to propose a model for computing any rule of $2D^{h}_{b}CA$ enabling the visualization of the corresponding data flow in real-time.

## 3 Current Model: GPU-Accelerated $2D^{h}_{b}CA$

Using equation 2 we can determine that $\rho_{(2D^{h}_{b}CA)}$ is equal to $2^{128}$ which is a large number of possible Boolean CA. A huge majority of these CA seem to behave in a very chaotic way such that it is not possible to identify geometrical characteristics of their respective behavior. That is why in this study we concentrate on symmetrical CA. We will see in the next section that the number of symmetrical CA is much smaller ($2^{28}$) than all possible Boolean CA, which narrows the search space. Nevertheless, trying to identify one by one the corresponding 268.435.456 behaviors is not the best approach.

**Fragment Shaders Model.** As previously stated the algorithm of the hexagonal CA is based on the relationship between a CPU program (using MSVC++) and Shaders codes. To make such a software, three main functions are required: CA computations, the hexagonal rendering, and the human machine interface (HMI, also called *user interface*) for the interactive choice of CA. As the first two functions are highly computationally expensive, both are implemented using Shaders. Furthermore, when CA's number of neighbors is superior to 6 key-code, it becomes too large to be transmitted. To solve this issue we encode the key-code as a 1D-texture that can be easily transmitted to the Shader program. Here is the pseudo-code of the fragment Shaders for the computation of the CA:

Based on the coordinates shown in Figure 2(d), we use the symbol "$\diamond$" for the texel (texture pixel), $\Gamma$ for the key-code texture, $l$ the parity of the matrix line, and $i$ for the intensity threshold to assume input images as Boolean matrices:

- $F_{xy} \leftarrow C_{xy} \leftarrow P_{xy} \leftarrow \diamond_{xy}$
- $(C,F)_x \leftarrow (C,F)_x \pm \delta_\diamond$
- if ($l$): $A_{xy} \leftarrow E_{xy} \leftarrow F_{xy}$ and $(A,E)_y \pm \delta_\diamond$ and $B_{xy} \leftarrow D_{xy} \leftarrow P_{xy}$ and $(B,D)_y \pm \delta_\diamond$
- else: $A_{xy} \leftarrow E_{xy} \leftarrow P_{xy}$ and $(A,E)_y \pm \delta_\diamond$ and $B_{xy} \leftarrow D_{xy} \leftarrow C_{xy}$ and $(B,D)_y \pm \delta_\diamond$
- $\forall$ cells $X$: set its state $X_s$ to true if intensity of $\diamond_X > i$
- define state code $\gamma = P_s + 2A_s + 4B_s + 8C_s + 16D_s + 32E_s + 64F_s$
- $P_s^{t+1} \leftarrow \Gamma(\gamma)$

**Table 1.** Corresponding symmetrical shape for the 128 bits $2D_b^hCA$ key-code

| $(FEDCBAP)_d$ | Corresponding $2D_b^hCA$ symmetrical shape –see Figure 3 |
|---|---|
| 0..31 | 1 2 3 4 3 4 5 8 3 4 6 9 5 8 11 15 3 4 7 10 6 9 12 16 5 8 13 17 11 15 19 22 |
| 32..63 | 3 4 6 9 7 10 13 17 6 9 14 18 12 16 20 23 5 8 12 16 13 17 21 24 11 15 20 23 19 22 25 26 |
| 64..95 | 3 4 5 8 6 9 11 15 7 10 12 16 13 17 19 22 6 9 13 17 14 18 20 23 12 16 21 24 20 23 25 26 |
| 96..127 | 5 8 11 15 12 16 19 22 13 17 20 23 21 24 25 26 11 15 19 22 20 23 25 26 19 22 25 26 25 26 27 28 |

**Symmetrical Rules.** As illustrated in [9], symmetrical CA can be very useful in the domain of real-time imaging, especially in computer graphics (CG) for automatic surface texturing or analysis of image flow. To understand the relationship between the key-code (rule) of a CA and its equivalent symmetrical code (if any), here is a practical example based on the emerging behavior in Figure 1$^{[left]}$. This CA shows a growing of ramification spreading regularly but not symmetrically all over the surface. To do so we applied a specific, selected symmetrical key-code (*i.e.* 1 to 28 possible $\phi'$ set to $true$) equivalent to a CA-rule with a key-code of $32\phi$. In this particular case the relationship is described by equation 3.

$$\begin{aligned} \varphi'_{true} &= [2.4.6.7.9.10.12.13.15.16.17.18.20.23.24.26.28] \equiv \\ \tau(\varphi_{(1..32)} &= [A2E8EFC2EFBFCBE82BF3FEBB83FA2B8A]) \end{aligned} \tag{3}$$

This global behavior generating root-like patterns can be associated to non-photorealistic renderings (NPR). Notice that only cells that have changed state are shown, furthermore to improve printing effects surrounding cells are blurred.

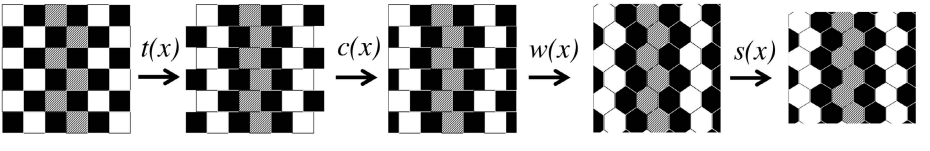



**Fig. 4.** Hexagonal rendering pipeline

### Hexagonal Grid Rendering

$$w(x) = \sum_{i=1}^{n} (\cos(\frac{k.x}{k^2})) \text{ with } k = 2i - 1 \tag{4}$$

Figure 4 presents the pipeline for the rendering Shaders enabling real-time and very accurate visualisation of the hexagonal grid. The four functions are $t(x)$ for translation, $c(x)$ for clipping, $w(x)$ for triangle-wave, and $s(x)$ for scaling. The first two allow the center of the cells to be consistent relative to the hexagonal grid. We used the *triangle wave* (*i.e.* Fourier series) shown in equation 4 enabling to simulate hexagons for $w(x)$ and the last function scales the $(y)$-axe by $\frac{\sqrt{3}}{2}$ so that hexagons are regular in all directions. An example of hexagonal grid rendering is illustrated in the pictures (e) and (f) of Figure 5.

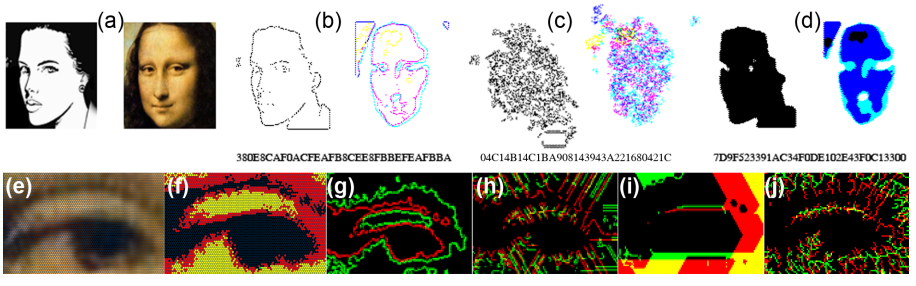



**Fig. 5.** From (a) to (d) a comparative test of five $2D_b^hCA$s on two input images with different levels of complexity (the right one being Da Vinci's "La Joconde"); from (e) to (j) Mona Lisa's left eye where triple CA (R,G,B) were applied, from left to right, a hexagonal discretization of the eye, the $RGB$ Boolean equivalent, and four examples of hCA

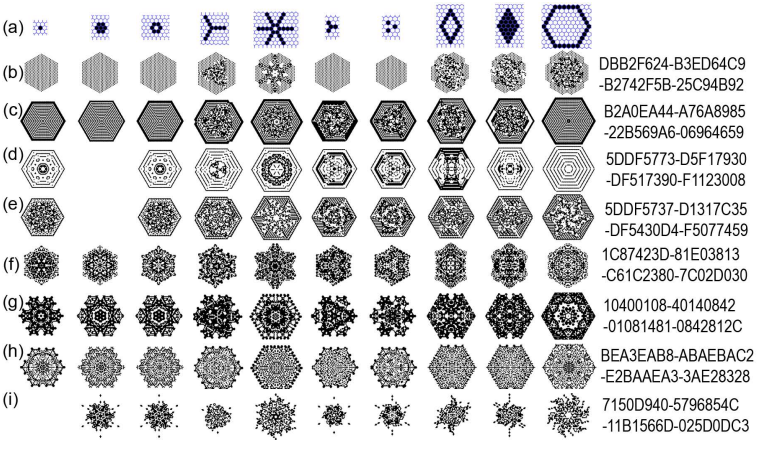



**Fig. 6.** Results of a $2D_b^hCA$ after 25 steps using 10 basic roots shows row (a)

**Graphical Results and Observations.** The last four pictures in Figure 5 (g) to (j) propose interesting $hCA$ emerging behaviors respectively contour detection, artistic vectorization, fragment-Voronoi patterns, and –similarly to Figure 1– another CA of root-like pattern. Another example of $hCA$ set is presented Figure 5 where three different $hCA$ are simultaneously tested on two type of faces (with 25 steps): picture (b) representing a converging edge detection CA; picture (c) with symmetrical rule $[5.6..18.19.23.27]$ gives an excellent representation of the famous *game of life* CA – converging after 2324 steps. To study $hCA$ behaviors more systematically, we focus on CA that grows from simple to complex seeds. Figure 6 presents a set of eight of such CA with ten different seeds –row (a)– from simple dot (left) to a very small hexagon (right). In this palette of patterns, we can observe these non exhaustive list of properties: silhouette and texture symmetries, chaos, fractal, spiral formation, crystal-like patterns (*e.g.* snowflakes see also Figure 1 (e) for comparison).

## 4 Performance

In this section, we present our contribution in acceleration of CA computations by comparing different types of algorithms (CPU or GPU based), machine configurations (with a number of different graphical cards), and texture sizes.

**Six Configurations.** All algorithms presented in this paper were developed in $C$++ on Microsoft Windows XP-pro using $MSVC$++, the graphics library OpenGL [16], and the OpenGL Shading Language ($GLSL$) [15]. To highlight the efficiency of the method presented in this paper, we demonstrate that our technique offers high real-time performance on common everyday computers which are available in reasonable prices (within the price range of regular home and office computers). Table 2 presents the six hardware configurations used to test our model.

**Table 2.** General specifications of six configurations with their respective graphical test points based on 3DMark06 ranking system –(*): overclocked processor

| Config | Comp.type | OS | Graphical Card | Memory | Processor | 3DMark06 pts |
|---|---|---|---|---|---|---|
| 1 | PC (HP) | Vista | GeForce 8600 GS, 512MB | 3072MB | 2 proc. Q6600, 2.39 GHz | 2650 |
| 2 | PC | WinXP | GeForce 7600 GT, 256MB | 1024MB | 2 proc. Q6300, 1.86 GHz | 3432 |
| 3 | PC (Dell) | WinXP | Quadro FX 3500, 256MB | 2048MB | 2 proc. 6700, 2.66 GHz | 4326 |
| 4 | PC (Dell) | WinXP | Quadro FX 3500, 256MB | 3072MB | 4 proc. Xeon, 3.20 GHz | 5198 |
| 5 | NB () | WinXP | GeForce 7900M GTX, 512MB | 2048MB | 2 proc. T7200, 2.00 GHz | 4700 |
| 6 | PC () | WinXP | GeForce 8800 GTX, 768 MB | 2048MB | 2 proc. E6750, 2.90* GHz | 12500 |

**Texture Limitations and Influence.** As explained in previous sections, graphical textures are used as a computational buffer for the CA matrices as well as the CA keycodes. The size of these textures depend on the graphics card and pixel Shaders drivers. For GeForce 7000s and Quadro 3000s, buffer sizes are limited –for any dimension–to 4096 pixels, and can reach 8192 for high-end graphical cards (GC) such as the NVidia GeForce 8800 GTX. Therefore, when using GPU for fast buffer computation, one buffer

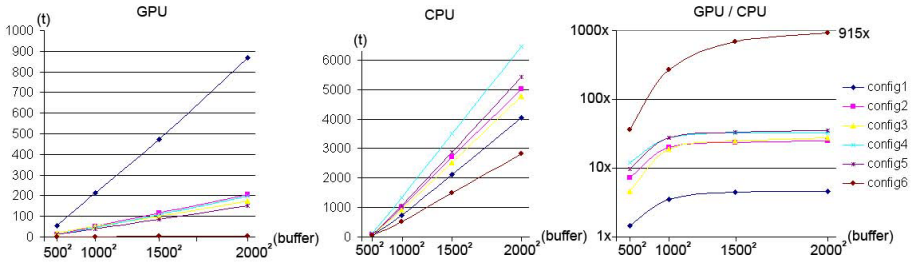



**Fig. 7.** Comparing performance between six different configurations: (a) and (b) time to compute ten thousand iterations on different buffer sizes, respectively for the GPU and the CPU; (c) comparison of the speed of GPU and CPU –using a logarithmic scaling.

cannot exceed a matrix bigger than $8192^2$ –only available for high-end GC in 2D. In practice, the matrix cannot exceed the GC memory, *i.e.* in the best case 768Mb. Nevertheless there are ways to work this around to enable access to even large 3D matrices, some of these will be presented in our next publication on $3D_b^{vN}CA$.

**GPU and CPU Results.** The left graph of Figure 7 presents the time (in seconds) that the six configurations spent for computing (using the GPU method) CA over different matrix sizes (from $500^2$ to $2000^2$) for ten thousand iterations. For the largest matrix, the slowest was configuration 1 with $869.2s$ and the fastest configuration 6 with $3.1s$. Notice that first results are linear and that the four intermediate machines are regrouped very close to each other at values from $152.3$ to $204.2$ seconds. The middle graph of Figure 7 proposes also the same task but computed by the CPU. Results are also linear but this time are ranged closer to each other from $2828s$ for configuration 6 (again the fastest) to $4042s$ the Xeon processor.

**Comparison Between GPUs.** Three categories of computers is clearly shown by right graph of Figure 7 representing the ratio between GPU and CPU computational power for this particular task. In one particular case (PC 1, GeForce 8000 series) , we suspect the operating system (MS Windows Vista) might be slowing down the graphics card, but to conclude this for certain, more tests are needed. We were also surprised by the excellent result of configuration 6, showing a computation ratio $915$ faster than its own CPU which is the fastest of all other PCs. The rest of the configurations $c_2, c_3, c_4, c_5$ provided fairly good results which corresponds to the similar cases in literature, respectively: $24.6, 27.4, 32.7, 35.6$ times faster than their CPU. As demonstrated by the reported results, our proposed algorithm greatly increases the computational performance on all platforms.

## 5 Conclusions and Future Work

Following up on our recent publications [9], in this paper we have presented an extended method to simulate –i.e. compute and visualize– Boolean CA using a graphics

card accelerated method for hexagonal grids. After introducing the programming technique using both CPU and GPU and summarizing concepts used in our previous work, we proposed a novel method to encode large CA key-codes allowing a generalized Boolean CA algorithm to be performed on a GPU –restricted to memory size limitations. We then have presented an original method to automatically sort symmetrical CA for any dimension based on their symmetrical structure. We have detailed how to encode generalized algorithms for $2D_hCA$ presenting the pseudo-code of the Shaders algorithms. Finally, to show the capabilities of our model we have presented examples of characteristic patterns, common global behaviors, and computational statistics done on six different hardware and software configurations. We demonstrated that our model allows to compute up to 915 times faster than if CPU alone was utilized. We are convinced that this method can lead to an accelerated and generalized model for 3D-surface CA [8] which would be a breakthrough for automatic texturing and simulations. The future work in our agenda regarding this research is to report the findings on an extended model for three-dimensional Boolean CA with direct neighbors ($3D_{vN}CA$). This work also explores new issues such as memory management and real-time graphical rendering of 3D with GPU. Furthermore, our current research team works on a range of topics on virtual reality (VR) and we are developing a system to explore CA models interacting with virtual scenes.

**Acknowledgment.** We would like to thank Dr. Arzu Coltekin for the proofreading of the text.

## References

1. Adamatzky, A., Wuensche, A., De Lacy Costello, B.: Glider-based computing in reaction-diffusion hexagonal cellular automata. ScienceDirect, Chaos Solutions and Fractals 27, 287–295 (2006)
2. Alonso-Sanz, R., Martín, M.: A structurally dynamic cellular automaton with memory in the hexagonal tessellation. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 30–40. Springer, Heidelberg (2006)
3. Chopard, B., Lagrava, D.: A cellular automata model for species competition and evolution. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 277–286. Springer, Heidelberg (2006)
4. Hernández Encinas, L., Hoya White, S., del Rey, A.M., Rodríguez Sánchez, G.: Modelling forest of fire spread using hexagonal cellular automata. ScienceDirect, Applied Mathematical Modelling 31, 1213–1227 (2007)
5. Fan, Z., Qiu, F., Kaufman, A., Yoakum-Stover, S.: GPU cluster for high performance computing. In: SC 2004. IEEE Computer Society, Los Alamitos (2004)
6. Foley, J.D., van Dam, A.F., Stephen, K., Hughes, J.F., Phillips, R.: Introduction to Computer Graphics, principles and practice, 2nd edn., 1175 pages. Addison Wesley, Reading (1993)
7. Gobron, S., Devillard, F., Heit, B.: Retina simulation using cellular automaton and GPU programming. Machine Vision and Applications Journal 66, 331–342 (2007)
8. Gobron, S., Finck, D.: Generating surface textures based on cellular networks. In: The Geometric Modeling and Imaging international conference (GMAI 2006), Londres, UK, July 5-7, 2006, pp. 113–120. IEEE Computer Society Press, Los Alamitos (2006)

9. Gobron, S., Mestre, D.: Information visualization of multi-dimensional cellular automata using GPU programming. In: 11th International Conference on Information Visualisation (iV 2007), Zurich, Switzerland, July 2-6, 2007, pp. 33–39. IEEE Computer Society Press, Los Alamitos (2007)
10. Harding, S., Banzhaf, W.: Fast genetic programming and artificial developmental systems on GPUs. In: 21st International Symposium on High Performance Computing Systems and Applications (HPCS 2007), Saskatoon, SK, Canada. IEEE Computer Society, Los Alamitos (2007)
11. Iovine, G., D'Ambrosio, D., Di Gregorio, S.: Applying genetic algorithms for calibrating a hexagonal cellular automata model for the simulation of debris flows characterised by strong inertia effects. Geomorphology 66, 287–303 (2005)
12. Lan, Y.L., Li, D.Z., Li, Y.Y.: Modeling austenite decomposition into ferrite at different cooling rate in low-carbon stell with cellular automaton model. ScienceDirect, Acta Materalia 52, 1721–1729 (2004)
13. Miyamoto, S., Sakai, H., Shiraishi, T., Morishita, S.: A flow modeling of lubricating greases under shear deformation by cellular automata. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 383–391. Springer, Heidelberg (2006)
14. Pagnutti, C., Anand, M., Azzouz, M.: Lattice geometry, gap formation and scale invariance in forests. ScienceDirect, Theoretical Biology 236, 79–87 (2005)
15. Rost, R.J.: OpenGL Shading Language, 2nd edn. Addison Wesley Professional, Reading (2006)
16. Shreiner, D., Woo, M., Neider, J., Davis, T.: OpenGL Programming Guide: the official Guide to learning OpenGL v2.0, 1st edn. Addison Wesley Professional, Reading (2005)
17. Stöcker, S.: Models for tuna school formation. Mathematical Biosciences 156, 167–190 (1999)
18. Toussaint, J.-C., Debierre, J.-M., Turban, L.: Deposition of particles in a two-dimensional lattice gas flow. Physical Review Letters 68(13) (1992)
19. Wainer, G.A.: Modeling robot path planning with cd++. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 595–604. Springer, Heidelberg (2006)
20. Wolfram, S.: A new kind of science, 1st edn. Wolfram Media Inc. (2002)
21. Wu, Y., Chen, N., Rissler, M., Jiang, Y., Kaiser, D., Alber, M.: Ca models of myxobacteria swarming. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 192–203. Springer, Heidelberg (2006)
22. Yongzhong, Z., Feng, Z.-.D., Tao, H., Liyu, W., Kegong, L., Xin, D.: Simulating wildfire spreading processes in a spatially heterogeneous landscapes using an improved cellular automaton model. In: Geoscience and Remote Sensing Symposium (IGARSS 2004). Proceedings. 2004 IEEE International, pp. 3371–3374 (2004)

# Automatic Design of FPGA Processor for the Backtracking of DNA Sequences Evolution Using Cellular Automata and Genetic Algorithms

Georgios Ch. Sirakoulis

Democritus University of Thrace, Department of Electrical and Computer Engineering,
Laboratory of Electronics,
GR 67100 Xanthi, Greece
gsirak@ee.duth.gr

**Abstract.** In several cases, the DNA sequences of an organism are available in different stages of its evolution and it is desirable to reconstruct the DNA sequence in a previous evolution stage for which the exact sequence is not known. A CAD tool for backtracking the DNA sequence evolution based on Cellular Automata (CA) and Genetic Algorithms (GAs) was developed. Furthermore, the proposed system is able of automatic production of synthesizable VHDL code corresponding to the CA model. More specifically, DNA is modeled as a one-dimensional CA with four states per cell, i.e. the four DNA bases A, C, T and G. Linear evolution rules, represented by square matrices, are considered. The evolution rule can be determined using the global state of the DNA sequence in various evolution steps. This determination is accomplished using GAs. Moreover, because of the final produced CA's binary states and its local rule simplicity, the hardware implementation of the proposed model is straightforward. Finally, the FPGA processor that executes the CA model was fully designed, placed and routed.

## 1 Introduction

Bioinformatics research has proven to be very successful. Thanks to the development of advanced biochemical and biophysical instrumentation methods, we are able to collect valuable information about genome and proteome sequences, and structures of biological macromolecules [1]. The collected data, however, are often noisy and ambiguous, and thus the need for better techniques to solve complex problems connected with proper interpretation and plausible reconstruction (in terms of models) of the obtained biochemical information. It requires more accurate and faster database and data processing technologies, and better computational intelligence algorithms. Biochemical and biophysical laboratories collect data only about constituent elements that must be combined, analysed, and processed in order to obtain valid bioinformatics models [2]. Fortunately, several of the existing computational intelligence techniques can be adopted for solving bioinformatics problems, and new methods are being developed almost daily. In general, we can use computational intelligence methods providing that they are wisely combined with bioinformatics, as illustrated by the authors in [2].

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 522–530, 2008.

Following the aforementioned outline, we figured out that DNA can be modelled as a one-dimensional Cellular Automaton (CA) [3]. In this model the phosphate chain corresponds to the CA lattice and the deoxyribose sugars to the CA cells. At each sugar molecule one of the four bases A (Adenine), C (Cytosine), T (Thymine) and G (Guanine) may bind. These four bases correspond to the four possible states of the CA cell. CAs appeared to be a promising model for DNA [3], because the DNA structure, function and evolution can be simulated using several mathematical tools (such as linear algebra and operators), introduced through the use of CAs. Following that line we developed a simulator, named *CAs for DNA*, for the study of DNA sequences with the help of CAs. *CAs for DNA* is an interactive simulation tool that includes a Graphical User Interface [GUI] which has been implemented using Matlab facilities. Moreover, in elementary CAs, the CA evolution rule can be extracted from a given number of CA evolution patterns. This method can also be applied to the CAs that model DNA. As a result, the developed simulator is able for modelling DNA evolution by extracting CA rules using Genetic Algorithms (GAs) [4]. The evolution rule can be determined by providing the global state of the DNA sequence in various evolution steps. Then, since the rule of evolution and the sequences of DNA are known for several evolution steps, it may be possible to determine the DNA sequence in previous evolution steps.

Finally, in order to speed up the application of CA to the study of DNA sequences the proposed tool is capable of producing Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) synthesizable code for the hardware implementation of the CA rules that model DNA. More specifically, *CAs for DNA* using a translation algorithm, that checks the CA parameters values previously determined by the user with the help of GA, automatically produces the synthesizable VHDL code that describes the CA algorithm. It should be mentioned that CAs are one of the computational structures best suited for hardware realization. The CA architecture offers a number of advantages and beneficial features such as simplicity, regularity, ease of mask generation, silicon-area utilization, and locality of interconnections. As a result, the automatically produced VHDL code can be fed into a commercial Field Programmable Gate Arrays (FPGA) CAD system, and the layout of the dedicated hardware that executes the CA algorithm can be designed to any FPGA Programmer. In this paper, the design processing of the finally produced VHDL code, i.e. analysis, elaboration and simulation, has been checked out with the help of the Quartus II, v. 7.2® design software of the ALTERA® Corporation. Test benches were automatically constructed by our system, for the simulation needs of the VHDL code, and the Simulator of Quartus® was used to simulate the operation of the dedicated processor described by the VHDL code obtained. Consequently, the implementation of the resulting VHDL code results in a FPGA, which is able to perform some real experiments, and to serve as a powerful "virtual lab" dedicated to the modelling of the backtracking of DNA sequences evolution.

## 2 Modeling DNA in Terms of Cellular Automata

In this session we suggest strategies for modelling DNA in terms of CAs by including a simultaneous translation of DNA properties into CAs. In the presented model the

phosphate chain corresponds to the CA lattice and the deoxyribose sugars to the CA cells. At each sugar molecule one of the four bases A, C, T and G may bind. These four bases correspond to the four possible states of the CA cell. In non-sexual reproduction, the DNA molecule is passed from an individual to its offspring, whereas in sexual reproduction, the DNA of the offspring consists of parts of the parental DNA. We define as an *evolution event* a change in state, which may occur in one or more CA cells. Therefore, mutation is an evolution event and it corresponds to cell state changes. The time step in CA evolution is the time interval between two CA cell changes and, therefore, the time flow is not uniform. A result of modeling DNA as a CA is that the DNA strand and the individuals passing it from one generation to the other may exist in different time scales and, therefore, the DNA evolution is time-like separated from the life of the individuals that carry it.

The main question that rises when one tries to model DNA is whether mutations are completely random or not. If mutations are completely random, then CAs, which are deterministic computational models, can not model DNA evolution. In this case probabilistic methods, such as Markov chains may be appropriate. Although the answer to this question is not known, there are some indications that mutations and, therefore, DNA evolution may not be completely random [5, 6].

We will proceed to the model construction by assuming that mutations, i.e. CA cell changes are not completely random, but depend on the states of some of the cells that are located near by. Neighbor-dependent mutation has been studied using Markov chains and revealed biases in mutation rates that depend on the neighboring bases. Suppose that a state change at the *i*th cell occurs, and a time step is taken. In the model presented here it is supposed that the state of this cell has changed as a result of the effect of the states of its neighbors. The new state of the *i*th cell at this time step (which is generally the $t+1$ step) is given by:

$$C_i^{t+1} = \hat{M}\left(C_{i-r}^t,...,C_{i-3}^t,C_{i-2}^t C_{i-1}^t,C_i^t,C_{i+1}^t,C_{i+2}^t,C_{i+3}^t,...,C_{i+r}^t\right) \quad (1)$$

where operator, $\hat{M}$, may be a mathematical function, a logic function, a matrix etc. In the case of linear evolution rules the operator $\hat{M}$ of equation (1) is a matrix, $M$ [3]. While, a vast number of evolution rules can be applied to the CA that models DNA, the study of linear rules reveals the dynamics of the CA evolution and provides a very good insight to the structures created by evolution. The use of linear rules is further justified by the fact that a linear algebra has already been successfully used to the analysis of mutation rates. Most of the studies on mathematical models of DNA are limited to nearest neighbor interaction. Because of that, we have chosen to use in our simulations an evolution rule that incorporates only nearest neighbor interaction. As a result, all the elements in a matrix row of $M$ are zero, except the three neighboring elements that are equal to one. In equation (1) cell states are one of the four bases A, C, T and G, which are represented by numbers of the quaternary number system, which contains only four numbers, i.e. *0, 1, 2* and *3*. We represent the bases with numbers as follows: $A \rightarrow 0$, $C \rightarrow 1$, $T \rightarrow 2$, and $G \rightarrow 3$. In elementary CAs, given an evolution pattern the evolution rule that generated it can be determined. It is very probable for such a method to exist for CAs that model DNA evolution. In this

case if the evolution of the DNA sequence at various time steps is given, it will be possible to determine the evolution rule (or rules) that generated this evolution. After that, since the evolution rule and the DNA sequence at present time are known, it may be possible to predict the next evolution event (or events) and, therefore, the DNA sequence at the next time step.

## 3 *CAs for DNA* CAD Tool

*CAs for DNA* is an automated simulation and hardware implementation tool for extracting with high success the CA evolution rule, or rules that model the evolution of DNA sequence with the usage of GAs. A vast number of evolution rules can be applied to the presented CA that models DNA evolution. The CA rule space comprises all the possible local rules that may be applied to the CA cells. For CAs with only two states per cell, the number of all possible rules is given by $2^{2^n}$, where n is the number of cells in the neighbourhood. In one-dimensional CAs with only two states per cell, the neighbourhood of which comprises the left, the right and the same cell, the number of all possible rules is $2^8$, while in the four-state CA these rules are extended to $4^{4^3}$ or $4^{64}$. The whole rule space of such CAs must be searched in order to find the possible CA evolution rules that model the DNA sequence evolution. In this work GAs are used to search the CA rule space. A possible evolution scheme of the proposed CA is shown in Fig. 1, where the first row gives all the possible states the cells within the neighbourhood could take. The $r_i$'s in the second row are the rule components which take values from the discrete set {0, 1, 2, 3}. The last row shows the coefficients associated with the corresponding components. The rule $R$ can be defined as $R=(r_0, r_1, r_2, r_3, \ldots, r_{63})$. The numerical label $D$ assigned to $R$ is given by: $D = \sum_{s=0}^{2^6-1} r_s 2^s$, which is simply the sum of the coefficients associated with all nonzero components.

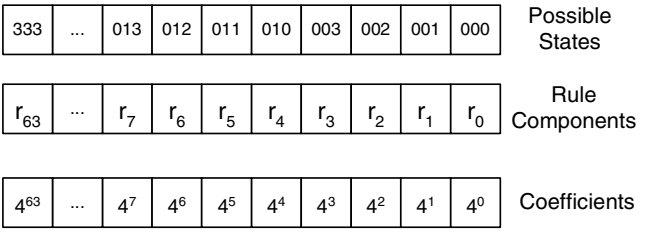



**Fig. 1.** Evolution rules of the four state classical CA model of DNA evolution

After the assignment of the origin DNA sequence by the user of the simulation tool, an initial population $P$ that contains $n$ possible solutions, meaning $n$ CA evolution rules is constructed randomly. The value of $n$ is user-defined and should be a compromise between accuracy and computer time and memory. For each possible solution $i$ of population $P$ with $n$ individuals an error function is given by:

$$Mer(i) = \sum_{j=0}^{SET} | \, y(i,j) - \hat{y}(i,j) \, | \tag{2}$$

where $y(i, j)$ is the measured state at data point $j$ for chromosome $i$ and $\hat{y}(i, j)$ is the predicted state, in correspondence. Each chromosome in the current population is ranked with respect to *Mer* of equation (2). The chromosome with the least *Mer* occupies the first position, the chromosome with the second least *Mer* occupies the second position and so on. Chromosomes with the same error share the same rank. After the final ranking we calculate the fitness function for each chromosome. The fitness function of the $i^{th}$ chromosome is defined as:

$$fit(i) = \frac{MAX(rank(i)) - rank(i)}{MAX(rank(i)) - MIN(rank(i))} \tag{3}$$

The pseudocode of the GA algorithm for the selection of CA evolution rules can be summarized as follows:

**Table 1.** Pseudocode of the GA algorithm for the selection of CA evolution rules

| Pseudocode | Comments |
|---|---|
| 1. Start | |
| 2. Set the current generation number $i$ =1. | /* Take a time step */ |
| 3. Set the GA algorithm parameters. | /* User defined GA parameters */ |
| 4. Generate the population set $P$ with $n$ individuals. | |
| 5. Compute *Mer* (modulus of error function) for each individual in $P$. | /* Use Equation 2 */ |
| 6. Rank the individual in $R$. | |
| 7. Calculate the fitness function. | /* Use Equation 3 */ |
| 8. Apply the parent selection technique to $P$. | |
| 9. Employ crossover and mutation to $P$ to produce the corresponding offspring set $P$'. | |
| 10. Calculate the corresponding fitness function for the chromosomes in the offspring set $P$'. Select the n fittest individual from both the population set $P$ and the corresponding set $P$', by comparing the fitness value. Reset $P$ using the corresponding newly selected n individuals and nullify the offspring set $P$'. | |
| 11. Set the generation number $i$=$i$+1. | |
| 12. If generation number less than a prespecified number of generations $G$<br>Go to 7 and repeat until has been reached<br>Else finish. | /* $G$ is the user defined number of generations */<br><br>/* Final state */ |
| 13. Stop | |

A paradigm of the functional operation of *CAs for DNA* is presented in Fig. 2. These simulations show that the evolution data visualization is straightforward, and the evolution patterns can be studied and interpreted [7].

## 4 Automatic FPGA Implementation

To implement the aforementioned CA model in hardware, synchronous very large scale integrated (VLSI) circuits should be used. These implementations could lead to dedicated FPGA processors that can be designed using commercially available FPGA CAD systems. Furthermore, the hardware implementation of the algorithms could be achieved after the manual translation of their parts into a synthesizable subset of a hardware description language (HDL).

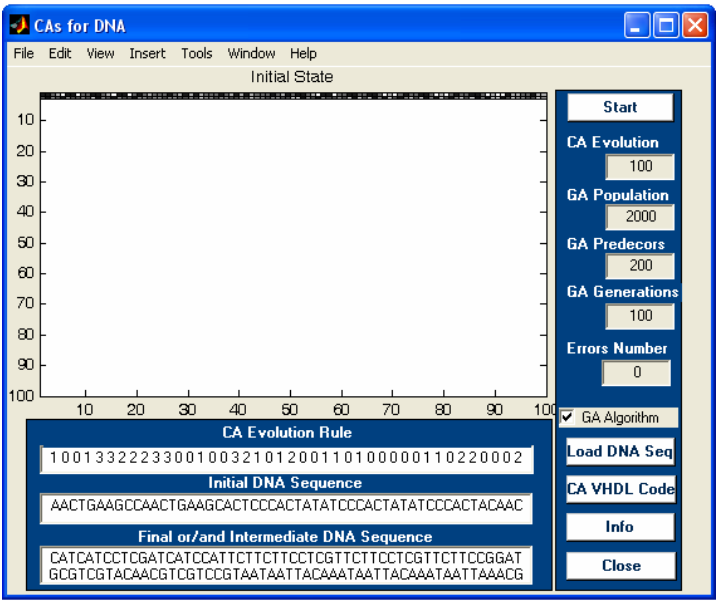



**Fig. 2.** The final screen of the *CAs for DNA* after the execution of the GA algorithm. (The graphical user interface of the simulator) (A: white, C: dark gray, T: light gray and G: black.)

The presented system *CAs for DNA* is able of automatic translation of the CA algorithm's code into synthesizable VHDL code based on the user's choice of simulation parameters. There are many reasons for implementing an algorithm, which simulates a system, using a hardware description language, and especially VHDL, instead of using standard VLSI design CAD tools. Mainly, because the VHDL models present the most reliable design process with the minimum cost and time and, furthermore, because they are capable of avoiding design errors. Furthermore, because the execution time in software depends on the complexity of the rule, while the execution time (throughput) in hardware is almost independent of the rule complexity. In our CAD tool, the primary parameters of the translation algorithm are used to produce the VHDL code. In the beginning, the CA algorithm is read by the translation algorithm. After the CA algorithm is read, the translation algorithm searches the CA code to detect the CA rule found by the GA algorithm in order to produce the VHDL code for the main component, i.e. the CA cell. This will be the behavioral part of the final

VHDL code, containing process and signal assignment statements. Subsequently, the translation algorithm searches the CA code to detect the lattice size, the boundary and initials CA conditions, in order to construct the structural part of the final VHDL code. The structural part implements the final module as a composition of subsystems, like the aforementioned main component (schematic Fig. 3). The final VHDL code produced by translation algorithm, including both the behavioral and structural parts, addresses all the basic VHDL concepts (i.e. interfaces, behavior, structure, test benches) included in the IEEE Standard 1076-2002. No previous knowledge of VHDL is required, since the VHDL code is directly produced from the high-level programming language code through the translation algorithm. However, there is always a possibility of functional simulation of the VHDL code with the use of the appropriate automatically generated test benches. The simulation results of the VHDL code are guaranteed to be found in complete agreement with the compilation results of the CA algorithm, produced during the phase of estimating the CAs algorithm performance and of verifying its functional correctness.

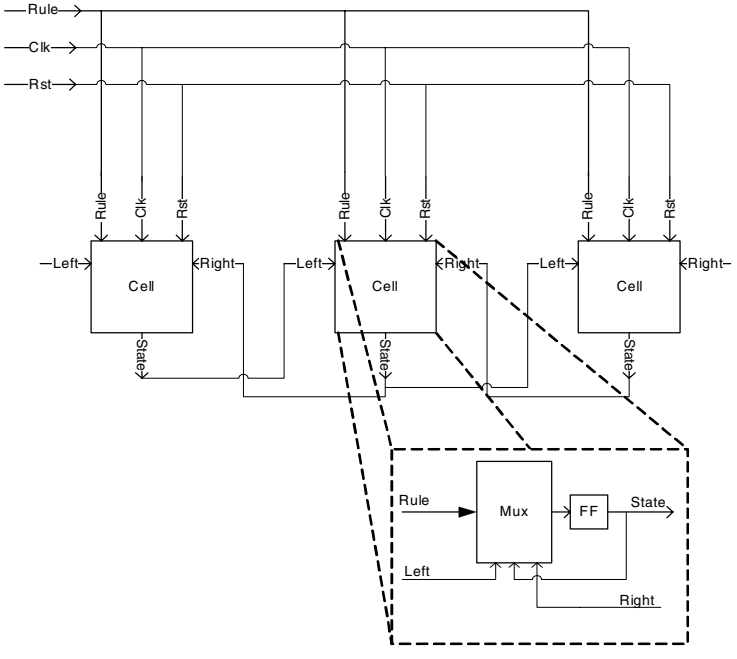



**Fig. 3.** Three (3) cells neighbourhood of the proposed CA architecture and its corresponding basic structural element

The automatically produced synthesizable VHDL CA code is translated into a hardware schematic of the defined architecture using predetermined timing constraints in Quartus II, v. 7.2® design software. The next step includes translation and mapping. In this phase the hardware schematic is mapped to the specific hardware of the FPGA and the communication channels between the generated components are

specified. The final phase is the generation of a configuration file. Finally, this translates the mapped design into a stream of bits that control the switchboxes, LUTs and other components of the FPGA. It should be mentioned that for hardware implementation purposes, the performance and the size of reconfigurable hardware such as FPGAs have been drastically improved in the last several years. With the latest FPGA chips, more than one hundred grids can be computed in one clock cycle (less than 50 ns), and the reconfigurability of FPGAs makes it possible to compute any kind of CA on the same chip.

Design of the proposed processor results in an ALTERA Stratix EP1S10F484C5 FPGA device, which indicates a maximum clock rate around 240MHz and consists of 100 CA cells. Initial data is loaded in a semi-parallel way and the automatic response of the processor provides the CA evolution of the DNA sequence under test. More specifically, inputs to the dedicated processor are the lines through which the initial conditions are transferred to the CA, the clock, the reset and load control signals, the boundary condition signals, as well as the power and ground connections. Furthermore, for comparison purposes we have evaluated the speed of the traditional software Matlab CA code implementation running on a typical Pentium IV 3GHz Windows XP computer system and the results justify the aforementioned integration of the FPGA processor. More specifically, a speed-up of 6 times for a medium length CA and a speed up of 23 times for an extra long length CA was measured concluding that the implementation of CA is significantly faster in FPGA hardware than in optimized software, thus enabling real parallel processing of data using custom digital structures. As a result, the proposed CA is running faster when implemented to a dedicated ASIC processor compared to a general purpose computer.

## 5 Conclusions

In this paper, *CAs for DNA*, an automated simulation and hardware implementation tool for DNA sequence evolution by extracting CA rules with the usage of proper GAs was developed. *CAs for DNA* was based on a CA DNA evolution model. Based on this model, a CAD tool of DNA evolution was developed, a GA methodology has been presented to determine the evolution rules generating given evolution patterns and a fast FPGA processor that executes the CA model was fully automatically designed, placed and routed. Speed is extremely significant in this application domain and it is really important to observe that hardware performance becomes available out of a general purpose FPGA card. As future work, the FPGA processor as well as the GA algorithm for the selection of the CA evolution rule, can be calibrated with real data (DNA sequences) of different microorganisms in various evolution steps targeting to the production of suitable drugs. Moreover, Register Transfer Level (RTL) design could be used by the proposed tool for the CA implementation giving better results. Furthermore, this work will facilitate the development of CA models of the self-organizing properties of DNA. As CAs models are developed, they are expected to contribute to the interpretation of DNA sequences, and possibly indicate new directions in the field of artificial intelligence for bioinformatics.

## References

1. Cios, K.J., Mamitsuka, H., Nagashima, T., Tadeusiewicz, R.: Computational intelligence in solving bioinformatics problems. Artificial Intelligence in Medicine 35, 1–8 (2005)
2. Baxevanis, A.D., Ouellette, B.F.: Bioinformatics, a practical guide to the analysis of genes and proteins. Wiley-Interscience, New York (1998)
3. Sirakoulis, G.C., Karafyllidis, I., Mizas, C., Mardiris, V., Thanailakis, A., Tsalides, P.: A cellular automaton model for the study of DNA sequence evolution. Computers in Biology and Medicine 33, 439–453 (2003)
4. Goldberg, D.A.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
5. McFadden, J., Al-Khalili, J.: A quantum mechanical model of adaptive mutation. BioSystems 50, 203–211 (1999)
6. Schwefel, H.P.: Deep insight from simple models of evolution. BioSystems 64, 189–198 (2002)
7. Mizas, C., Sirakoulis, G.C., Mardiris, V., Karafyllidis, I., Glykos, N., Sandaltzopoulos, R.: Reconstruction of DNA sequences using genetic algorithms and cellular automata: Towards mutation prediction? Biosystems 92, 61–68 (2008)

# From Data and Signals Cellular Automata to Self-organizing Circuits

André Stauffer and Joël Rossier

Logic Systems Laboratory, Ecole polytechnique fédérale (EPFL),
IN-Ecublens, CH-1015 Lausanne, Switzerland
{name.surname}@epfl.ch
lslwww.epfl.ch

**Abstract.** Self-organizing circuits are able to grow, to self-replicate, and to self-repair. These properties are implemented in hardware thanks to configuration, cloning and cicatrization mechanisms. They are realized by the configuration layer of the POEtic tissue, a data and signals cellular automata based circuit. Specified as a data-flow processor, the application and routing layers of the circuit compose a timer as an application example.

## 1 Introduction

Borrowing three structural principles (multicellular architecture, cellular division, and cellular differentiation) from living organisms, we have already shown how embryonic hardware [1] is able to implement bio-inspired properties in silicon. This hardware implementation leads to self-organizing circuits, based on data and signals cellular automata (DSCA), capable to deal with faults in a fully automatic way. In Section 2, the self-organizing mechanisms are implemented as a configuration layer in the POEtic tissue, a reconfigurable circuit that draws inspiration from the structure of complex biological organisms. The application and routing layers of the circuit are then specified in order to define a data-flow processor, the MOVE processor (Section 3). Using four such processors, a timer is realized as an application example. A brief conclusion (Section 4) summarizes our paper and opens new research avenues.

## 2 DSCA Based Reconfigurable Circuit

### 2.1 Circuit Characteristics

The *POEtic tissue* [2] is a reconfigurable circuit that draws inspiration from the structure of complex biological organisms to implement the three main models commonly used in bio-inspired systems: (1) *Phylogeny*, the history of evolution of the species through time, (2) *Ontogeny*, the development of an individual as directed by its genetic code, and (3) *Epigenesis*, the development of an individual through learning processes. This tissue is the first hardware substrate

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 531–536, 2008.

dedicated to the implementation of systems able to combine the three axes of bio-inspiration into one single circuit. Physically, the tissue is a 2-dimensional array of molecules and each molecule is composed of the three layers described in the following paragraphs.

### 2.2 Configuration Layer

The *configuration layer* implements the self-organizing mechanisms and their constituting processes [3]. This layer is designed as data and signals cellular automaton (DSCA) cell [4], resulting from the interconnection of a processing unit handling the data and a control unit computing the signals. In the detailed architecture of our layer (Fig. 1a), six resources implement the *growth* and *branching processes*:

- An input multiplexer DIMUX, selecting one out of the four input data $NDI$, $EDI$, $SDI$ or $WDI$.
- A 2N-level stack organized as N genotypic registers G1 to GN (for mobile data), and N phenotypic registers P1 to PN (for fixed data).
- An output buffer DOBUF producing the output data $DO$.
- An encoder ENC for the input signals $NSI$, $ESI$, $SSI$, and $WSI$.
- A transmission register I for the memorization of the input selection.
- A generator GEN producing the output signals $NSO$, $ESO$, $SSO$, and $WSO$.

In order to implement the *load*, *repair* and *reset processes*, the architecture of the configuration level (Fig. 1a) requires four supplementary resources:

- A decoder DEC defining the mode and the type of the molecule.
- A signal register S.
- A mode register M.
- A type register T.

To allow the bypassing of the spare, faulty or repair molecules, data and signals transmission multiplexers and demultiplexers are added to the configuration layer. Depending on its molecular type T, the layer finally controls its output signals with buffers. These buffers limit the propagation of the load and reset signals according to the boundaries of the cell.

### 2.3 Application Layer

The *application layer* implements the logic design of the system under development as well as its short range connections between neighboring molecules. The core of this layer (Fig. 1b) is made up of four resources:

- An input multiplexer AIMUX, selecting four inputs out of the four application data $NAI$, $EAI$, $SAI$, $WAI$, and the routing data $RO$.
- A 16-bit look-up table LUT.

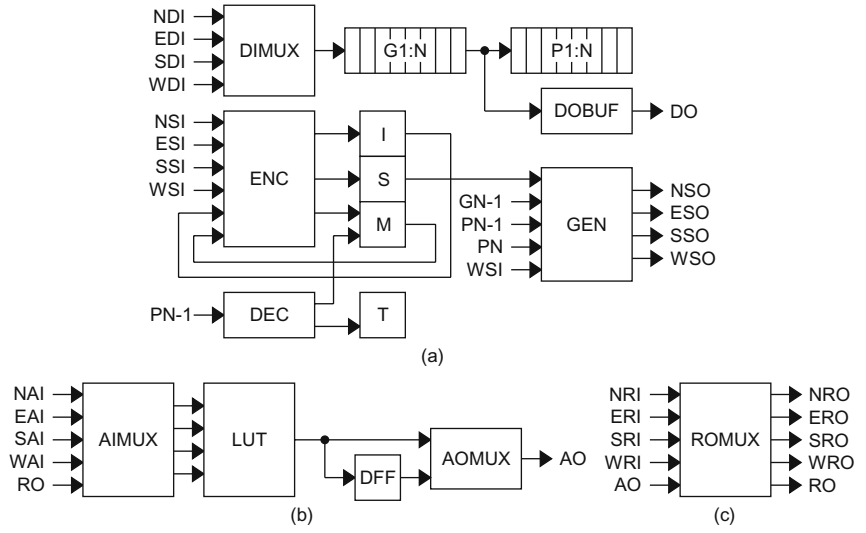



**Fig. 1.** Detailed architecture of a molecule. (a) DSCA cell corresponding to the configuration layer. (b) Application layer. (c) Routing layer.

– A D-type flip-flop DFF for the realization of sequential systems.
– An output multiplexer AOMUX selecting the combinational or the sequential data as application output $AO$.

To allow the bypassing of the spare, faulty or repair molecules, transmission multiplexers are added to the application layer.

### 2.4 Routing Layer

The *routing layer* handling the short range connections between distant molecules is made of a single resource (Fig. 1c):

– An output multiplexer ROMUX selecting the five outputs $NRO$, $ERO$, $SRO$, $WRO$, and $RO$ out of the four routing input data $NRI$, $ERI$, $SRI$, $WRI$, and the application output data $AO$.

To allow the bypassing of the spare, faulty or repair molecules, transmission multiplexers are added to the routing layer.

## 3 Multi-processor Application

### 3.1 MOVE Processor

The *MOVE processor*, originally developed as an application-specific dataflow processor [5], relies on a set of functional units connected together by a bus. Fig. 2 details the constituting resources of the processor:

– A program memory PM.
– An instruction fetch unit IF.
– Four functional units with input registers RI and output registers RO.
– Two communication units with address registers ADR and data registers DATA.

The instructions of the processor move operands into the input registers RI of the functional units and move the result from their output registers RO. Using their address registers ADR and their data registers DATA, the communication units are handled in the same way as the functional units.

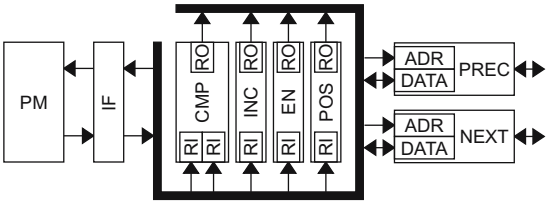



**Fig. 2.** Detailed architecture of the MOVE processor

### 3.2 Timer Application

A *timer* counting seconds (from 00 to 59) and minutes (from 00 to 59) is realized as a chain of four counters SU (units of seconds), ST (tens of seconds), MU (units of minutes), and MT (tens of minutes). This application involves four MOVE processors that are specified as modulo-10 counters for the units (seconds SU or minutes MU) and modulo-6 counters for the tens (seconds ST or minutes MT). Each processor contains consequently the following specialized functional units (Fig. 2):

– A comparator CMP.
– An incrementer INC.
– A count enable unit EN.
– A position unit POS for the location within the chain.

The chaining of the four processors is implemented at the programmable circuit level and realized by a distributed long range routing algorithm [6] that dynamically connects the output communication units NEXT to the input communication units PREC.

### 3.3 POEtic Implementation

Each MOVE processor, using one basic cell of the POEtic tissue for each of its $30 \times 12$ molecules, implements a counter cell of the timer organism. In order to build this organism (Fig. 3a), the structural configuration mechanism, the functional configuration mechanism, and the cloning mechanism are applied at the cellular level. Starting with the structural and functional configuration data of

the MOVE processor, these mechanisms generate successively the four counters of the timer.

The cicatrization mechanism results from the introduction in each cell of one column of spare molecules to the right (Fig. 3a), defined by the structural configuration of the MOVE processor, and the automatic detection of faulty molecules. Thanks to this mechanism, the faulty molecule of the upper right cell (Fig. 3b) is deactivated, isolated from the network, and replaced by the nearest right molecule, which will itself be replaced by the nearest right molecule, and so on until a spare molecule is reached. The functional reconfiguration mechanism takes then place in order to regenerate the counter cell of the timer organism. As shown in Fig. 3b, the display of the regenerated counter cell presents some graphical distortion.

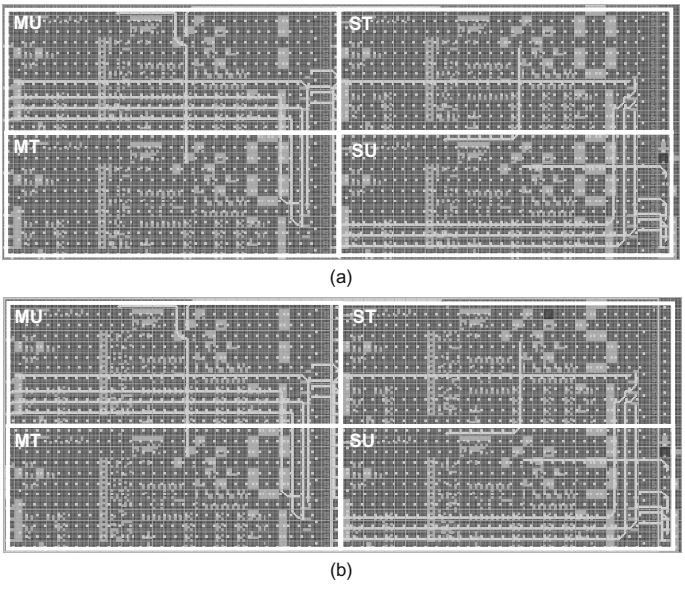



**Fig. 3.** POEtic implementation of the timer. (a) Original configuration of the two rows by two columns of processors displaying clockwise MT,MU:ST,SU=01:08. (b) Cicatrization of the upper right processor displaying a distorted ST=5.

## 4 Conclusion

The self-organizing mechanisms are made of simple processes like growth, load, branching, repair, and reset. They allow the cellular systems to possess bio-inspired properties such as:

- Cloning or self-replication at cellular and organismic levels.
- Cicatrization or self-repair at the cellular level.

Starting with the POETIC tissue, a DSCA based reconfigurable circuit, we described first the configuration layer of its basic cell implementing the self-organizing mechanisms and their underlying processes. We detailed then the corresponding application layer and routing layer as well as their specifications in order to define a MOVE processor. We finally realized a timer made up of four such processors as an application example.

In order to improve our systems, we intend to study additional hardware features such as:

- Automatic detection of faulty molecules, erroneous configuration data, and application dysfunction.
- Asynchronous implementation at the organismic level and synchronous implementation at the cellular level.

## References

1. Mange, D., Stauffer, A., Petraglio, E., Tempesti, G.: Embryonics Machines that Divide and Differentiate. In: Ijspert, A.J., Murata, M., Wakamiya, N. (eds.) BioADIT 2004. LNCS, vol. 3141, pp. 328–343. Springer, Heidelberg (2004)
2. Tyrrell, A., Sanchez, E., Floreano, D., Tempesti, G., Mange, D., Moreno, J.-M., Rosenberg, J., Villa, A.: Poetic Tissue: An Integrated Architecture for Bio-inspired Hardware. In: Tyrrell, A.M., Haddow, P.C., Torresen, J. (eds.) ICES 2003. LNCS, vol. 2606, pp. 129–140. Springer, Heidelberg (2003)
3. Stauffer, A., Mange, D., Rossier, J., Vannel, F.: Bio-inspired Systems with Self-developing Mechanisms. In: Kang, L., Liu, Y., Zeng, S. (eds.) ICES 2007. LNCS, vol. 4684, pp. 151–162. Springer, Heidelberg (2007)
4. Stauffer, A., Sipper, M.: The Data-and-Signals Cellular Automaton and its Application to Growing Structures. Artificial Life 10(4), 463–477 (2004)
5. Corporaal, H., Mulder, H.: MOVE: A Framework for High-performance Processor Design. In: Proceedings of the Int. Conference on Supercomputing, pp. 692–701 (2003)
6. Moreno, J.-M., Sanchez, E., Cabestani, J.: An In-system Routing Strategy for Evolvable Hardware Programmable Platforms. In: Keymeulen, D., Stoica, A., Lohn, J., Zebulum, R.S. (eds.) Proceedings of the Third NASA/DOD Workshop on Evolvable Hardware, pp. 157–166. IEEE Computer Society, Los Alamitos (2001)

# Integrated Simulation and Information Sharing System for Disaster Mitigation

Itsuki Noda

Information Technology Research Institute
National Institute of Advanced Industrial Science and Technology
i.noda@aist.go.jp

**Abstract.** When a huge disaster occurs, many phenomena, not only physical one like land-slides but also human factors like evacuation and rescue, emerge and influent with each other. In order to mitigate damages from such a disaster, we need a tool to understand and to investigat the disaster as a whole. I have been joined several research projects for disaster mitigation in Japan, in which we have been developing an integrated disaster-and-rescue simulation framework and a disaster mitigation information sharing platform. A key concept of these framework and platform is modularity and integration. Because no researcher can be a specialist of all phenomena of the disaster, we need a simple and flexible framework to combine simulation technique for each phenomenon. I show several example of application of these works like fire-fighting, traffic under disaster, and evacuation from terrors.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, p. 537, 2008.

# Towards an Ontology for Crowds Description: A Proposal Based on Description Logic

Stefania Bandini, Sara Manzoni, and Stefano Redaelli

CSAI - Complex Systems & Artificial Intelligence Research Centre
University of Milano–Bicocca
{bandini,manzoni,redaelli}@disco.unimib.it

The research context of this paper refers to bottom–up approaches to *crowd dynamics* that is, the study of how and where crowds form and move [1]. Several phenomena like *crowd aggregation, dispersion* and *self–organized movement* have been observed and studied by multiple disciplines interested to crowds (e.g. physics, sociology, ethology, social and behavioral psychology, building design, urban planning, security management, among others), each one with its specific viewpoint and ontological setting. SCA4CROWDS is an interdisciplinary research within this context that aims at contributing towards the development of a unifying ontology on crowds allowing the integration of contributions coming from several disciplines and that could be exploited for scientific and applicative issues (e.g. model comparison, validation, calibration). Potential exploitations of SCA4CROWDS results are towards the support of design and management of public crowded spaces and events to improve security, safety and comfort of people. SCA4CROWDS, in particular, aims at developing formal and computational tools to support the design, execution and analysis of crowds' behavior as effect of individual interactions (e.g. physical, social, emotional) according to Situated Cellular Agent (SCA) [2]. SCA is a modeling and simulation framework to model and study crowd dynamics phenomena with an approach based on Multi–Agent Systems (MAS) and Cellular Automata [3] principles.

In this paper we present the ontological framework for crowds' study we developed according to Elias Canetti work [4], in which a classification and ontological description of the crowd has been proposed as result of 40–years of empirical observations and studies from psychological and anthropological viewpoints. In crowds and CA literature, a formal analysis with CA-based models of theories developed within human sciences context has previously been proposed in [9]. Elias Canetti can be considered as belonging to the tradition of social studies that refer to *the crowd as an entity dominated by uniform moods and feelings*. We preferred this work among others (see for instance [5,6,7,8]) due to its clear semantics and explicit reference to concepts of *loss of individuality*, *crowd uniformity*, *spatio-temporal dynamics* and *discharge*, that could be fruitfully represented by modeling approaches like SCA and Cellular Automata in general. In the following we introduce a formal language belonging to the family of Description Logics (DL [10]) that we propose for the formal representation of conceptual description of crowds and their dynamic phenomena. Description Logics is a family of knowledge representation languages that are used to develop various applications to the issues of data management, including: expressing the

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 538–541, 2008.

conceptual domain model/ontology of the data source, integrating multiple data sources, and expressing and evaluating queries. First, we define the basic syntax and semantics of the DL language we propose and then we describe, according to this language, basic concepts of Canetti work [4].

## 1 DL Language for Ontology Representation

A DL language is defined by a set of *concepts*, the elements of the described subject, and a set of *roles* that specify the type of relationships that can hold between concepts. The box on the left hand side of Figure 1 shows the graphical representation of concepts and roles we defined. Atomic concepts (i.e $ENTI \sqcap QUAL \sqcap PSYC \sqcap PHEN \sqsubseteq \bot$) are:

- $ENTI$, entities representing the involved objects;
- $QUAL$, qualities, characterizing an $ENTI$;
- $PSYC$, psychological states;
- $PHEN$, phenomena involving $ENTI$s.

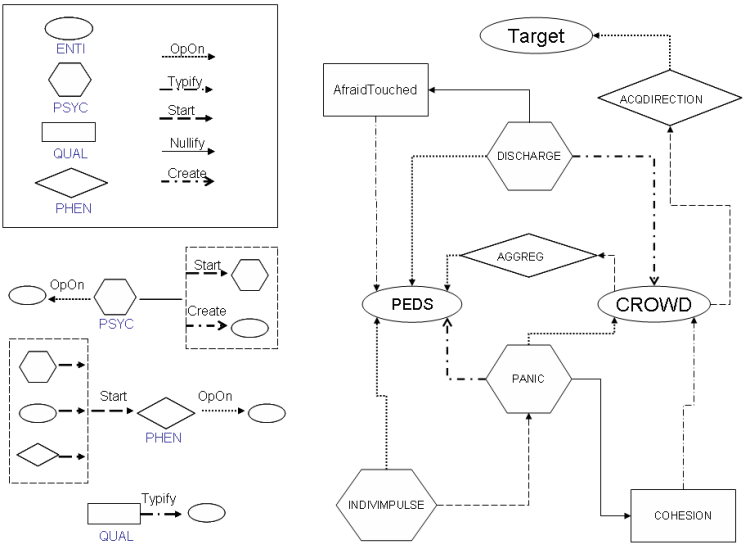



**Fig. 1.** On the left hand side the graphical representation of the used language basic elements and their relations, composed by the basic concepts and roles shown in the box. On the right hand side a schematic representation of the principal elements of Canetti's crowd ontology.

All atomic roles, relationships between atomic concepts, are not–symmetrical, transitive, and reflexive (with the exception of $Typify$ that is not reflexive):

- *OpOn* specifying that a concept "operate on" another concept;
- *Create*, when a concept is the cause of the birth of another one;

- $Typify$ that indicates a concept as an attribute of another;
- $Start$ when a concept is the starting condition for the occurrence of another;
- $Nullify$ specifying that a concept is the cause of the termination of another.

Language semantics is defined by the following relationships (graphically represented on the left hand side of Figure 1):

- a psychological state (i.e. $PSYC$) is a concept that operates on an entity (i.e. $ENTI$), and starts a psychological state (i.e. $PSYC$) or creates an entity ($ENTI$):
  $PSYC \sqsubseteq (=1)OpOn.ENTI \sqcap ((=1)Start.PSYC \sqcup (=1)Create.ENTI)$
- a phenomenon (i.e. $PHEN$) is a concept that is started by an entity (i.e. $ENTI$) or a psychological state (i.e. $PSYC$) or a phenomenon (i.e. $PHEN$), and operates on an entity (i.e. $ENTI$):
  $PHEN \sqsubseteq (=1)Start^{-}.(ENTI \sqcup PSYC \sqcup PHEN) \sqcap (=1)OpOn.ENTI$
- a quality (i.e. $QUAL$) is a concept that typifies an entity (i.e. $ENTI$):
  $QUAL \sqsubseteq (=1)Typify.ENTI$

## 2 Elias Canetti's Crowd

Elias Canetti's definition of "*crowd*" can be summed up as follows:

> ... a unic entity dominated by uniform moods and feelings; it is characterized by the spontaneous will of growing and aggregating other pedestrians, and has a target, that is identified as a location of the environment or an object that all the individuals aggregated into the crowd desire. The *aggregation* phenomenon describes the growing effect that starts from an aggregative psychological impulse called the "*discharge*" . The "*discharge*" occurs spontaneously in people and overcomes the natural social repulsive behavior of the "*fear to be touched*". On the other side, crowd disgregation is the result of an other psychological impulse called "*panic*", rising as the result of "*individulistic impulses*".

According to the DL language above described, crowd basic entities are $CROWD \sqcup PEDS \sqcup TARGET \sqsubseteq ENTI$, where $PEDS$ are "pedestrians", $CROWD$ is the aggregation of pedestrians that Canetti considers as a single entity, $TARGET$ is the location or the object desired by the crowd. Qualities are $AFRAIDTOUCHED \sqcup COHESION \sqsubseteq QUAL$ (where $AFRAIDTOUCHED$ represents the social repulsive behavior, i.e. *afraid to be touched*), and $COHESION$ is the crowd desire to remain compact. Psychological states are $DISCHARGE \sqcup PANIC \sqcup INDIVIMPULSE \sqsubseteq PSYC$, representing Canetti's *discharge*, *panic* and *individualistic impulses*, respectively. Finally, phenomena are: $AGGREG \sqcup ACQDIRECTION \sqsubseteq PHEN$, where $AGGREG$ represents the "aggregation" phenomenon and $ACQDIRECTION$ represents the crowd motion towards its "target". Relationships between concepts are graphically expressed in the right hand side of Figure 1, and in the DL language above defined:

- $PEDS \sqsubseteq \forall OpOn^{-}.DISCHARGE \sqcap \forall OpOn^{-}.INDIVIMPULSE$
- $DISCHARGE \sqsubseteq \forall Create.CROWD$
- $CROWD \sqsubseteq \forall OpOn^{-}.PANIC \sqcap \forall Start.AGGREG \sqcap \forall Start.ACQDIRECTION$
- $PANIC \sqsubseteq \forall Create.PEDS$
- $AFRAIDTOUCHED \sqsubseteq \forall Typify.PEDS \sqcap \forall Nullify^{-}.DISCHARGE$
- $INDIVIMPULSE \sqsubseteq \forall Start.PANIC$
- $AGGREG \sqsubseteq \forall OpOn.PEDS$
- $ACQDIRECTION \sqsubseteq \forall OpOn.TARGET$
- $COHESION \sqsubseteq \forall Typify.CROWD \sqcap \forall Nullify^{-}.PANIC$

## 3 Concluding Remarks

We presented a formal language belonging to the family of Description Logics that we proposed for the representation of the ontological description of crowds according to Elias Canetti psychological and anthropological studies. The main contribution of this work is to provide a semantically clear reference ontology for crowds' models based on SCA (and on any CA-like approach). The formal definition of such a reference ontology can be exploited, from one hand, for their calibration and validation with available data sets, and on the other hand, for their comparison with other CA-based models. Currently, we are integrating the above presented ontology into SCA formal and computational framework, e.g. to study crowds' aggregation phenomenon.

## References

1. Still, K.: Crowd Dynamics. PhD thesis. University of Warwick (2000)
2. Bandini, S., Federici, M., Vizzari, G.: Crowd modeling and simulation: the situated cellular agents approach. Cybernetics and Systems 38(7), 729–753 (2007)
3. Bandini, S., Manzoni, S., Simone, C.: Enhancing cellular spaces by multilayered multi agent situated systems. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493. Springer, Heidelberg (2002)
4. Canetti, E.: Crowds and Power. Farrar, Straus and Giroux (1984)
5. Blumer, H.: Collective Behavior. Irvington Publishers (1993)
6. Bon, G.L.: The Crowd: a study of the pupular mind. Dover Publications (2002)
7. Turner, L., Killian, L.: Collective Behavior. Prentice Hall College Div. (1987)
8. Levy, L.: A study of sports crowd behavior: The case of the great pumpkin incident. Journal of Sport and Social Issues (1989)
9. Adamatzky, A.: Dynamics of Crowd–Minds:Patterns of Irrationality in Emotions, Beliefs and Actions. Series on Nonlinear Science, vol. 54. World Scientific, Singapore (2005)
10. Staab, S., Studer, R.: Handbook on ontologies. Birkhäuser, Basel (2004)

# Towards a Multi-agent Model for Planning and Design of Exposition Spaces

Ivan Blecic[1], Arnaldo Cecchini[1], Giuseppe A. Trunfio[1],
Jian Zhou[2], Wei Zang[2], and Bing Liu[2]

[1] Department of Architecture and Planning - University of Sassari
Palazzo del Pou Salit, Piazza Duomo, 6 I07041 Alghero (SS), Italy
{ivan,cecchini,trunfio}@uniss.it
[2] College of Architecture and Urban Planning - Tongji University
1239 Siping Road, Shanghai 200092, P.R. China
{zhou,zang,Liu}@mail.tongji.edu.cn

**Abstract.** In this paper we present a multi-agent model for supporting the planning, design and organisation of exposition spaces. The model and its related software implementation simulates the behaviour of agents during their visit of an exposition with the objective to evaluate their overall comfort and satisfaction. It takes into account both the architectural and environmental aspects as well as the interaction among agents simulating possible phenomena of crowding and other forms of competition. The representation of the relevant phenomena in the model relies on the coexistence of both vector and cellular (raster) simulation approaches, taking into account the trade-off between the computational efficiency and the need for precision.

## 1 Introduction

The objectives of the research programme whose assumptions, approach and preliminary results are hereby outlined is to develop an agent-based simulation model (*i*) to assist the design of exposition spaces and environments, (*ii*) to support the organisation and planning of public expositions with massive participation, and (*iii*) to help the management of flows of visitors in expositions, museums, public interactive spaces. For these purposes, the presented simulation model, together with its operative software implementation, is designed and developed (*i*) to simulate and predict the behaviour and paths of movement of visitors, (*ii*) to estimate the time of stay in places and to discover potential phenomena of crowding, (*iii*) to evaluate the comfort and the satisfaction of visitors, and finally (*iv*) to evaluate the effectiveness of services and information systems designed for the exposition environment.

## 2 Outline of the Model

In the simulation model, the architectural environment is populated with stationary objects (e.g. walls, doors, attractions, etc.) and agents moving in the

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 542–545, 2008.

continuous space. Every entity is characterised by a shape with its vector representation. Without losing essential geometric information, the walkable surfaces are mapped onto horizontal planes and a uniform grid is used to compute and store auxiliary data. The coexistence of both raster and vector representations supports different agents' capabilities of spatial perception, in an attempt to use an appropriate combination of the two, given the trade-off between computational efficiency and precision. In particular, the raster space is mainly used to manage agents' micro-level planning activities (like choosing an attraction in a room or selecting suitable view-points), while those spatial activities which require greater accuracy (such as agents' visual perception of attractions for their appreciation) are managed in the vector space. To model the theoretical fruition in absence of other agents, we assume that every attraction projects a static fruition space around itself, accounting for the presence of obstacles, lighting conditions and other phenomena. Every point in that space has a static fruition suitability (see Fig. 1). The suitability of every cell is pre-computed once at the beginning of the simulation, and it considers the raster space as a cellular automaton where the vector fields emitted by attractions propagate along the cells. Thus, the neighbourhood pattern and the rules of propagation, that have been designed in order to allow obstacles to block the diffusion of visual signals [1], determine the static fruition suitability of every cell for every attraction. However, the actual fruition of an attraction by an agent is further influenced by the presence and the position of other agents (which can partially or totally obstruct the view, provoke crowding, and compete for best view-points). This brings about the need to pass from the static to a dynamic fruition space representing the actual fruition suitability. During a simulation, a raster representation of the dynamic fruition space is obtained through the use of a cellular automaton where each cell holding an agent emits a field propagating in the direction opposite to the attraction. For each cell, such field represents the decay of the static fruition suitability, and leads to the actual suitability value. Furthermore, there is a decay of suitability around every agent in order to account for the decrease of fruition comfort due to the crowdedness (see Fig. 2).

### 2.1 Behaviour of Agents

Every agent in the model expresses a personal interest for different categories of attractions (e.g. different painting styles and periods) and for specific attractions within these categories (e.g. one specific painting). The goal of each agent is hence to maximise a satisfaction function through the fruition of attractions, given the limited time available. The fruition of an attraction is largely based on a process of visual perception, whose quality and effectiveness depends (*i*) on the position of the agent relative to the attraction, (*ii*) on the topology of the room and (*iii*) on the presence of other agents which may be the source of view obstruction, crowding and generally decay of the fruition comfort. To maximise the satisfaction function, an agent must explore the environment looking for interesting attractions, given its system preferences and given its knowledge about the space and about the position of attractions within that space.

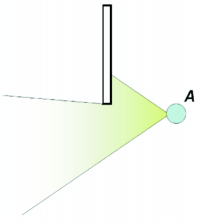



**Fig. 1.** The static fruition space is shaped by the physical obstacles

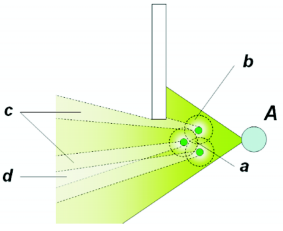



**Fig. 2.** The dynamic fruition space accounts for the presence of agents

At a macro-level, each agent has the ability to build and memorise a rough topological map (RTM) of the entire exposition environment. Practically, RTM is an undirected graph whose nodes are associated to accessible places (like rooms, corridors, halls, etc.) and arcs are associated to gates. The nodes of RTMs can hold information on attractions in different places, as well as other data related to agents' exploration activity (like already visited nodes or already used gates). Such RTMs are at the basis of agents' macro-level planning process: as soon as some information is present in its RTM, the agent is able to elaborate a plan of visit, choosing which rooms and attractions to visit, what general path to follow, and how to roughly allocate the available time among different rooms and attractions. This macro-plans can be provisional and very short-term, since incomplete and inaccurate RTMs get updated and expanded as agents better explore and collect more information about the environment; which in turn can make the macro-plan to change accordingly. On a micro-level, during the visit of a room, an agent creates a list of visible potential targets (LVPT), which is the set of relevant objects visible from the agent's position. Typically, the relevant objects are attractions of interest for the agent, or gates to other rooms and regions of the environment. A LVPT has two main purposes: (*i*) once built, it is used by agents to update and make their RTMs more accurate (e.g. as soon a new gate has been discovered, it is added to the agent's RTM); (*ii*) some or all elements of the LVPT are visited or used by the agent, according to the decision procedure described below. When the agent changes significantly its position, its LVPT is rebuilt using the static visibility map (SVM) [2] which is computed in a pre-processing phase at the beginning of the simulation. With the use of the LVPT, the room-level behaviour of an agent is articulated into two phases. During the first phase, upon the entrance of the agent into the room, the LVPT is built, and the RTM is updated if necessary. Then, the agent selects one target among those present in the LVPT, following a decision procedure based on a gravitation-like model [3], where the elements of the LVPT are assumed to have masses and thus attracting the agent. If the chosen target is an attraction, the agent first moves towards it (using the heuristics described in [4,5], and then it selects the favourable position for the fruition of the attraction, taking into account the presence of other agents.

Otherwise, if the chosen target is a gate, the agent moves towards it and exits the room. During the fruition of the attraction the agent computes a satisfaction

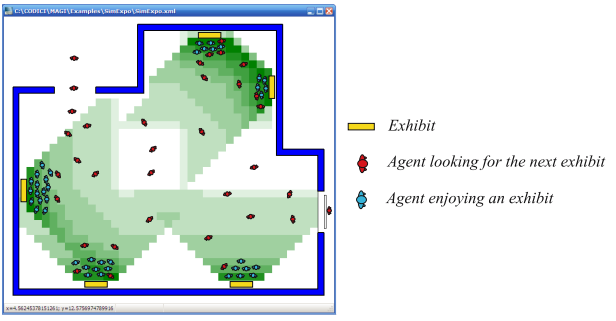



**Fig. 3.** A screenshot of the simulation. The colour gradient refers to the static suitability map.

parameter on the basis of a visibility test which is carried out through exact geometric calculations accounting for the presence of the other agents.

## 3 Model Development

A preliminary version of the model (see Fig. 3) has been developed in C++, using several open source components for spatial data integration, geometric computation and fast spatial queries support. In particular, the software has been designed for efficiently managing both raster and vector representation of simulated entities. Some of the strategies adopted for this purpose was the use a specific geometry engine, implementing a core set of operations on spatial data through robust geometric algorithms [6], together with an efficient spatial indexing strategy for moving objects (i.e. the Leaf-prior Update R-Tree [7]), allowing therefore fast visual-perception algorithms.

## References

1. Bandini, S., Manzoni, S., Vizzari, G.: Multi-agent approach to localization problems:The case of multilayered multi-agent situated system. Web Intelligence and AgentSystems 2, 155–166 (2004)
2. Moulin, B., Chaker, W., Perron, J., Pelletier, P., Hogan, J., Gbei, E.: MAGS project:Multi-agent geosimulation and crowd simulation. In: Kuhn, W., Worboys, M.F., Timpf, S. (eds.) COSIT 2003. LNCS, vol. 2825, pp. 151–168. Springer, Heidelberg (2003)
3. Bandini, S., Manzoni, S., Simone, C.: Enhancing cellular spaces by multilayered multi agent situated systems. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493, pp. 156–167. Springer, Heidelberg (2002)
4. Shao, W., Terzopoulos, D.: Autonomous pedestrians. Graph. Models 69, 246–274 (2007)
5. Reynolds, C.: Steering behaviours for autonomous characters. In: Proceedings of the Game Developers Conference, pp. 763–782 (1999)
6. GEOS: Geometry engine - open source (2007), http://geos.refractions.net/
7. Kwon, D., Lee, S., Lee, S.: Efficient update method for indexing locations of moving objects. J. Inf. Sci. Eng. 21, 643–658 (2005)

# Potential Field Approach of a Cellular Automaton Evacuation Model and Its FPGA Implementation

Ioakeim G. Georgoudas, Georgios Ch. Sirakoulis, and Ioannis Th. Andreadis

Democritus University of Thrace, Department of Electrical and Computer Engineering,
Laboratory of Electronics,
GR 67100 Xanthi, Greece
{igeorg,gsirak,iandread}@ee.duth.gr

**Abstract.** Crowd evacuation from constructions can be modelled with the use of Cellular Automata. The crowd consists of individuals and its behaviour is modelled by the response of each individual to a simple updating rule that directs a person to the nearest exit as facilitates any object avoidance. Individual interactions are defined locally while the crowd motion evolves. In this paper, this motion approach is based on the concept of virtual potential fields that are defined by the location of exits, other pedestrians or objects. Electric charges located at exit and obstacle positions generate such fields. Characteristic features of crowd dynamics, such as incoherent-to-coherent pedestrian motion, blockings in front of exits and mass behaviour are successfully simulated. Finally, the paper presents the main architectural concepts of the corresponding dedicated FPGA processor, as the major structural part of an integrated, interactive, decision-support system.

**Keywords:** Cellular Automata, Evacuation, Potential Fields, Hardware, FPGA.

## 1 Introduction

Safety and security in human social activities are issues of major and constantly increasing concern. Tragedies demonstrated during crowd evacuation processes, involving huge human casualties, prove in the most painful way the importance of consistent development of efficient egress systems able to confront with such phenomena. Research community effectively turns its attention to this issue, presenting and updating sophisticated models that try to approach egress in a more elaborate way [1]. Sociologists and physicists have tried to study and understand crowd behaviour; mathematicians have provided tools while architects and engineers have tried to implement knowledge to more applicable patterns, such as building construction or development of decision support systems.

In this paper, crowd evacuation from buildings is modelled with the use of a potential-field cellular automata (CA) model. The idea of the proposed model can be found in Georgoudas et al. [2, 3] where a CA model is presented capable of simulating earthquake activity in correspondence to the quasi-static two-dimensional version of the Burridge-Knopoff spring-block model of earthquakes, as well as, to the Olami, Feder

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 546–549, 2008.

and Christensen (OFC) earthquake model, applying the concept of self-organised criticality in earthquake process. In the earthquake CA model the system balances through the exercitation of electrostatic Coulomb–forces among charges, without the existence of any other form of interconnection in–between. As a result, in the proposed CA, crowd motion approach is based on the concept of virtual potential fields that are generated by the location of virtual charges at exits and objects locations. Simple and easily hardware applicable updating rules, implemented locally, demonstrate global behavioural patterns that distinctly characterise mass egress. Exit seeking, obstacle avoidance, incoherent-to-coherent pedestrian motion, blockings in front of exits and collective effects have been successfully presented during simulation.

In Section 2, the virtual potential field approach of the motion description and how it is applied to the CA evacuation model is discussed. In Section 3, the basic architectural concepts of the corresponding dedicated FPGA processor are presented. Finally, conclusions are drawn in Section 4.

## 2 Potential Field Approach to Pedestrian Movement

Virtual potential fields generated by different charges that are located at exits and obstacles positions define motion approach. The method was originally proposed by Khatib in 1986 [4] as a real-time collision avoidance model, and later extended as a motion planning for mobile robotics [5]. In the CA based evacuation model each pedestrian responds locally to the field. Pedestrian motion is influenced by an artificial potential field induced by the exit and obstacles charges. Furthermore, walls are modelled by successively located charges that repel pedestrians and prevent pedestrian routes from unintentional sticking. The charges occupying exits and obstacles positions are responsible for the generation of electric fields. Moreover, the charges are stable and time independent. Consequently, the corresponding vector fields of electric intensity $\vec{E}(x, y)$ are time-independent, and the fields induced electrostatics thus conservative as well. Such fields can be described by a scalar function, namely the potential function $f(x, y)$. The gradient of the latter at every point M of the field is equal to function $\vec{E}(x, y)$ of the vector field.

$$\nabla f(M) = \vec{E}(M) \ . \tag{1}$$

Motion is controlled by a law that derives from the gradient descent of the potential function. The gradient points in the direction where the derivative has the largest value (i.e. the greatest rate of increase in the value of $f$). The gradient descent optimisation algorithm searches the opposite direction of the gradient to find the minimum of the function. Potential field methods employ a similar approach.

A characteristic potential field around the exit is mathematically described by the following equations:

$$E(x) = \frac{1}{2}(\vec{x} - \vec{x}_{exit})^T(\vec{x} - \vec{x}_{exit}) \ , \tag{2}$$

$$\nabla E(x) = \vec{x} - \vec{x}_{exit} \ , \tag{3}$$

where $\vec{x}$ is a position vector in the two-dimensional workspace. Vector $\vec{x}_{exit}$ acts as an attractor for the pedestrians, thus defining their route towards the exit (Fig. 1).

Similarly, the obstacle acts as repulsor for the pedestrians when described by a potential field as:

$$E(x) = -\frac{1}{2}(\vec{x} - \vec{x}_{obstacle})^T(\vec{x} - \vec{x}_{obstacle}) \ , \tag{4}$$

since following the negative gradient of $E$ the pedestrian will move away from the obstacle (Fig. 1).

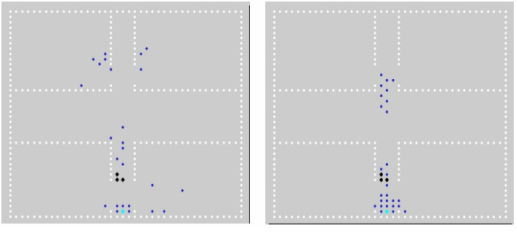


**Fig. 1.** Simulation of evacuation, introducing virtual potential fields around objects and exits

The negative gradient of the superimposed potential field described by Eqs. (2) and (3) describes the motion of each pedestrian moving towards an exit, avoiding an obstacle as well.

## 3 Architecture of the Dedicated FPGA Processor

In this section, the main concepts of the architecture of the hardware implementation of the CA based evacuation model are presented. The motivation is the exploitation of massive parallelism that characterises CA. They can be effectively defined as Turing machines, characteristic examples of finite-state machines that are connected together and operating in parallel. Hence, CA ensure that all operations are performed simultaneously, thus extensively enhancing the speed of the modelled process.

FPGA technology has been applied to design the processor using the Altera Quartus II software multiplatform, a comprehensive environment available for system-on-a-programmable-chip (SOPC) design. Each cell is separated into two parts; the combinational one that includes all computations taking place into cells and the memory part that passes the combinational results to the adjacent cells at successive time steps (Fig. 2). The combinational part comprises of three structures that are connected one right after the other; the closest exit identification circuit, the closest neighbourhood coordinates calculation circuit and the route tracking circuit. At the latter part the exit coordinates are loaded to two multiplexers and the control input defines each time the response. Initial data is provided to the processor by the external cells, which

engage the very left column of the CA grid and it proceeds to the rest of the grid. A peripheral, circuit defines the clock cycles required by this initialisation process to be completed and the application of the updating rule to start.

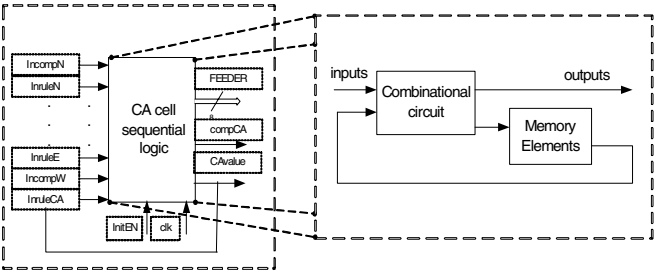



**Fig. 2.** Generic diagram of the CA cell

Secondary circuits have also been designed to operate as one-to-eight or eight-to-one bit converters in order to avoid multiple-pins input driving, as well as to preserve the functionality of the design. Furthermore, the pipelining notion has also been used to enhance the throughput, hence the speed, of the design and registers have been used at cells interconnection to prevent combinational loops from emerging.

## 4 Conclusions

The proposed CA evacuation model realises pedestrians route choice based on artificial potential fields induced by different charges located at exits and obstacles locations. The process is completed in two states; initially, the appropriate potential field is defined and then the pedestrian response to the field is specified. The model, efficiently demonstrates prominent features of evacuation process, such as a phase transition from a disordered to an ordered situation, blockings in front of exits and herding behaviour. Inherent advantages of CA are exploited by the hardware implementation of the corresponding dedicated FPGA processor leading to an integrated, effective, decision-support system that can provide active guidance at mass egress.

## References

1. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407, 487–490 (2000)
2. Georgoudas, I.G., Sirakoulis, G.C., Andreadis, I.: Modelling earthquake activity features using cellular automata. Mathematical and Computer Modelling 46(1-2), 124–137 (2007)
3. Georgoudas, I.G.: Cellular Automata Modelling of Large Scale Systems and VLSI Implementation Perspectives. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 88–93. Springer, Heidelberg (2006)
4. Khatib, O.: Real-time obstacle avoidance for robot manipulator and mobile robots. Internat. J. Robotics Res. 5(1), 90–98 (1986)
5. Arkin, R.C.: Behavior-Based Robotics. MIT Press, Cambridge (1998)

# Evolving Multi-creature Systems for All-to-All Communication

Rolf Hoffmann and Patrick Ediger

Technische Universität Darmstadt
FB Informatik, FG Rechnerarchitektur
Hochschulstr. 10, 64289 Darmstadt, Germany
{hoffmann,ediger}@ra.informatik.tu-darmstadt.de

**Introduction.** Several creatures are moving around in a cellular automata grid. At a certain point of time all creatures want to exchange their information with all others (all-to-all communication). The goal is to find an optimal rule for the movement of the creatures in order to exchange their information as fast as possible. The information exchange is only possible when the creatures meet each other and when they form certain defined local patterns (communication situations). Possible communication situations are exemplarily shown in Fig. 1. In the cases a, b, c the creatures are directly in contact. But it is a matter of definition whether such situations allow communication. For this investigation we have defined the communication patterns d, e, f. A reason could be that communication can only take place if a mediator/negotiator is used between them. Furthermore the mediator may perform a particular computation (e. g., average, maximum, priority select). Such conflicts occur when creatures want to move to the same target position, like vehicles which are meeting in a cross-way. The center of the crossing can be interpreted as the mediator.

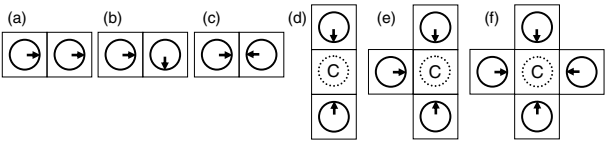



**Fig. 1.** Possible ommunication situations. Communication is only allowed for the situations d, e, f using a mediator C.

In former investigations [1] we have tried to find the best algorithms for the creature's exploration problem, in which the creatures have the task to visit all empty cells in shortest time. The presented problem is related to it in the way how creatures can move. But the task is different: The creatures shall exchange their information in shortest time taking advantage out of the conflicts which are the defined communication situations. – All-to-all communication is a very common task in distributed computing. The problem's specification can depend on many fixed or dynamic varying parameters like the number and location of nodes, the number and location of processes, the number, users and properties

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 550–554, 2008.

of the communication channels and so on. All-to-all communication in multi-creature systems is related to multi-agent problems like finding a consensus, synchronizing oscillators, flocking theory or rendezvous in space [2], or in general to distributed algorithms with robots [3].

**Modeling.** The whole system is modeled by cellular automata (CA). It consists of an environment ($n \times m$ grid) with borders or without borders (wrap-around) and $k$ uniform creatures. A creature has a certain moving direction and it can only read the information from one cell ahead (target cell, front cell). If it detects a border cell or a creature in front or a conflict, it will turn right ($R$) or left ($L$). A conflict occurs when two or more creatures want to move to the same front cell (crossing point, cell in conflict, mediator). The conflict detection is realized by an arbitration logic [4] which is available in each cell. The arbitration logic evaluates the move requests coming from the creatures and replies asynchronously by an acknowledge signal in the same clock cycle. In the case that the creature can move forward it is simultaneously turning right ($Rm$) or to the left ($Lm$). Thus a creature performs the rule:

1. (*Evaluate move condition* $x$):
   If (front cell == OBSTACLE $\vee$ CREATURE $\vee$ CONFLICT) then $x = 0$ else $x = 1$
2. (*React*): If ($x$) then $R/L$ else $Rm/Lm$

The decision which of the actions $R/L$ respectively $Rm/Lm$ will be performed depends on the behavior of the creature. The behavior (algorithm) of a creature is defined by a finite state machine. Input of the state machine is the move condition $x$. Output of the state machine is the signal $y$. The action $R/L$ is performed if $y = 1/0$ and $x = 0$. The action $Rm/Lm$ is performed if $y = 1/0$ and $x = 1$. The actions were defined in this way in order to keep the control automaton as simple as possible – A state machine is defined by a state transition table (Fig. 2) with current input $x$, current state $s$, next state $s'$ and current output $y$. It is represented by concatenating the contents to a string, e. g.: 1R5L3L4R5R3L-1Lm2Rm3Rm4Lm5Rm0Lm or in a simplified form 1R5L3L4R5R3L-1L2R3R4L5R0L.

To solve the problem very general either theoretical or practical with respect to all interesting parameters is too difficult. Therefore we have specialized our investigation. The grid size was set to 35 by 35. This size was taken over from former investigations allowing to distribute equally a varying number of creatures

| x | 0 | | | | | | 1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s | 0 | 1 | 2 | 3 | 4 | 5 | 0 | 1 | 2 | 3 | 4 | 5 |
| s',y | 1,R | 5,L | 3,L | 4,R | 5,R | 3,L | 1,Lm | 2,Rm | 3,Rm | 4,Lm | 5,Rm | 0,Lm |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| j | 0,1 | 2,3 | 4,5 | 6,7 | 8,9 | 10,11 | 12,13 | 14,15 | 16,17 | 18,19 | 20,21 | 22,23 |

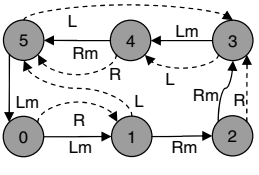



**Fig. 2.** A state table, defining the behavior (algorithm) of a creature and the corresponding state graph

at the borders. The number of creatures was set to $k = 16$. From former investigations in multi-creature systems we know that a number of creatures between approx. 8 and 64 can lead to good synergy effects and a sufficient number of conflicts which are required here. – The ultimate goal is to find the optimal behavior on average for all possible initial configurations. As we cannot test all possible configurations we will be satisfied if we can find the best behaviors for a first test set of 10 randomly generated initial configurations which have to confirm their quality using a second set of 100 random initial configurations. As the search space for different behaviors is very large we are not able to check all possible behaviors. Therefore we used a genetic procedure and tried to find the best behavior within a reasonable computational time limit. – The fitness of a multi-creature system is defined as the number of generations which are necessary to distribute (all-to-all) the information, averaged over all initial configurations (start positions and direction of the creatures) under test. In other words we search for state algorithms which can solve the problem with a minimum number of generations. – In order to model the distribution of information we are using a bit vector with $k$ bits which is stored in each creature. At the beginning the bits are set mutually exclusive (bit($i$)=1 for creature($i$)). When two, three or four creatures form a communication situation they exchange their information by simply OR-ing their bit vectors together. The task is successfully solved when the bit vectors of all creatures obtain $11 \ldots 1$.

**Genetic Procedure.** The behavior is described by a state table (Fig. 2) using 6 control states, 2 inputs ($x = 0/1$) and 2 outputs ($y = 0/1$). The string representation of the state table defines the genome (individual, possible solution). $P$ populations of $N$ individuals are updated in each iteration (optimization cycle). During each cycle $M$ offsprings are produced in each population. The union of the current $N$ individuals and the $M$ offsprings are sorted according to their fitness and the $N$ best are selected to form the next population. An offspring is produced as follows: (1. Get Parents) Two parents are chosen for each population. Each parent is chosen from the own population with a probability of $p_1$ and from an arbitrary other population with the probability of $(1 - p_1)$. (2. Crossover) Each new component $(s'_i, y_i)$ of the genome string is taken from either the first parent or the second parent with a probability of 50%. This means that the tuple (next state, output) for the position $i$=(input, state) is inherited from any parent. (3. Mutation) The string being modified by the crossover is afterwords mutated with a probabiltity of $p_2$. If a mutation shall be performed, an arbitrary position $j$ is chosen and a new value (randomly choosen from the set of valid values) is replacing the existing one. Thereby either the next state or the output is changed at position $i$.

**Results.** The algorithms were optimized using $P = 7$ populations with $N = 100$ individuals each, $M = 10$ offsprings, $p_1 = 98\%$ and $p_2 = 9\%$. 800 iterations were performed starting with randomly generated state algorithms. The fitness was tested for each offspring by simulating the multi-creature systems with 10 initial random configurations. The 700 best behaviors which were produced after

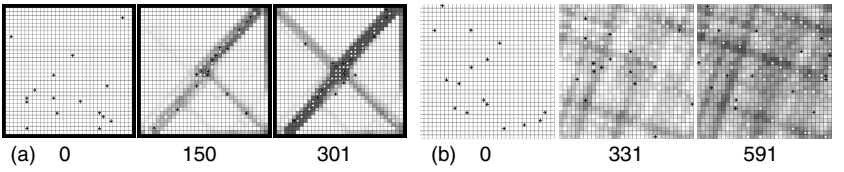



**Fig. 3.** Simulation patterns of the best algorithms $A_b$ (a) respectively $A_w$ (b) (see also Tab. 1) on an arbitrary start configuration. The numbers indicate the generations of the CA when the situations were observed. The rightmost images show the final generation when the algorithms accomplished all-to-all communication.

**Table 1.** The top 3 fastest alg. with and without borders averaged over 100 env

| | | Algorithm | Generations | not visited cells | Communication Situations d | e | f |
|---|---|---|---|---|---|---|---|
| with borders | $A_b$ | 2L5L5R1R3L4L-2R3R3L4R3L1R | 391.76 | 29.43% | 73.16 | 0.49 | 0.02 |
| | $B_b$ | 2L5L5L1R3L4L-2R3R3L4R3L1R | 392.41 | 29.39% | 72.72 | 0.51 | 0.02 |
| | $C_b$ | 2L5L3R1R3L2L-4R3R3L4R3L1R | 394.06 | 29.48% | 71.39 | 0.6 | 0.03 |
| without borders | $A_w$ | 1R5L3L4R5R3L-1L2R3R4L5R0L | 707.09 | 0.017% | 82.68 | 0.35 | 0 |
| | $B_w$ | 4L2R3L1R5R3L-1L2R3R4L5R0L | 722.08 | 0.031% | 88.03 | 0.43 | 0 |
| | $C_w$ | 4R5L3L1R5R0L-1L2R3R4L5R0L | 723.22 | 0.013% | 83.58 | 0.34 | 0 |

800 iterations were used for an additional test set consisting of 100 initial configurations, also randomly generated in advance. – All algorithms which could successfully communicate in all the 100 environments show a similar global behavior: The creatures first move to a corner and from there on a diagonal path back and forth (Fig. 3(a)). Thereby two trails of communication are established, forming a cross. When a creature meets another creature it is slightly deviated, but manages to move back to a diagonal trail. The points of communication mainly lie on these trails and they are marked in Fig. 3 as communication spots in white. – We were not satisfied by this result when we realized that the communication trails were induced by the shape of the borders, in particular by the corners. Therefore we started another experiment: We removed the borders by wrap-around in order to get more interesting results to be independent of the border's shape. By the use of the same genetic procedure we could also evolve a bundle of good algorithms (the three best with borders and with wrap-around are shown in Tab. 1). Now the communication trails form a sort of grid, which is slightly turned to the right or to the left (Fig. 3(b)). Similar patterns have emerged for all the algorithms out of the top 100 we have checked. The best algorithm $A_w$ is shown in Fig. 2. Analyzing the behavior of a single creature showed that it is following a trail which forms a spiral with wrap-around in the torus. – Future work: Further investigations are considered with different actions, a different number of control states, time-shuffled algorithms, non-uniform creatures [5], specialized fixed communication cells, and complete quality tests or formal proves for small environments.

## References

1. Halbach, M., Hoffmann, R.: Optimal behavior of a moving creature in the cellular automata model. In: Malyshkin, V.E. (ed.) PaCT 2005. LNCS, vol. 3606, pp. 129–140. Springer, Heidelberg (2005)
2. Lin, J., Morse, A.S., Anderson, B.D.O.: The Multi-Agent Rendezvous Problem. An Extended Summary. In: Cooperative Control. LNCS, vol. 309, pp. 257–289. Springer, Heidelberg (2005)
3. Principe, G., Santoro, N.: Distributed Algorithms for Autonomous Mobile Robots. In: 4th IFIP Int. Conf. on TCS. IFIP, vol. 209, pp. 47–62. Springer, Heidelberg (2006)
4. Halbach, M., Hoffmann, R., Both, L.: Optimal 6-state algorithms for the behavior of several moving creatures. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 571–581. Springer, Heidelberg (2006)
5. Ediger, P., Hoffmann, R., Halbach, M.: Is a non-uniform system of creatures more efficient than a uniform one? In: NIDISC at IPDPS Proceedings. IEEE, Los Alamitos (2008)

# Counterflow Extension for the F.A.S.T.-Model

Tobias Kretz[1,3], Maike Kaufman[2,3], and Michael Schreckenberg[3]

[1] PTV AG
Stumpfstraße 1 – D-76131 Karlsruhe – Germany
Tobias.Kretz@PTV.De
[2] Robotics Research Group – Dept. of Engineering Science – University of Oxford
Parks Road – OX1 3PJ Oxford – UK
Maike@Robots.Ox.Ac.UK
[3] Physics of Transport and Traffic – University of Duisburg-Essen
Lotharstraße 1 – D-47057 Duisburg – Germany
Schreckenberg@PTT.Uni-DuE.De

**Abstract.** The F.A.S.T. (Floor field and Agent based Simulation Tool) model is a microscopic model of pedestrian dynamics [1], which is discrete in space and time [2,3]. It was developed in a number of more or less consecutive steps from a simple CA model [4,5,6,7,8,9,10]. This contribution is a summary of a study [11] on an extension of the F.A.S.T-model for counterflow situations. The extensions will be explained and it will be shown that the extended F.A.S.T.-model is capable of handling various counterflow situations and to reproduce the well known lane formation effect.

## 1 Introduction

Counterflow situations [12,13,14,15,16,17,18,19,20,21,22] pose a special problem to models of pedestrian dynamic. To avoid deadlocks a certain degree of self-organization is necessary, which typically leads to the formation of lanes, where people in the same lane follow each other and different lanes have opposing movement directions. CA models of pedestrian motion are particularly susceptible to deadlocks, if the main orientation axis of a corridor is aligned to one of the discretization axes. The reason for this is the given lane structure which leads to many head-on collision situation which lack any asymmetry to decide about the side of mutual passing. One modeling ansatz is to let opposing pedestrians exchange their cells with a certain probability [23]. While this method is capable of reproducing some elements of reality, one still might be unhappy, as in reality people just do not walk through each other. A major temptation in modelling pedestrian counterflow using discrete models is to make use of the lanes preset by the CA lattice and align the two opposing groups' movement along one lattice axis. While such simulations can increase the understanding of a model or method, they are merely of academic interest and not sufficient for general use in safety, traffic or city engineering. The other big difficulty is combining counterflow situations with speeds larger than one cell per round. The method presented here aims at being independent of the underlying lattice and it includes speeds larger 1.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 555–558, 2008.

## 2 Extension of the F.A.S.T.-Model by a Comoving Dynamic Potential

The probability $p_{xy}$ for an agent $i$ to select a cell $(x, y)$ as desired cell is multiplied by an additional factor $p^f_{xy}$:

$$\bar{p}_{xy} = p_{xy} \cdot p^f_{xy} \tag{1}$$

$$p^f_{xy} = \exp\left( k_f \sum_{j \in \mathcal{NN}_i}^{N} \frac{\sigma(i,j)}{|\sigma(i,j)|} P^j_{xy} \right) \tag{2}$$

where $k_f$ is the coupling parameter that determines the strength of the effect. $\mathcal{NN}_i$ is the set of nearest neighbors of agent $i$. $N$ is the maximum number of nearest neighbors which is considered. $\mathcal{NN}_i$ only consists of agents within a distance $r_{max}$, which are visible to agent $i$ (i.e. not hidden by walls and in the field of view $\pi/2$ to the left and right of the direction of motion of agent $i$. $\sigma(i,j) = \boldsymbol{v}_i \cdot \boldsymbol{v}_j$ is the scalar product of the velocities $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ of agent $i$ and agent $j$, so the fraction in equation 2 is $+1$, if the agents rather move into the same direction and $-1$, if the rather move into opposite direction. Finally $P^j_{xy}$ is the value of the comoving potential induced by agent $j$ at position $(x, y)$. Assuming motion of agent $j$ at position $(x_0, y_0)$ along the x-axis, $P^j_{xy}$ has the following form (for any other direction of motion, the potential has to be rotated accordingly):

$$P^j_{xy} = 2h \left( \frac{v_j}{v_j^{max}} + \delta \right) \left( 1 - \frac{|y - y_0|}{a} \right), \text{ if } \frac{|y - y_0|}{|x - x_0|} \leq \frac{a}{b} \tag{3}$$

$$P^j_{xy} = 2h \left( \frac{v_j}{v_j^{max}} + \delta \right) \left( 1 - \frac{|x - x_0|}{b} \right), \text{ if } \frac{|y - y_0|}{|x - x_0|} > \frac{a}{b} \tag{4}$$

where $h$ is the strength (height) of the potential. The content of the first brackets models the velocity dependence of the potential, $a$ is the basewidth of the potential orthogonal to the direction of motion, $b$ is the baselength alongside the direction of motion of agent $j$.

The values $N = 12$, $h = 4.0$, $\delta = 0.2$, $a = 2$, $b = r_{max} = 15$, and $k_f = 0.8$ have been established as a reasonable choice of parameters. Interestingly the simulation results worsened not only, if one reduced $N$, but also as it was increased. All simulations were done with $v_{max} = 3$. Due to larger fluctuations the method works less well for speeds $v_{max} \geq 5$.

## 3 Simulation Results

Figure 1 clearly shows, how lanes are formed and by that deadlocks are avoided and figure 2 the effect on the fundamental diagram by variation of $k_f$ and $\mathcal{NN}_i$ is shown.

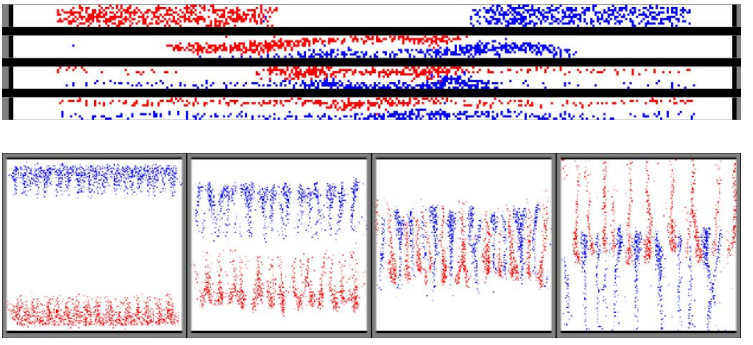


**Fig. 1.** Lane formation in a corridor and on an area

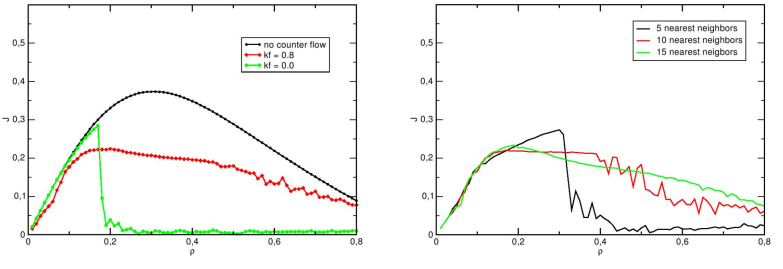



**Fig. 2.** Fundamental diagrams in dependence of $k_f$ and $\mathcal{NN}_i$

## 4 Conclusions and Outlook

This contribution presented a model extension to the F.A.S.T.-model for the simulation of counterflow. It was shown that lane formation was reproduced and deadlocks were avoided for speeds up to 3 cells per round and at densities well above the deadlock density of the earlier model. For even larger speeds the model needs to be stabilized, i.e. fluctuations must be restricted to some maximal value. An interesting empirical question would be to find the number of maximally considered nearest neighbors for real pedestrians.

## References

1. Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., Seyfried, A.: Evacuation Dynamics: Empirical Results, Modeling and Applications. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and System Science, Springer, Heidelberg (to be published, 2009); arXiv:0802.1620v1
2. Kretz, T., Schreckenberg, M.: The F.A.S.T.-Model. In: El Yacoubi., et al. (eds.) [27], pp. 712–715
3. Kretz, T.: Pedestrian Traffic - Simulation and Experiments. PhD thesis, Universität Duisburg-Essen (2007)

4. Burstedde, C., Klauck, K., Schadschneider, A., Zittarz, J.: Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. Phys. A 295, 507 (2001)
5. Kirchner, A., Schadschneider, A.: Cellular Automaton Simulations of Pedestrian Dynamics and Evacuation Processes. In: Fukui, et al. (eds.) [25], pp. 531–536
6. Kirchner, A., Schadschneider, A.: Simulation of Evacuation Processes Using a Bionics-inspired Cellular Automaton Model for Pedestrian Dynamics. Phys. A 312(1-2), 260–276 (2002)
7. Schadschneider, A.: Cellular Automaton Approach to Pedestrian Dynamics - Theory. In: Schreckenberg, Sharma (eds.) [24], pp. 76–85
8. Schadschneider, A.: Bionics-Inspired Cellular Automaton Model for Pedestrian Dynamics. In: Fukui, et al. (eds.) [25], pp. 499–509
9. Kirchner, A., Nishinari, K., Schadschneider, A.: Friction Effects and Clogging in a Cellular Automaton Model for Pedestrian Dynamics. PRE 67(056122) (2003)
10. Nishinari, K., Kirchner, A., Namazi, A., Schadschneider, A.: Extended Floor Field CA Model for Evacuation Dynamics. IEICE Trans. Inf. & Syst. E87-D, 726–732 (2004)
11. Kaufman, M.: Lane Formation in Counterflow Situations of Pedestrian Traffic. Master's thesis. Universität Duisburg-Essen (2007)
12. Muramatsu, M., Irie, T., Nagatani, T.: Jamming transition in pedestrian counterflow. Phys. A 267, 487–498 (1999)
13. Helbing, D., Farkas, I.J., Vicsek, T.: Freezing by Heating in a Driven Mesoscopic System. Phys. Rev. Lett. 84, 1240–1243 (2000)
14. Muramatsu, M., Nagatani, T.: Jamming transition in two-dimensional pedestrian traffic. Phys. A 275, 281–291 (2000)
15. Blue, V.J., Adler, J.L.: Cellular Automata Microsimulation of Bi-Directional Pedestrian Flows. TRR, TRB 1678, 135–141 (2000)
16. Schadschneider, A., Kirchner, A., Nishinari, K.: CA Approach to Collective Phenomena in Pedestrian Dynamics. In: Bandini, et al. (eds.) [26], pp. 239–248
17. Tajima, Y., Takimoto, K., Nagatani, T.: Pattern formation and jamming transition in pedestrian counter flow. Phys. A 313, 709–723 (2002)
18. Isobe, M., Adachi, T., Nagatani, T.: Experiment and simulation of pedestrian counter flow. Phys. A 336, 638–650 (2004)
19. John, A., Schadschneider, A., Chowdhury, D., Nishinari, K.: Collective effects in traffic on bi-directional ant trails. J. Theor. Biol. 231, 279–285 (2004)
20. Nagai, R., Fukamachi, M., Nagatani, T.: Experiment and simulation for counter flow of people going on all fours. Phys. A 358, 516–528 (2005)
21. Kretz, T., Wölki, M., Schreckenberg, M.: Characterizing correlations of flow oscillations at bottlenecks. JSTAT, P02005 (2006)
22. Kretz, T., Grünebohm, A., Kaufman, M., Mazur, F., Schreckenberg, M.: Experimental study of pedestrian counterflow in a corridor. JSTAT, P10001 (2006)
23. Blue, V.J., Adler, J.L.: Cellular Automata Microsimulation For Modeling Bi-Directional Pedestrian Walkways. TRB, 35(293) (2001)
24. Schreckenberg, M., Sharma, S.D. (eds.): Pedestrian and Evacuation Dynamics. Springer, Heidelberg (2002)
25. Fukui, M., Sugiyama, Y., Schreckenberg, M., Wolf, D.E. (eds.): Traffic and Granular Flow 2001. Springer, Heidelberg (2003)
26. Bandini, S., Chopard, B., Tomassini, M. (eds.): ACRI 2002. LNCS, vol. 2493. Springer, Heidelberg (2002)
27. El Yacoubi, S., Chopard, B., Bandini, S. (eds.): ACRI 2006. LNCS, vol. 4173. Springer, Heidelberg (2006)

# Conflicts and Friction in Pedestrian Dynamics

Andreas Schadschneider

Institut für Theoretische Physik, Universität zu Köln, 50937 Köln, Germany
as@thp.uni-koeln.de

**Abstract.** "Conflicts" occur generically in particle-hopping models with parallel dynamics when multiple occupation of sites is forbidden. For cellular automata models of pedestrian dynamics we argue that such conflicts represent an important aspect of the real dynamics. Clogging at bottlenecks is described more realistically if one introduces "friction", i.e. conflicts in which none of the involved agents is allowed to move.

**Keywords:** cellular automata, evacuation simulation, floor field model.

## 1 Introduction

Usually, cellular automata (CA) models are based on discrete time dynamics which is realized in computer simulations through a synchronous (parallel) updating scheme. In models with moving particles (*particle-hopping models*) [1,2], this causes inherent problems if an *exclusion principle* has to be satisfied, i.e. if a site can not be occupied by more than one particle. This leads to *conflicts* when two or more particles try to move to the same destination cell within the same timestep (Fig. 1). Since multiple occupations are not allowed a procedure to resolve these conflicts has to be defined. Obviously this is a complication of the dynamics which has a direct influence on the efficiency of simulations. Therefore it is often tried to avoid conflicts by modifying the dynamics. However, such modifications might cause problems in identifying timescales needed for the calibration of the models.

Here we investigate the problem of conflicts in more detail in the context of the *floor field model* [3,4,5] of pedestrian dynamics which provides a rather realistic description of crowd motion and the related collective effects [6,7,8].

## 2 The Floor Field Model

In CA models of pedestrian dynamics space is discretized into cells which can be occupied by at most one pedestrian which leads to a typical cell size of $40*40$ cm$^2$ [3]. The timestep is identified with the reaction time of a pedestrian. Models that yield realistic crowd behaviour are based on stochastic rules where the motion of particles is determined by transition probabilities to neighbour cells.

The dynamics of the floor field model [3,4,5] takes inspiration from the process of *chemotaxis* used by ants for communication [9]. This translates effects of

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 559–562, 2008.

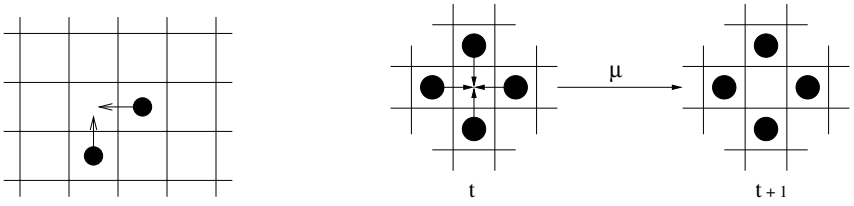



**Fig. 1.** Left: Typical conflict in a particle-hopping model: Two or more particles try to move to the same destination cell within the same timestep. Right: Refused movement due to the friction parameter $\mu$ for a conflict involving four particles.

longer-ranged interactions into purely local ones by introducing a kind of memory. The transition probabilities are determined by three contributions: (i) the preferred walking direction of each pedestrian, (ii) interactions with other pedestrians, and (iii) interactions with the infrastructure (walls, doors etc.). The last two contributions are incorporated via floor fields. These act like virtual chemotaxis by enhancing transitions in the direction of stronger fields.

The *dynamic floor field* $D_{ij} = 0, 1, 2, \ldots$ at site $(i, j)$ represents a virtual trace left by moving pedestrians. Similar to chemotaxis this trace has its own dynamics (diffusion and decay), leading to its broadening and dilution. The *static floor field* $S_{ij} = 0, 1, 2, \ldots$ does not change in time. It reflects the surrounding infrastructure. For evacuation processes, $S_{ij}$ describes the shortest distance to an exit door and increases in the direction of the exit. The relative influence of the two fields is determined by sensitivity parameters $k_s$ and $k_d$. $k_d$ controls the tendency to follow in the footsteps of others, sometimes called *herding*, and $k_s$ determines the effective velocity of a single agent in the direction of its destination.

## 3 Conflicts and Friction

In the original model variant [3] conflicts are resolved by chosing one particle randomly which is allowed to move whereas the others stay at their positions. The details of this choice have only weak influence on the overall dynamics [3].

Conflicts might appear to be undesirable effects that should be avoided by using a different update scheme. However, it turns out that they are important for a correct description of crowd dynamics [5]. Although conflicts are local phenomena they can have a strong influence on global quantities like evacuation times, especially in clogging situations near intersections and bottlenecks. In real life this often leads to dangerous situations and injuries. We will show that the inclusion of conflicts in the model is important for a correct reproduction of the dynamics observed empirically.

For a more realistic description of clogging effects the resolution of conflicts has to be considered in more detail [5]. In real life, conflict situations often lead to a moment of hesitation before a conflict is resolved which reduces the effective velocities of all involved pedestrians. This can be incorporated in the model in a

simple way: With some probability $\mu$ the movement of all involved pedestrians is denied, i.e. all remain at their site (Fig. 1). Thus with probability $1-\mu$ one of the individuals moves to the desired cell. Usually the moving pedestrian is chosen randomly. We call this effect *friction* and $\mu$ *friction parameter* since it has similar consequences as contact friction, e.g. in granular materials.

The use of a parallel update is essential as any sequential update will disguise the number of conflicts in the system. In any model with continuous time these effects have to be implemented in a different way, e.g. through contact friction.

## 4 Evacuation Simulations

The influence of friction effects has been tested in simple evacuation scenarios [4,5]. Generically evacuation times will increase with increasing $\mu$ due to the conflicts near the exit which have a direct influence on the outflow [5,10]. Even at high densities the outflow shows an intermittent behaviour which is well-known from granular flow and is typical for clogging situations [11].

However, friction also leads to counterintuitive effects. Fig. 2 shows the influence of the coupling strength $k_S$ for fixed $\mu$. For $k_S \to 0$ pedestrians perform random walks and evacuation times are almost independent of $\mu$ since conflicts are not very important for the dynamics. In contrast, for $k_S \to \infty$ the shortest way to the exit is chosen. Then evacuation times should decrease with increasing $k_S$ since also the effective velocity in the direction of the door is increased. This is only true for small friction $\mu$, but for $\mu \to 1$ the number of unsolved conflicts increases due to strong jamming at the exit. Then one finds a minimal evacuation time for an intermediate coupling ($k_S \approx 1$) ($\mu = 0.9$ in Fig. 2). This means that a larger $k_S$, which implies a larger average velocity of freely moving pedestrians, leads to larger evacuation times. This collective phenomenon is known as *faster-is-slower effect* [7,12].

Experiments have shown [13] that the motivation level of passengers has a significant influence on the egress time from an aircraft. Egress times were measured

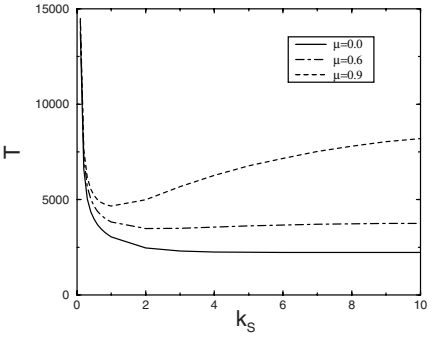



**Fig. 2.** Evacuation time as function of $k_S$ for different values of $\mu$ and $k_D = 0$, $\rho = 0.3$

for various exit widths $w$ in two different situations: (i) competitive, and (ii) cooperative (non-competitive). In the competitive case a bonus was paid for the first passengers to reach the exit. The main result is that $t_{\rm comp} > t_{\rm non-comp}$ for $w < w_c$, whereas $t_{\rm comp} < t_{\rm non-comp}$ for $w > w_c$, where $w_c \approx 70$ cm. Thus competition is beneficial for wide exits, but harmful for narrow ones.

This surprising result is reproduced by the floor field model. Competition is described by increased assertiveness (large $k_S$) and strong hindrance in conflicts (large $\mu$) whereas cooperation is represented by small $k_S$ and $\mu = 0$ [14].

## 5 Conclusions

We have shown that conflicts in particle-hopping models with discrete time update and particles obeying an exclusion principle be considered as an essential part of the dynamics. Their occurance is important for an accurate description of several aspects, e.g. for clogging at bottlenecks. Close to exits, conflicts are most harmful because they have a direct influence on evacuation times.

To further improve the realism of CA models of pedestrian dynamics the concept of "friction" has been introduced, i.e. with probability $\mu$ none of the particles involved in a conflict is allowed to move. This leads to effects like formation of arches near exits and has an important influence on quantities like the evacuation time. It also leads to counterintuitive phenomena like the faster-is-slower effect and an unusual dependence of egress times on the exit width and motivation level.

## References

1. Chowdhury, D., Santen, L., Schadschneider, A.: Phys. Rep. 329, 199 (2000)
2. Chowdhury, D., Nishinari, K., Schadschneider, A.: Physics of Life Reviews 2, 318 (2005)
3. Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J.: Physica A 295, 507 (2001)
4. Kirchner, A., Schadschneider, A.: Physica A 312, 260 (2002)
5. Kirchner, A., Nishinari, K., Schadschneider, A.: Phys. Rev. E 67, 056122 (2003)
6. Helbing, D.: Rev. Mod. Phys. 73(1067) (2001)
7. Helbing, D., Farkas, I., Molnár, P., Vicsek, T.: In: Schreckenberg, M., Sharma, S.D. (eds.) Pedestrian and Evacuation Dynamics. Springer, Heidelberg (2002)
8. Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., Seyfried, A.: In: Meyers, R.A. (ed.) Encyclopedia of Complexity and System Science. Springer, Heidelberg (2009), http://arxiv.org/abs/0802.1620
9. Hölldobler, B., Wilson, E.O.: The Ants, Belknap (1990)
10. Yanagisawa, D., Nishinari, K.: Phys. Rev. E 76, 061117 (2007)
11. Wolf, D.E., Grassberger, P. (eds.): Friction, Arching, Contact Dynamics. World Scientific, Singapore (1996)
12. Helbing, D., Farkas, I., Vicsek, T.: Nature 407, 487 (2000)
13. Muir, H.C., Bottomley, D.M., Marrison, C.: Intern. J. Aviat. Psych. 6, 57 (1996)
14. Kirchner, A., Klüpfel, H., Nishinari, K., Schadschneider, A., Schreckenberg, M.: Physica A 324, 689 (2003)

# Fundamental Diagram and Validation of Crowd Models

Armin Seyfried[1] and Andreas Schadschneider[2]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, 52425 Jülich, Germany
a.seyfried@fz-juelich.de
[2] Institut für Theoretische Physik, Universität zu Köln, 50937 Köln, Germany
as@thp.uni-koeln.de

**Abstract.** In recent years, several approaches for crowd modeling have been proposed. However, so far not much attention has been paid to their *quantitative* validation. The fundamental diagram , i.e. the density-dependence of the flow or velocity, is probably the most important relation as it connects the basic parameter to describe the dynamic of crowds. But specifications in different handbooks as well as experimental measurements for the fundamental diagram differ considerably. We give a review of the experimental data base and the causes for the discrepancies discussed in the literature. Up to now it was neglected that the way of measurement can cause variations between the results of different studies. To shed some light on this problem we studied by means of experimental trajectories of the single file movement how different measurement methods influence the resulting fundamental diagram.

**Keywords:** empirical data, model validation, fundamental diagram.

## 1 Introduction

The number of models for pedestrian dynamics has grown in the past years, but the experimental data to test them and to discriminate between these models is still to a large extent uncertain and contradictory (see e.g. [1]). In most models, pedestrians are considered to be autonomous mobile agents, hopping particles in a cellular automaton or self-driven particles in a continuous space. However, if one wants to make quantitative predictions (e.g. evacuation or travel times) the model has to be calibrated with empirical data, independent from the model type. One of the most important characteristics of pedestrian dynamics is the fundamental diagram giving the relation between pedestrian flow and density. Beside its importance for the dimensioning of pedestrian facilities it is associated with every qualitative self-organization phenomenon, like the formation of lanes or the occurrence of jams. However, specifications of different experimental studies, guidelines and handbooks, display non negligible differences concerning maximal flow values, the assigned density and the density where the flow is expected to become zero due to overcrowding. Although a large variety of models for pedestrian dynamics has been proposed, so far there have been only limited attempts to calibrate and validate these approaches with the fundamental

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 563–566, 2008.

diagram. In part, one reason is the unclear situation of the empirical data, as described above. This situation is very unsatisfactory and poses serious limitations to the use of such models e.g. in the area of safety planning. To improve the current state of affairs it is necessary to have more reliable data that can be used as basis for validation and calibration which then would allow to make even quantitative predictions based on computer simulations.

## 2 Specifications and Measurements

The fundamental diagram describes the empirical relation between density $\rho$ and flow $J$ (or specific flow per unit width $J_s = J/b$). Due to the hydrodynamic relation $J = \rho v b$ there are three equivalent forms: $J_s(\rho)$, $v(\rho)$ and $v(J_s)$. In applications the relation is a basic input for engineering methods developed for the design and dimensioning of pedestrian facilities [2,3,4]. For various facilities like floors, stairs or ramps the shape of the diagrams differ, but in general it is assumed that the fundamental diagrams for the same type of facilities but different widths merge into one diagram for the specific flow $J_s$. In this contribution we will concentrate on planar facilities like sidewalks, corridors or halls. The comparison in Fig. 1 reveals that specifications and measurements disagree considerably. In particular the maximum of the function giving the capacity $J_{s,\max}$ ranges from 1.2 $(\mathrm{ms})^{-1}$ to 1.8 $(\mathrm{ms})^{-1}$, the density value where the maximum flow is reached $\rho_c$ ranges from 1.75 $\mathrm{m}^{-2}$ to 7 $\mathrm{m}^{-2}$ and, most notably, the density $\rho_0$ where the velocity approaches zero due to overcrowding ranges from 3.8 $\mathrm{m}^{-2}$ to 10 $\mathrm{m}^{-2}$. Several explanations for these deviations have been suggested, including cultural and population differences [7], differences between uni- and multidirectional flow [10,8], short-ranged fluctuations [8], influence of

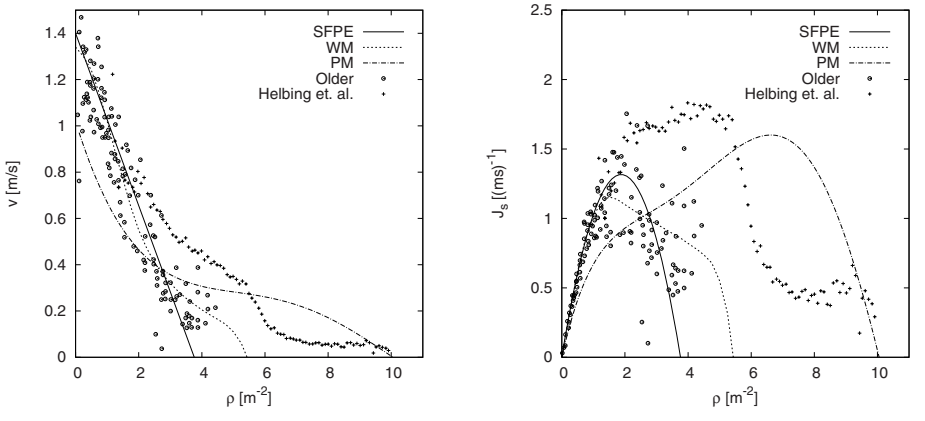



**Fig. 1.** Fundamental diagrams for pedestrian movement in planar facilities. The lines refer to specifications according to planing guidelines (SFPE Handbook [4] [SFPE], Predtechenskii and Milinskii [2] [PM], Weidmann [5] [WM]). Data points give the range of experimental measurements (Older [6] and Helbing *et al.,* [7]).

psychological factors given by the incentive of the movement [2] and, partially related to the latter, the type of traffic (commuters, shoppers) [9].

The most elaborate fundamental diagram has been given by Weidmann who collected 25 data sets. An examination of the data which were included in Weidmann's analysis shows that most measurements with densities larger then $\rho = 1.8$ m$^{-2}$ are performed on multidirectional streams. Weidmann neglected differences between uni- and multidirectional flow in accordance with Fruin, who states in his often cited book [3] that the fundamental diagrams of multidirectional and unidirectional flow differ only slightly. This disagrees with results of Navin and Wheeler [10] who found a reduction of the flow in dependence of directional imbalances. But bidirectional pedestrian flow includes unordered streams as well as lane-separated and thus quasi-unidirectional streams in opposite directions. Another explanation is given by Helbing et al. [7] who argue that cultural and population differences are responsible for the deviations between Weidmann and their data. In contrast to this interpretation the data of Hanking and Wright [11] gained by measurements in the London subway (UK) are in good agreement with the data of Mori and Tsukaguchi [12] measured in the central business district of Osaka (Japan), both on strictly uni-directional streams. This brief discussion clearly shows that up to now there is no consensus about the origin of the discrepancies between different fundamental diagrams and how one can explain the shape of the function.

## 3 Influence of the Measurement Method

Partially the discussion outlined in the previous section loose it's importance due to two reasons. First it is important to notice that in the majority of cases the data come without fluctuations and error margins and thus, strictly speaking, there is no contradiction between the data. Second it is well known in vehicular traffic that different measurement methods can lead to deviations for the resulting relations [13,14]. The deviations depend on the fact that the velocity distributions measured at a certain location and averaged over time do not necessarily conform with velocity distributions measured at a certain point of time averaged over space. For pedestrian traffic it was never analyzed whether there exits or how large the deviations due to different measurement methods are.

Fig. 2 shows a comparison of fundamental diagrams for the single file movement. In our experiment [15] we performed 12 runs with varying number of pedestrians, $N = 17$ to $N = 70$. It is important to note that the different measurements shown in Fig. 2 are basing on the same set of trajectories determined automaticaly from video recordings of the measurement area with high accuracy ($x_{err} \pm 0.02m$). The data analysis is restricted to the stationary state. More details will be given in [16]. Even at this very simple and regular system of pedestrians moving along a line it is astounding how large the deviations are. Thus it is important to consider the measurement method for the validation of models as well as for the comparison of different experimental studies.

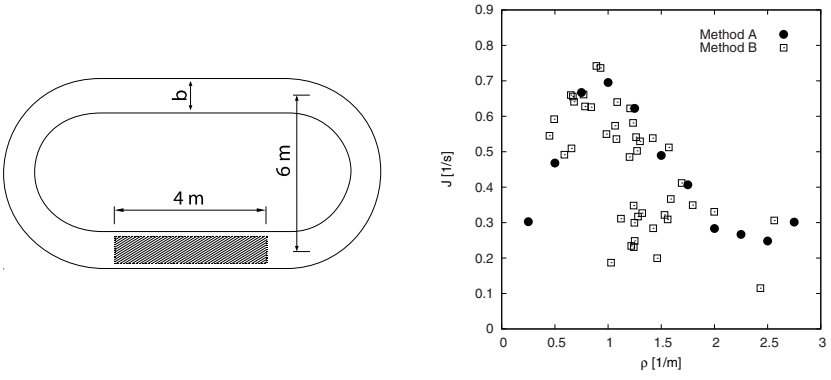



**Fig. 2. Left:** Sketch of the experimental setup to determine the fundamental diagram for the movement of pedestrians along a line ($b = 0.7m$). The measurement area is dashed. **Right:** Fundamental diagram determined by different measurement methods. *Method A:* Measurement of the density $\rho$ and velocity $v$ at a certain point of time $t$ averaging over space $\Delta x$. The flow is given by $J = \rho v$. *Method B:* Measurement of the flow $J = N/\Delta t$ and velocity $v$ at a certain location $x$ averaging over a time interval $\Delta t$. The density is given by $\rho = J/v$.

## References

1. Schadschneider, A. et. al.: Evacuation dynamics: Empirical results, modeling and applications. In Meyers, R.A. (editor), Encyclopedia of Complexity and System Science. Springer (2008)
2. Predtechenskii, V. M., Milinskii, A. I.: Planing for foot traffic flow in buildings. Amerind Publishing, New Dehli (1978)
3. Fruin, J. J: Pedestrian Planning and Design. Metropolitan Association of Urban Designers and Environmental Planners, New York (1971)
4. Nelson H. E., Mowrer, F. W.: Emergency movement. In DiNenno, P. J. (editor), SFPE Handbook of Fire Protection Engineering. Third edition (2002)
5. Weidmann, U.: Transporttechnik der Fussgänger. Schriftenreihe des IVT Nr. 90, ETH Zürich (1993)
6. Older, S. J.: Traffic Engineering and Control 10, 160-163 (1968)
7. Helbing, D. et. al.: Phys. Rev. E, 75:046109 (2007)
8. Pushkarev, B., Zupan J. M.: Transportation Research Record 538, 1–15 (1975)
9. Oeding, D.: Verkehrsbelastung und Dimensionierung von Gehwegen und anderen Anlagen des Fußgängerverkehrs. Forschungsbericht 22, Technische Hochschule Braunschweig (1963)
10. Navin, P. D., Wheeler R. J.: Traffic Engineering 39, 31–36 (1969)
11. Hankin, B. D., Wright R. A.: Operational Research Quarterly 9, 81–88 (1958)
12. Mori, M., Tsukaguchi, H.: Transp. Res. 21A(3), 223–234 (1987)
13. Leutzbach, W.: Introduction to the Theory of Traffic Flow. Springer, Berlin (1988)
14. Kerner, B. S.: The Physics of Traffic. Springer, Berlin (2004)
15. http://www.fz-juelich.de/jsc/math/RD/projects/ped_dynamics/
16. Seyfried, A., et. al.: Conference proceedings PED2008. Springer, Berlin (2009) in preparation

# Multi-agent Frame of Social Distances Model

Jarosław Wąs

Institute of Automatics
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
jarek@agh.edu.pl

**Abstract.** The article focuses on a multi-agent frame of Social Distances model. It introduces characteristics of active and passive pedestrian behavior. It also presents possibility of using Social Distances model for complex modeling of pedestrian traffic. The presented model also includes a concept of familiar groups.

## 1 Introduction

Over the last years, a lot scientists and practitioners are looking for new realistic and effective models of pedestrian dynamics, for instance [1,2,3,5,9,8].

Social Distances model presented in [10] is an attempt at building realistic model which would take into consideration the principles of proxemics. The model applies the sociological theory of *Social Distances* introduced by E. T. Hall [6,7].

The model is based on nonhomogeneous CA and simultaneously it is also a multi-agent system. The principles of movement in this model depend on *an actual state of pedestrian*.

The aim of the article is twofold: to present generalization and extensions of Social Distances model and to highlight Multi-agent frame of the model.

### 1.1 Short Description of Social Distances Model

A pedestrian in the model is represented by an ellipse, whose center coincides with the center of the cell occupied by that person. A movement in the model is realized in Moore neighborhood of radius 1 per one time-step.

A person occupying the cell can take one out of four allowed positions: $H$, $R$, $V$ and $L$ which correspond to the turning of turning the ellipsis around by: $\pm 0$, $\pm 45$, $\pm 90$ and $\pm 135$ degrees respectively. The crucial issue is to establish the set of forbidden and allowed positions for all cells in Moore neighborhood of radius 1, each cell being occupied by one person [10].

Social areas are also represented as ellipses, but the eccentricities of both ellipses (pedestrian and social) area can differ, because "social configuration" in front of the person has much more influence on them behavior than the configuration behind them. Thus, geometrical centers of both the ellipses are

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 567–570, 2008.

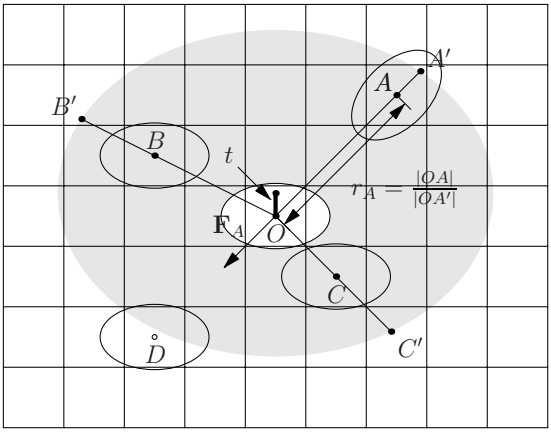



**Fig. 1.** Pedestrian $O$–observer, social area – grey ellipse and other pedestrians – "intruders"

different. The ellipse representing the social area is shifted forward along line of vision of the considered pedestrian by some distance $t$ (see Fig. 1) [10].

The method of calculating the distance between the "observer" $O$ and "intruders": $A$, $B$, $C$ and $D$ is presented on fig 1. If the intruder enters the social area of the observer (on Fig. 1 only $A$, $B$ and $C$) the normalized distance $r$ within the social area is calculated as a ratio of the distance between the centers of persons (e.g. $|OA|$) to the distance between the observer and the point of projection of the intruder's center on the boundary of the social area (respectively $|OA'|$). The normalized distance belongs to the interval $[0, 1]$.

The interaction between the observer and a single intruder is described by "social distance force" $\mathbf{F}_s$. The absolute value of $\mathbf{F}_s$ depends only on the normalized distance between them, $\mathbf{F}_s = F_i(r)$ where $F_i$ is one of some assumed models for social distance force. $\mathbf{F}_s$ has reverse sense than the vector observer-intruder. The total social force affecting the observer is calculated simply as a vector sum of social forces calculated for each intruder (in the presented case: $\mathbf{F}_s = \mathbf{F}_A + \mathbf{F}_B + \mathbf{F}_C$) [10].

## 2 Adaptation of Social Distances for Complex Modeling of Pedestrians Traffic

Social Distances model was dedicated to pedestrian dynamics within a certain limited area (for instance passenger behavior in a tram). The model could be adapted for pedestrian traffic modeling in streets, shopping centers, restaurants and other facilities.

There are three possible states of each pedestrian in the model: *Go to*, *Wait in intermediate aim* and *Wait*. The only state when social distances have a direct influence on pedestrians is the *Wait* state. Every pedestrian in this state is under the influence of all other pedestrians.

When a pedestrian heads to an aim, we deal with the *active* state. In this case, the most important thing for the pedestrian is to *reach the aim*. This is confirmed by a simple observation: when we head to an exit on a crowded tram, we generally do not have a problem of violating our intimate area *for a moment*, because we would like to reach the aim (simply to leave a vehicle). An important issue in the model is choosing the aim (door, seat, table in restaurant) from a set of available aims using rules of proxemics.

*Wait in intermediate aim* is also classified as an active state. Why? Because we perform a specific activity (validating a ticket, eating a meal in a restaurant, sitting on a seat). Thus, in the model, the influence of social distances is not possible in this state, rules of proxemics are anticipated during process of choosing the aim.

*Wait* state is the *passive* state. It is observed when we stand on a tram, go up or go do down in a lift. The situations of talking with somebody in a street is also classified as passive. It is confirmed by an observation, that during longer period of "inactivity" we tend to situate (allocate) ourselves in the possibly most comfortable way, because we are under influence of social distances.

### 2.1 Rules of Movement

Rules of movement in the model depend on an actual state of pedestrian. In active state it is based on cellular automata (concept of intermediate aims and potential fields [10]).

In passive state a particular pedestrian could be under influence of others. It is a influence of a force (Social Distance force), but movement in this case is still realized in cellular automata lattice.

## 3 Familiarities in Social Distances Model

The previous version of the model took into consideration only repulsive forces. But according to the theory of proxemics, there is a need of introducing attractive forces for modeling "familiar" behaviors for instance for couples, parents and children or group of friends.

In the initial part of the simulation there is a need of point a leader of a familiar group. Because it is a frame of asynchronous CA. In the process of implementation the leader of a group is always placed in the list before other members. Thus, the leader makes movement as the first in the group. The members follow her/him maintaining a possibly short distance (including allowed configurations![10]). Group behavior is shown in each state: jointly reaching and profiting aims (neighbor seats, tables etc.) and jointly waiting in *Wait* state. If the group was divided during *Wait* state, attractive forces operate in the next time step in order to connect members.

## 4 Concluding Remarks

The distinction of two main states in pedestrian dynamics: *active* and *passive* make it possible to apply the rules of proxemics. During *active* state pedestrians

anticipate reaching resources (seats) or they use these resources. In *passive* state pedestrians are under influence of Social Distances.

In the previous version of the model, only repulsive forces were applied. The current version brings a definition of familiar groups including a concept "follow a leader" with attractive forces.

A multi-agent frame of pedestrian dynamics simulation makes it more realistic and easier for interpretation.

## References

1. Bandini, S., Manzoni, S., Vizzari, G.: Situated Cellular Agents a Model to Simulate Crowding Dynamics. Special Issues on Cellular Automata E87-D, 669–676 (2004)
2. Burstedde, C., Klauck, K.: Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. Physica A 235, 507–525
3. Dijkstra, J., Jessurun, A.J., Timmermans, H.: A Multi-Agent Cellular Automata System for Visualising Simulated Pedestrian Activity. In: Proceedings of ACRI, pp. 29–36 (2000)
4. Dudek–Dyduch, E., Wąs, J.: Knowledge Representation of Pedestrian Dynamics in Crowd. In: Formalism of Cellular Automata. Proceedings of ICAISC. LNCS (LNAI). Springer, Heidelberg (2006)
5. Gloor, C., Stucki, P., Nagel, K.: Hybrid Techniques for Pedestrian Simulations. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 581–590. Springer, Heidelberg (2004)
6. Hall, E.T.: The Silent Language. Garden City, New York (1959)
7. Hall, E.T.: The Hidden Dimension. Garden City, New York (1966)
8. Narimatsu, K., Shiraishi, T., Morishita, S.: Acquisiting of Local Neighbour Rules in the Simulation of Pedestrian Flow by Cellular Automata. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 211–219. Springer, Heidelberg (2004)
9. Schadshneider, A., Kirchner, A., Nishinari, K.: CA Aproach to Collective Phenomena in Pedestrian Dynamics. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493, pp. 239–248. Springer, Heidelberg (2002)
10. Wąs, J., Gudowski, B., Matuszyk, P.J.: Social Distances Model of Pedestrian Dynamics. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 492–501. Springer, Heidelberg (2006)
11. Wąs, J.: Cellular Automata Model of Pedestrian Dynamics for Normal and Evacuation Conditions. In: Proceedings of Intelligent Systems Design and Applications, Wroclaw, pp. 154–159. IEEE CS, Washington (2005)
12. Wąs, J.: Experiments on evacuation dynamics for different classes of situations. In: International Conference Pedestrian and Evacuation Dynamics – PED 2008, Wuppertal (in press, 2008)

# Multi-agent Frame of Social Distances Model

Jarosław Wąs

Institute of Automatics
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
jarek@agh.edu.pl

**Abstract.** The article focuses on a multi-agent frame of Social Distances model. It introduces characteristics of active and passive pedestrian behavior. It also presents possibility of using Social Distances model for complex modeling of pedestrian traffic. The presented model also includes a concept of familiar groups.

## 1 Introduction

Over the last years, a lot scientists and practitioners are looking for new realistic and effective models of pedestrian dynamics, for instance [1,2,3,5,9,8].

Social Distances model presented in [10] is an attempt at building realistic model which would take into consideration the principles of proxemics. The model applies the sociological theory of *Social Distances* introduced by E. T. Hall [6,7].

The model is based on nonhomogeneous CA and simultaneously it is also a multi-agent system. The principles of movement in this model depend on *an actual state of pedestrian*.

The aim of the article is twofold: to present generalization and extensions of Social Distances model and to highlight Multi-agent frame of the model.

### 1.1 Short Description of Social Distances Model

A pedestrian in the model is represented by an ellipse, whose center coincides with the center of the cell occupied by that person. A movement in the model is realized in Moore neighborhood of radius 1 per one time-step.

A person occupying the cell can take one out of four allowed positions: $H$, $R$, $V$ and $L$ which correspond to the turning of turning the ellipsis around by: $\pm 0$, $\pm 45$, $\pm 90$ and $\pm 135$ degrees respectively. The crucial issue is to establish the set of forbidden and allowed positions for all cells in Moore neighborhood of radius 1, each cell being occupied by one person [10].

Social areas are also represented as ellipses, but the eccentricities of both ellipses (pedestrian and social) area can differ, because "social configuration" in front of the person has much more influence on them behavior than the configuration behind them. Thus, geometrical centers of both the ellipses are

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 571–574, 2008.

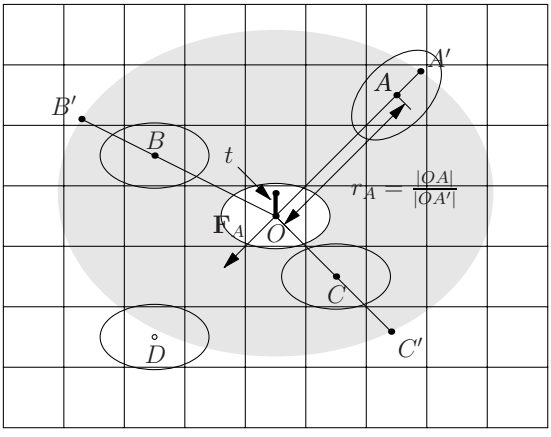



**Fig. 1.** Pedestrian $O$–observer, social area – grey ellipse and other pedestrians – "intruders"

different. The ellipse representing the social area is shifted forward along line of vision of the considered pedestrian by some distance $t$ (see Fig. 1) [10].

The method of calculating the distance between the "observer" $O$ and "intruders": $A$, $B$, $C$ and $D$ is presented on fig 1. If the intruder enters the social area of the observer (on Fig. 1 only $A$, $B$ and $C$) the normalized distance $r$ within the social area is calculated as a ratio of the distance between the centers of persons (e.g. $|OA|$) to the distance between the observer and the point of projection of the intruder's center on the boundary of the social area (respectively $|OA'|$). The normalized distance belongs to the interval $[0, 1]$.

The interaction between the observer and a single intruder is described by "social distance force" $\mathbf{F}_s$. The absolute value of $\mathbf{F}_s$ depends only on the normalized distance between them, $\mathbf{F}_s = F_i(r)$ where $F_i$ is one of some assumed models for social distance force. $\mathbf{F}_s$ has reverse sense than the vector observer-intruder. The total social force affecting the observer is calculated simply as a vector sum of social forces calculated for each intruder (in the presented case: $\mathbf{F}_s = \mathbf{F}_A + \mathbf{F}_B + \mathbf{F}_C$) [10].

## 2 Adaptation of Social Distances for Complex Modeling of Pedestrians Traffic

Social Distances model was dedicated to pedestrian dynamics within a certain limited area (for instance passenger behavior in a tram). The model could be adapted for pedestrian traffic modeling in streets, shopping centers, restaurants and other facilities.

There are three possible states of each pedestrian in the model: *Go to*, *Wait in intermediate aim* and *Wait*. The only state when social distances have a direct influence on pedestrians is the *Wait* state. Every pedestrian in this state is under the influence of all other pedestrians.

When a pedestrian heads to an aim, we deal with the *active* state. In this case, the most important thing for the pedestrian is to *reach the aim*. This is confirmed by a simple observation: when we head to an exit on a crowded tram, we generally do not have a problem of violating our intimate area *for a moment*, because we would like to reach the aim (simply to leave a vehicle). An important issue in the model is choosing the aim (door, seat, table in restaurant) from a set of available aims using rules of proxemics.

*Wait in intermediate aim* is also classified as an active state. Why? Because we perform a specific activity (validating a ticket, eating a meal in a restaurant, sitting on a seat). Thus, in the model, the influence of social distances is not possible in this state, rules of proxemics are anticipated during process of choosing the aim.

*Wait* state is the *passive* state. It is observed when we stand on a tram, go up or go do down in a lift. The situations of talking with somebody in a street is also classified as passive. It is confirmed by an observation, that during longer period of "inactivity" we tend to situate (allocate) ourselves in the possibly most comfortable way, because we are under influence of social distances.

### 2.1 Rules of Movement

Rules of movement in the model depend on an actual state of pedestrian. In active state it is based on cellular automata (concept of intermediate aims and potential fields [10]).

In passive state a particular pedestrian could be under influence of others. It is a influence of a force (Social Distance force), but movement in this case is still realized in cellular automata lattice.

## 3 Familiarities in Social Distances Model

The previous version of the model took into consideration only repulsive forces. But according to the theory of proxemics, there is a need of introducing attractive forces for modeling "familiar" behaviors for instance for couples, parents and children or group of friends.

In the initial part of the simulation there is a need of point a leader of a familiar group. Because it is a frame of asynchronous CA. In the process of implementation the leader of a group is always placed in the list before other members. Thus, the leader makes movement as the first in the group. The members follow her/him maintaining a possibly short distance (including allowed configurations![10]). Group behavior is shown in each state: jointly reaching and profiting aims (neighbor seats, tables etc.) and jointly waiting in *Wait* state. If the group was divided during *Wait* state, attractive forces operate in the next time step in order to connect members.

## 4 Concluding Remarks

The distinction of two main states in pedestrian dynamics: *active* and *passive* make it possible to apply the rules of proxemics. During *active* state pedestrians

anticipate reaching resources (seats) or they use these resources. In *passive* state pedestrians are under influence of Social Distances.

In the previous version of the model, only repulsive forces were applied. The current version brings a definition of familiar groups including a concept "follow a leader" with attractive forces.

A multi-agent frame of pedestrian dynamics simulation makes it more realistic and easier for interpretation.

## References

1. Bandini, S., Manzoni, S., Vizzari, G.: Situated Cellular Agents a Model to Simulate Crowding Dynamics. Special Issues on Cellular Automata E87-D, 669–676 (2004)
2. Burstedde, C., Klauck, K.: Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. Physica A 235, 507–525
3. Dijkstra, J., Jessurun, A.J., Timmermans, H.: A Multi-Agent Cellular Automata System for Visualising Simulated Pedestrian Activity. In: Proceedings of ACRI, pp. 29–36 (2000)
4. Dudek–Dyduch, E., Wąs, J.: Knowledge Representation of Pedestrian Dynamics in Crowd. In: Formalism of Cellular Automata. Proceedings of ICAISC. LNCS (LNAI). Springer, Heidelberg (2006)
5. Gloor, C., Stucki, P., Nagel, K.: Hybrid Techniques for Pedestrian Simulations. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 581–590. Springer, Heidelberg (2004)
6. Hall, E.T.: The Silent Language. Garden City, New York (1959)
7. Hall, E.T.: The Hidden Dimension. Garden City, New York (1966)
8. Narimatsu, K., Shiraishi, T., Morishita, S.: Acquisiting of Local Neighbour Rules in the Simulation of Pedestrian Flow by Cellular Automata. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 211–219. Springer, Heidelberg (2004)
9. Schadshneider, A., Kirchner, A., Nishinari, K.: CA Aproach to Collective Phenomena in Pedestrian Dynamics. In: Bandini, S., Chopard, B., Tomassini, M. (eds.) ACRI 2002. LNCS, vol. 2493, pp. 239–248. Springer, Heidelberg (2002)
10. Wąs, J., Gudowski, B., Matuszyk, P.J.: Social Distances Model of Pedestrian Dynamics. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 492–501. Springer, Heidelberg (2006)
11. Wąs, J.: Cellular Automata Model of Pedestrian Dynamics for Normal and Evacuation Conditions. In: Proceedings of Intelligent Systems Design and Applications, Wroclaw, pp. 154–159. IEEE CS, Washington (2005)
12. Wąs, J.: Experiments on evacuation dynamics for different classes of situations. In: International Conference Pedestrian and Evacuation Dynamics – PED 2008, Wuppertal (in press, 2008)

# Author Index