Stefania Bandini
Bastien Chopard
Marco Tomassini (Eds.)

LNCS 2493

Cellular Automata

5th International Conference on Cellular Automata for Research and Industry, ACRI 2002
Geneva, Switzerland, October 2002
Proceedings

Springer

Lecture Notes in Computer Science 2493

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Stefania Bandini   Bastien Chopard
Marco Tomassini (Eds.)

Cellular Automata

5th International Conference on Cellular Automata
for Research and Industry, ACRI 2002
Geneva, Switzerland, October 9-11, 2002
Proceedings

Springer

Volume Editors

Stefania Bandini
University of Milano-Bicocca, Department of Computer Science, Systems
and Communication, Via Bicocca degli Arcimboldi, 8, 20126 Milan, Italy
E-mail: bandini@disco.unimib.it

Bastien Chopard
University of Geneva, Computer Science Department
24, Rue du General Dufour, 1211 Geneva 4, Switzerland
E-mail: Bastien.Chopard@cui.unige.ch

Marco Tomassini
University of Lausanne, Computer Science Institute
Faculty of Sciences, 1015 Lausanne, Switzerland
E-mail: marco.tomassini@iis.unil.ch

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Cellular automata : proceedings / 5th International Conference on Cellular Automata for Research and Industry, ACRI 2002, Geneva, Switzerland, October 9 - 11, 2002 / Stefania Bandini ... (ed.). - Berlin ; Heidelberg ; New York ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; 2493)
ISBN 3-540-44304-5

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

ISSN 0302-9743
ISBN 3-540-44304-5 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP Berlin, Stefan Sossna e. K.
Printed on acid-free paper SPIN: 10870685 06/3142 5 4 3 2 1 0

Preface

This volume contains the proceedings of the Fifth International Conference on Cellular Automata for Research and Industry (ACRI 2002) that was held in Geneva on October 9–11, 2002. After more modest beginnings in 1994 as a largely Italian conference, over the years ACRI has gradually become firmly established as one of the premier conferences in the field of cellular automata in Europe and beyond.

Although the field of cellular automata is a relatively old and established one, these simple but powerful systems and their newer variations continue to attract the interest of researchers after more than half a century since the seminal work of Ulam and Von Neumann. The ACRI series of conferences has the ambition of being an internationally renowned forum for all those interested in the theory and applications of cellular systems.

The contributions collected in this volume concern cellular automata in various fields such as theory, implementations and applications. In addition, several fields of research (e.g. the multi-agents approach) adopt methodologies that show strict affinities to cellular automata, but without the label “Cellular Automata”. Therefore, one of our intentions was to enlarge the cellular automata community to include new related techniques.

The papers in these proceedings demonstrate the wide and varied applicability of cellular automata. These papers have been selected from among about 50 submitted contributions. Each paper was reviewed by at least two members of the program committee. We are extremely grateful to these reviewers for their willingness to offer their expertise to ensure the decision-making process was as fair as possible. The results of the selection are seen in the high quality of the papers published within this volume, many of which are by internationally recognized researchers.

The published papers range from theoretical contributions to applications of cellular automata in various fields, some classical and some novel, including lattice gases, pattern classification, cryptography and authentication. Less well known models have received attention, such as probabilistic, asynchronous, and multilevel automata. Among the new applications and models, one may mention highway traffic, pedestrian and spatial population dynamics, new environmental applications and collective intelligence.

We would like to express our sincere thanks to the invited speakers Olga Bandman, Serge Galam and Kai Nagel, who gave keynote talks on hot topics in the context of standard cellular automata and beyond.

This conference would have been considerably poorer without the support of many people and organizations who helped in different ways to make this event possible. We already remarked on the important role of the Program Committee. We would also like to thank Illycaffè (Trieste) and the Troisième Cycle Romand d'Informatique (Fribourg) for their generous financial support. We also thank Sissa, the International School for Advanced Studies (Trieste), which hosted a successful previous edition of this conference, for its support in the final production of this volume.

We feel that these proceedings amply demonstrate that cellular automata are a lively research topic that, far from being exhausted, will bear new fruits and new models in the pure and applied sciences.

October 2002

Stefania Bandini,
Bastien Chopard,
Marco Tomassini

Organization

Organizing Committee

Stefania Bandini	(University of Milan–Bicocca, Italy)
Bastien Chopard	(University of Geneva, Switzerland)
Marco Tomassini	(University of Lausanne, Switzerland)
Thomas Worsch	(University of Karlsruhe, Germany)

Program Committee

Paul Albuquerque	(University of Geneva, Switzerland)
Stefania Bandini	(University of Milan–Bicocca, Italy)
Olga Bandman	(Russian Academy of Sciences, Russia)
Mathieu Capcarrère	(University of Lausanne, Switzerland)
Giampiero Cattaneo	(University of Milan–Bicocca, Italy)
Bastien Chopard	(University of Geneva, Switzerland)
Michel Droz	(University of Geneva, Switzerland)
Andreas Deutsch	(Max Planck Institute, Germany)
Samira El Yacoubi	(University of Perpignan, France)
Mario Giacobini	(University of Lausanne, Switzerland)
Salvatore Di Gregorio	(University of Calabria, Italy)
Tamás Legendi	(University of Budapest, Hungary)
Furio Suggi Liverani	(Illycaffè, Italy)
Victor E. Malyshkin	(Russian Academy of Sciences, Russia)
Giancarlo Mauri	(University of Milan–Bicocca, Italy)
Jacques Mazoyer	(École Normale Supérieure de Lyon, France)
Paola Rizzi	(Istituto di Architettura di Venezia, Italy)
Roberto Serra	(Centro Ricerche Ambientali Montecatini, Italy)
Moshe Sipper	(Ben-Gurion University, Israel)
Giandomenico Spezzano	(University of Calabria, Italy)
Gianluca Tempesti	(Swiss Federal Institute of Technology, Switzerland)
Marco Tomassini	(University of Lausanne, Switzerland)
Giuseppe Tratteur	(University of Naples, Italy)
Hiroshi Umeo	(Osaka Electro-Communication University, Japan)
Roland Vollmar	(University of Karlsruhe, Germany)
Thomas Worsch	(University of Karlsruhe, Germany)

Sponsoring Institutions

Illycaffè (Trieste)
Troisième Cycle Romand d'Informatique (Fribourg)
Sissa – International School for Advanced Studies (Trieste)

Table of Contents

Invited Papers

Spontaneous Coalition Forming. Why Some Are Stable? 1
Serge Galam

Simulating Spatial Dynamics by Probabilistic Cellular Automata 10
Olga Bandman

Cellular Automata Models for Transportation Applications 20
Kai Nagel

Contributed Papers

An Evolutionary Approach to the Study of Non-trivial Collective Behavior in Cellular Automata 32
Francisco Jiménez-Morales

Artificially Evolved Asynchronous Cellular Automata for the Density Task 44
Marco Tomassini, Mattias Venzi

Evolving Cellular Automata as Pattern Classifier 56
Niloy Ganguly, Pradipta Maji, Sandip Dhar, Biplab K. Sikdar, P. Pal Chaudhuri

An Efficient Mapping Scheme for Embedding Any One-Dimensional Firing Squad Synchronization Algorithm onto Two-Dimensional Arrays .. 69
Hiroshi Umeo, Masashi Maeda, Norio Fujiwara

Chaotic Subshifts Generated by One Dimensional Elementary CA. The Role of Transitivity. 82
Gianpiero Cattaneo, Alberto Dennunzio

Stochastic Analysis of Cellular Automata and the Voter Model 92
Heinz Mühlenbein, Robin Höns

Universality Class of Probabilistic Cellular Automata 104
Danuta Makowiec, Piotr Gnaciński

Kinetic Approach to Lattice Quantum Mechanics 114
Sauro Succi

Emergence of Self-Replicating Loops in an Interactive, Hardware-Implemented Game-of-Life Environment 123
André Stauffer, Moshe Sipper

Spontaneous Emergence of Robust Cellular Replicators 132
Iker Azpeitia, Jesús Ibáñez

Emergence of Macro Spatial Structures in Dissipative Cellular Automata 144
Andrea Roli, Franco Zambonelli

Enhancing Cellular Spaces by Multilayered Multi Agent Situated Systems 156
Stefania Bandini, Sara Manzoni, Carla Simone

Perturbing the Regular Topology of Cellular Automata: Implications for the Dynamics 168
Roberto Serra, Marco Villani

A Path-Planner for Mobile Robots of Generic Shape with Multilayered Cellular Automata 178
Fabio M. Marchese

Dynamics of Populations in Extended Systems 190
Michel Droz, Andrzej Pękalski

Simulation of Vegetable Populations Dynamics Based on Cellular Automata 202
Stefania Bandini, Giulio Pavesi

A Fish Migration Model 210
Birgitt Schönfisch, Michael Kinder

A Parallel Cellular Ant Colony Algorithm for Clustering and Sorting 220
Paul Albuquerque, Alexandre Dupuis

A Multiparticle Lattice Gas Automata Model for a Crowd 231
Stefan Marconi, Bastien Chopard

CA Approach to Collective Phenomena in Pedestrian Dynamics 239
Andreas Schadschneider, Ansgar Kirchner, Katsuhiro Nishinari

Opinion Formation and Phase Transitions in a Probabilistic Cellular Automaton with Two Absorbing States 249
Franco Bagnoli, Fabio Franci, Raúl Rechtman

Cellular Automata Based Authentication (CAA) 259
Monalisa Mukherjee, Niloy Ganguly, P. Pal Chaudhuri

Cellular Automata Machine for Pattern Recognition 270
Pradipta Maji, Niloy Ganguly, Sourav Saha, Anup K. Roy, P. Pal Chaudhuri

Cellular Automata Model of Drug Therapy for HIV Infection 282
Peter Sloot, Fan Chen, Charles Boucher

Cellular Automata Approaches to Enzymatic Reaction Networks 294
Jörg R. Weimar

Modelling Surface Flows for Macroscopic Phenomena by Cellular Automata: An Application to Debris Flows 304
Donato D'Ambrosio, Salvatore Di Gregorio, Giulio Iovine, Valeria Lupiano, Rocco Rongo, William Spataro

Simulation Framework for the Autobahn Traffic in North Rhine-Westphalia 315
Sigurður F. Marinósson, Roland Chrobok, Andreas Pottmeier, Joachim Wahle, Michael Schreckenberg

Cellular Automata Based Temporal Process Understanding of Urban Growth 325
Jianquan Cheng, Ian Masser

Playing with Automata. An Innovative Perspective for Gaming Simulation 337
Ivan Blecic, Arnaldo Cecchini, Paola Rizzi, Giuseppe A. Trunfio

Urban Cellular Automata: The Inverse Problem 349
Giovanni A. Rabino, Alessandra Laghi

Regional Controllability with Cellular Automata Models 357
Samira El Yacoubi, Abdelhaq El Jai, Nezha Ammor

Author Index 369

Spontaneous Coalition Forming. Why Some Are Stable?

Serge Galam

Laboratoire des Milieux Désordonnés et Hétérogènes,
Université Pierre et Marie Curie,
Tour 13 - Case 86, 4 place Jussieu, 75252 Paris Cedex 05, France
Laboratoire associé au CNRS (UMR n° 800)
galam@ccr.jussieu.fr

Abstract. A model to describe the spontaneous formation of military and economic coalitions among a group of countries is proposed using spin glass theory. Between each couple of countries, there exists a bond exchange coupling which is either zero, cooperative or conflicting. It depends on their common history, specific nature, and cannot be varied. Then, given a frozen random bond distribution, coalitions are found to spontaneously form. However they are also unstable making the system very disordered. Countries shift coalitions all the time. Only the setting of macro extra national coalition are shown to stabilize alliances among countries. The model gives new light on the recent instabilities produced in Eastern Europe by the Warsow pact dissolution at odd to the previous communist stability. Current European stability is also discussed with respect to the European Union construction.

1 Introduction

Twenty years ago, using physics to describe political or social behavior was a very odd approach. Among very scarce attempts, one paper was calling on to the creation of a new field under the name of “Sociophysics” [1]. It stayed without real continuation. Only in the last years did physicists start to get involved along this line of research [2]. Among various subjects [3,4], we can cite voting process [5,6], group decision making [7], competing opinion spreading [8,9,10], and very recently international terrorism [11].

In this paper we adress the question of spontaneous coalition forming within military alliances among a set of independant countries [12,13,14]. A model is built from the complexe physics of spin glasses [15]. While coalitions are found to form spontaneously, they are unstable. It is only the construction of extra-territory macro organizations which are able to produce stable alliances.

The following of the paper is organised as follows. The second part contains the presentation of the model. Basic features of the dynamics of spontaneous froming bimodal coalitions are outlined. The building of extra-territory coaltions is described in Section 3. The cold war situation is then analysed in Section 4. Section 5 is devoted to the situation in which only one world coalition is active. A

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 1–9, 2002.

new explanation is given in Section 6 to Eastern European instabilities following the Warsaw pact dissolution as well as to Western European stability. Some hints are also obtained on how to stabilize these Eastern Europe instabilities. Last Section contains some concluding remarks.

2 Presentation of the Model

We start from a group of N independant countries [12]. From historical, cultural and economic experience, bilateral propensities $J_{i,j}$ have emerged between pairs of countries i and j. They are either favoring cooperation ($J_{i,j} > 0$), conflict ($J_{i,j} < 0$) or ignorance ($J_{i,j} = 0$). Each propensity $J_{i,j}$ depends solely on the pair (i, j) itself. Propensities $J_{i,j}$ are local and independant frozen bonds. Respective intensities may vary for each pair of countries but are always symmetric, i.e., $J_{ij} = J_{ji}$.

From the well known saying “the enemy of an enemy is a friend” we get the existence of only two competing coalitions. They are denoted respectively by A and B. Then each country has the choice to be in either one of two coalitions. A variable η_i where index i runs from 1 to N, signals the i actual belonging with $\eta_i = +1$ for alliance A and $\eta_i = -1$ for alliance B. From bimodal symmetry all A-members can turn to coalition B with a simultaneous flip of all B-members to coalition A.

Given a pair of countries (i, j) their respective alignment is readily expressed through the product $\eta_i \eta_j$. The product is $+1$ when i and j belong to the same coalition and -1 otherwise. The associated “cost” between the countries is measured by the quantity $J_{ij}\eta_i\eta_j$ where J_{ij} accounts for the amplitude of exchange which results from their respective geopolitical history and localization.

Here factorisation over i and j is not possible since we are dealing with competing bonds [15]. It makes teh problem very hard to solve analytically. Given a configuration X of countries distributed among coaltions A and B, for each nation i we can measure its overall degree of conflict and cooperation with all others $N - 1$ countries via the quantity,

$$E_i = \sum_{j=1}^{N} J_{ij}\eta_j \, , \tag{1}$$

where the summation is taken over all other countries including i itself with $J_{ii} \equiv 0$. The product $\eta_i E_i$ then evaluates the “cost” associated with country i choice with respect to all other country choices. Summing up all country individual “cost” yields,

$$E(X) = \frac{1}{2}\sum_{i=1}^{N} \eta_i E_i \, , \tag{2}$$

where the 1/2 accounts for the double counting of pairs. This “cost” measures indeed the level of global satisfaction from the whole country set. It can be recast as,

$$E(X) = \frac{1}{2} \sum_{<i,j>} J_{ij} \eta_i \eta_j \, , \tag{3}$$

where the sum runs over the $N(N-1)$ pairs (i,j). At this stage it sounds reasonable to assume each country chooses its coalition in order to minimize its indivual cost. Accordingly to make two cooperating countries ($J_{i,j} > 0$) in the same alliance, we put a minus sign in from of the expression of Eq. (3) to get,

$$H = -\frac{1}{2} \sum_{<i,j>} J_{ij} \eta_i \eta_j \, , \tag{4}$$

which is indeed the Hamiltonian of an Ising random bond magnetic system. There exist by symmetry $2^N/2$ distinct sets of alliances each country having 2 choices for coalition. Starting from any initial configuration, the dynamics of the system is implemented by single country coalition flips. An country turns to the competing coalition only if the flip decreases its local cost. The system has reached its stable state once no more flip occurs. Given $\{J_{ij}\}$, the $\{\eta_i\}$ are thus obtained minimizing Eq. (4).

Since the system stable configuration minimizes the energy, we are from the physical viewpoint, at the temperature $T = 0$. In practise for any finite system the theory can tell which coalitions are possible. However, if several coalitions have the same energy, the system is unstable and flips continuously from one coalition set to another one at random and with no end.

For instance, in the case of three conflicting nations like Israel, Syria and Iraq, any possible alliance configuration leaves always someone unsatisfied. Let us label them respectively by 1, 2, 3 and consider equal and negative exchange interactions $J_{12} = J_{13} = J_{23} = -J$ with $J > 0$ as shown in Fig. (1). The associated minimum of the energy is equal to $-J$. However this minimum value is realized for several possible and equivalent coalitions which are respectively (A, B, A), (B, A, A), (A, A, B), (B, A, B), (A, B, B), and (B, B, A). First three are identical to last ones by symmetry since here what matters is which countries are together within the same coalition. This peculiar property of a degenerate ground state makes the system unstable. There exists no one single stable configuration which is stable. Some dynamics is shown in Fig. (1). The system jumps continuously and at random between (A, B, A), (B, A, A) and (A, A, B).

To make the dynamics more explicit, consider a given site i. Interactions with all others sites can be represented by a field,

$$h_i = \sum_{j=1}^{N} J_{ij} \eta_j \tag{5}$$

resulting in an energy contribution

$$E_i = -\eta_i h_i \, , \tag{6}$$

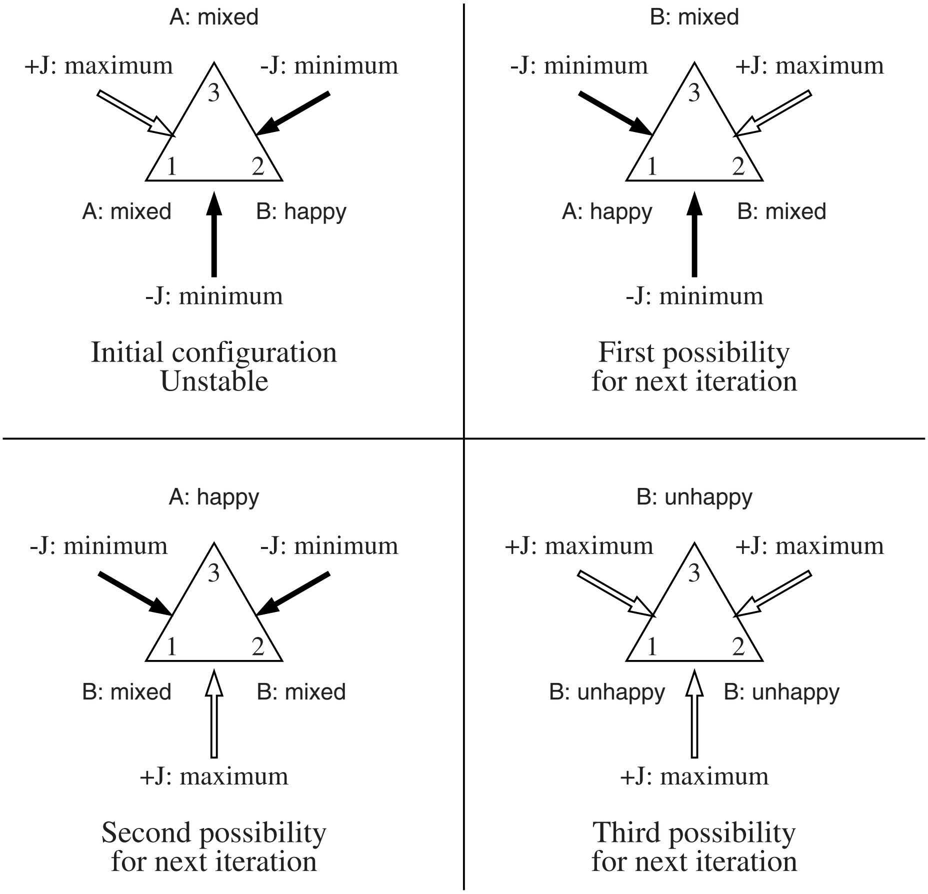


Fig. 1. Top left shows one possible configuration of alliances with countries 1 and 2 in A and country 3 in B. From it, countries 1 and 2 being in a mixed situation with respect to optimzing their respective bilateral interations, three possible and equiprobable distributions are possible. In the first possible following configuration (top right), country 1 has shifted alliance from A to B. However its move keeps it in its mixed situation while making country 2 happy and country 3 mixed. Instead it could have been country 2 which had shifted alliance (low right) making 1 happy and 3 mixed. Last possibility (low left) is both 1 and 2 shifting simultaneously. It is the worse since each country is unhappy.

to the Hamiltonian $H = \frac{1}{2}\sum_{i=1}^{N} E_i$. Eq. (6) is minimum for η_i and h_i having the same sign. For a given h_i there exists always a well defined coalition choice except for $h_i = 0$. In this case site i is unstable. Then both coalitions are identical with respect to its local energy which stays equal to zero. An unstable site flips continuously with probability $\frac{1}{2}$ (see Fig. (1)).

3 Setting up Extra Territory Coalitions

In parallel to the spontaneous emergence of unstable coalitions, some extra territory organizations have been set in the past to create alliances at a global world level. Among the recent more powerfull ones stand Nato and the former Warsow pact. These alliances were set above the country level and produce economic and military exchanges. Each country is then adjusting to its best interest with respect to these organisations. A variable ϵ_i accounts for each country i natural belonging. For coalition A it is $\epsilon_i = +1$ and $\epsilon_i = -1$ for B. The value $\epsilon_i = 0$ marks no apriori. These natural belongings are also induced by cultural and political history.

Exchanges generated by these coalitions produce additional pairwise propensities with amplitudes $\{C_{i,j}\}$. Sharing resources, informations, weapons is basically profitable when both countries are in the same alliance. However, being in opposite coalitions produces an equivalent loss. Therefore a pair (i, j) propensity is $\epsilon_i \epsilon_j C_{i,j}$ which can be positive, negative or zero to mark respective cooperation, conflict or ignorance. It is a site induced bond [15]. Adding it to the former bond propensity yields an overall pair propensity,

$$J_{i,j} + \epsilon_i \epsilon_j C_{i,j} \; , \tag{7}$$

between two countries i and j.

At this stage an additional variable $\beta_i = \pm 1$ is introduced to account for benefit from economic and military pressure attached to a given alignment. It is still $\beta_i = +1$ for A and $\beta_i = -1$ for B with $\beta_i = 0$ in case of no pressure. The amplitude of this economical and military interest is measured by a local positive field b_i which also accounts for the country size and its importance. At this stage, the sets $\{\epsilon_i\}$ and $\{\beta_i\}$ are independent.

Actual country choices to cooperate or to conflict result from the given set of above quantites. The associated total cost becomes,

$$H = -\frac{1}{2} \sum_{<i,j>} \{J_{i,j} + \epsilon_i \epsilon_j C_{ij}\} \eta_i \eta_j - \sum_{i=1}^{N} \beta_i b_i \eta_i \, . \tag{8}$$

An illustration is given in Fig. (2) with above exemple of Israel, Syria and Iraq labeled respectively by 1, 2, 3 with $J_{12} = J_{13} = J_{23} = -J$ and $b_1 = b_2 = b_3 = 0$. Suppose an arab coalition is set against Israel with $\epsilon_1 = +1$ and $\epsilon_2 = \epsilon_3 = -1$. The new propensities become respectively $-J + C$, $-J - C$, $-J - C$. They are now minimized by $\eta_1 = -\eta_2 = \eta_3$ to gives an energy of $-J - C$ to all three couplings.

4 Cold War Scenario

The cold war scenario means that the two existing world level coalitions generate much stonger couplings than purely bilateral ones, i.e., $|J_{i,j}| < C_{i,j}$ since to

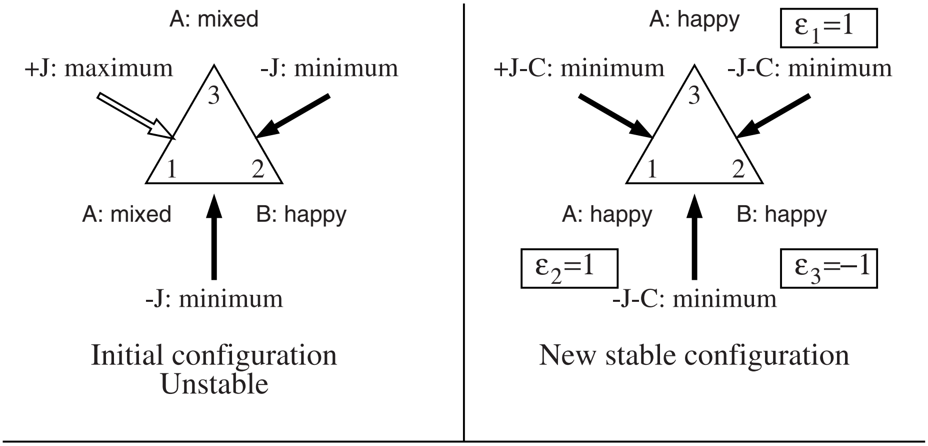


Fig. 2. Starting from one possible unstable configuration of alliances (left) with countries 1 and 2 in A and country 3 in B, the stabilization is shown to result from the existence of the various ϵ with $\epsilon_1 = \epsilon_2 = -\epsilon_3$.

belong to a world level coalition produces more advantages than purely local unproper relationship. Local bond propensities are neutralized since overwhelmed by the two block site exchanges. The overall system is very stable. There exists one stable distribution between both competing alliances.

We consider first the coherent case in which cultural and economical trends go along the same coalition, i.e., $\beta_i = \epsilon_i$. Then from Eq. (8) the minimum of H is unique with all country propensities satisfied. Each country chooses its coalition according to its natural belonging, i.e., $\eta_i = \epsilon_i$. This result is readily proven via the variable change $\tau \equiv \epsilon_i \eta_i$ which turns the energy to,

$$H_1 = -\frac{1}{2} \sum_{<i,j>} C_{ij} \tau_i \tau_j - \sum_{i=1}^{N} b_i \tau_i \, . \tag{9}$$

Above Hamiltonian representd a ferromagnetic Ising Hamiltonian in positive symmetry breaking fields b_i. Indeed it has one unique minimum with all $\tau_i = +1$.

The remarkable result here is that the existence of two apriori world level coalitions is identical to the case of a unique coalition with every country in it. It shed light on the stability of the Cold War situation where each country satisfies its proper relationship. Differences and conflicts appear to be part of an overall cooperation within this scenario.

The dynamics for one unique coalition including every country, or two competing alliances, is the same since what matters is the existence of a well defined stable configuration. However there exists a difference which is not relevant at this stage of the model since we assumed $J_{i,j} = 0$. In reality $J_{i,j} \neq 0$ makes the existence of two coalitions to produce a lower "energy" than a unique coalition since then, more $J_{i,j}$ can also be satisfied.

It worth to notice that field terms $b_i \epsilon_i \eta_i$ account for the difference in energy cost in breaking a pair proper relationship for respectively a large and a small country. Consider for instance two countries i and j with $b_i = 2b_j = 2b_0$. Associated pair energy is

$$H_{ij} \equiv -C_{ij} \epsilon_i \eta_i \epsilon_j \eta_j - 2b_0 \epsilon_i \eta_i - b_0 \epsilon_j \eta_j \,. \tag{10}$$

Conditions $\eta_i = \epsilon_i$ and $\eta_j = \epsilon_j$ give the minimum energy,

$$H_{ij}^m = -J_{ij} - 2b_0 - b_0 \,. \tag{11}$$

From Eq. (11) it is easily seen that in case j breaks proper alignment shifting to $\eta_j = -\epsilon_j$ the cost in energy is $2J_{ij} + 2b_0$. In parallel when i shifts to $\eta_i = -\epsilon_i$ the cost is higher with $2J_{ij} + 4b_0$. Therfore the cost in energy is lower for a breaking from proper alignment by the small country ($b_j = b_0$) than by the large country ($b_j = 2b_0$). In the real world, it is clearly not the same for instance for the US to be against Argentina than to Argentina to be against the US.

We now consider the uncoherent case in which cultural and economical trends may go along opposite coalitions, i.e., $\beta_i \neq \epsilon_i$. Using above variable change $\tau \equiv \epsilon_i \eta_i$, the Hamiltonian becomes,

$$H_2 = -\frac{1}{2} \sum_{<i,j>} J_{ij} \tau_i \tau_j - \sum_{i=1}^{N} \delta_i b_i \tau_i \,, \tag{12}$$

where $\delta_i \equiv \beta_i \epsilon_i$ is given and equal to ±1. H_2 is formally identical to the ferromagnetic Ising Hamiltonian in random fields $\pm b_i$.

The local field term $\delta_i b_i \tau_i$ modifies the country field h_i in Eq. (9) to $h_i + \delta_i b_i$ which now can happen to be zero. This change is qualitative since now there exists the possibility to have "unstability", i.e., zero local effective field coupled to the individual choice. Moreover countries which have opposite cultural and economical trends may now follow their economical interest against their cultural interest or vice versa. Two qualitatively different situations may occur.

- Unbalanced economical power: in this case we have $\sum_i^N \delta_i b_i \neq 0$.
  The symmetry is now broken in favor of one of the coalition. But still there exists only one minimum.
- Balanced economical power: in this case we have $\sum_i^N \delta_i b_i = 0$.
  Symmetry is preserved and H_2 is identical to the ferromagnetic Ising Hamiltonian in random fields which has one unique minimum.

5 Unique World Leader

Very recently the Eastern block has disappeared. However it the Western block is still active as before. In this model, within our notations, denoting A the Western alignment, we have still $\epsilon_i = +1$ for countries which had $\epsilon_i = +1$. On the opposite, countries which had $\epsilon_i = -1$ have now turned to either $\epsilon_i = +1$ if joining Nato or to $\epsilon_i = 0$ otherwise.

Therefore above $J_{i,j} = 0$ assumption based on the inequality $|J_{i,j}| < |\epsilon_i \epsilon_j| C_{i,j}$ no longer holds for each pair of countries. In particular propensity $p_{i,j}$ become equal to $J_{i,j}$ in all cases where $\epsilon_i = 0$, $\epsilon_j = 0$ and $\epsilon_i = \epsilon_j = 0$.

A new distribution of countries results from the collapse of one block. On the one hand A coalition countries still determine their actual choices between themselves according to $C_{i,j}$. On the other hand former B coalition countries are now determining their choices according to competing links $J_{i,j}$ which did not automatically agree with former $C_{i,j}$.

This subset of countries has turned from a random site spin glasses without frustration into a random bond spin glasses with frustration. The former B coalition subset has jumped from one stable minimum to a highly degenerated unstable landscape with many local minima. This property could be related to the fragmentation process where ethnic minorities and states are shifting rapidly allegiances back and forth while they were formerly part of a stable structure just few years ago.

While the B coalition world organization has disappeared, the A coalition world organization did not change and is still active. The condition $|J_{i,j}| < C_{i,j}$ is still valid for A pair of countries with $\epsilon_i \epsilon_j = +1$. Associated countries thus maintain a stable relationship and avoid a fragmentation process. This result supports a posteriori argument against the dissolution of Nato once Warsaw Pact was disolved. It also favors the viewpoint that former Warsaw Pact countries should now join Nato.

Above situation could also shed some light on the current European debate. It would mean European stability is mainly the result of the existence of European structures with economical reality and not the outcome of a new friendship among former ennemies. These structures produce associated propensities $C_{i,j}$ much stronger than local competing propensities $J_{i,j}$ which are still there. European stability would indeed result from $C_{i,j} > |J_{i,j}|$ and not from all having $J_{i,j} > 0$. An eventual setback in the European construction ($\epsilon_i \epsilon_j C_{i,j} = 0$) would then automatically produce a fragmentation process analogous of what happened in former Yugoslavia with the activation of ancestral bilateral local conflicts.

6 Conclusion

In this paper we have proposed a new way to describe alliance forming phenomena among a set of countries. It was shown that within our model the cold war stabilty was not the result of two opposite alliances but rather the existence of alliances which neutralize the conflicting interactions within allies. It means also that having two alliances or just one is qualitatively the same with respect to stability.

From this viewpoint the strong instabilies which resulted from the Warsow pact dissolution are given a simple explanation. Simultaneously some hints are obtained about possible policies to stabilize world nation relationships. Along this line, the importance of European construction was also underlined. At this stage, our model remains rather basic. However it opens some new road to explore and to forecast international policies.

References

1. S. Galam, Y. Gefen and Y. Shapir, Sociophysics: "A mean behavior model for the process of strike" *Math. J. of Sociology* **9**, 1-13 (1982)
2. S. Moss de Oliveira, P.M.C. de Oliveira, and D. Stauffer, Evolution, Money, War, and Computers - Non-Traditional Applications of Computational Statistical Physics, Teubner, Stuttgart-Leipzig, ISBN 3-519-00279-5 5 (1999)
3. S. Solomon, G. Weisbuch, L. de Arcangelis, N. Jan and D. Stauffer, Social percolation models, *Physica* **A 277** (1-2), 239-247 (2000)
4. F. Schweitzer and J. Holyst, Modelling Collective Opinion Formation by Means of Active Brownian Particles, *Eur. Phys. J.* **B 15**, 723-742 (2000)
5. S. Galam, Social paradoxes of majority rule voting and renormalization group, *J. Stat. Phys***61**, 943-951 (1990)
6. S. Galam, Real space renormalization group and totalitarian paradox of majority rule voting, Physica A 285, 66-76 (2000)
7. S. Galam, Rational group decision making: a random field Ising model at $T = 0$, *Physica* **A 238**, 66-80 (1997)
8. S. Galam, B. Chopard, A. Masselot and M. Droz, Competing Species, *Dynamics Eur. Phys. J.* **B 4** 529-531 (1998)
9. B. Chopard, M. Droz and S Galam, An evolution theory in finite size systems, Eur. Phys. J. **B 16** 575-578 (2000)
10. S. Galam, Minority Opinion Spreading in Random Geometry, *Eur. Phys. J.* **B 25**, Rapid Note, 403-406 (2002)
11. S. Galam, The September 11 attack: A percolation of individual passive support, *Eur. Phys. J.* **B 26**, Rapid Note, 269-272 (2002)
12. S. Galam Fragmentation versus stability in bimodal coalitions, *Physica* **A230**, 174-188 (1996)
13. R. Axelrod and D. S. Bennett A landscape theory of aggregation, *British J. Political Sciences* **23**, 211-233 (1993)
14. S. Galam, Comment on A landscape theory of aggregation, *British J. Political Sciences* **28**, 411-412 (1998)
15. K. Binder and A. P. Young, Spin glasses: experimental facts, theoretical concepts, and open questions, *Review of Modern Physics* **58**, 801-911 (1986)

Simulating Spatial Dynamics by Probabilistic Cellular Automata

Olga Bandman

Supercomputer Software Department
ICMMG, Siberian Branch
Russian Academy of Science
Pr. Lavrentieva, 6, Novosibirsk, 630090, Russia
bandman@ssd.sscc.ru

Abstract. A method is proposed, which is intended for constructing a probabilistic cellular automaton (CA), whose evolution simulates a spatially distributed process, given by a PDE. The heart of the method is the transformation of a real spatial function into a Boolean array whose averaged form approximates the given function. Two parts of a given PDE (a differential operator and a function) are approximated by a combination of their Boolean counterparts. The resulting CA transition function has a basic (standard) part, modeling the differential operator and the updating part modifying it according to the function value. Special attention is paid to the reaction-diffusion type of PDE. Some experimental results of simple processes simulation are given and perspectives of the proposed method application are discussed.

1 Introduction

Simulating nonlinear spatial dynamics is the main task of mathematical physics. By tradition, the process under simulation is represented by a system of partial differential equations (PDE). It is well known that it is practically impossible to obtain their analytical solutions (except very simple cases). So, numerical methods are to be used. Even though numerical analysis is progressing rapidly, its practical use meets certain difficulties. Particularly, implicit schemes of discretization lead to algorithms, which have no efficient parallel realizations. When using explicit schemes much effort is to be put to provide computational stability.

Cellular automata models having appeared as an alternative to the PDEs [1], are free of these shortages. They are absolutely stable, have no rounding off errors and are less time consuming. Moreover, border conditions in cellular automata are straightforward, which makes them appropriate to simulate fluids in porous media. Of course, cellular automata have their own problems, which are not yet completely solved. Among them the most important are the elimination of the automata noise and the account of physical parameters (concentration, viscosity, pressure) in terms of state-transition functions. Moreover, construction of a CA-model for a given phenomenon simulation is a nonformalized task whose solution

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 10–19, 2002.

requires a profound understanding of the process nature and much experience in dealing with cellular automata.

Meanwhile, most spatial dynamical processes have a well studied mathematical representation in the form of PDEs. Thus, it seems very natural to look for a formal method of transforming a given PDE system into a CA, whose evolution approximates its nonstationary solution. Such a method resulting in a probabilistic CA is proposed and presented below. The resulting CA is suggested to possess computation features of CA-models, and, hence, there is a hope that the method may be efficiently used instead of obtaining numerical PDE solution. Though, in order to obtain a quantitative assessment of the method efficiency much theoretical and experimental work is to be done.

Apart from the Introduction the paper contains three sections and the Conclusion. In the second section the procedure of transforming a real spatial function into a Boolean cellular array is given. Based on it the method for construction a probabilistic CA approximation of a PDE system is developed in the third section. In the forth section a few results of reaction-diffusion processes simulation are given. In the Conclusion some considerations are presented about the applicability of the proposed approach.

2 Boolean Discretization of a Continuous Spatial Function

Since the main objective is to obtain a discrete model (CA) intended to replace a continuous one (PDE), that is to be done first is to construct a transition between these two forms of data representation. More specifically, a transformation of a continuous spatial function into a Boolean array and vice versa is to be found.

Let x, y, z be real numbers, determining Cartesian coordinates in a continuous space, $u(x, y, z)$ be a differentiable function with the range in the interval [0,1]. Discretization of the space yields $x = h_x i, y = h_y j, z = h_z k$, where h_x, h_y, h_z are spatial steps, i, j, k - are integers. The function $u(i, j, k)$ may now be considered as a *cellular array* Ω_R, which is a set of cells. A *cell* is a pair of the form (u_m, m), where $m \in M$ is a cell name, $M = \{(i, j, k)\}$ being a *naming set*, which is a countable set of coordinate vectors considered as *names* and denoted by a single symbol m for short. The function values q are real numbers, $u_m \in\mathbf{R}$, representing *cell states* as the abstraction of a certain physical value (density, velocity, concentration, etc).

The spatial function $u(i, j, k)$ may be also represented by a Boolean array $\Omega_B = \{(v_m, m) : v_m \in \{0, 1\}, m \in M\}$. Such a representation is the most profound discretization of a continuous function, where all variables are discrete with the domain in the set of natural numbers and the range in the Boolean one. Transformation of the Boolean array into a usual real spatial function is performed by the procedure of *averaging.* Averaging is done for each cell over a certain area around it, determined by the *averaging neighborhood* in the following form.

$$N_{Av}(m) = \{(v_j(m), \phi_j(m)) : j = 0, \ldots, q-1\}. \tag{1}$$

where $\phi_j(m)$ is a *naming function* indicating the name of a jth neighbor of the cell (v_m, m). Let's agree that $\phi_0(m) = m$.

Averaging procedure consists of computation for each $m \in M$ the state values

$$v'_m = \frac{1}{q} \sum_{N_{Av}(m)} v_j(m), \tag{2}$$

resulting in $\Omega_{Av} = \{(v'_m, m) : v'_m \in \mathbf{R}, m \in M\}$

The inverse procedure which transforms a cellular array $\Omega_R = \{(u_m, m) : m \in M, u_m \in \mathbf{R}\}$, into a Boolean array $\Omega_B = \{(v_m, m) : v_m \in \{0,1\}, m \in M\}$, is called a *Boolean discretization.* As distinct to the averaging which is a deterministic and precise procedure, the Boolean discretization is an approximate and probabilistic one. The formal constraint on the resulting Boolean array is that for any $m \in M$

$$u_m - v'_m < \epsilon_m \tag{3}$$

where $v'_m \in\mathbf{R}$ is the averaged value of Ω_B, and $\epsilon \ll q$ is the admissible approximation error.

There is no exact solution of this problem. The approximate one is obtained by constructing Ω_B according to the following rule: the probability of the fact that $v_m = 1$ is equal to u_m, i.e.

$$P_{(v_m=1)} = u_m. \tag{4}$$

The above simple rule is a straightforward from the probability definition provided u_m is constant on the averaging neighborhood. In the general case the expected value $\mathrm{M}(v'_m)$ is the mean cell state over the averaging neighborhood, i.e.

$$\mathrm{M}(v'_m) = \frac{1}{q} \sum_{N_{Av}(m)} v_j(m) P_{(v_j(m)=1)} = \frac{1}{q} \sum_{N_{Av}(m)} u_j(m). \tag{5}$$

So, the approximation error ϵ vanishes only in a number of trivial cases, where the following holds.

$$u_m = \frac{1}{q} \sum_{N_{Av}(m)} u_j(m). \tag{6}$$

The set of such cases includes, for example, all linear functions and parabolas of odd degrees when considered relative to coordinate system with the origin in m. In general case $\delta_j(m) = u_j(m) - u_m \neq 0, \quad j = 1, \ldots, q-1$, and

$$\mathrm{M}(v'_m) = \frac{1}{q} \sum_{N_{Av}(m)} (u_m + \delta_j(m)) = u_m + \frac{1}{q} \sum_{N_{Av}(m)} \delta_j(m). \tag{7}$$

The second term in the right-hand side of (7) is the averaging error of the Boolean discretization in the cell named m.

$$\epsilon'_m = \frac{1}{q} \sum_{N_{Av}(m)} (u_j(m) - u_m). \tag{8}$$

From (8) it is clear, that the Boolean discretization error depends on the spatial function smoothness. The largest errors are in those cells, where the spatial function $u(m)$ has sharply defined extremes on their averaging areas. Such extremes may be caused both by the given initial cellular array and by the reaction function. The errors may be decreased by the appropriate choice of spatial discretization steps h_x, h_y, h_z and averaging area size.

3 Approximation of a PDE by a Probabilistic CA

The most general form of PDE representation is as follows

$$\frac{\partial u_g}{\partial t'} = \nabla^n u_g + f_g(u_1, \dots, u_l), \quad n = 1, 2, \quad g = 1, \dots, l, \tag{9}$$

where ∇^n is an n-order differential operator, $f_g(u_1, \dots, u_l)$ – an arbitrary function with the domain and the range in the interval (0,1).

If $n = 1$, the equation system of the type (9) describes, for example, electromagnetic process. If $n = 2$, i.e. ∇^2 is a Laplacian, the system (9) represents a reaction-diffusion phenomenon. When numerical methods of PDE solution are used, it is precisely these operators which cause the main troubles in providing stability of computation. Nevertheless, for both above types of differential operators there exist well known cellular automata, whose evolution simulate the corresponding processes. The examples may be found in [2] where a CA-model is proposed for elecrtromagnetic field, and in [3,4] where CA-diffusion models are studied. This brings up to the idea of replacing the finite-difference form of $\nabla^n u$ by simple and reliable CA-models, which are further referred to as *standard CAs*. Clearly, this transfers the computation process to the Boolean domain, which requires to perform the addition of $f_g(u_1, \dots, u_l)$ to a standard CA result in the Boolean form. Such a procedure is further referred to as *updating*. Its formal representation is as follows.

After time and space discretization with $t' = t\tau, t = 0, 1, \dots, ...$ and $x = h_x i,\ y = h_y j,\ z = h_z k$, the system (9) takes the form

$$u_g(t+1) = u_g(t) + d'\overline{\nabla}^n u_g(t) + \tau f_g(u_1(t), \dots, u_l(t)), \quad g = 1, \dots, l. \tag{10}$$

where $d'\overline{\nabla}^n u_g$ is a finite difference representation of the differential operator, d' depends on τ, d, h. Particularly, in the case of $n = 2,\ d' = \tau d/h^2$. In the right-hand side of (10) two first terms are responsible for a process to be simulated by a standard CA, while the function $f_g(u_1, \dots, u_l)$ should be turned out into a Boolean form and added to the standard CA result at each iteration of the simulation process. Since there are l variables to be simulated, the naming set

M is considered to be composed of l parts, $M = \cup_{g=1}^{l} M_g$, forming a layered structure so, that each $m_g = (i,j,k)_g \in M_g$ has a single name with the same (i,j,k) in each g-th layer.

Let the values of $u_g(t,x,y,z),\ g = 1,\ldots,l$, be the solutions of (9), which are taken as reference. Then the problem of constructing a CA, which simulates the same process is stated as follows. Given a PDE in the form of (10) and an initial array $\Omega_R(0) = \{(u_m(0), m)\}$, a CA should be constructed whose evolution starting from the Boolean discretization $\Omega_B(0) = \{(v_m(0), m)\}$ of $\Omega_R(0)$ provides at each t-th iteration for any $m \in M$ that

$$u_m(t) - v'_m(t) < \epsilon, \tag{11}$$

where v' is the averaged value over a certain averaging neighborhood. The latter may be different in different layers, so, $N_{Av}(m_g) = \{(u_j(m_g), \phi_j(m_g)) : j = 0,\ldots,q_g\}$ with $\phi_j(m_g) \in M_g$.

As it was mentioned above, the transition rule of a resulting CA is a combination of two procedures: 1) computation of the next state of a standard part and 2) updating it according to the functions $f_1,\ldots,f_l$ values. The first procedure follows the chosen standard CA-model, which is not described here, but a representative example is given in detail in the next section. The updating procedure relies upon the same probabilistic rule than that used for Boolean discretization of a real function, but with the account that the function value constitutes only a part of the total averaged cell state.

If $f_g(v'_{m_1}(t),\ldots,v'_{m_l}(t)) = f_{g_m} > 0$, then the updating should increase the amount of ones in $N_{Av}(m_g)$. Hence, a cell $(0, m_g)$ may, probably, be changed into $(1, m_g)$. Since in any averaging neighborhood $N_{Av}(m_g)$ there are $q_g(1 - w'_{m_g})$ zeros, then the probability of that change is

$$P_{(0,m_g)\to(1,m_g)} = \frac{f_{m_g}(t)}{1 - w'_{m_g}(t)}. \tag{12}$$

When $f_{m_g} < 0$ the updating should decrease the resulting averaged value, which is done by changing the cell $(1, m_g)$ into $(0, m_g)$ with the probability

$$P_{(1,m_g)\to(0,m_g)} = \frac{|f_{m_g}(t)|}{w'_{m_g}(t)}. \tag{13}$$

Denoting the right-hand sides of (12) and (13) as $T^+(f_{m_g})$ and $T^-(f_{m_g})$, respectively, the updating procedure is as follows.

$$v_{m_g}(t+1) = \begin{cases} 1, & if \quad w_{m_g}(t) = 0,\ f_{m_g}(t) > 0,\ T^+(f_{m_g}) > rand(1), \\ 0, & if \quad w_{m_g}(t) = 1,\ f_{m_g}(t) < 0,\ T^-(f_{m_g}) > rand(1), \\ v_{m_g}(t) & \text{otherwise} \end{cases} \tag{14}$$

where $rand(1)$ is a random number from the interval (0,1).

Let the result of a standard CA's tth iteration in each gth layer be $\Omega_S(t) = \{(w_g(t), m_g)\}$ and its averaged form – $\Omega'_S(t) = \{(w'_g(t), m_g)\}$, then the updating procedure according to (11) should meet the following condition. For any cell $m_g \in M_g,\ g = 1, \ldots, l,$ and any $t = 1, 2, \ldots$

$$v'_{m_g}(t) - \big(w'_{m_g}(t)) + f_g(v'_{m_1}(t), \ldots, v'_{m_l}(t))\big) < \epsilon, \tag{15}$$

where all terms are real values in the interval (-1,1).

From above it follows that a CA, whose evolution simulates the same process than a given PDE, is a multilayer cellular array, each layer corresponding to an equation of the system. Each cell is capable to perform two functions: a Boolean function of a standard CA, and updating the result according to (14). Both functions are computed by all cells in parallel, the whole array changing its global state iteratively. Since (14) may be considered as a function of a probabilistic neuron, such a cellular array is also called a *cellular-neural automaton* [5].

4 CA Simulation of Reaction-Diffusion Processes

A number of computer experiments have been carried out in order to obtain the assurance that the proposed method works properly. Two examples of 2D reaction-diffusion processes simulation by CAs, constructed according to the proposed method are given below. The first displays fire propagation usually given by a fist-order PDE. The second shows the development of a chemical reaction described by a second order PDE. In both cases the CA-diffusion [3, 6] called in [4] Block-Rotation (BR-diffusion) model, is used as a standard CA part.

BR-diffusion is a synchronous two-step CA, processing the cellular array Ω_B with the naming set $M = \{(i,j) : i,j = 0, \ldots\}$. Two partitions into square 2×2 blocks are defined on the array. The first one consists of *even* blocks, their diagonal cell name component sums $(i + j)$ being even. The second partition is the *odd* one. Its blocks have the diagonal cell names with odd $(i + j)$. In both cases the cell neighborhood $N(i,j)$ of a cell $(v, (i,j)) \in \Omega_B$ is the block, where this cell is in its left top corner, i.e.

$$N(i,j) = \{(v_0, (i,j)), (v_1, (i+1,j)), (v_2, (i+1,j+1)), (v_3, (i,j+1))\}. \tag{16}$$

CA rules are the following probabilistic ones.

$$N_{ij}(t+1) = \begin{cases} S'_{ij}(t) & if \quad \text{rand}(1) < p, \\ S''_{ij}(t) & if \quad \text{rand}(1) \geq (1-p), \\ N_{ij}(t) & \text{otherwise}, \end{cases} \tag{17}$$

where rand(1) is a random number in the interval (0,1), and the subarrays in the right-hand side are as follows

$$S'_{ij}(t) = \{((v_1, (i,j)), (v_2, (i+1,j)), (v_3, (i+1,j+1)), (v_0, (i,j+1))\}$$
$$S''_{ij}(t) = \{((v_3, (i,j)), (v_0, (i+1,j)), (v_1, (i+1,j+1)), (v_2, (i,j+1))\}.$$

At the first step of each iteration the above rules are applied to the even blocks, rotating them on $\pi/2$ either clockwise with the probability equal to p, or counterclockwise with the same probability. At the second step the same rules are applied to the odd blocks. In [3] the model with $p = 1/2$ is proved to simulate diffusion with the value d' from (10) equal to 3/2. Another diffusion coefficient value may be obtained by appropriate choice of p, τ, and h [4].

Fire propagation. Equation (9) with $f(u) = \alpha u(1-u)$ is a simplified model of reaction-diffusion process. With ($\alpha \leq 1/2$) and the following initial conditions

$$u(x,y) = \begin{cases} 1 & \text{if} \quad x,y < K, \\ 0 & \text{if} \quad x,y \geq K, \end{cases}$$

$K = const$, the equation represents an active wave with a center in $x = y = 0$, which simulates the propagation of fire, of weeds, of epidemics etc. Simulation process starts with initial array construction. Let's assume, that it is given as a 2D array $\Omega_R = \{(u,m) : u \in R, m \in M'\}$, where $M' \in M$ is a finite set of names, given as pairs of Cartesian coordinates, i.e. $M' = \{(i,j) : i = 0,\ldots,n,\ j = 0,\ldots,m\}$. In Fig.1 the first snapshot is the initial cellular array $\Omega_R(0)$ with $n = m = 200$. The black pixels correspond to the cells $(1,(ij))$, the white pixels – to the cells $(0,(i,j))$, the gray ones being inside the interval. The initial flash of fire is given by a black area at the top of the vertical channel formed by two walls, the right one having a hole with an opened cover. The walls are given by cells $(c,(i,j)),\ c \neq \{0,1\}$. The width of walls should be chosen larger than the radius ρ of the averaging area ($\rho = 10, q = (2\rho)^2 = 400$) to avoid hoping the "ones" over them in the process of averaging.

To begin the computation the initial Boolean array is also required. Hence, Ω_R is transformed into Ω_B , according to (4), but, since the array has a finite cardinality and there are cells belonging to the walls, the following border conditions should be taken into account. If in the averaging neighborhood of a cell $(u,(ij)) \in \Omega_R(t)$ there occur $r,\ r > 0$, cells, which do not belong to M', i.e. their states or names are not in $\{0,1\}$ or in M', respectively, then the probability of $(v,(i,j))$ to be $(1,(i,j))$ is as follows.

$$P_{(v_{i,j}=1)} = \frac{u_{i,j}}{q-r}, \tag{18}$$

The spatial structure of cell interactions is determined by two neighborhoods: $N(i,j)$ given as (16) needed for performing BR-diffusion, and $N_{Av}(i,j)$ for performing averaging and Boolean discretization.

Each tth iteration of CA evolution consists of the following procedures over $\Omega_B(t)$ and $\Omega_R(t)$.

1) The diffusion part is computed by applying the rule (17) both to all even and all odd blocks, border conditions being as follows. If a cell $(v,(i,j))$, such that either v or (i,j) do not belong to $\{0,1\}$ or to M', respectively, occurs in the neighborhood of $(v,(i,j))$, then no CA-rule is applied to it. The result of diffusion part computation is $\Omega_S(t+1) = \{(w,(i,j))\}$.

2) The function $f(v')$ for all cells in $\Omega_R(t)$ is computed forming a cellular array $\Omega_F(t) = \{(f_{i,j},(i,j))\}$.

3) Updating $\Omega_S(t+1)$ is performed according to (15), the result $\Omega_B(t+1)$ being the initial Boolean array for the next iteration.

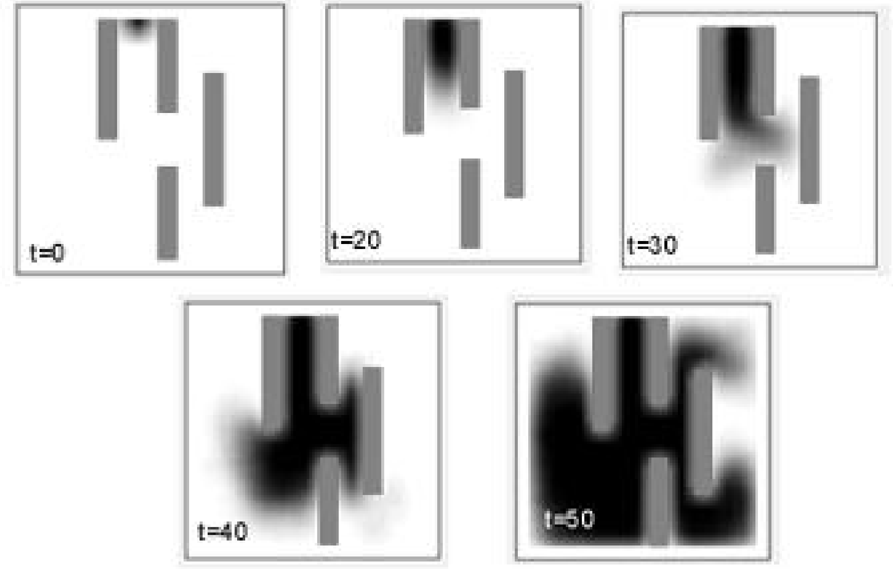


Fig. 1. Snapshots of CA evolution simulating fire propagation in 2D space, borgered by a two walls

4) The array $\Omega_B(t+1)$ is averaged according to (2) for cells whose neighborhood $N_{Av}(i,j) \in M'$. If in $N_{Av}(i,j)$ there are r cells out of M', then

$$v'_{(i,j)} = \frac{1}{q-r} \sum_{N_A v(i,j)} v_{i,j}.$$

The resulting $\Omega_R(t+1)$ is the initial averaged array for the next iteration.

In Fig.1 some snapshots are given displaying the CA evolution.

Chemical reaction. A great number of second order PDE systems modeling a reaction of Belousov-Zhabotinsky has been proposed and studied. One of them (a simplified one) [7] is chosen as an example for replacing the PDE system by a probabilistic CA. Using the notation of the paper the PDEs may be expressed as follows.

$$\begin{aligned} \tfrac{\partial u_1}{\partial t} &= d\big(\tfrac{\partial^2 u_1}{\partial x^2} + \tfrac{\partial^2 u_1}{\partial y^2}\big) + f_1(u_1, u_2), \\ \tfrac{du_2}{dt} &= f_2(u_1, u_2). \end{aligned} \qquad (19)$$

In (20) d is a diffusion coefficient,

$$f_1(u_1, u_2) = \frac{1}{d}(u_1(1-u_1) - zu_2(u_1 - s)/(u_1 + s)), \quad f_2(u_1, u_2) = u_1 - u_2$$

where s is the initial density of the substances in the medium, i.e. $(u_1(0) = s, u_2(0) = s)$, $z(x, y)$ is the spatial nonlinear function of both concentrations. With $x = 0, y = 0$ being chosen to be waves center, the constraints on $z(x, y)$ are expressed as follows: $0.5 < z(0, 0) < (1 - s)$. The process begins when in the homogeneous mixture of two substances with a certain gel there appears a little spot of saturated u_1.

According to the order of a PDE, the discrete array is a 2-layered one, i.e. $M = M'_1 \cup M'_2$, $M'_g = \{(i, j)_g : i, j = -n \ldots, 0, \ldots, n,\ g = 1, 2\ n = 150\}$. The initial arrays $\Omega_{R_g}(0) = \{(s, (i, j)_g\},\ g = 1, 2,\ s = 0.1$, except that in the center of $\Omega_{R_1}(0)$ there is a spot 5×5 of cells with $u_1 = 1$.

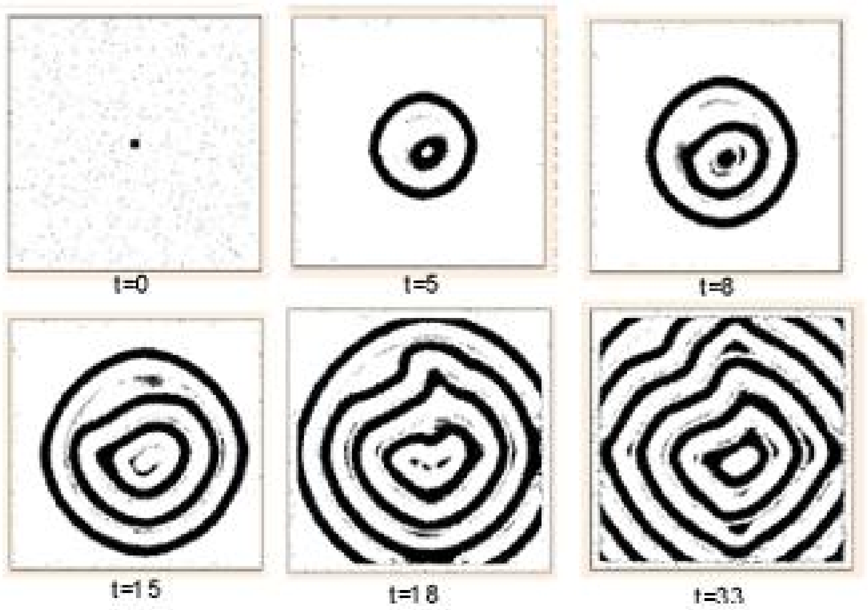


Fig. 2. Snapshots of CA evolution simulating the chemical reaction, generating concentric waves in 2D space

Boolean discretization of $\Omega_{R_1}(0)$ is achieved by setting cell states $v = 1$ with the probability $P_{(v_g=1)} = s$, except the square 5×5 in the center of Ω_{R_1}, where $v = 1$ determinately. Border conditions are taken into account according to (18). Since the second equation in (19) has no diffusion part, there is no need to perform Boolean discretization of $\Omega_{R_1}(0)$.

Each iteration of the CA approximating the PDE system (19) consists of the following computations.

1) An iteration of BR-diffusion is applied to Ω_{R_1} resulting in $\Omega_{S_1}(t + 1)$.

2) The functions f_1 and f_2 are computed for all cells $(i, j) \in M'$ forming cellular arrays Ω_{F_1} and Ω_{F_2}. The latter is also used as $\Omega_{R_2}(t + 1)$.

3) Updating $\Omega_{S_1}(t+1)$ is performed according to (15) resulting in $\Omega_{B_1}(t+1)$.
4) $\Omega_{B_1}(t+1)$ is averaged according to (2) to obtain $\Omega_{R_1}(t+1)$.
In Fig.2 some snapshots of the obtained CA evolution are shown.

5 Conclusion

A method of construction a CA, whose evolution simulates spatial dynamics given as a PDE. The resulting CA is a paradigm of fine-grained parallel computational model, intended for being included in a unified technology for parallel programming. Apart from the inherent parallelism the resulting CA model has some advantages derived from discreteness of the standard part, which provides the absence of rounding-off errors and improvement of computational stability.

As for the total efficiency of the simulation by the obtained probabilistic CA as compared with known numerical methods, in this stage it is possible to make a tentative qualitative estimates. The quantitative ones may be made after a long and hard work both of theoretical and experimental types. Nevertheless, the minor experience gained during the proposed method investigation enables us to believe that the proposed method is promising for natural phenomena simulation.

Acknowledgment. This work is supported by Russian Fund of Basic Research, grant 00-01-00026.

References

1. Toffolli T. Cellular Automata as an Alternative to (rather than Approximation) to Differential Equations in Modeling Physics. Physica, vol.10 D, (1984) 117-127.
2. Simons N.R.S., Bridges G.E., and Cuhachi M.: A Lattice Gas Automaton Capable of Modeling Three-Dimensional Electromagnetic Fields. Journal of Computational Physics, vol.151 (1999) 816-835.
3. Malinetski G.G., Stepantsov M.E.: Modeling Diffusive Processes by Cellular Automata with Margolus Neighborhood. Zhurnal Vychislitelnoy Matematiki i Matematicheskoy phiziki, vol. 36, N 6 (1998) 1017-1021 (in Russian).
4. Bandman O.: Comparative Study of Cellular-Automata Diffusion Models. In: Malyshkin V.(ed.):Lecture Notes in Computer Science, 1662. Springer-Verlag, Berlin (1999), 395-409.
5. Bandman O.: Cellular-Neural Automaton. A Discrete Model of Active Media Dynamics. In: Proceedings of the Conference, Devoted to 90th Aniversary of A.A.Liapunov (Novosibirsk, 2001). http://www.sbras.ru/ws/Lyap2001/23-34.
6. Toffolli T., Margolus N.: Cellular Automata Machines. MIT Press (1987) 280 p.
7. Tyson J.J., Fife.P.C. Target Patterns in a Realistic Model of the Belousov-Zhabotinskii Reaction. Journal of Chemical Physics, 73, (1980) 2224-2230.

Cellular Automata Models for Transportation Applications

Kai Nagel

Institute for Scientific Computing, Swiss Federal Institute of Technology Zürich (ETHZ),
8092 Zürich, Switzerland

Abstract. This paper gives an overview of the use of CA modes for transportation applications. In transportation applications, the CA dynamics is embedded within several other concepts, such as the fact that the dynamics unfolds on a graph instead of on flat 2d space, or multi-agent modeling. The paper also discusses the the limits of the CA technology in traffic.

1 Introduction

Cellular automata methods have their applications primarily in areas of spatio-temporal dynamics. Transportation simulations, with travelers and vehicles moving through cities, fall into that category. There are however also important differences between a “standard” CA simulation and those used in traffic. These differences are that in traffic, the dynamics is normally considered as unfolding on a graph instead of on flat space, and that particles in transportation simulations are better characterized as “intelligent agents”. These aspects will be discussed in Secs. 3 and 4. This is followed by a discussion of the limits of the CA technology and relations to other methods (Sec. 5), and a short outlook on a simulation of “all of Switzerland” (Sec. 6). The paper is concluded by a summary.

2 CA Rules for Traffic

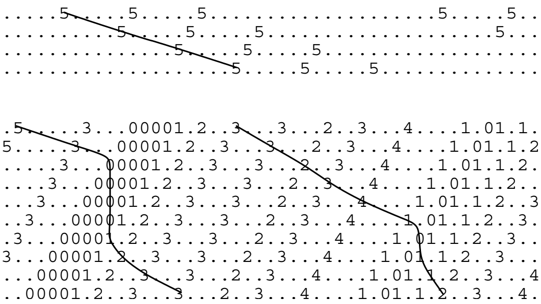
In CA models for traffic, space is typically coarse-grained to the length a car occupies in a jam ($\ell = 1/\rho_{jam} \approx 7.5$ m), and time typically to one second (which can be justified by reaction-time arguments [1]). One of the side-effects of this convention is that space can be measured in “cells” and time in “time steps”, and usually these units are assumed implicitly and thus left out of the equations. A speed of, say, $v = 5$, means that the vehicle travels five cells per time step, or 37.5 m/s, or 135 km/h, or approx. 85 mph.

Deterministic Traffic CA. Typical CA for traffic represent the single-lane road as a 1-dimensional array of cells of length ℓ, each cell either empty or occupied by a single vehicle. Vehicles have integer velocities between zero and v_{max}. A possible update rule is [2]

Car following:	$v_{t+1} = \min\{g, v_t + 1, v_{max}\}$
Movement:	$x_{t+1} = x_t + v_{t+1}$

The first rule describes deterministic car-following: try to accelerate by one velocity unit except when the gap is too small or when the maximum velocity is reached. g is the gap,

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 20–31, 2002.



```
.....5.....5.....5....................5.....5..
..........5.....5.....5....................5...
...............5.....5.....5...................
....................5.....5.....5..............

.5.....3...00001.2..3...3...2..3...4....1.01.1.
5.....3...00001.2..3...3...2..3...4....1.01.1.2
.....3...00001.2..3...3...2..3...4....1.01.1.2.
....3...00001.2..3...3...2..3...4....1.01.1.2..
...3...00001.2..3...3...2..3...4....1.01.1.2..3
..3...00001.2..3...3...2..3...4....1.01.1.2..3.
.3...00001.2..3...3...2..3...4....1.01.1.2..3..
3...00001.2..3...3...2..3...4....1.01.1.2..3...
...00001.2..3...3...2..3...4....1.01.1.2..3...4
..00001.2..3...3...2..3...4....1.01.1.2..3...4.
```

Fig. 1. Sequence of configurations of CA 184. Lines show configurations of a segment of road in second-by-second time steps; traffic is from left to right. Integer numbers denote the velocities of the cars. For example, a vehicle with speed "3" will move three sites (dots) forward. Via this mechanism, one can follow the movement of vehicles from left to right, as indicated by some example trajectories. TOP: Uncongested traffic. BOTTOM: Congested traffic.

i.e. the number of empty cells between the vehicle under consideration and the vehicle ahead, and v is measured in "cells per time step".

This rule is similar to the CA rule 184 in the Wolfram classification [3]; indeed, for $v_{max} = 1$ it is identical. This model has some important features of traffic, such as start-stop waves, but it is unrealistically "stiff" in its dynamics.

For this CA, it turns out that, after transients have died out, there are two regimes, depending on the system-wide density ρ_L (Fig. 1):

- Laminar traffic. All vehicles have gaps of v_{max} or larger, and speed v_{max}. Flow in consequence is $q = \rho \, v_{max}$.
- Congested traffic. All vehicles have gaps of v_{max} or smaller. It turns out that their speed is always equal to their gap. This means that $\sum v_i = \sum g_i = N_{veh} \times \langle g \rangle$. Since density $\rho = 1/(\langle g \rangle + 1)$, this leads to $q = \rho \, v = 1 - \rho$.

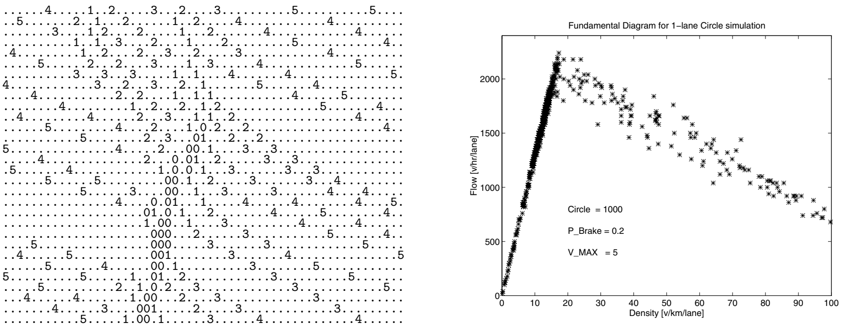
The two regimes meet at $\rho_c = 1/(v_{max} + 1)$ and $q_c = v_{max}/(v_{max} + 1)$; this is also the point of maximum flow.

Stochastic Traffic CA (STCA). One can add noise to the CA model by adding a randomization term [4]:

Car following:	$v_{t+\frac{1}{2}} = \min\{v_t + 1, g_t, v_{\max}\}$
Randomization:	$v_{t+1} = \begin{cases} \max\{v_{t+\frac{1}{2}} - 1, 0\} & \text{with probability } p_n \\ v_{t+\frac{1}{2}} & \text{else} \end{cases}$
Moving:	$x_{t+1} = x_t + v_{t+1}$

t and $t+1$ refer to the actual time-steps of the simulation; $t + \frac{1}{2}$ denotes an intermediate result used during the computation.

With probability p_n, a vehicle ends up being slower than calculated deterministically. This parameter simultaneously models effects of (i) speed fluctuations at free driving,



```
......4.....1..2.....3...2....3..........5..........5....
..5.......2..1...2......1..2.....4............4..........
.......3....1.2....2.....1...2.......4............4......
..........1..1..3....2....1....2.........5............4..
.4.........1..2....2...3...2.....3............4..........
.....5......2...2....3....1..3......4.............5......
..........3...3...3.....1..1....4.......4..............5.
4............3...2...3...2..1.......5.......4............
....4...........2..2....1..1.1...........4......5........
........4.........1..2...2..1.2..............5.......4...
..4.........4......2...3...1.1..2.................4......
......5.........4....2....1.0.2...2...................4..
...........5........2..3...01...2...2....................
5...............4.....2...00.1....3...3..................
.....4..............2...0.01..2......3...3...............
.5.......4............1.0.0.1...3.......3...3............
......5......5.........00.1..2.....3.......3...4.........
...........5......3....00..1...3......3.......4....4.....
................4....0.01...1.....4......4........4....5.
....................01.0.1...2........4......5........4..
....................1.00..1....3..........4.......4......
.....................000...2......3...........5.......4..
....5................000.....3.......3.............5.....
..4......5...........001........3.......4...............5
......5.......4......00.1..........3........5............
5..........5......1..01..2............3..........5.......
.....5..........2..1.0.2...3.............3............5..
...4......4.......1.00...2....3.............3............
.......4......3....001.....2.....3.............3.........
...........5.....1.00.1......3......4.............4......
..4.............1.000..1........3.......5.............5..
```

Fig. 2. Stochastic CA. LEFT: Jam out of nowhere leading to congested traffic. RIGHT: One-lane fundamental diagram as obtained with the standard cellular automata model for traffic using $p_{noise} = 0.2$; from [6].

(ii) over-reactions at braking and car-following, and (iii) randomness during acceleration periods.

This makes the dynamics of the model significantly more realistic (Fig. 2). $p_{noise} = 0.5$ is a standard choice for theoretical work (e.g. [5]); $p_{noise} = 0.2$ is more realistic with respect to the resulting value for maximum flow (capacity), see Fig. 2 (right) [6].

Slow-to-Start (S2s) Rules/Velocity-Dependent Randomization (VDR). Real traffic has a strong hysteresis effect near maximum flow: When coming from low densities, traffic stays laminar and fast up to a certain density ρ_2. Above that, traffic "breaks down" into start-stop traffic. When lowering the density again, however, it does not become laminar again until $\rho < \rho_1$, which is significantly smaller than ρ_2, up to 30% [7,8]. This effect can be included into the above rules by making acceleration out of stopped traffic weaker than acceleration at all other speeds, for example by making the probability p_n in the STCA velocity-dependent: If $p_n(v=0) > p_n(v \geq 1)$, then the speed reduction through the randomization step is more often applied to vehicles with speed zero than to other vehicles. Such rules are called "slow-to-start" rules [9,10].

Time-Oriented CA (TOCA). A modification to make the STCA more realistic is the so-called time-oriented CA (TOCA) [11]. The motivation is to introduce a higher amount of elasticity in the car following, that is, vehicles should accelerate and decelerate at larger distances to the vehicle ahead than in the STCA, and resort to emergency braking only if they get too close. The rule set is easier to write in algorithmic notation, where $v := v + 1$ means that the variable v is increased by one at this line of the program. For the TOCA velocity update, the following operations need to be done in sequence for each car:

1. if ($g > v \cdot \tau_H$) then, with probability p_{ac}: $v := \min\{v + 1, v_{max}\}$;
2. $v := \min\{v, g\}$
3. if ($g < v \cdot \tau_H$) then, with probability p_{dc}: $v := \max\{v - 1, 0\}$.

Typical values for the free parameters are $(p_{ac}, p_{dc}, \tau_H) = (0.9, 0.9, 1.1)$. The TOCA generates more realistic fundamental diagrams than the original STCA, in particular when used in conjunction with lane-changing rules on multi-lane streets.

Dependence on the Velocity of the Car Ahead. The above rules use gap alone as the controlled variable. More sophisticated rules will use more variables, for example the first derivative of the gap, which is the velocity difference. The idea is that if the car ahead is faster, then this adds to one's effective gap and one may drive faster than without this. In the CA context, the challenge is to retain a collision-free parallel update. Ref. [12] achieved this by going through the velocity update twice, where in the second round any major velocity changes of the vehicle ahead were included. Ref. [13] instead also looked at the gap of the vehicle ahead. The idea here is that, if we know both the speed and the gap of the vehicle ahead, and we make assumptions about the driver behavior of the vehicle ahead, then we can compute bounds on the behavior of the vehicle ahead in the next time step.

Traffic Breakdown. An interesting topic is the transition from laminar to congested traffic. For the deterministic model, things are clear: The laminar regime is when all vehicles move at full speed; the congested regime is when at least one vehicle in the system does not move at full speed. Deterministic models can also display bi-stability, i.e. density ranges where both the laminar and the congested phase are stable. This is for example the case with deterministic slow-to-start models [14]. This characterization is the same as for deterministic fluid-dynamical models [15].

For stochastic models, things are less clear since even in the laminar regime there may be slow vehicles, their slowness caused by random fluctuations. Often, the analogy to a gas/liquid transition is used, meaning that traffic jams are droplets of the liquid phase interdispersed in the gaseous phase of laminar traffic. However, the question of a phase transition in stochastic models has not been completely settled [16,17,18]. The main problem seems to be that questions of meta-stability and of phase separation are not treated separately, although they should be, as our own recent investigations show [19].

Lane Changing. Lane changing is implemented as an additional sub-timestep before the velocity update. Lane changing consists of two parts: the reason to change lanes, and the safety criterion. The first one can be caused by slow cars ahead, or by the desire to be in the correct lane for a turn at the end of a link. The safety criterion means that there should be enough empty space around a vehicle which changes lanes. A simple symmetric implementation of these principles is:

- Reason to change lanes (incentive criterion): $g \leq v$ .AND. $g_o > g$, where g is the standard gap, and g_o is the gap ahead on the other lane. The rule means that a reason to change lanes is given when the gap on the current lane inhibits speed and when the gap on the other lane is larger. – This is a simple symmetric criterion based on gaps, more complicated and/or asymmetric criteria are possible [20].
- Safety criterion: $g_o \geq v$ .AND. $g_{b,o} > v_{b,o}$, where the index $_{b,o}$ refers to the vehicle coming from from behind on the other lane. This safety criterion is fulfilled if the gap on the other lane is larger than the current velocity, and the backwards gap on the other lane is larger than the oncoming vehicle's velocity.

The safety criterion is in fact important in order to maintain laminar traffic [21], an aspect that should not be forgotten if one has spent considerable effort in designing rules for stable laminar high flow traffic on single lanes [9].

3 Dynamics on a Graph

A big difference between typical CA models and those used for transportation applications is that the latter typically operate on a **graph**. A graph consists of nodes and links. Nodes for transportation applications have ID-numbers and geographical coordinates. Links connect two nodes, and they have attributes such as speed limit or number of lanes. Obviously, nodes correspond to intersections and links to the roads connecting them.

Traffic on links can be represented through 2d arrays, with one dimension being the length and the other one being the number of lanes, and using the driving models from Sec. 2. The only addition is to include lane changes for plan following, which forms an additional incentive to change lanes as discussed in Sec. 2. The remaining parts of the driving logic concern themselves with intersections.

Intersections. An easy way to deal with intersections is to treat intersections as "black boxes" without internal dynamics. In this case, the prioritization is handled when vehicles are about to enter the intersection. There are two important cases: turning movements which are "protected" by traffic signals, and unprotected turns. These will be discuseed in the following.

Protected turns are straightforward, since the signal schedule is assumed to take care of possible conflicts. If vehicles can brake to zero speed in one time step (as is assumed in most CA models for traffic), then a yellow phase is not needed. The only other condition for a vehicle to move through an intersection is that there needs to be space on the outgoing link.

Unprotected turns (yield, stop, merging, etc.) are more advanced. In general, for each turning movement a list of conflicting lanes needs to be available, which is normally generated via pre-processing and is part of the network coding. A vehicle that wants to go through an intersection checks for each conflicting lane if there is a conflicting vehicle, and only if there is none and if in addition the outgoing link has space, then the vehicle can move.

The rules for conflicting lanes are normally treated in terms of gap acceptance, very similar to the safety criterion in lane changing. For example, one can demand that for each interfering lane, a conflicting vehicle needs to be at least $n \times v$ cells away, where n is a small number, and v is the speed of the conflicting vehicle. If the simulation has a time step of 1 sec, then n corresponds to the time gap of the conflicting vehicle in seconds. In reality, this time gap is of the order of 5 sec; in CA-based simulations, we found that 3 sec yields more realistic dynamics.

Unexpected Side Effects and Calibration/Validation. Sometimes, an arbitrary rule, as plausible as it may be, can have unexpected side effects. For example, $g > n \times v$ means that with $v = 0$ the gap still needs to be larger than or equal to one. In contrast, with $g \geq n \times v$ the turn will be accepted when the gap is zero and the conflicting vehicle is not

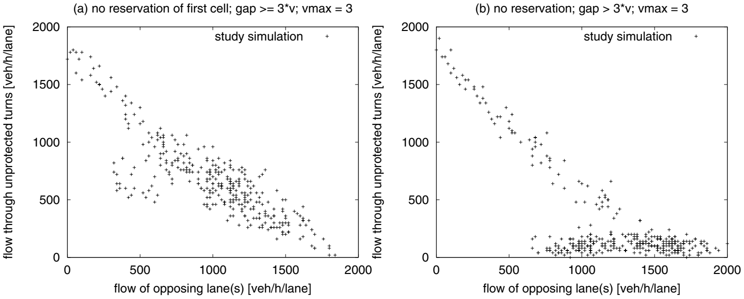


Fig. 3. Different yield rules. LEFT: Vehicles accept turn if $g \geq 3\,v$. RIGHT: Vehicles accept turn if $g > 3\,v$. Note the large difference in the congested regime.

moving. The resulting differences in fundamental diagrams (see Fig. 3) are enormous. The latter turns out to model “zip-lock” dynamics, which is in fact the desired behavior under congested conditions.

In protected turns during the green phase as well as for unprotected turns which have the priority (such as a freeway link connecting to another freeway link at the position of an off-ramp), care has to be taken that free traffic flows unobstructed through the connection. This means, for example, that for CA logic with $v_{max} > 1$, up to v_{max} cells of the outgoing link need to be considered.

Care has also to be taken when different incoming links compete for space on the same outgoing link. Although in principle the prioritization given by traffic rules should take care of this, in practice such conflicts can rarely be completely avoided, for example because of small network coding errors. In order to have a robust implementation, it is thus desirable that vehicles reserve cells where they intend to go.

This can again lead to unexpected effects. For example, we noticed above that the condition $g \geq n \times v$ is very different from $g > n \times v$ under congested conditions. In TRANSIMS, however, it turns out that there is in fact no difference at all between the two rules. Why is that? The answer is that in TRANSIMS, vehicles with velocity zero on the main road reserve space on the outgoing link on the assumption that they might accelerate to speed one. In consequence, vehicles from the minor road cannot move to that same space, even if it turns out that the vehicle on the major road does not move after all.

In order to find out about such unexpected effects, driving logic should be systematically tested. In fact, there should be standardized test cases that each micro-simulation should do and which should be publicly available, e.g. on the internet. A minimum set of tests would consist of the fundamental diagrams for 1-lane, 2-lane, and 3-lane traffic (such as Fig. 2 right), and for all unprotected merging movements (such as Fig. 3). Such tests should be done with the production version of the code so that differences between the specification and the actual implementation could be detected. These tests should be available as an easy-to-configure run of the software package, and the results should be available on the Internet.

The simplest version of the TRANSIMS driving logic consists in fact of the rules described above. Most discussed variations, such as different gap acceptance at unprotected turns, or different p_{noise}, can be changed by global parameters.

The "Roundabout" Solution. An elegant solution to many of these conflicts is the use of small roundabouts at intersections [22]. The advantage of roundabouts is that the high complexity of interfering lanes of standard intersections is decomposed into smaller sub-units, where in each sub-unit only the conflict between the roundabout and the incoming lane needs to be resolved.

Alternative Implementation of Graph Dynamics. The above description assumes that a street network is given as a graph. It was already said that this yields realistic description at links, but may become problematic at nodes. The software package VISSIM [23] therefore dispenses with nodes and connects links via a second type of links, called connectors. Such connectors start somewhere on a link and end somewhere on a (usually different) link. There is no need that they start at beginnings or ends of links. Any vehicle which encounters an outgoing connector somewhere on the link can decide to go there, or to continue on the link. However, a vehicle need to select one of the connectors eventually, since there is nowhere to go once the vehicle reaches the end of the link.

Similarly, incoming traffic is modeled via connectors which end somewhere on the link, not necessarily at its beginning. There need to be rules which resolve conflicts between incoming vehicles and vehicles which are already on a link.

As a radically different approach, it is possible to dispense with the graph dynamics completely. The whole transportation system then is overlayed by a CA grid structure, and vehicles always move within cells. The typical artifacts with off-axis movement are compensated for by smoothing techniques. CITY-TRAFFIC [24] seems to be using this technology; we use a similar approach for the simulation of tourists hiking in the Alps [25].

4 Moving Particles and Moving Agents

There is a stark difference between typical physics particle hopping models and the transportation models: in transportation, the particles are **intelligent agents**, meaning that they have individual and potentially complex rules of how they react to the environment. This means that in implementations some pointer to these complicated rule structures needs to be maintained when the particles move. This immediately excludes the use of single bit coding, since these pointers typically occupy 32 or even 64 bits of memory.

In consequence, a simple typical vehicle grid in a transportation looks as follows

```
class veh {
    int ID ;
    double Speed ;
    ...
};
veh* Road[200];  // memory allocation for 200 pointers
```

which means that Road consists of 200 pointers pointing to vehicles. Memory for the pointers is immediately allocated; memory for the vehicles is only allocated when it is used. For example, when the code decides to put a vehicle into cell number ii, the code fragment may look as

```
Road[ii] = new veh ; // memory allocation for vehicle
Road[ii]->ID = SomeID ;
Road[ii]->Speed = 0. ;
```

Movement is still done in a relatively standard way:

```
speed = int(Road[ii]->Speed) ;
Road[ii+speed] = Road[ii] ;
Road[ii] = NULL ;
```

The big advantage of this is that all information belonging to the vehicle is always being moved with it.

Clearly, many improvements to the above are possible or even recommended, such as using vectors instead of arrays, making clean definitions of constants such as “200”, making IDs constant, explicitely defining constructors and destructors, etc.

The currently most important application of the agent technology in transportation simulations is that agents know where they are going. More precisely, it is possible to give each agent its full route, for example consisting of a sequence of nodes. A related but different area of research is to generate those strategic decisions for the agents. All this results in additional computational modules which are part of a complete transportation simulation package (e.g. [26]).

5 Limits of the CA Technology and Relations to Other Methods

More Realistic Representations. A standard problem with CA methods is that they may be difficult to calibrate against realistic values. Take for example the STCA as described above in Sec. 2. The length of a cell is straightforward: this needs to be the space a vehicle occupies in a jam in the average. The time step is traditionally taken as 1 sec, which is justified by reaction time arguments [27]. This implies that speeds come in increments of 7.5 m/s; 5 cells per second $=$ 37.5 m/s $=$ 135 km/h is a convenient maximum speed. The remaining free parameter, p_{noise}, is now selected such that the maximum flow comes out at 2000 veh/sec; this results in $p_{noise} = 0.2$. Lane changing rules can be calibrated similarly, and can even reproduce the density inversion which happens on German freeways when they are close to capacity [28].

So far so good. The problems start if for some reason the above is not good enough. For example, the existing speed classes are not fine enough to resolve a difference between a 55mph and a 50mph speed limit, a common occurence in the U.S. Similarly, although the fundamental diagram comes out plausibly, acceleration of vehicles turns out to be too high, which is a problem for emissions calculations.

And it is difficult to resolve those problems via a clever choice of the probability p_{noise}. For example, increasing p_{noise} leads to lower acceleration (which is desired), but also to lower throughput (which is not desired). A possible way out is to have p_{noise} dependent on the velocity: A small p_{noise} at low velocities together with a large p_{noise}

at high velocities leaves the fundamental diagram nearly unchanged while leading to a much lower average acceleration. However, unfortunately such measures also change the fluctuations of the system – for example, such a reduced acceleration will lead to a much wider spread of the times that vehicles need to accelerate from 0 to full speed. Also note that in slow-to-start models, the modifications of p_{noise} are exactly the other way round.

As an alternative, it would be possible to make the resolution of the cells finer, for example to introduce cells of length 3.75 m and make vehicles occupy two cells. It is unclear if this would be worthwhile; it would certainly be slower than the standard method because twice the number of cells needs to be treated.

A possible method that seems to work well in many cases in practice are hybrid simulations. Here, one leaves the cellular structure intact, but allows for offsets of particles against the cellular structure. For directional traffic, it seems that one can ultimately completely dispense with the grid and work with a method that still has a 1 sec time resolution but a continuous resolution in space [27]. The reason why this works for traffic is that it is computationally relatively cheap to keep track of neighbors since a link is essentially one-dimensional. For higher-dimensional simulations, keeping some cellular structure is normally advantagous for that task alone – see for example the parallel code for molecular dynamics which turned out to also handle the problem of neighbor finding very efficiently.

(Even) Less Realistic Representations. Another problem with microscopic simulations often is that the necessary input data is not available. For example, for a CA-based traffic microsimulation one would need at least the number of lanes and some idea about the signal schedules. Most transportation network databases, in particular if they were put together for transportation planning, only contain each link's capacity. It is difficult to construct CA links so that they match a given capacity. The only way seems to be a heuristic approach, by selecting the right number of links and then to restrict the flow on the link for example by a (fake) traffic light. Still, this leaves many questions open. For example, signals phases need to be coordinated so that not two important incoming links try to feed into the same outgoing link at the same time. Furthermore, from the above it is not clear which incoming lane feeds into which outgoing lane (lane connectivities).

In consequence, there are situations where a CA representation is still too realistic, and a simpler representation is useful. A possibility to do this is the queue model. This is essentially a queuing model with added queue spillback. Links are characterized by free speed travel time, flow capacity, and storage capacity. Vehicles can enter links only when the storage capacity is not exhausted. Vehicles which enter a link need the free speed travel time to arrive at the other end of the link, where they will be added to a queue. Vehicles in that queue are moved accross the intersection according to the capacity constraint, and according to availability of space on the next link.

This describes only the most essential ingredients; care needs to be taken to obtain fair intersections and for parallelization [29]. Also, there are clearly unrealistic aspects of the queue model, such as the fact that openings at the downstream end of the link are immediately transmitted to the upstream end. This has for example the consequence that queue resolution looks somewhat unrealistic: queues break up along their whole length simultaneously, instead of from the downstream end. Nevertheless, the queue simulation is an excellent starting point for large scale transportation simulations.

6 A Simulation of All of Switzerland

One of the current main goals in our group is a simulation of "all of Switzerland". By this we mean a microscopic 24h simulation of a typical workday of all traffic in Switzerland. Fig. 4 contains an early result of this.

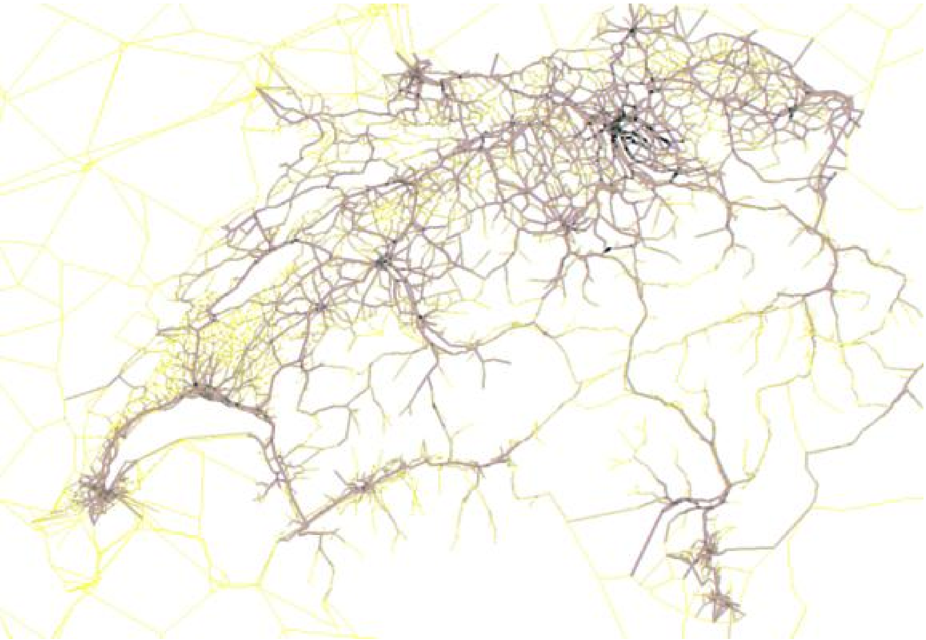


Fig. 4. Most of Switzerland at 8am, simulation result. The graphics shows individual vehicles, but they are so small that they cannot be distinguished. Areas in dark contain traffic jams.

The network that is used was originally developed for the Swiss regional planning authority (Bundesamt für Raumentwicklung), and has since been modified by Vrtic at the IVT and by us. The network has the fairly typical number of 10 572 nodes and 28 622 links. Also fairly typical, the major attributes on these links are type, length, speed, and capacity.

Demand is obtained from a 24-hour origin-destination matrix with 3066 zones, also from the Bundesamt für Raumentwicklung. This matrix is converted to 24 separate hourly matrices by a several-step heuristic. In the long run, it is intended to move to activity-based demand generation. Then, as explained above one would start from a synthetic population, and for each population member, one would generate the chain of activities for the whole 24-hour period.

Routes are obtained via iterations between simulation and time-dependent fastest path routing. The simulation behind Fig. 4 is the queue simulation as described in Sec. 5.

7 Summary

This paper has outlined the most important elements of CA use in transportation applications. Besides the standard CA rules of "traffic on a link", the important aspects are, that the dynamics unfolds on a graph instead of on flat space, and that the particles are intelligent. Both aspects make simulation packages considerably more complicated, the first since intersections need to be modeled; the second since the "intelligence" of the travelers (route choice, destination choice, activity generation, etc.) needs to be modeled. Finally, the limits of the CA technology were discussed. These limits exist in two directions: (1) The driving logic of the CA rules may not be realistic enough, and making it more realistic may be computationally as expensive as moving to coupled map lattices (discrete time, contiuous state space). (2) The available real world data may not be detailed enough to feed a realistic CA-based micro-simulation.

Acknowledgments. This paper and in particular the work on the simulation of "all of Switzerland" would not have been possible without the help of Kay Axhausen, Nurhan Cetin, Christian Gloor, Bryan Raney, Milenko Vrtic, Res Voellmy, and Marc Schmitt.

References

1. S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, University of Cologne, Germany, 1997. See www.zpr.uni-koeln.de.
2. K. Nagel and H. J. Herrmann. Deterministic models for traffic jams. *Physica A*, 199:254, 1993.
3. S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific, Singapore, 1986.
4. K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221, 1992.
5. D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329(4–6):199–329, May 2000.
6. K. Nagel, P. Stretz, M. Pieck, S. Leckey, R. Donnelly, and C. L. Barrett. TRANSIMS traffic flow characteristics. Los Alamos Unclassified Report (LA-UR) 97-3530, Los Alamos National Laboratory, see transims.tsasa.lanl.gov, 1997.
7. B. S. Kerner and H. Rehborn. Experimental features and characteristics of traffic jams. *Physical Review E*, 53(2):R1297–R1300, 1996.
8. B. S. Kerner and H. Rehborn. Experimental properties of complexity in traffic flow. *Physical Review E*, 53(5):R4275–R4278, 1996.
9. R. Barlovic, L. Santen, A. Schadschneider, and M. Schreckenberg. Metastable states in cellular automata. *European Physical Journal B*, 5(3):793–800, 10 1998.
10. D. Chowdhury, L. Santen, A. Schadschneider, S. Sinha, and A. Pasupathy. Spatio-temporal organization of vehicles in a cellular automata model of traffic with 'slow-to-start' rule. *Journal of Physics A – Mathematical and General*, 32:3229, 1999.
11. W. Brilon and N. Wu. Evaluation of cellular automata for traffic flow simulation on freeway and urban streets. In W. Brilon, F. Huber, M. Schreckenberg, and H. Wallentowitz, editors, *Traffic and Mobility: Simulation – Economics – Environment*, pages 163–180. Aachen, Germany, 1999.
12. D.E. Wolf. Cellular automata for traffic simulations. *Physica A*, 263:438–451, 1999.

13. C. L. Barrett, M. Wolinsky, and M. W. Olesen. Emergent local control properties in particle hopping traffic simulations. In D.E. Wolf, M. Schreckenberg, and A. Bachem, editors, *Traffic and granular flow*, pages 169–173. World Scientific, Singapore, 1996.
14. M. Takayasu and H. Takayasu. Phase transition and $1/f$ type noise in one dimensional asymmetric particle dynamics. *Fractals*, 1(4):860–866, 1993.
15. B. S. Kerner and P. Konhäuser. Structure and parameters of clusters in traffic flow. *Physical Review E*, 50(1):54, 1994.
16. L. Roters, S. Lübeck, and K.D. Usadel. Critical behavior of a traffic flow model. *Physical Review E*, 59:2672, 1999.
17. M. Sasvari and J. Kertesz. Cellular automata models of single lane traffic. *Physical Review E*, 56(4):4104–4110, 1997.
18. D. Chowdhury et al. Comment on: “Critical behavior of a traffic flow model”. *Physical Review E*, 61(3):3270–3271, 2000.
19. K. Nagel, Chr. Kayatz, and P. Wagner. Breakdown and recovery in traffic flow models. In Y. Sugiyama et al, editor, *Traffic and granular flow ’01*. Springer, Heidelberg, in press.
20. K. Nagel, D.E. Wolf, P. Wagner, and P. M. Simon. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2):1425–1437, 1998.
21. M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour. Two lane traffic simulations using cellular automata. *Physica A*, 231:534, 1996.
22. A. Dupuis and B. Chopard. Cellular automata simulations of traffic: a model for the city of Geneva. *Networks and Spatial Economics*, forthcoming.
23. Planung Transport und Verkehr (PTV) GmbH. See www.ptv.de.
24. City-traffic. Fraunhofer Institut Autonome Intelligente Systeme (AIS). See www.citytraffic.de.
25. See www.inf.ethz.ch/˜nagel/projects/alpsim.
26. K. Nagel, J. Esser, and M. Rickert. Large-scale traffic simulation for transportation planning. In D. Stauffer, editor, *Annual Reviews of Computational Physics*, pages 151–202. World Scientific, 2000.
27. S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, University of Cologne, Germany, 1997. See www.zpr.uni-koeln.de.
28. P. Wagner. Traffic simulations using cellular automata: Comparison with reality. In D E Wolf, M.Schreckenberg, and A.Bachem, editors, *Traffic and Granular Flow*. World Scientific, Singapore, 1996.
29. N. Cetin and K. Nagel, in preparation.

An Evolutionary Approach to the Study of Non-trivial Collective Behavior in Cellular Automata

Francisco Jiménez-Morales

Departamento de Física de la Materia Condensada. Universidad de Sevilla.
P. O. Box 1065, 41080-Sevilla, Spain.
jimenez@us.es

Abstract. A genetic algorithm (GA) is used to evolve two and one dimensional cellular automata (CA) to perform a non-trivial collective behavior task. Using as fitness function the average area in the iterative map, the GA is able to discover several rules with the desired behavior. In $d = 2$ we study the scaling of the attractor versus lattice size and noise. In $d = 1$, using the tools of the computational mechanics, the structural organization of the CA dynamics is uncovered.

1 Introduction

Cellular Automata (CA) are fully discrete dynamical systems, where the states are chosen in a finite set and distributed on a discrete grid, the time evolution is run synchronously in all the sites of a regular lattice and each site changes its state $s_i(t)$ (0 or 1) according to a local rule ϕ that only depends upon its neighbor values. Despite the simplicity of their construction CA are found to be capable of diverse and complex behavior and are often used as a prototype for the analysis of spontaneous emergence of ordered behavior in spatially extended systems that are locally coupled.

As CA are governed by local interactions and subjected to noise, it was expected that any globally observable, such as the concentration of activated cells $c(t) = \frac{1}{N}\sum_i^N s_i(t)$ would show a trivial time dependence in the limit of infinite size [1]. But several exceptions to this have been found. The most remarkable one is a quasiperiod three behavior (QP3) that exhibits the concentration of rule-33 automaton in d=3 [7] and other CA in high space dimensions [3]. This behavior is neither transient nor due to the finite size of the lattice and has been obtained for deterministic and probabilistic rules [9]. Several attempts have been made to understand its phenomenology and have addressed the possible mechanisms by which this puzzling collective behavior emerges [2] but at the moment there is not any answer to the question of how this non-trivial collective behavior can be predicted from the local rule.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 32–43, 2002.

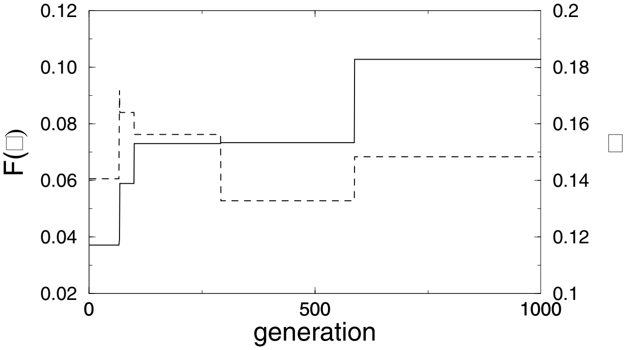


Fig. 1. Best fitness rule and λ-parameter versus generation in the run in which rule ϕ_a^2 was discovered in generation 590. Lattice size of 32^2 cells.

In this work we use a Genetic Algorithm to evolve a population of two and one dimensional CA. The survival of an individual CA rule is determined by its ability to perform a "P3 task". The goal is to find a CA that starting from a random initial configuration reaches one in which the concentration oscillates among three different values, i.e. the GA will select rules with P3 or QP3 collective behavior.

2 The Genetic Algorithm

Our GA begins with a population of $P = 20$ randomly generated chromosomes, listing the rule-table output bits in lexicographic order of neighborhood patterns [8]. We consider binary CA with periodic boundary conditions. Each CA is represented by a bit string delineating its rule table ϕ, containing the output bits for all possible neighborhood configurations. The bit string is of size $2^7 = 128$, resulting in a huge space of 2^{128} possible rules. The fitness evaluation for each CA rule is carried out on a lattice of $N = L^d$ cells starting from a random initial condition of concentration 0.5. After a transient time of $N/2$ time steps, we allow each rule to run for a maximum number of $N/2$ iterations. The values of concentration are assembled in groups of 4 consecutive values and the fitness function $F(\phi)$ is defined by:

$$F(\phi) = \frac{4}{M}\sum_{i}^{M/4}\frac{1}{2}abs[(c_2-c_1)(c_4-c_2)-(c_3-c_2)(c_3-c_1)]_i$$

The rule's fitness $F(\phi)$ is taken from a geometrical point of view and it is an average area in the iterative map, i.e. the graph of $c(t+1)$ versus $c(t)$. In this iterative map the area of a period-2 behavior is too small, almost 0, the area of a noisy period-1 and the area of an intermittent P2 is higher than that of a P2 and finally QP3 and P3 behaviors have the highest values.

Table 1. The best evolved rules, the rule table hexadecimal code, the type of non-trivial collective behavior and the Langton's parameter. To recover the 128-bit string giving the output bits of the rule table, expand each hexadecimal digit to binary. The output bits are then given in lexicographic order. The arrangement of neighbors for ϕ_a^2 is $|s_{i,j}|s_{i-3,j}|s_{i+3,j}|s_{i,j-2}|s_{i,j+2}|s_{i-2,j}|s_{i+2,j}|$, while for ϕ_b^2 is $|s_{i,j}|s_{i-3,j}|s_{i+3,j}|s_{i,j-1}|s_{i,j+1}|s_{i-2,j}|s_{i+2,j}|$. In $d = 1$ the arrangement of neighbors is $|s_{i-3}|s_{i-2}|s_{i-1}|s_i|s_{i+1}|s_{i+2}|s_{i+3}|$.

	Symbol	**Rule Table Hexadecimal Code**	**NTCB**	λ
$d = 2$	ϕ_a^2	10000008-1004000a-10000048-108e0c43	QP3	0.148
	ϕ_b^2	10000008-1000000a-100000cc-10860cc3	QP3	0.156
$d = 1$	ϕ_a^1	21088418-01091108-41038844-10c18080	P3	0.211
	ϕ_b^1	ffbe84bc-10874438-c6a08204-9d1b800b	P3	0.414
	ϕ_c^1	146157d1-fbb53fec-7dfbeffc-eaf0fa28	QP3(P3)	0.625
	ϕ_d^1	f193800-c06b0eb0-e000461c-80659c11	P3	0.336

In each generation: (i) $F(\phi)$ is calculated for each rule ϕ in the population. (ii) The population is ranked in order of fitness. (iii) A number $E = 5$ of the highest fitness ("elite") rules is copied without modification to the next generation. (iv) The remaining $P - E = 15$ rules for the next generation are formed by single-point crossover between randomly chosen pairs of elite rules. The offsprings from each crossover are each mutated with a probability $m = 0.05$. This defines one generation of the GA; it is repeated $G = 10^3$ times for one run of the GA.

3 Results

We performed more than 500 different runs of the GA each with a different random-number seed. The dynamics of a typical run is shown in Figure 1 which plots the fittest rule of each generation and the Langton's parameter λ which is the fraction of 1s in the rule table. Though the study of the generational progression of the GA can give important information about the design of specific CA rules, here we focus mainly on the behavior of the last evolved rule. Table 1 shows the best evolved rules in $d = 2$ and $d = 1$, the rule table hexadecimal code, the kind of collective behavior observed and the λ parameter.

3.1 Experiments in d=2

In most runs the GA ends up with rules that show a noisy P1 or a P4 collective behavior, but in a few cases the GA is able to detect some rules with a QP3 collective behavior. Figure 2 shows the iterative map of the fittest rule in the run in which ϕ_a^2 was found. In the initial generations, Figure 2a-b, the GA detects a rule with a cloudy P1 behavior. At generation 253 (Figure 2c) the GA finds a rule for which its iterative map shows a dense triangular object. Finally at

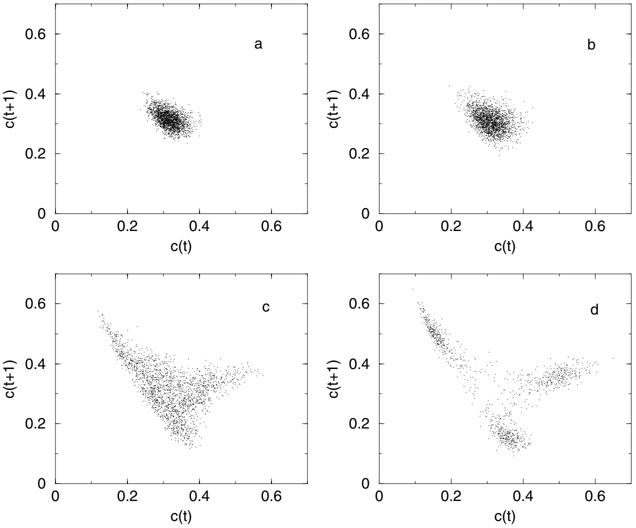


Fig. 2. The iterative map for the best evolved rule in generation: (a) 10; (b) 100; (c) 263 and (d) 637 when ϕ_b^2 was found. Lattice size of 32^2 cells.

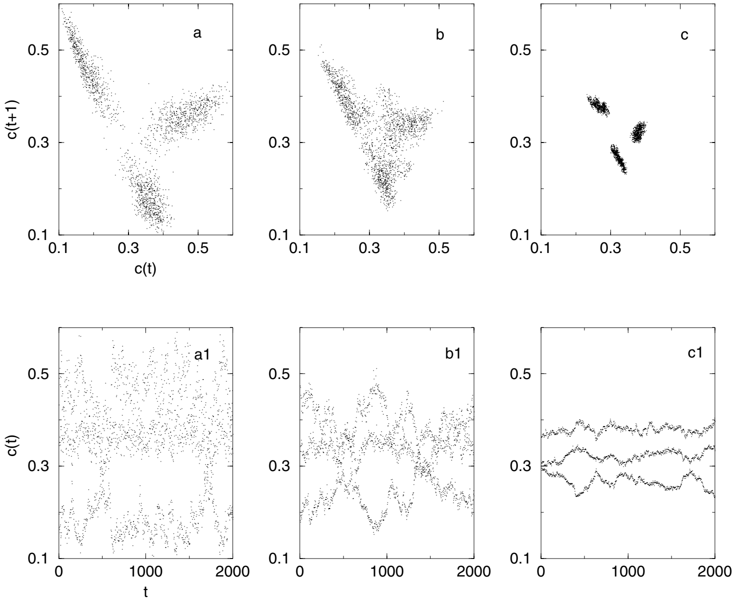


Fig. 3. The iterative map and the time series of the concentration for rule ϕ_a^2 in d=2 for different lattice size. (a) 32^2; (b) 64^2 and (c) 256^2. Transient discarded

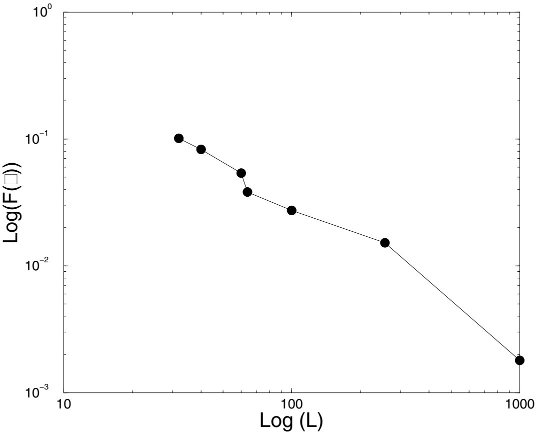


Fig. 4. Log-log plot of the Fitness function for rule ϕ_a^2, versus lattice size $L = N^{\frac{1}{2}}$.

generation 637 rule ϕ_a^2 is found, its iterative map shows a triangular object which is less dense than the previous one which results in a higher value of the fitness function $F(\phi)$.

According to [2] $d = 2$ is a critical space dimension above which coherent (quasiperiodic)oscillations can be observed. But surprisingly the GA has discovered rules like ϕ_a^2 and ϕ_b^2 which shows a QP3. It might be that this quasiperiodic behavior is only metastable or only seen on small lattices, so it is interesting to see how the attractor scales with the lattice size, but a more complete study of the existence of this behavior in the thermodynamic limit is out of the scope of this work.

The return plot and the time series of the concentration for rule ϕ_a^2 are plotted in Figure 3 for three different lattices sizes, where it can be seen a fuzzy limit cycle that looks like previous studied CAs in $d = 3$ with QP3 behavior. The amplitude of the oscillations and the fitness function shrink as lattice size increases, see Figure 4, however the attractor is found to become thinner and sharper the larger the system. The largest runs were made on lattices of 10^6 cells over 25000 time steps, the behavior is observed in Figure 5 where starting from a random initial condition, the concentration oscillates among three different branches that can be clearly distinguished initially, but as the system evolves the three branches mix irregularly and the size of the attractor diminishes.

The stability of the attractor against external noise is shown in Figure 6 that plots the fitness function $F(\phi)$ versus the amount of noise η in a log-log plot. Noise is introduced in the system by flipping a fraction η of randomly selected spins before each time the rule is applied. In d=2 the attractor is very sensitive to very small amounts of noise which produces the shrinking of the attractor from the triangular object initially obtained to a noisy cloud of points.

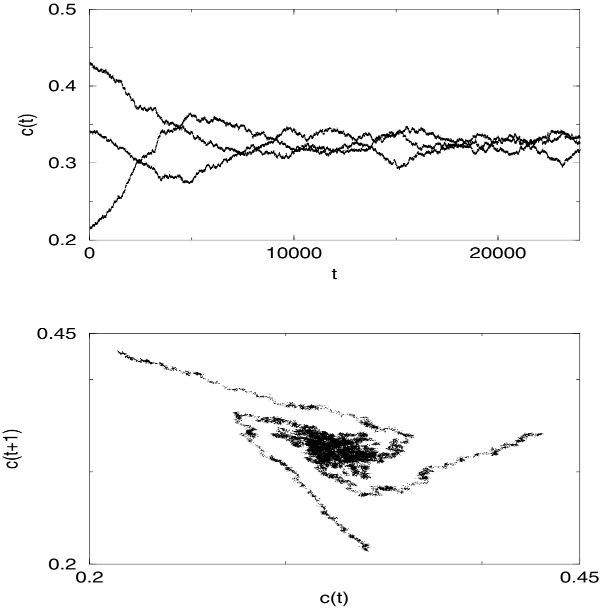


Fig. 5. The time series of the concentration and the iterative map for rule ϕ_a^2 in d=2. Lattice of 10^6 cells.

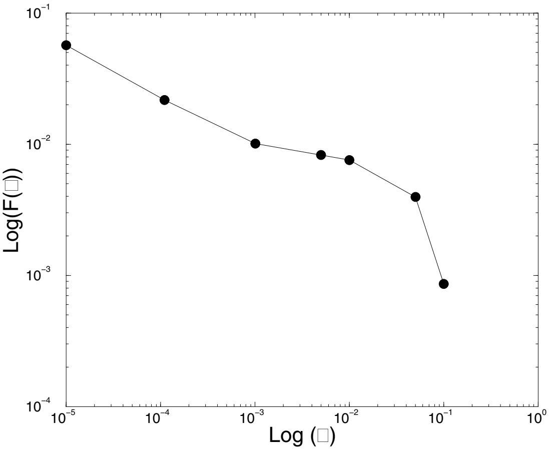


Fig. 6. Log-log plot of the Fitness function for rule ϕ_a^2, versus the amount of noise. Lattice size of 100^2 cells.

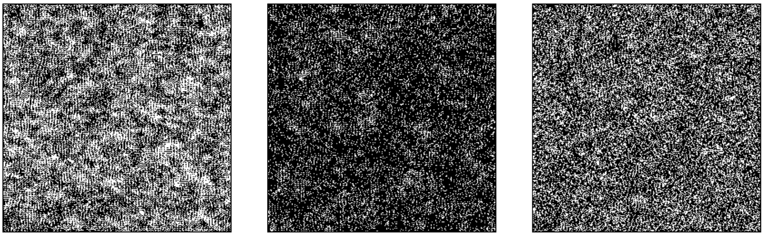


Fig. 7. Snapshots of rule ϕ_a^2 running on a d=2 lattice size of 10^6 cells. Three consecutive time steps are shown. Black cells correspond to state 1 and white cells to state 0.

Finally Figure 7 shows three consecutive snapshots of the evolution of rule ϕ_a^2. The spatial structure is very homogeneous with some spot-like inhomogeneities but there is not any propagating structures such as "gliders" observed in others low-dimensional systems.

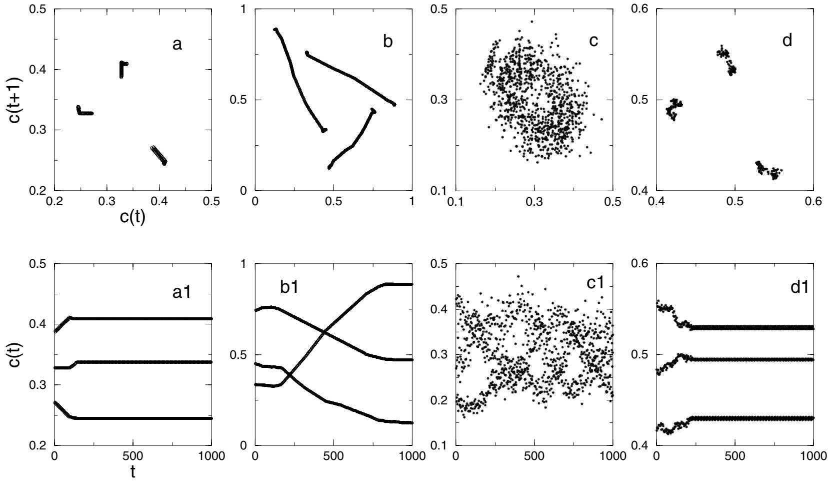


Fig. 8. The iterative map and the corresponding time series of the concentration for the best evolved rules in $d = 1$ **(a)** ϕ_a^1. **(b)** ϕ_b^1. **(c)** ϕ_c^1 .**(d)** ϕ_d^1. Lattice size is 2000 cells. Transient of 1000 time steps.

3.2 Experiments in d=1

In $d = 1$ the GA is able to find many rules with the desired behavior, about 30% of the runs ended up with a rule that showed a P3 collective behavior. Figure 8 shows the iterative map and the time series of the concentration for some of the best evolved rules ϕ_a^1, ϕ_b^1, ϕ_c^1 and ϕ_d^1.

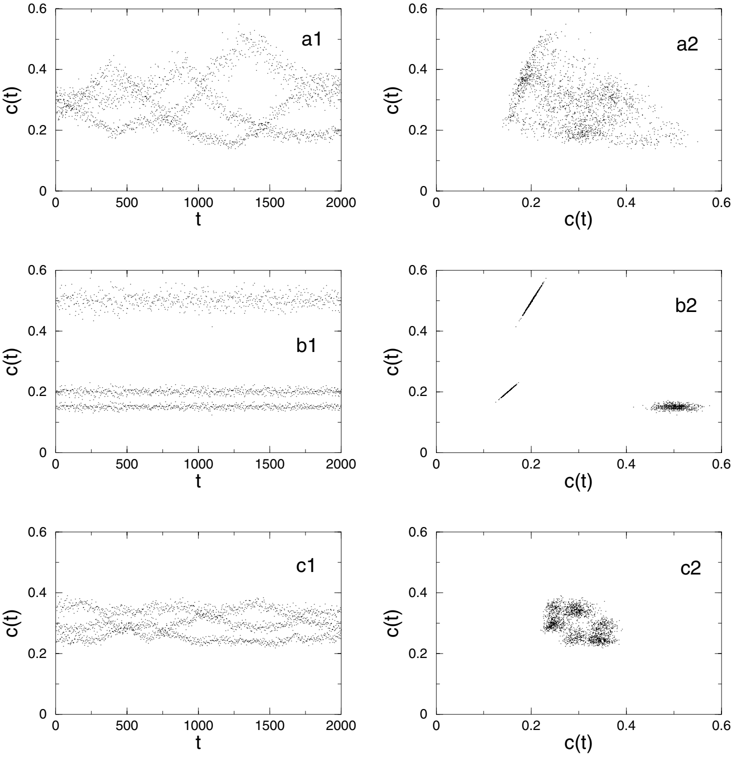


Fig. 9. The time series of the concentration and the iterative map for rule ϕ_c^1, for three different lattice sizes: (a) N=3000, (b) N=5000 y (c) N=10000. The transient time of $5 \cdot N$

Rules ϕ_a^1, ϕ_b^1 and ϕ_d^1 shows a P3 behavior that can be seen as three clouds of points in the iterative map and in the time series of the concentration as three branches that interact and mix among them. The attractor of rule ϕ_c^1 (Figure 8-c) is more complex and consists of a fuzzy cycle. The time series of the concentration shows three branches that can be distinguished clearly. The triangular shape of the attractor is maintained as the lattice size increases up to $L = 10^4$ and the time series of the concentration shows a quasiperiod-3 or a P3 collective behavior as can be seen in Figure 9. However the QP3 behavior seems to be metastable because after a long time the final state is a P3. For example Figure 10 shows the time series of the concentration of ϕ_c^1 during 10^4 time steps where it can be observed a sharp transition from a QP3 to a P3 collective behavior.

Under the fitness function $F(\phi)$ in $d = 1$ the evolutionary process selects rules that starting from a random initial condition synchronize the whole system to a three-state cycle. To see how such a synchronization is obtained we

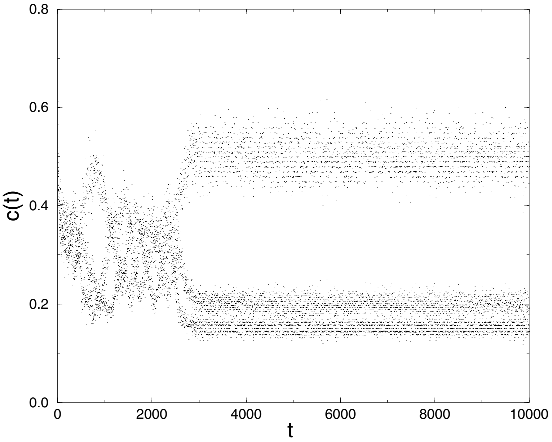


Fig. 10. The time series of the concentration for rule ϕ_c^1 starting from a random initial concentration of 0.5. It can be observed that the quasiperiodic regimen is metastable. Lattice size is 2000 cells.

use the tools of the "computational mechanics" developed by Crutchfield and Hanson [6]. This point of view describes the computation embedded in the CA space-time configuration in terms of domains, defects and defect interactions. Here we describe two of the best evolved rules: ϕ_a^1 and ϕ_c^1. Figure 11-a shows a space-time diagram of rule ϕ_a^1. Time starts on $t = 0$ and goes from up down and space is displayed on the horizontal axis. Figure 11-a shows patterns in which there is an easily recognized spatio-temporally periodic background -the domain- on which some dislocations move. In the simplest case a domain consist of a set of cells in the space-time diagram that are always repeated; for example, the domain for rule ϕ_a^1 is shown in Table 2. If over a long time all the cells of the space-time diagram are in the domain then the concentration of activated cells will be oscillating among three values 1/2, 1/3 and 1/6. Displacements of the domain shown in Table 2 along the temporal axis give place to other domains and then at the boundaries between them there are some defects cells. For rule ϕ_a^1 these defects cells are spatially localized and time-invariant and they are considered to be "particles".

Table 2. Domain for rule ϕ_a^1.

1	1	0	1	0	0
1	1	0	0	0	0
0	1	0	0	0	0
1	0	0	1	1	0
0	0	0	1	1	0
0	0	0	0	1	0

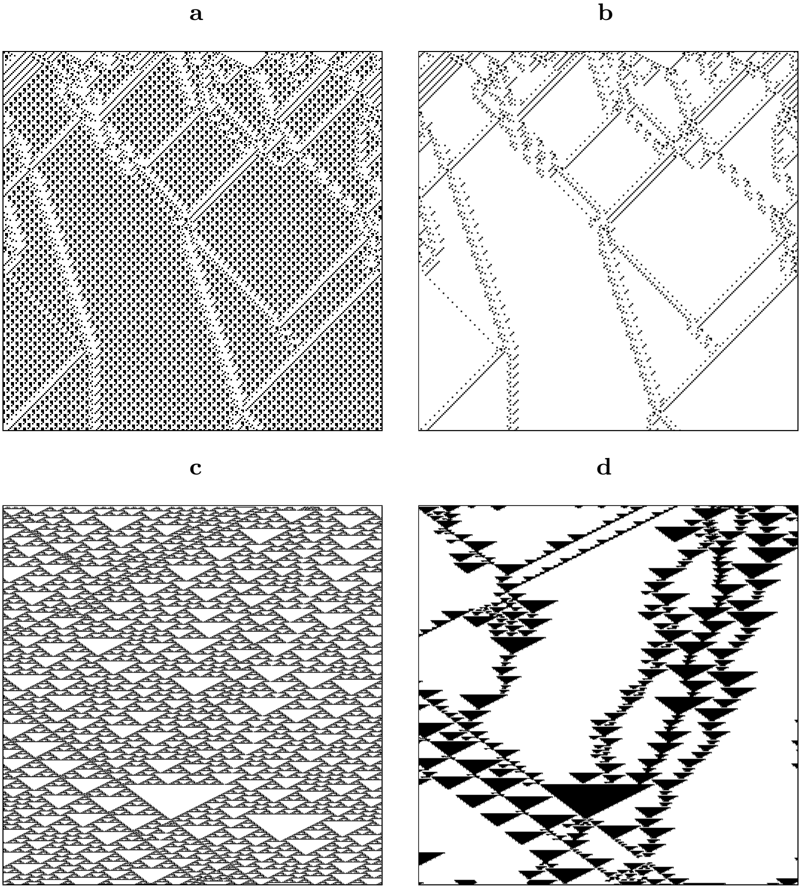


Fig. 11. Space-Time diagram and filtered diagram of rules: **(a)-(b)** ϕ_a^1, **(c)-(d)** ϕ_c^1. It is shown a window of 256x256 points. Black represents a cell in state 1 while white is a cell in state 0.

The corresponding filtered space-time diagram of Figure 11-a is shown in Figure 11-b where cells in a domain are plotted in white. The filtered diagrams reveal defect cells that interact among them and are transmitted from distant parts of the lattice until the synchronization of the whole system is obtained.

Figure 11-c shows a space-time diagram of rule ϕ_c^1. The domain for ϕ_c^1 is not a regular pattern as it is for ϕ_a^1, but a chaotic pattern of triangles of different sizes. To construct a filter for ϕ_c^1 we left the rule to run for a long time (10^5 time steps) and then we measure the frequencies of appearance of spatial sequences of symbols of a given length (we use words of 6 bits). Then the sequences of symbols with small frequencies are considered as defects. Figure 11-d shows the filtered diagram corresponding to Figure 11-c. It appears that some defects in ϕ_c^1 move randomly which suggests a stochastic model of dislocation motion like it happens for some chaotic CA [5,4].

Finally Figure 12 shows in a log-log plot the decaying of the density of defects ρ_d, where for the chaotic rule ϕ_c^1 only the sequence of symbols 010110 is counted and its average density decreases due to recombinations with $t^{-0.39}$. For rules ϕ_a^1, ϕ_b^1 and ϕ_d^1 the density of defects ρ_d decreases as the time grows until a non-zero asymptotic value.

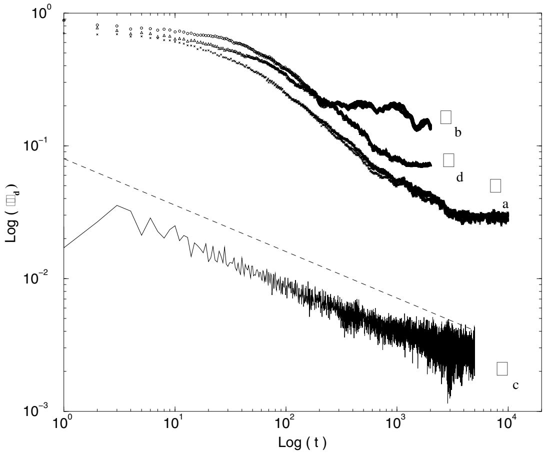


Fig. 12. Log-log plot of the density of defects versus time. Data points are averaged over 10 different experiments in a lattice of 2000 cells. For rule ϕ_c^1 the experiments were done in a lattice of 10^5 cells and only defects of type "010110" were counted. The slope of the dashed line is -0.39.

4 Conclusion

Non-trivial collective behavior in cellular automata (CA) is a striking phenomena not well understood yet. In this work we presented results in which a genetic algorithm (GA) is used to evolve one and two-dimensional binary-state CA to perform a non-trivial collective task in which the concentration of activated cells oscillates among three different values. We found that with an appropriate fitness function the artificial evolutionary process is able to detect several CA rules with the desired behavior.

In $d = 2$ the stability of the collective behavior is very sensitive to small amounts of noise and as the lattice size increases the size of the attractor diminishes. In $d = 1$ the GA detects some rules that organize the space-time diagram in regular patterns. In this case the mechanism used to synchronize the entire system is the propagation and interaction of particles. Particles were also shown to be central to the GA's solutions for the density and synchronization tasks. However it is worth pointing out that for the task studied here the final state the CA must reach is not necessarily a periodic one. Some evolved rules show chaotic patterns and the final state is obtained through the interactions among particles and random-like defects, but this is a new mechanism that will require a more detailed study.

Acknowledgments. Many thanks to the EvCA group of Santa Fe Institute for their assistance and suggestions.This work was supported by Grant No.PR2002-0176 of the Spanish Ministerio de Educacion, Cultura y Deporte.

References

1. C. H. Bennet, G. Grinstein, Yu He, C. Jayaprakash, and D. Mukamel. Stability of temporally periodic states of classical many-body systems. *Phys. Rev. A*, 41:1932–1935, 1990.
2. H. Chaté, G. Grinstein, and P. Lei-Hang Tan. Long-range correlations in systems with coherent(quasi)periodic oscillations. *Phys.Rev.Lett.*, 74:912–915, 1995.
3. H. Chaté and P.Manneville. Collective behaviors in spatially extended systems with local interactions and synchronous updating. *Progress Theor. Phys.*, 87(1):1–60, 1992.
4. J. P. Crutchfield and J. E. Hanson. Attractor vicinity decay for a cellular automaton. *Chaos*, 3(2):215–224, 1993.
5. P. Grassberger. New mechanism for deterministic diffusion. *Phys. Rev. A*, 28:3666–3667, 1983.
6. J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata: An example. *Physica D*, 103:169–189, 1997.
7. J. Hemmingsson. A totalistic three-dimensional cellular automaton with quasiperiodic behaviour. *Physica A*, 183:225–261, 1992.
8. F. Jiménez-Morales. Evolving three-dimensional cellular automata to perform a quasiperiod-3(p3) collective behavior task. *Phys. Rev. E*, 60(4):4934–4940, 1999.
9. F. Jiménez-Morales and J. J. Luque. Collective behaviour of a probabilistic cellular automaton with two absorbing phases. *Phys. Lett. A*, 181:33–38, 1993.

Artificially Evolved Asynchronous Cellular Automata for the Density Task

Marco Tomassini and Mattias Venzi

Computer Science Institute, University of Lausanne,
1015 Lausanne, Switzerland.

Abstract. In this paper we study the evolution of asynchronous automata for the density task. We compare our results with those obtained for synchronous automata and we describe the influence of various asynchronous update policies on the computational strategy. We also investigate how synchronous and asynchronous cellular automata behave under noisy conditions and show that asynchronous ones are more fault-tolerant.

1 Introduction

Cellular automata(CAs) are well-known and widely used sytems [10]. In this work we concentrate on the customary simultaneous i.e, *synchronous* updating of the CA cells. This update mode is conceptually simple and it is easier to deal with in mathematical terms. However, perfect synchronicity is only an abstraction: if CAs are to model physical or biological situations or are to be considered physically embodied computing machines then the synchronicity assumption is untenable. In fact, in any spatially extended system signals cannot travel faster than light. Hence, it is impossible for a signal emitted by a global clock to reach any two computing elements at exactly the same time. In this study we relax the synchronicity constraint and work with various kinds of *asynchronous* CA updating modes on a well-known problem: density classification by a CA. The few existing studies on asynchronous CAs have shown that asynchronous update often gives rise to completely different time evolutions for the CA. For instance, cyclic attractors are no longer possible and generally there is a loss of the rich structures commonly found in synchronous CAs (see e.g. [1,4]).

In systems with many components faulty behavior is a common occurrence. In the second part of the paper we compare the dynamics of synchronous and asynchronous CAs for the density task in the presence of random errors in order to ascertain their respective robustness.

The paper is organized as follows. The following section 2 summarizes definitions and facts about standard CAs and their asynchronous counterparts. Section 3 deals with the artificial evolution of asynchronous CAs for the density task and compares their behavior and solution strategies with those of known synchronous CAs. In section 4 we study their fault-tolerance aspects. Section 5 presents our conclusions and hints to further work.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 44–55, 2002.

2 Synchronous and Asynchronous Cellular Automata

Cellular automata are discrete dynamical systems. A standard cellular automaton consists of an array of cells, each of which can be in one of a finite number of possible states, updated synchronously in discrete time steps, according to a local, identical interaction rule. Here we will only consider Boolean automata for which the cellular state $s \in \{0, 1\}$. The state of a cell at the next time step is determined by the current states of a surrounding neighborhood of cells. The regular cellular array (grid) is d-dimensional, where $d = 1, 2, 3$ is used in practice; in this paper we shall concentrate on $d = 1$, i.e. one-dimensional grids. The identical rule contained in each cell is usually specified in the form of a rule table with an entry for every possible neighborhood configuration of states. For one-dimensional CAs, a cell is connected to r local neighbors (cells) on either side; each cell thus has $2r + 1$ neighbors. When considering a finite grid, spatially periodic boundary conditions are frequently applied, resulting in a circular grid for the one-dimensional case. The term *configuration* refers to an assignment of ones and zeros to each cell at a given time step.

There are many ways for sequentially updating the cells of a CA (for an excellent discussion, see [7]). The most general one is *independent random ordering* of updates in time, which consists in randomly choosing the cell to be updated next, with replacement. This corresponds to a binomial distribution for the update probability, the limiting case of which for large n is the Poisson distribution (n is the number of cells in the grid).

For comparison purposes we also employ two other update methods: *fixed random sweep* and *random new sweep* ([7]). In the fixed random sweep update, each cell to be updated next is chosen with uniform probability without replacement; this will produce a certain update sequence $(c_1^j, c_2^k, \ldots, c_n^m)$, where c_q^p means that cell number p is updated at time q and $(j, k, \ldots, m)$ is a permutation of the n cells. The same permutation is then used for the following update cycles. The random new sweep method is the same except that each new sweep through the array is done by picking a different random permutation of the cells. A time step consists in updating n times, which corresponds to updating all the n cells in the grid for fixed random sweep and random new sweep, and possibly less than n cells in the binomial method, since some cells might be updated more than once. It should be noted that our chosen asynchronous updating being non-deterministic, the same CA may reach a different configuration after n time steps on the same initial distribution of states, which is not the case for synchronous CAs, since there is a single possible sequence of configurations for a synchronous CA for a given initial configuration of states.

3 Evolving 1-D Asynchronous CAs for the Density Task

In this section we define the density task and we describe how asynchronous CAs for performing this task can be evolved by genetic algorithms (GA).

The density task is a prototypical computational task for CAs that has been much studied due to its simplicity and richness of behavior. For one-dimensional finite CA of size n (with n odd for convenience) it is defined as follows: the CA must relax to a fixed-point pattern of all 1s if the initial configuration of states contains more 1s than 0s and, conversely, it must relax to a fixed-point pattern of all 0s otherwise, after a number of time steps of the order of the grid size. This computation is trivial for a computer having a central control. However, the density task is non-trivial for a small radius 1-D CA since such a CA can only transfer information at finite speed relying on local information exclusively, while density is a global property of states configuration [6]. It has been shown that the density task cannot be solved perfectly by a uniform, two-state CA with finite radius [5], although a slightly modified version of the task can be shown to enjoy perfect solution by such an automaton [2]. In general, given a desired global behavior for a CA, it is extremely difficult to infer the local CA rule that will give rise to the emergence of the computation sought. Since exhaustive evaluation of all possible rules is out of the question except for elementary ($d = 1, r = 1$) automata, one possible solution of the problem consists in using evolutionary algorithms, as proposed by Mitchell *et al.* [6] for uniform and synchronous CAs and by Sipper for non-uniform (the rules need not be all the same) ones [8].

3.1 Artificial Evolution of Cellular Automata

Here we use a genetic algorithm similar to the one described in [6] for synchronous CAs, with the aim of evolving asynchronous CAs for the density task. Each individual in the population represents a candidate rule and is represented simply by the output bits of the rule table in lexicographic order of the neighborhood (see section 2). Here $r = 3$ has been used, which gives a chromosome length of $2^{2r+1} = 128$ and a search space of size 2^{128}, far too large to be searched exhaustively. The population size is 100 individuals, each represented by a 128-bit string, initially randomly generated from a uniform density distribution over the interval $[0, 1]$. The fitness of a rule in the population has been calculated by randomly choosing 100 out of the 2^n possible initial configurations (IC) with uniform density in the same manner as for the initial population and then iterating the rule on each IC for $M = 2n$ time steps, where $n = 149$ is the grid size. The fitness of the rule is the fraction of ICs for which the rule produced the correct fixed point, given the known IC density. At each generation a different set of ICs is generated for each rule. After ranking the rules in the current population according to their fitness, the 20% top rules are copied in the next population without change. The remaining 80 rules are generated by crossover and mutation. Crossover is single-point and is performed between two individuals randomly chosen from the top 20 rules with replacement and is followed by single-bit mutation of the two offspring. The best 80 rules after the application of the genetic operators enter the new population.

The performance of the best rules found at the end of the evolution is evaluated on a larger sample of ICs and it is defined as the fraction of correct classifications over 10^4 randomly chosen initial configurations. Moreover, the ICs are sampled according to a binomial distribution, (i.e. each bit is independently drawn with probability 1/2 of being 0). Clearly, this distribution is strongly peaked around $\rho_0 = 1/2$ and thus it makes a much more difficult case for the CA (ρ_0 is the density of 0s in the initial configuration).

Due to the high computational cost, we have performed 15 runs, each lasting for 100 generations, for each of the asynchronous update policies. This is not enough to reach very good results, but it is sufficient for studying the emergence of well-defined computational strategies, which has been our main objective here.

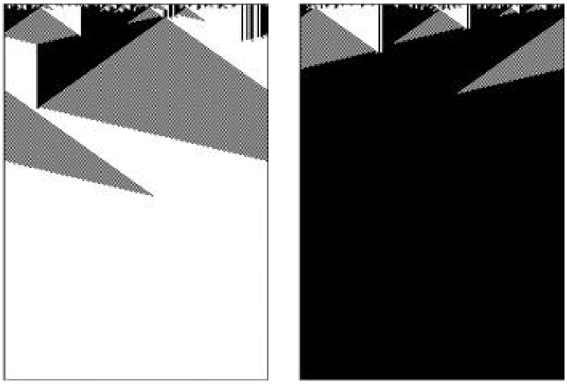
3.2 Evolutionary Dynamics and Results: Synchronous CAs

Mitchell and co-workers performed a number of studies on the emergence of synchronous CAs strategies for the density task during evolution (see e.g. [6], where more details can be found). In summary, these findings can be subdivided into those pertaining to the evolutionary history and those that are part of "final" evolved automata. For the former, they essentially observed that, in successful evolution experiments, the fitness of the best rules increase in time according to rapid jumps, giving rise to what they call "epochs" in the evolutionary process. Each epoch corresponds roughly to a new, increasingly sophisticated solution strategy. Concerning the final CA produced by evolution, it was noted that, in most runs, the GA found non-sophisticated strategies that consisted in expanding sufficiently large blocks of adjacent 1s or 0s. This "block-expanding" strategy is unsophisticated in that it mainly uses local information to reach a conclusion. As a consequence, only those IC that have low or high density are classified correctly since they are more likely to have extended blocks of 1s or 0s. In fact, these CAs have a performance around 0.6. However, some of the runs gave solutions that presented novel, more sophisticated features that yielded better performance (around 0.77) on a wide distribution of IC. These new strategies rely on travelling signals that transfer spatial and temporal information about the density in local regions through the lattice. An example of such a strategy is given in figure 1, where the behavior of the so-called GKL rule is depicted [6]. The GKL rule is a hand-coded one but its behavior is similar to that of the best solutions found by evolution.

3.3 Evolutionary Dynamics and Results: Asynchronous CAs

For the evolution of asynchronous CAs we have used GA parameters as described in section 3.1. As expected, the evolved asynchronous CAs find it more difficult to solve the density task due to their stochastic nature. In fact, a given CA could classify the same initial configuration in a different way depending on the update sequence, and indeed, although synchronous CAs are delocalized systems, a kind of central control is still present, because of the presence of a global clock. This is not the case for asynchronous CAs. Nevertheless, for all the asynchronous update methods CAs with fair capabilities were evolved. In table 1 we list the best rules found by the GA for the three update modes. We note that the performance of the solutions are lower than those for synchronous CAs.

The behavior of the CAs evolved with all three asynchronous updating modes were very similar both from the point of view of the performance, as well as



(a) (b)

Fig. 1. Space-time diagram for the GKL rule. CA cells are depicted horizontally, while time goes downward. The 0 state is depicted in white; 1 in black. The density of zeros ρ_0 is 0.476 in (a) and $\rho_0 = 0.536$ in (b).

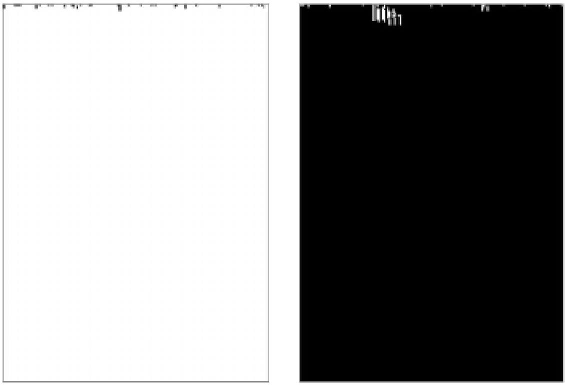
Table 1. Performance of the best evolved asynchronous rules calculated over 10^4 binomially distributed initial configurations. Rule numbers are in hexadecimal.

Update Mode	**Rule**	**Performance**
IndependentRandom	00024501006115AF5FFFBFFDE9EFF95F	67.2
FixedRandomSweep	114004060202414150577E771F55FFFF	67.7
RandomNewSweep	00520140006013264B7DFCDF4F6DC7DF	65.5

from the point of view of the solution strategies that evolved. Since independent random ordering, i.e. uniform update, is in some sense the more natural, we will describe it here, although most of what we say also applies to the other two methods. During most evolutionary runs we observed the presence of periods in the evolution in which the fitness of the best rules increase in rapid jumps. These "epochs" were observed in the synchronous case too (see section 3.2) and correspond to distinct computational innovations, i.e. to major changes in the strategies that the CA uses for solving the task.

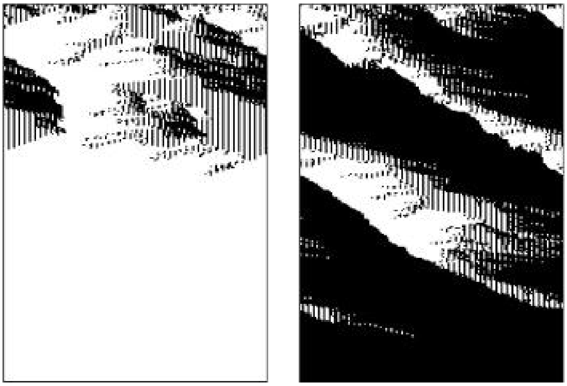
In epoch 1 the evolution only discovers local naive strategies that only work on "extreme" densities, (i.e. low or high) but most often not on both at the same time. Fitness is only slightly over 0.5. In the following epoch, epoch 2, rules specialize on low or high densities as well and use unsophisticated strategies, but now they give correct results on both low and high densities. This can be seen, for instance, in figure 2.

In epoch 3, with fitness values between 0.80 and 0.90, one sees the emergence of block-expanding strategies, as in the synchronous case, but more noisy. Moreover, narrow vertical strips appear (see figure 3).



(a) (b)

Fig. 2. Space-time diagrams for an epoch 2 rule. (a) $\rho_0 = 0.194$, (b) $\rho_0 = 0.879$. The rule only classifies low or high densities.



(a) (b)

Fig. 3. Space-time diagrams for an epoch 3 rule. (a) $\rho_0 = 0.489$, (b) $\rho_0 = 0.510$. Block-expanding and vertical strips make their appearance.

The following, and last, epoch 4 sees the refinement of the vertical strips strategy with fitness above 0.9 and steadily increasing. The propagating patterns become less noisy and the strategy is little affected by the intrinsic stochasticity of the update rule. Figure 4 illustrates the best solution found by evolution at the end of epoch 4. The "zebra-like" moving patterns, which represent the most efficient strategies for evolved asynchronous automata, are different from those found in the synchronous case. In fact, the asynchronous updating modes have the effect of destroying or delaying the propagation of the long-range transversal signals that carry information in the synchronous case (see figure 1). Thus, the CA expands blocks of zeros and ones, which collide and annihilate. Small-block propagate in time, which gives the characteristic zebra-like patterns. These strips are stable and propagate further to the right or to the left.

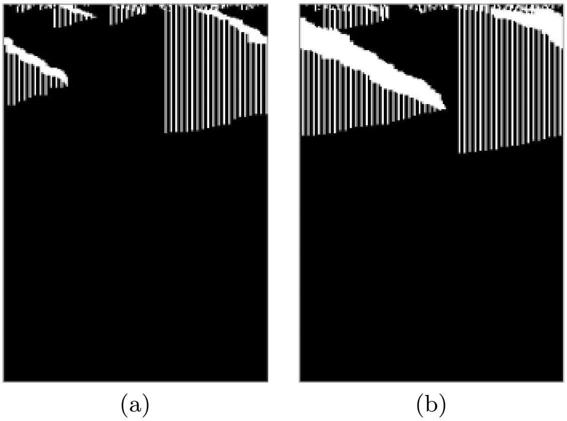


Fig. 4. Space-time diagrams for the best asynchronous rule found. The density $\rho_0 = 0.55$ in both (a) and (b) and the initial state configuration is the same. The different time evolution points out the stochasticity of the updating policy.

3.4 Scalability of Evolved Automata

It is interesting to study the behavior of the evolved density classification rules as the number n of cells in the grid increases. If the solution is scalable then, as n increases, performances should remain more or less the same, although the CA has been evolved for $n = 149$. Crutchfield and Mitchell [3] found that their synchronous evolved CAs do not scale very well with the grid size, as their performance decreases slightly when n goes from 149 to 599, and to 999. We have found that the behavior of the best evolved asynchronous CAs are less influenced by the grid size than their synchronous counterparts. As an example, in figure 5 we plot the performance of the best evolved CA with the uniform update method as a function of the density for three lattice sizes, $n = 149, 599, 999$.

It is apparent that the larger grid sizes have a smaller region of error, i.e. the performance is better for a larger interval around $\rho_0 = 0.5$, except in the immediate vicinity of 0.5, where the performance is slightly less. On the whole, it is clear that the performance stays constant or even it increases slightly with increasing grid size.

4 The Effect of Noise

There are at least two possible sources of indeterminism in CAs: the first is random asynchronous updating, as explained in the previous sections; the second

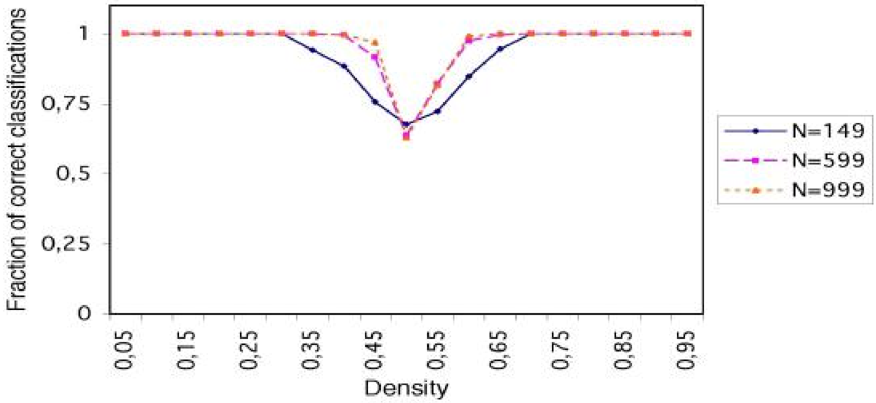


Fig. 5. Experimental performance of the best asynchronous evolved CA as a function of the configuration density for three values of the lattice size.

is faulty functioning of the rules, or of the cells, or both. In this section we explore in some detail the behavior of evolved CAs subject to the second type of noise. This kind of considerations will be important in the future, when it is likely that self-organizing computational systems composed of an enormous number of parts will be used in order to design systems that are partially or totally tolerant to such faults. Of course, the comparatively small and simple systems studied here are only toys with respect to real future computing machines. Nonetheless, their study is certainly a worthy first step. It should be noted that we will not try to correct the errors, which is an important but very complicated issue. Rather, we will focus on the self-recovering capabilities of the systems under study.

We will study two kinds of perturbations and their effect on the density task: the first is *probabilistic updating* and the second is *intermittent* faults. They are defined as follows:

- probabilistic updating: a CA rule may yield the incorrect output bit with probability p_f, and thus the probability of correct functioning will be $(1-p_f)$. Futhermore, we assume that errors are uncorrelated.
- intermittent faults: at time t a given cell has a certain probability of being inactive, that is of keeping its current state.

For probabilistic updating usually two initially identical copies of the system are used. One evolves undisturbed with $p_f = 0$, while the second is submitted to a non-zero probability of fault (see e.g. [9] and references therein, where the case of synchronous, non-uniform CAs is examined). Figure 6 depicts the typical behavior of the best evolved synchronous rule under noise. We see that even for relatively low ($p_f = 0.001$) values of the fault probability the CA does not work correctly. For higher values ($p_f = 0.01$) the CA is so perturbed that it cannot accomplish the task any longer. Clearly, the propagation of transversal signals is more and more hindered as the automaton becomes more noisy.

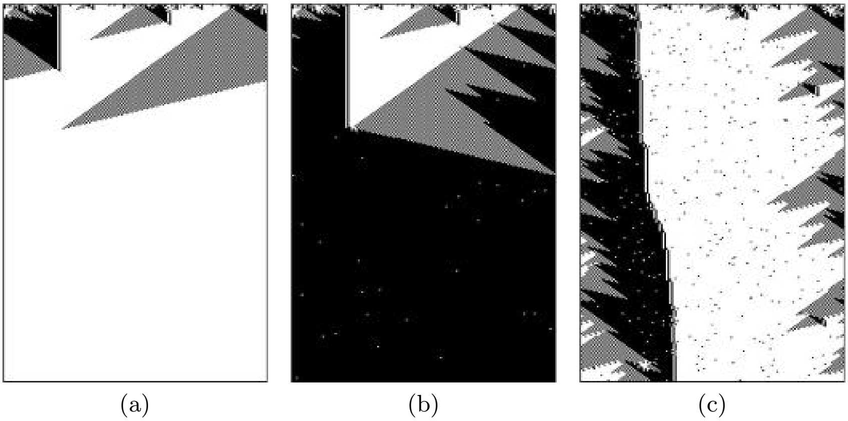
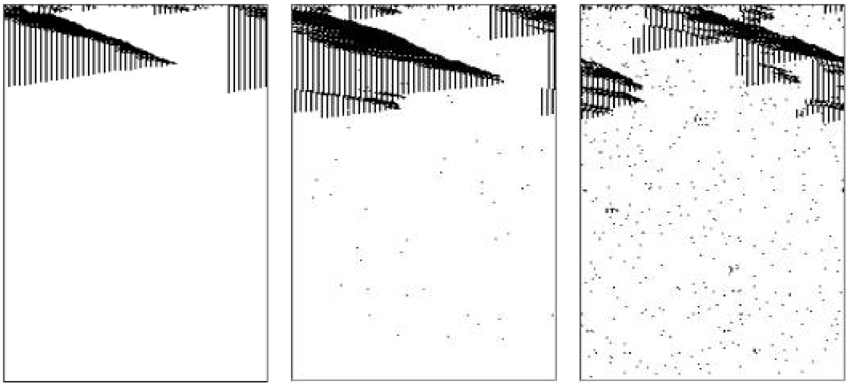


Fig. 6. Typical behavior of the EvCA CA under probabilistic updating. The density ρ_0 is 0.416 and the probabilities of fault p_f in (a), (b), and (c) are, respectively, 0, 0.001 and 0.01.

The following figure 7 shows the same evolution for the best evolved asynchronous CA using the uniform choice update policy. The visual inspection already confirms that the CA is much less perturbed by random noise in the rules. Even relatively high levels of faults do not prevent the CA from recovering and finding the correct classification in many cases. This is clearly due to the fact that asynchronous CAs were evolved in a noisy environment (the randomness associated with the sequential update order) and thus, to some extent, this allows them to cope better with errors.

Although the previous examples are single cases, they are typical of what happens. The following figure 8 shows an histogram of the ratio of the success rate of the best evolved synchronous CA and of two evolved asynchronous CAs as a function of the fault probability, with respect to the unperturbed versions. Each CA has been tested on 1000 IC. The ranking is relative, since we only kept the successful runs of the unperturbed automata to calculate the ratio. One sees clearly that, already for $p_f = 1.0 \times 10^{-4}$, the synchronous CA starts to degrade, while both asynchronous versions maintain good performance, especially the uniform choice one, for values of p_f up to the order of 10^{-3}.

In the case of intermittent faults, we have tested 1000 IC for each of a number of probability values of cell inactivity. We have used three CA rules: the best evolved synchronous CA (EvCA [6]), the GKL rule, and the best evolved uniform choice asynchronous automaton. The results are reported in figure 9. We observe that for low values of the fault probability the three rules are almost equivalent in that they keep a very good level of performance. However, as soon as the



(a) (b) (c)

Fig. 7. Typical behavior of an asynchronous CA under probabilistic updating. The density ρ_0 is 0.416 and the probabilities of fault p_f in (a), (b), and (c) are, respectively, 0, 0.001 and 0.01.

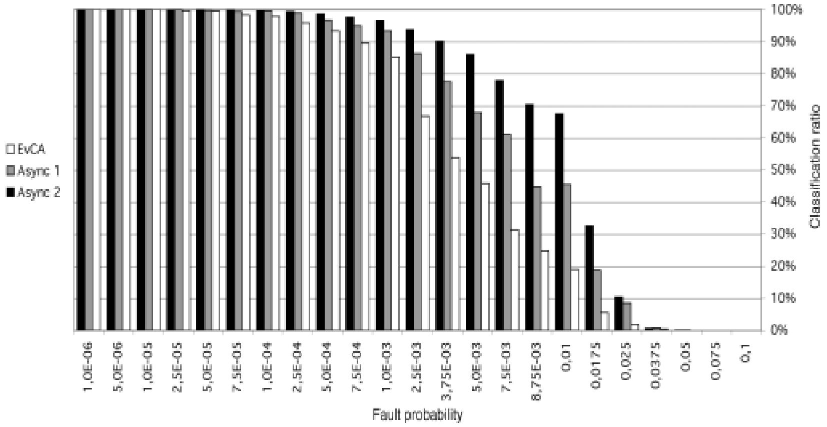


Fig. 8. Histogram representing the percentage of success of three noisy automata with respect to the unperturbed versions. The probability of fault is on the horizontal axis. Async 1 (gray bar) is the new random sweep automaton, while Async 2 (black bar) corresponds to the uniform choice CA. EvCA (white bar) is the best evolved synchronous CA.

probability exceeds 0.01, the two synchronous rules collapse, especially GKL, while the asynchronous rule does not seem to suffer much from the increasing level of noise and keeps a good performance level in the whole range, except for high probability values (note the logarithmic scale on the horizontal axis).

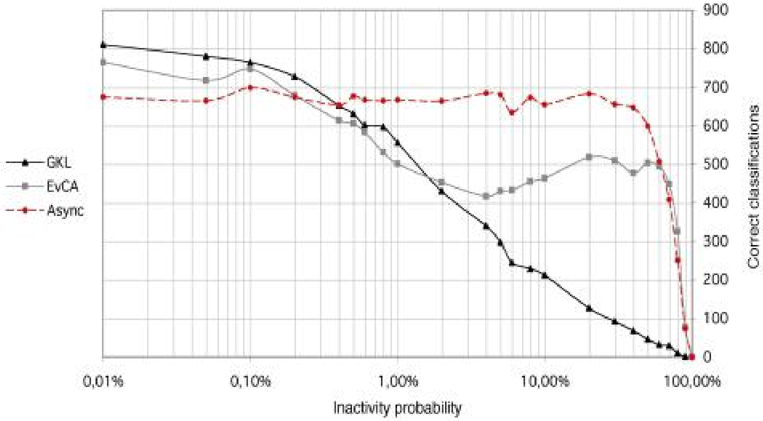


Fig. 9. Number of correct classification as a function of inactivity probability. The curves refer to the GKL rule and to two asynchronous CAs.

Once again, the results of this section confirm that asynchronous CAs degrade much more gracefully than synchronous ones in noisy environments. They thus intrinsically offer more resilience and robustness.

5 Conclusions

In this work we have shown that physically more realistic asynchronous CAs of various kinds can be effectively evolved for the density task using genetic algorithms, although their performance is lower than that obtained by evolved synchronous CAs. We have also shown that the computational strategies discovered by the GA in the asynchronous case are different from those of synchronous CAs due to the presence of a stochastic component in the update. This very reason makes them more resistant to changes in the environment and thus potentially more interesting as computational devices in the presence of noise. In the same vein, they seem to have better scalability to larger computational grids than evolved synchronous CAs. Other important aspects that we are studying but are not included here are further investigations into their fault-tolerance properties.

Acknowledgment. We acknowledge the Swiss National Fund for Scientific Research for financial support under the grant 21-58893.99.

References

1. H. Bersini and V. Detour. Asynchrony induces stability in cellular automata based models. In R. A. Brooks and P. Maes, editors, *Artificial Life IV*, pages 382–387, Cambridge, Massachusetts, 1994. The MIT Press.

2. M. S. Capcarrere, M. Sipper, and M. Tomassini. Two-state, r=1 cellular automaton that classifies density. *Physical Review Letters*, 77(24):4969–4971, December 1996.
3. J. P. Crutchfield and M. Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Sciences USA*, 92(23):10742–10746, 1995.
4. T. E. Ingerson and R. L. Buvel. Structure in asynchronous cellular automata. *Physica D*, 10:59–68, 1984.
5. M. Land and R. K. Belew. No perfect two-state cellular automata for density classification exists. *Physical Review Letters*, 74(25):5148–5150, June 1995.
6. M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7:89–130, 1993.
7. B. Schönfisch and A. de Roos. Synchronous and asynchronous updating in cellular automata. *BioSystems*, 51:123–143, 1999.
8. M. Sipper. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach.* Springer-Verlag, Heidelberg, 1997.
9. M. Sipper, M. Tomassini, and O. Beuret. Studying probabilistic faults in evolved non-uniform cellular automata. *International Journal of Modern Physics C*, 7(6):923–939, 1996.
10. S. Wolfram. *Cellular Automata and Complexity.* Addison-Wesley, Reading, MA, 1994.

Evolving Cellular Automata as Pattern Classifier

Niloy Ganguly[1], Pradipta Maji[2], Sandip Dhar[2], Biplab K. Sikdar[2], and
P. Pal Chaudhuri[2]

[1] Computer centre, IISWBM, Calcutta, West Bengal, India 700073,
n_ganguly@hotmail.com
[2] Department of Computer Science & Technology, Bengal Engineering College (D U),
Howrah, West Bengal, India 711103
{pradipta,sdhar,biplab,ppc}@cs.becs.ac.in

Abstract. This paper reports a high speed, low cost pattern classifier based on the sparse network of Cellular Automata. High quality of classification of patterns with or without noise has been demonstrated through theoretical analysis supported with extensive experimental results.

1 Introduction

The internetworked society has been experiencing a explosion of data that is acting as an impediment in acquiring knowledge. The meaningful interpretation of these data is increasingly becoming difficult. Consequently, researchers, practitioners, entrepreneurs from diverse fields are assembling together to develop sophisticated techniques for knowledge extraction. Study of data classification models form the basis of such research. A classification model comprises of two basic operations - classification and prediction. The evolving CA based classifier proposed in this paper derives its strength from the following features:

- The special class of CA referred to as Multiple Attractor Cellular Automata ($MACA$) is evolved with the help of genetic algorithm to arrive at the desired model of CA based classifier.
- In the prediction phase the classifier is capable of accommodating noise based on distance metric.
- The classifier employs the simple computing model of three neighborhood Additive CA having very high throughput. Further, the simple, regular, modular and local neighborhood sparse network of Cellular Automata suits ideally for low cost $VLSI$ implementation.

The Cellular Automata (CA) preliminaries follows in the next section.

2 Cellular Automata Preliminaries

The fundamentals of Cellular Automata we deal with is reported in the book [1]. The classifier reported in this work has been developed around a specific class of CA referred to as *Multiple Attractor CA* ($MACA$).

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 56–68, 2002.

2.1 Multiple Attractor Cellular Automata

The state transition graph of an $MACA$ consists of a number of *cyclic* and *non-cyclic* states. The set of non-cyclic states of an $MACA$ forms inverted trees rooted at the cyclic states. The *cycles* are referred to as *attractors. Fig.1* depicts the state transition diagram of a 5-cell $MACA$ with four attractors {00000,00011,00110,00101} having self loop. In rest of the paper, by an attractor we will refer to a cycle of length 1. The states of a tree rooted at the cyclic state α forms the *α-basin.*

With reference to the state transition diagram of a CA, the *depth d* of the CA is the number of edges between a non-reachable state and the attractor. The depth d of the 5-cell $MACA$ of *Fig.1* is 3.

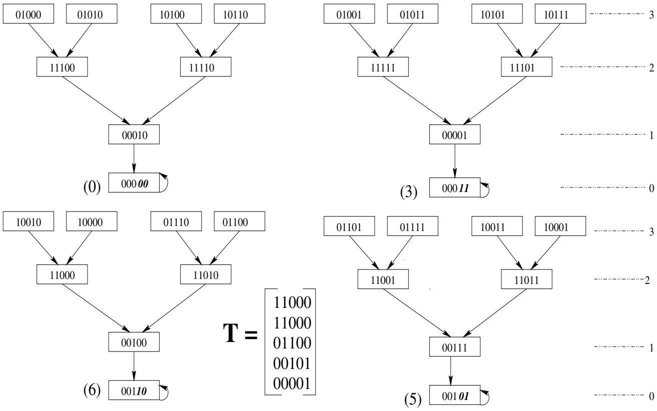


Fig. 1. State transition diagram of a 5-cell $MACA$

The detailed characterization of $MACA$ is available in [1]. A few fundamental results for an n-cell $MACA$ having k number of attractors is next outlined.

Result I: The characteristic polynomial of the $MACA$ is $x^{n-m}(1 + x)^m$, where $m = log_2(k)$.

Result II: The characteristic polynomial noted above can be also written in elementary divisor form as
$(1 + x)(1 + x) \cdot \cdot m \ times \quad x^{d_1}x^{d_2} \cdot \cdot x^{d_p}$ where $d_1 \geq d_2 \cdot \cdot \geq d_p$ and $d_1 + d_2 \cdot \cdot \cdot \cdot + d_p = n - m$.

Result III: The minimal polynomial of an $MACA$ is $x^{d_1}(1 + x)$, where depth $= d_1$.

Definition 1 *An m-bit field of an n-bit pattern set is said to be pseudo-exhaustive if all possible 2^m patterns appear in the set.*

Theorem 1 *[1] In an n cell $MACA$ with $k = 2^m$ attractors, there exists* m-bit *positions at which the attractors give pseudo-exhaustive 2^m patterns.*

Theorem 2 *[1] The* modulo-2 *sum of two states is the non-zero predecessor of* 0-state *(pattern with all 0's) if and only if the two states lie in the same $MACA$ basin.*

Example 1 *The example $MACA$ of* Fig.1 *is used to illustrate the above results.*

- *It is a 5-cell $MACA$ having 4 number of attractors and the depth of the $MACA$ is 3.*
- Result I: *The characteristic polynomial is $x^3 \cdot (1+x)^2$. Therefore, m=2. This is consistent with the result in the* Fig.1 *where attractor(k) is 4.*
- Result II: *The characteristic polynomial in elementary divisor form is $x^3 \cdot (1+x) \cdot (1+x)$.*
- Result III: *The minimal polynomial is $x^3 \cdot (1+x)$.*
- Result of Theorem 1: *In* Fig.1, *two least significant bit positions constitute the PEF.*
- Result of Theorem 2: *We take an attractor 00011 and any two states 11111, 11101 of the attractor basin. The* modulo-2 *sum of these two patterns is 00010 which is a state in $0-basin$. By contrast, if we take two states 00001 and 11000 belonging to two different attractor basins 00001 and 11000 respectively, their* modulo-2 *sum is 11011 which is a state in a non-zero attractor (00101) basin.*

2.2 *MACA* – As a Classifier

An n-bit $MACA$ with k-attractors can be viewed as a natural classifier (*Fig.2*). It classifies a given set of patterns into k-distinct classes, each class containing the set of states in the attractor basin.

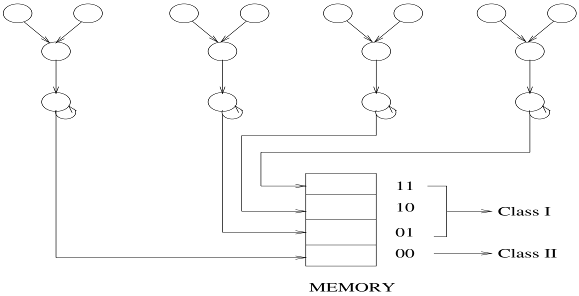


Fig. 2. $MACA$ based Classification Strategy

For an ideal classifier, to distinguish between two classes, we would need one bit. The classifier of *Fig 2* requires two bits to distinguish between two classes which is a memory overhead of the design. A k-attractor two class classifier needs $log_2(k)$ bits.

To classify pattern set into two classes, we should ideally find an $MACA$ with two attractor basins - each basin having the members of the specific class. Even if this ideal situation is not attained, the algorithm should design a $MACA$ based classifier having minimum number of attractor basins - while one subset of basins accommodates the elements of one class, the remaining subset houses the elements of the other class.

MACA based two class classifier. The design of the $MACA$ based Classifier for two pattern sets P_1 and P_2 should ensure that elements of one class (say P_1) are covered by a set of attractor basins that do not include any member from the class P_2. Any two patterns $a \in P_1$ and $b \in P_2$ should fall in different attractor basins. According to *Theorem 2*, the pattern derived out of *modulo-2* sum of a and b $(a \oplus b)$ should lie in a non-zero attractor basin. Let X be a set formed from *modulo-2* sum of each member of P_1 with each member of P_2 that is, $X = \{x_l \mid x_l = (a_i \in P_1) \oplus (b_j \in P_2) \, \forall_{i,j}\}$. Therefore, all members of X should fall in non-zero basin. This implies that the following set of equations should be satisfied.

$$T^d \cdot X \neq 0 \tag{1}$$

where T is a valid $MACA$ to be employed for designing two class classifier.

Design of Multi-class Classifier. A two class classifier can be viewed as a single stage classifier. For designing a multi-class classifier, this scheme of single stage classification will be repeatedly employed leading to a multi-stage classifier consisting of multiple CA, each CA corresponds to a single stage (*Fig.3*).

Hence, in the rest of the paper, we concentrate on designing an efficient CA based two class classifier.

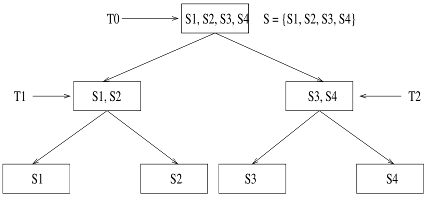


Fig. 3. $MACA$ based multi-class classifier. Note : A leaf node represents a class in the input set $S = \{S_1,S_2,S_3,S_4\}$

3 Genetic Algorithm for Evolution of $MACA$

The aim of this evolution scheme is to arrive at the desired $MACA$ (T matrix) that can perform the task of classification with minimum number of attractors.

This section describes the GA based solution to evolve the $MACA$ with the desired functionality. The three major functions of GA - Random Generation of initial population, Crossover and Mutation, as developed in the current GA formulation, are next discussed.

3.1 Random Generation of Initial Population with Desired Format

To form the population, it must be ensured that each solution randomly generated is an n bit $MACA$ with 2^m number of attractors where m can assume any value from 1 to n. From *Result II* of *Section 2.1*, the elementary divisor form of $MACA$ is $(1+x)(1+x)\cdot\cdot m\ times\ \ x^{d_1}x^{d_2}\cdot\cdot x^{d_p}$ & $d_1+d_2\cdots+d_p=(n\ -\ m)$, where the number of $(1+x)$ determines the number of attractors.

As per [3], the elementary divisors, if arranged in different order, produces $MACA$ with different attractor basins. For synthesis, the elementary divisors are first randomly distributed among itself forming a sequence. *Fig 4* shows one such sequence for the characteristic polynomial $x^7(1+x)^3 : x^2\cdot x^2\cdot(1+x)\cdot x\cdot(1+x)\cdot(1+x)\cdot x^2$. This is referred to as *pseudo-chromosome format* has been detailed in *Section 3.2*.

Each elementary divisor ($\phi_i(x)$) can be converted to a CA [1]. A tri-diagonal T matrix with characteristic polynomial $\phi_i(x)$ is accordingly synthesized. Let T_i matrices correspond to elementary divisors x^{d_i} (it will be a $d_i\times d_i$ matrix) and T_j matrices correspond to each elementary divisors $(x+1)$. If T_is and T_js are randomly arranged in block diagonal form, the characteristic polynomial of the resultant T is $x^{n-m}\cdot(1+x)^m$ and the minimal polynomial is $x^{d_p}\cdot(1+x)$ and it generates the $MACA$[1].

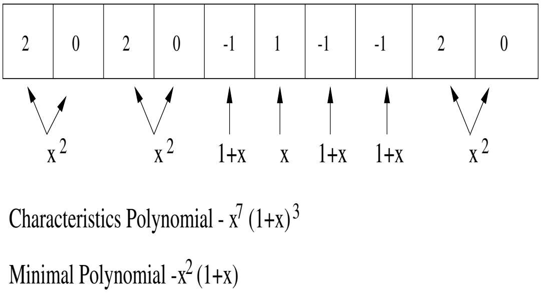

2
0
2
0
-1
1
-1
-1
2
0
x^2
x^2
1+x
x
1+x
1+x
x^2
Characteristics Polynomial - x^7 (1+x)^3
Minimal Polynomial -x^2 (1+x)

Fig. 4. MACA in Diagonal Representation Form

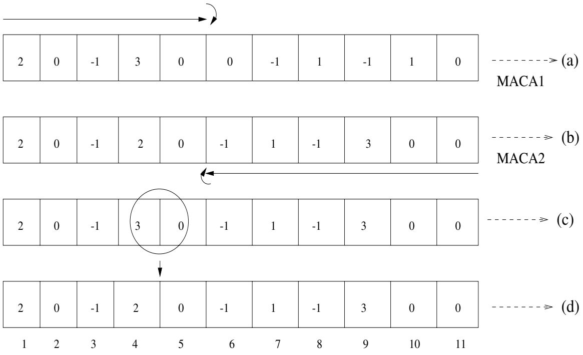
3.2 Pseudo-Chromosome Format

It is a method of representing an $MACA$ with respect to the sequence in which its $x^{d_i}{'}s$ and $(1+x){'}s$ are arranged. It gives a semblance of the chromosome and hence termed as *pseudo-chromosome format*. It a string of n bits where (a)

d_i positions occupied by a x^{d_i} is represented by d_i followed by $(d_i - 1)$ zeros (for example, $x^3 = [300]$), and (b) $(1 + x)$ is represented by -1. The *pseudo-chromosome format* of the *MACA* is illustrated in *Fig. 4*.

3.3 Crossover Algorithm

The crossover algorithm implemented is similar in nature to the conventional one normally used for *GA* framework with minor modifications as illustrated in *Fig.5*. The algorithm takes two *MACA* from the present population (*PP*) and forms the resultant *MACA*. The *pseudo-chromosome format* has x^{d_i} represented by d_i followed by $(d_i - 1)$ zeros. But in the case of *Fig 5c*, we have 3 followed by a single zero. This is a violation since the property of *MACA* is not maintained. So we take out those two symbols and form a *CA* of elementary divisor x^2 and adjust it. The resultant *MACA* after adjustment is shown in *Fig 5d.*



2	0	-1	3	0	0	-1	1	-1	1	0	(a) MACA1
2	0	-1	2	0	-1	1	-1	3	0	0	(b) MACA2
2	0	-1	3	0	-1	1	-1	3	0	0	(c)
2	0	-1	2	0	-1	1	-1	3	0	0	(d)
1	2	3	4	5	6	7	8	9	10	11	

Fig. 5. An Example of Cross-over Technique

3.4 Mutation Algorithm

The mutation algorithm emulates the normal mutation scheme (*Fig.6*. It makes some minimal change in the existing *MACA* of *PP* (Present Population) to a new *MACA* for *NP* (Next Population). Similar to the single point mutation scheme, the *MACA* is mutated at a single point.

In mutation algorithm, an $(x + 1)$'s position is altered. Some anomaly crops up due to its alteration. The anomaly is resolved to ensure that after mutation the new *CA* is also an *MACA*. The inconsistent format, as shown in the *Fig 6b* is the mutated version of *Fig 6a*. The inconsistency of the *pseudo-chromosome format* of *Fig 6b* can be resolved to generate the format of *Fig 6c*.

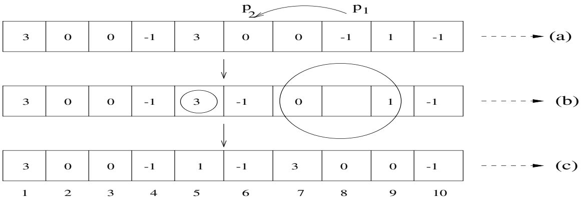


Fig. 6. An Example of Mutation Technique

3.5 Fitness Function

The fitness $\mathcal{F}(s)$ of a particular $MACA$ in a population is determined by the weighted mean of two factors - F_1 and F_2. The fitness criteria F_1 of the $MACA$ is determined by the percentage of patterns satisfying the *relation 1*. F_2 has been defined as -

$$F_2 = 1 - [(m-1)/n]^l \tag{2}$$

where 2^m denotes the number of attractor basins for the n cell CA, and l is equal to 1·8. The value of l is set empirically.

Subsequent to extensive experimentation, we have fixed up the relative weighage of F_1 and F_2 to arrive at the following empirical relation for the fitness function

$$\mathcal{F}(s) = 0.8 \cdot F_1 + 0.2 \cdot F_2 \tag{3}$$

4 Characterization of Attractor Basin – Its Capacity to Accommodate Noise

A classification machine is supposed to identify the zone, the trained patterns occupy. That is, let P_i be a representative pattern of n-bit learnt by the classifier as (say) Class A. Then a new n-bit pattern $\tilde{P_i}$ also belongs to the same class, if the hamming distance (r) between P_i and $\tilde{P_i}$ is small ($r << n$). **The hamming distance r is termed as noise while the pattern $\tilde{P_i}$ is termed as noisy pattern**.

This section provides a comprehensive study on the capacity of the classifier to accommodate noise. The study is developed in four phases

– *Phase I:* Establishes the fact that probing into the nature of pattern distribution of zero basin is equivalent to studying the noise accommodating capacity of the classifier.
– *Phase II:* Establishes the fact that characterization of the zero basin of $MACA$ with two attractor basins can be easily extended to $MACA$ with more than two attractors.
– *Phase III:* Characterizes the zero basin of $MACA$ with two attractor basins.
– *Phase IV:* Generalizes the study with 2^m number of basins, m varying from 1 to n.

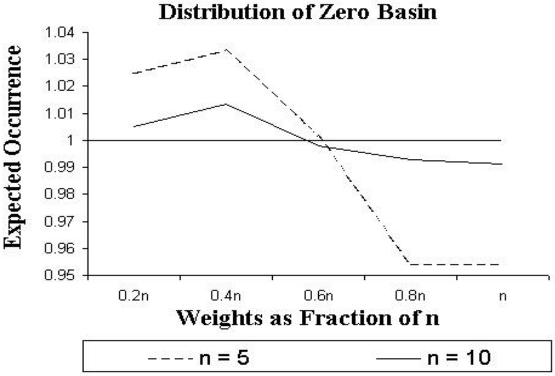


Fig. 7. Expected Distribution of MACA with two attractors ($m = 1$)

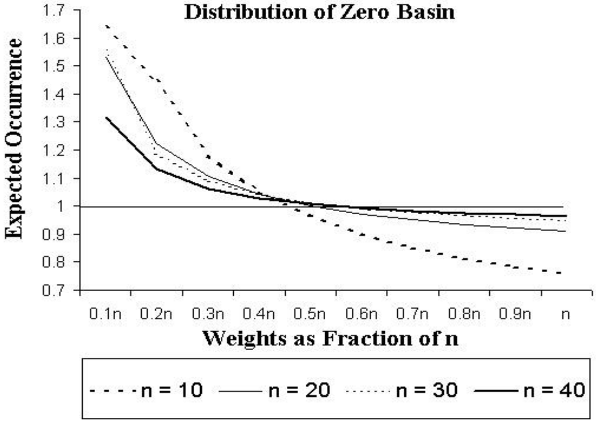


Fig. 8. Expected Distribution of MACA with 4 attractors ($m = 2$)

For shortage of space we only report the end results. The details are available in [3]. Both *Phase III & IV* show that the zero basin has a definite bias for patterns with lesser weight. *Phase III* establishes the relation on $MACA$ with two attractors, while *Phase IV* generalizes the result for multiple attractor $MACA$.

The bias gets reflected in the graphs reported in the *Fig 7-9* for $MACA$ with 2^m attractor basins. The graphs are based on the relation denoting the Expected Occurrence (EO) of a pattern with particular weight (r) in the zero basin of an $MACA$ over the normal unbiased expectation. The general relationship can be noted as

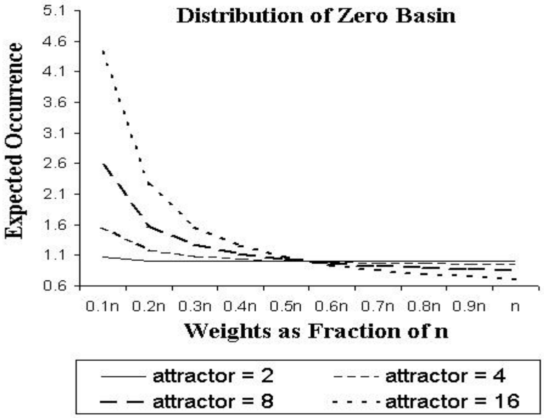


Fig. 9. Expected Distribution of MACA (n =30) with multiple attractors (m = 1,2,3,4)

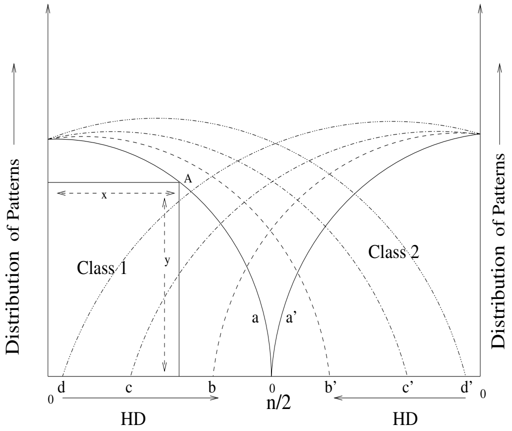


Fig. 10. Distribution of patterns in class 1 and class 2

$$EO(r, m) = N_{MACA}(r, m)/N_{UB}(r, m) \tag{4}$$

where $N_{UB}(r, m)$ shows the expected number of patterns with weight r in an $(n - m)$ dimensional vector subspace. $N_{MACA}(r, m)$ = the expected number of patterns with weight r in the zero basin of an $MACA$ with 2^m attractors. The entire equation is specific for a particular n. It has been shown [3] that the bias of the zero basin for low weight (r) states ($r << n$) creeps up due to the three neighborhood constraints of the CA [3].

The graph in *Fig 7-9* plots the expected occurrence - EO(r,m) - denoted by *relation 4* in the y-axis, while the weight of patterns is plotted on x-axis as a fraction of n - the number of bits in a pattern.

In *Fig 7*, where $m = 1$, it is seen that the expected occurrence doesn't follow a monotonically decreasing function but reaches the peak at slightly higher weight value. However, as m value is increased, (for *graph -8, m = 2*), the function becomes monotonically decreasing.

The gradient becomes steeper as the value of m is increased further (*Fig 9*). In graph of *Fig 9* which is plotted for different values of m keeping n ($= 30$) constant. It can be seen that the expectation of lower weight patterns occurring in zero basin increases manifold.

5 Performance Analysis of $MACA$ Based Classifier

For the sake of convenience of performance analysis, distributions of patterns in two classes are assumed as shown in *Fig.10.* Each pair of sets on whom classifiers are run are characterized by the curves ($a{-}a'$, $b{-}b'$, $c{-}c'$, $d{-}d'$). The ordinate of the curves represents number of pairs of patterns having the specified hamming distance. For example, at point A (on the curve for a) has y number of pairs of patterns which are at hamming distance x. The abscissa has been plotted in both direction, from left to right for *Class I* while from right to left for *Class II.* The curves of *Class I* & *II* overlap if $D_{min} < d_{max}$. An ideal distribution $a - a'$ is represented by the continuous line without any overlap of two classes.

In each distribution various values of n are taken. For each value of n, 2000 patterns are taken for each class. Out of this, 1000 patterns are taken from each class to build up the classification model. The rest 1000 patterns are used to test the prediction accuracy of the model. For each value of n, 10 different pairs of pattern sets are built.

The *Table 1* represents the classification efficiency of data set $a - a'$, $b - b'$, $c{-}c'$. *Column II* represents the different values of m (number of attractor basins) for which GA finds the best possible solution. *Column III* to *VI* represent the classification efficiency of training and test data set respectively. Classification efficiency of training set is the percentage of patterns which can be classified in different attractors while that of test data implies the percentage of data which can be correctly predicted. The best result of classification efficiency corresponding to each m in the final generation is taken. This is averaged over for the 10 different pairs of pattern set taken for each value of n.

The following experiments validate the theoretical foundations of the classifier performance reported in earlier sections.

5.1 Expt 1: Study of GA Evolution

The GA starts with various values of m. But it soon begins to get concentrated in certain zone of values. The genetic algorithm is allowed to evolve for 50 generations. In each case 80% of the population in the final solution assumes the two or three values of m noted in *Column II* of *Table 1*. The classification performance improves at a very slow gradient on further increase of m.

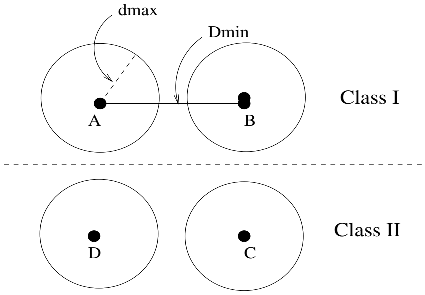


Fig. 11. Clusters Detection by two-class classifier

Table 1. Experiment to find out the value of m

Size	Value	Curve a-a'		Curve b-b'		Curve c-c'	
(n)	of m	Training	Testing	Training	Testing	Training	Testing
20	2	85.40	85.60	83.20	82.00	81.20	72.40
	3	96.10	94.35	92.20	93.35	92.20	83.35
40	3	98.20	97.75	97.60	96.80	77.60	66.80
	4	98.35	98.85	97.20	97.45	94.28	87.45
60	3	98.55	97.75	96.90	96.05	86.98	77.55
	4	98.50	98.00	96.90	96.05	91.90	86.60
80	3	98.81	98.65	98.70	97.70	81.70	76.35
	4	99.15	99.20	98.70	97.70	88.79	87.30
	5	99.75	99.70	98.30	97.30	91.65	87.10
100	3	99.65	99.25	98.30	97.45	86.40	77.95
	4	99.67	99.35	98.40	97.30	83.10	80.35

5.2 Expt 2: Experiment Done on $a - a'$ and $b - b'$

It shows that classification accuracy is above 98% in most of the cases. Moreover, the prediction accuracy is also almost equal to the classification accuracy. This validates the theoretical foundation that *MACA* basins are natural clusters.

5.3 Expt 3: Experiment Done on Data Set $c - c'$

With increasing overlap of data set (curve $c - c'$), a significant gap is created between classification efficiency of test and trained data. The *Column V & VI* of *Table 1* shows the result. But even in this case the classification efficiency and the number of attractors required to represent the data set doesn't deteriorate much.

Table 2. Clusters Detection by MACA based Classifier, $n = 100$, $D_{min} = 20$, $d_{max} = 5$

Combi^n of Clusters	Value of m	Performance (%)	
		Training	Testing
A & *B*, *C* & *D*	2	95.90	92.30
	4	99.82	97.10
A & *C*, *B* & *D*	2	94.50	92.30
	4	98.70	96.62
A & *D*, *B* & *C*	2	94.60	90.40
	4	99.20	96.82

5.4 Expt 4a: Experiment with Data Sets Having Implicit Clusters

Even if the data set between two classes have high overlap $(d - d')$, the classifier functions well if the classes have implicit clusters among them. The following experiment is performed to validate this observation.

To perform this experiment, we first randomly generate four pivot patterns (*Fig.11*) which are separated by a fixed hamming distance (say D_{min}). Then around each pivot, we randomly generate a cluster of p number of patterns within d_{max} distance (maximum hamming distance between a pivot and an element in the associated cluster); where $d_{max} <= D_{min}/2$. It is seen that $MACA$ based classifier perform classification task very well even though the distribution of patterns of each class is not ideal. The *Table 2* gives the detail result of cluster detection.

The *Column III* and *IV* depict the classification efficiency of the training and testing data set respectively at different values of m. In the two attractor basins (say a & b) which *Cluster A* & *B* occupy, it is observed that 99% of patterns of *Cluster A* occupies attractor a, while 99% of patterns of *Cluster B* occupies the attractor b. It is similar in case of the class formed from C and D. This is in line with the theoretical formulation that each attractor basin generates a cluster of patterns having lesser hamming distance.

6 Conclusion

The paper presents the detailed design and analysis of Cellular Automata based Pattern Classification technique. Theoretical formulation and experimental results prove beyond doubt the scope of the machine becomes really popular in industry in the near future.

References

1. P.Pal Chaudhuri, D.Roy Choudhury, S. Nandi and S. Chattopadhyay. *Additive Cellular Automata Theory and Applications.* IEEE Computer Society Press, California, USA, 1997.
2. S. Wolfram, *Theory and applications of cellular automata.* Singapore: World Scientific, 1986. ISBN 9971-50-124-4 pbk.
3. Niloy Ganguly, *"Cellular Automata Evolution : Theory and Applications in Pattern Recognition and Classification",* Ph.D thesis (proposal) *CST* Dept. BECDU India.

An Efficient Mapping Scheme for Embedding Any One-Dimensional Firing Squad Synchronization Algorithm onto Two-Dimensional Arrays

Hiroshi Umeo, Masashi Maeda, and Norio Fujiwara

Osaka Electro-Communication University,
Neyagawa-shi, Hatsu-cho 18-8, Osaka, 572-8530, Japan
umeo@umeolab.osakac.ac.jp

Abstract. An efficient mapping scheme is proposed for embedding any one-dimensional firing squad synchronization algorithm onto 2-D arrays, and some new 2-D synchronization algorithms based on the mapping scheme are presented. The proposed mapping scheme can be readily applied to the design of synchronization algorithms with fault tolerance, algorithms operating on multi-dimensional cellular arrays, and for the generalized case where the general is located at an arbitrary position on the array. A six-state algorithm is developed that can synchronize any $m \times n$ rectangular array in $2(m+n)-4$ steps. In addition, we develop a nine-state optimum-time synchronization algorithm on square arrays. We progressively reduce the number of internal states of each cellular automaton on square and rectangular arrays, achieving nine states for a square array and six states for a rectangular array. These are the smallest number of states reported to date for synchronizing rectangular and square arrays.

1 Introduction

The famous firing squad synchronization problem [13] is stated as follows: Given an array of n identical cellular automata, including a *general* at the left end that is activated at time $t = 0$, we want to design the automata such that, *at some future time*, all the cells will *simultaneously* and, *for the first time*, enter a special *firing* state. The problem was originally proposed by J. Myhill in 1957 to synchronize all parts of a self-reproducing machine, and it was subsequently presented in detail by Moore [13] in 1964. The firing squad synchronization problem has been studied extensively in more than forty years [1-25].

The present authors are involved in research on firing squad synchronization algorithms on two-dimensional (2-D) cellular arrays. Several synchronization algorithms on 2-D arrays have been proposed, including Grasselli [4], Kobayashi [6], Shinahr [16] and Szwerinski [17]. To date, the smallest number of cell states for which a synchronization algorithm has been developed is 17 for a square array and 28 for rectangular array, achieved by Shinahr [16].

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 69–81, 2002.

In this paper, we propose a simple and efficient mapping scheme for embedding any one-dimensional firing squad synchronization algorithm onto 2-D arrays and present some new 2-D synchronization algorithms. The first 6-state algorithm we develop can synchronize any $m \times n$ rectangular array in $2(m+n)-4$ steps. Although the algorithm is $\min(m, n) - 1$ steps slower than the optimum case, the number of internal states has been reduced considerably. Our mapping scheme can be easily applied to the design of synchronization algorithms with fault-tolerance, for algorithms operating on multi-dimensional cellular arrays, and for the generalized case where the general is located at any position on the array. The proposed mapping scheme allows us to obtain the first fault-tolerant synchronization algorithm on 2-D arrays. In addition, we develop a 9-state optimum-time synchronization algorithm on square arrays through progressive reduction of the number of internal states of each cellular automaton.

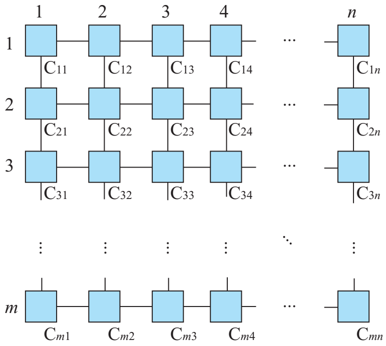


Fig. 1. Two-dimensional array of cellular automata

2 Firing Squad Synchronization Problem on Two-Dimensional Arrays

Figure 1 shows a finite two-dimensional cellular array consisting of $m \times n$ cells. Each cell is an identical (except the border cells) finite-state automaton. The array operates in lock-step mode in such a way that the next state of each cell (except border cells) is determined by both its own present state and the present states of its north, south, east and west neighbors. All cells (*soldiers*), except the north-west corner cell (*general*), are initially in the quiescent state at time $t = 0$ with the property that the next state of a quiescent cell with quiescent neighbors is the quiescent state again. At time $t = 0$, the north-west corner cell $C_{1,1}$ is in the *fire-when-ready* state, which is the initiation signal for the array. The firing squad synchronization problem is to determine a description (state set and next-state function) for cells that ensures all cells enter the *fire* state at exactly the same time and for the first time. The set of states must be independent of m and n.

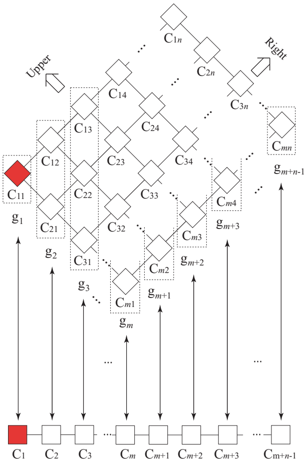


Fig. 2. Correspondence between 1-D and 2-D cellular arrays

3 Mapping Scheme for Embedding Any One-Dimensional Firing Squad Synchronization Algorithm onto Two-Dimensional Arrays

The proposal is a simple and efficient mapping scheme that enables us to embed any one-dimensional firing squad synchronization algorithm onto two-dimensional arrays without introducing additional states. We consider a 2-D array of size $m \times n$. We divide mn cells into $m+n-1$ groups g_k, $1 \leq k \leq m+n-1$, defined as follows.

$$g_k = \{C_{i,j} | (i-1) + (j-1) = k-1\}, \text{ i.e.,}$$

$g_1 = \{C_{1,1}\}$, $g_2 = \{C_{1,2}, C_{2,1}\}$, $g_3 = \{C_{1,3}, C_{2,2}, C_{3,1}\}$, . . . , $g_{m+n-1} = \{C_{m,n}\}$. Figure 2 shows the division into $m+n-1$ groups. For convenience, we define $g_0 = \{C_{0,0}\}$ and $g_{m+n} = \{C_{m+1,n+1}\}$.

Let $M = (Q, \delta_1, w)$ be any one-dimensional CA that fires ℓ cells in $T(\ell)$ steps, where Q is the finite state set of M, $\delta_1 : Q^3 \rightarrow Q$ is the transition function, and $w \in Q$ is the state of the right and left ends. We assume that M has $m+n-1$ cells, denoted by C$_i$, where $1 \leq i \leq m+n-1$. For convenience, we assume that M has a left and right end cells, denoted by C$_0$ and C$_{m+n}$, respectively. Both end cells C$_0$ and C$_{m+n}$ always take the end state $w (\in Q)$. We consider the one-to-one correspondence between the ith group g_i and the ith cell C_i on M

such that $g_i \leftrightarrow C_i$, where $1 \leq i \leq m+n-1$ (see Fig. 2). We can construct a 2-D CA $N = (Q, \delta_2, w)$ such that all cells in g_i simulate the ith cell C_i in real-time and N can fire any $m \times n$ arrays at time $t = T(m+n-1)$ if and only if M fires 1-D arrays of length $m+n-1$ at time $t = T(m+n-1)$, where $\delta_2 : Q^5 \rightarrow Q$ is the transition function, and $w \in Q$ is the border state of the array. Note that the set of internal states of N is the same as M. The transition function δ_2 is constructed as follows:

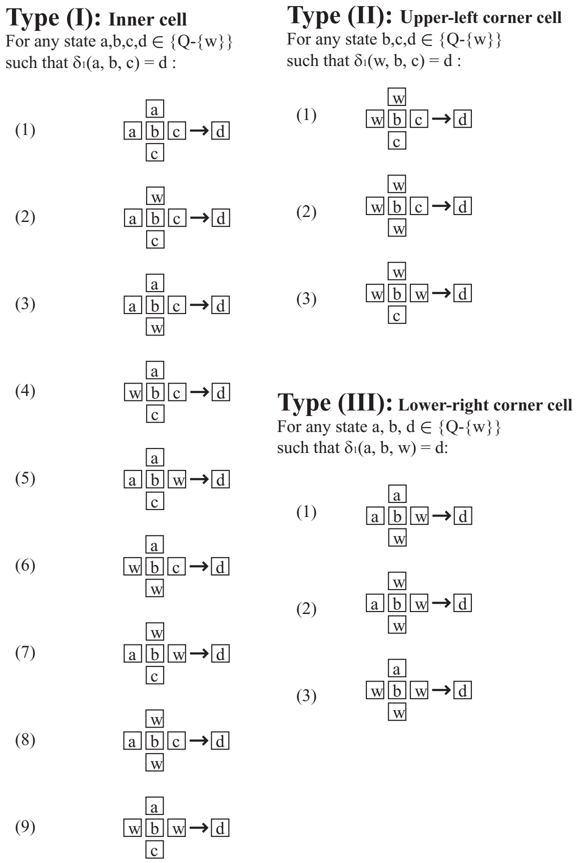

Type (I): Inner cell
For any state a,b,c,d ∈ {Q-{w}}
such that δ1(a, b, c) = d :
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
Type (II): Upper-left corner cell
For any state b,c,d ∈ {Q-{w}}
such that δ1(w, b, c) = d :
(1)
(2)
(3)
Type (III): Lower-right corner cell
For any state a, b, d ∈ {Q-{w}}
such that δ1(a, b, w) = d:
(1)
(2)
(3)

Fig. 3. Construction of transition rules for 2-D firing squad synchronization algorithm

Let $\delta_1(a, b, c) = d$ be any transition rule of M, where $a, b, c, d \in \{Q - \{w\}\}$. Then, N has nine transition rules, as shown in Fig. 3 Type (I). The first rule (1) in Type (I) is used by an inner cell that does not include border cells amongst its four neighbors. Rules (2)-(7) are used by an inner cell that has a border cell as its upper, lower, left, right, lower left, or upper right neighbor, respectively.

Here the terms *upper*, *right* etc. on the rectangular array are interpreted in a usual way, shown in Fig. 2, although the array is rotated by 45° in the counterclockwise direction. Rules (8) and (9) correspond to the case where $m = 1$, $n \geq 2$ and $m \geq 2$, $n = 1$, respectively. When $a = w$, that is, $\delta_1(w, b, c) = d$, where $b, c, d \in \{Q - \{w\}\}$, then N has three rules, as shown in Type (II). These rules are used by the cell located in the upper left corner. Rules (2) and (3) in Type (II) are special cases corresponding to rules (8) and (9) in Type (I). When $c = w$, that is, $\delta_1(a, b, w) = d$, where $a, b, d \in \{Q - \{w\}\}$, then N has three rules, as shown in Type (III). These rules are used by the cell located in the lower right corner. Rules (2) and (3) in Type (III) are special cases corresponding to rules (8) and (9) in Type (I).

Now let M have $m + n - 1$ cells. Here we show that the construction of 2-D CA N can generate the configuration of M in real-time. Specifically, for any i, $1 \leq i \leq m + n - 1$, the state of any cell in g_i at any step is the same and is identical to the state of C_i at the corresponding step. Let S_i^t, $S_{i,j}^t$ and $S_{g_i}^t$ denote the state of C_i, $C_{i,j}$ at step t and the set of states of the cells in g_i at step t, respectively. Then, we can establish the following lemma.

[Lemma 1] Let i and t be any integers such that $1 \leq i \leq m + n - 1$, $0 \leq t \leq T(m + n - 1)$.

1. $\| S_{g_i}^t \| = 1$. That is, the set $S_{g_i}^t$ is a singleton and all cells in g_i at step t are in the same state. We denote the state as $S_{g_i}^t$.
2. $S_{g_i}^t = S_i^t$.

(Proof sketch) This is proved by mathematical induction on t. At time $t = 0$, it is easily seen that the lemma holds, because the cell C_1 assumes a general state and other cells C_j, $2 \leq j \leq m+n-1$, assume a quiescent state, while $C_{1,1}$ takes a general state and all other cells are in a quiescent state at time $t = 0$. We assume that $S_{g_{i-1}}^k = S_{i-1}^k = a$, $S_{g_i}^k = S_i^k = b$, $S_{g_{i+1}}^k = S_{i+1}^k = c$, and $\delta_1(a, b, c) = d$, a, b, c, d$\in Q$, for some integer k such that $0 \leq k \leq T(m + n - 1) - 1$, and then we show that, for any i, $1 \leq i \leq m + n - 1$, $\| S_{g_i}^{k+1} \| = 1$ and $S_{g_i}^{k+1} = S_i^{k+1}$. Without loss of generality, we assume that $m \geq n$. We then prove the case of $1 \leq i \leq n - 1$. Considering any cell $C_{x,y}$ in g_i, where $x + y = i + 1$:

1. Case $2 \leq x \leq i - 1$, $2 \leq y \leq i - 1$: It is clear that $C_{x-1,y}, C_{x,y-1} \in g_{i-1}$ and $C_{x+1,y}, C_{x,y+1} \in g_{i+1}$. From the assumption, $S_{x-1,y}^k = S_{x,y-1}^k =$ a, $S_{x+1,y}^k = S_{x,y+1}^k = c$. Following rule (1) in Type (I), we have that $S_{x,y}^{k+1} = d$. In addition, from the assumptions $S_{i-1}^k = a$, $S_i^k = b$, $S_{i+1}^k = c$, and $\delta_1(a, b, c) = d$, we obtain $S_i^{k+1} = d$. Thus, we have $S_{x,y}^{k+1} = S_i^{k+1}$.
2. Case $x = 1, y = i$: The cell $C_{1,i}$ has a border cell as its north neighbor. It can be seen that $C_{1,i-1} \in g_{i-1}$ and $C_{2,i}, C_{1,i+1} \in g_{i+1}$. From the assumption, $S_{1,i-1}^k = a$, $S_{1,i}^k = b$, $S_{2,i}^k = S_{1,i+1}^k = c$. Following rule (2) in Type (I), we have that $S_{1,i}^{k+1} = d$. In addition, from the assumptions $S_{i-1}^k = a$, $S_i^k = b$, $S_{i+1}^k = c$, and $\delta_1(a, b, c) = d$, we obtain $S_i^{k+1} = d$. Thus, we have $S_{1,i}^{k+1} = S_i^{k+1}$.

3. Case $x = i, y = 1$: The cell $C_{i,1}$ has a border cell as its left neighbor. It can be seen that $C_{1,i-1} \in g_{i-1}$ and $C_{i,2}, C_{i+1,1} \in g_{i+1}$. From the assumption, $S^k_{1,i-1} = a$, $S^k_{i,1} = b$, $S^k_{i,2} = S^k_{i+1,1} = c$. Following rule (4) in Type (I), we have that $S^{k+1}_{i,1} = d$. In addition, from the assumptions $S^k_{i-1} = a$, $S^k_i = b$, $S^k_{i+1} = c$, and $\delta_1(a, b, c) = d$, we obtain $S^{k+1}_i = d$. Thus, we have $S^{k+1}_{i,1} = S^{k+1}_i$.

Other cases such as $i = 1$, $i = n$, $n + 1 \leq i \leq m - 1$, $i = m$, $m + 1 \leq i \leq m + n - 2$, and $i = m + n - 1$ can be treated similarly. Thus, we have proved the lemma. ■

We see that any configuration on a 1-D CA consisting of $m + n - 1$ cells can be mapped onto 2-D $m \times n$ arrays. Therefore, when the embedded 1-D CA fires $m + n - 1$ cells in $T(m + n - 1)$ steps, the corresponding 2-D CA fires the $m \times n$ array in $T(m+n-1)$ steps. Thus, we can embed any 1-D synchronization algorithm on 2-D arrays without increasing the number of internal states. We complete the lemma in the next theorem.

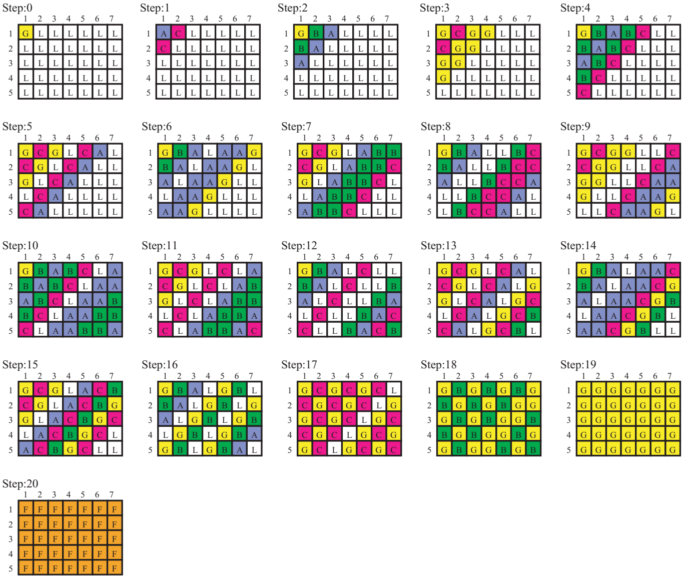


Fig. 4. Snapshots of the proposed 6-state linear-time firing squad synchronization algorithm on rectangular arrays

[Theorem 2] Let A be any s-state firing synchronization algorithm operating in $T(n)$ steps on 1-D n cells. Then, there exists a 2-D s-state cellular automaton that can synchronize any $m \times n$ rectangular array in $T(m + n - 1)$ steps.

In the long history of the study of the firing squad synchronization problem, many optimum-time synchronization algorithms have been proposed, including Balzer [1], Goto [3], Mazoyer [10] and Waksman [24]. The 6-state optimum-time algorithm on 1-D arrays developed by Mazoyer [10] is the smallest number of internal states for which such an algorithm has been developed. We can obtain a 6-state rectangular firing algorithm based on the 6-state Mazoyer's synchronization algorithm through lemma 3 below. Our next rectangular synchronization algorithm fires any $m \times n$ array in $2(m+n)-4$ steps, and it is $\min(m,n)-1$ steps slower than the optimum 28-state algorithm given in Shinahr [16]. However, the number of internal states is considerably smaller. In Fig. 4, we show snapshots of our 6-state firing synchronization algorithm running on a rectangular array of size 5×7.

[Lemma 3][10] There exists a 6-state 1-D cellular automaton that can synchronize n cells in the optimum $2n-2$ steps.

[Theorem 4] There exists a 6-state firing squad synchronization algorithm that can synchronize any $m \times n$ rectangular array in $2(m+n)-4$ steps.

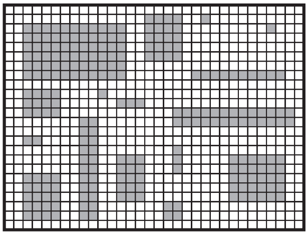


Fig. 5. A 2-D cellular array with faulty rectangular holes

4 Applications of the Mapping Scheme

Some applications and extensions of our mapping scheme are given below. The first problem we consider is the design of a fault-tolerant synchronization algorithm on 2-D arrays.

4.1 Fault-Tolerant Synchronization Algorithm on 2-D Arrays

A firing squad synchronization problem on 1-D arrays with faulty cells has been studied by Kutrieb and Vollmar [7] and Umeo [18]. A similar problem on 2-D arrays has never been discussed or studied due to the difficulties in designing such synchronization algorithms. Here we present first a fault-tolerant synchronization algorithm on 2-D arrays. The array we consider includes some faulty regions, each consisting of faulty cells that cannot transmit any information or change state. Each faulty region is rectangular, and isolated from each other and

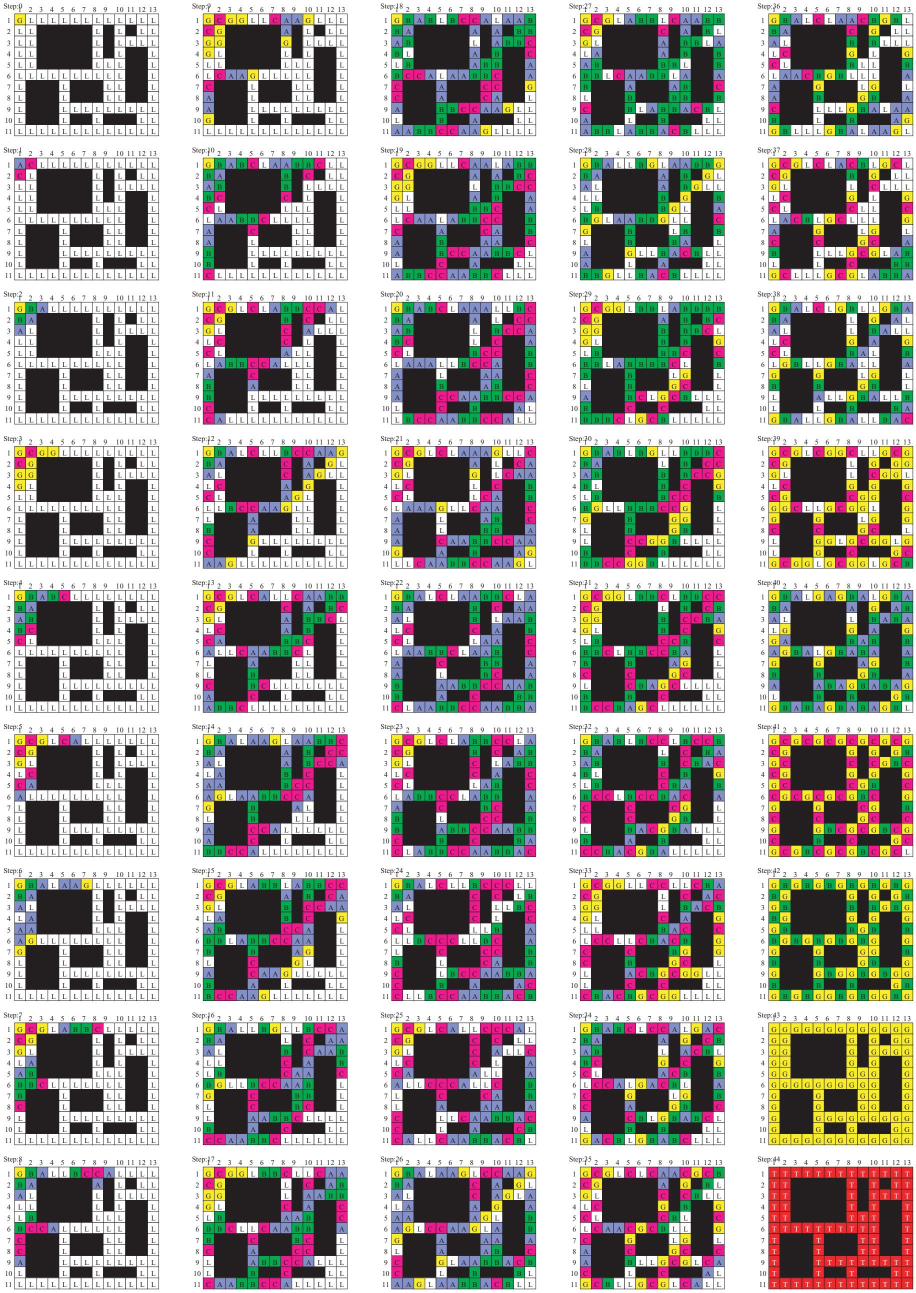


Fig. 6. Snapshots of the proposed 6-state fault-tolerant linear-time firing squad synchronization algorithm on rectangular arrays

from the boundary encircling the array. The faulty regions can be regarded as obstacles or holes in the array. To simplify the definition of a faulty region in our construction, each faulty cell is assumed to be marked by a boundary symbol. A typical 2-D rectangular array with 16 isolated holes (obstacles) is shown in Fig. 5. The problem is to design a firing squad synchronization algorithm such that non-faulty cells on the array fire simultaneously. The algorithm given in Theorem 4 can run without modification on any 2-D rectangular array containing isolated rectangular holes. Snapshots of our 6-state fault-tolerant synchronization algorithm running on a rectangular array of size 11 × 13 including 8 holes are shown in Fig. 6.

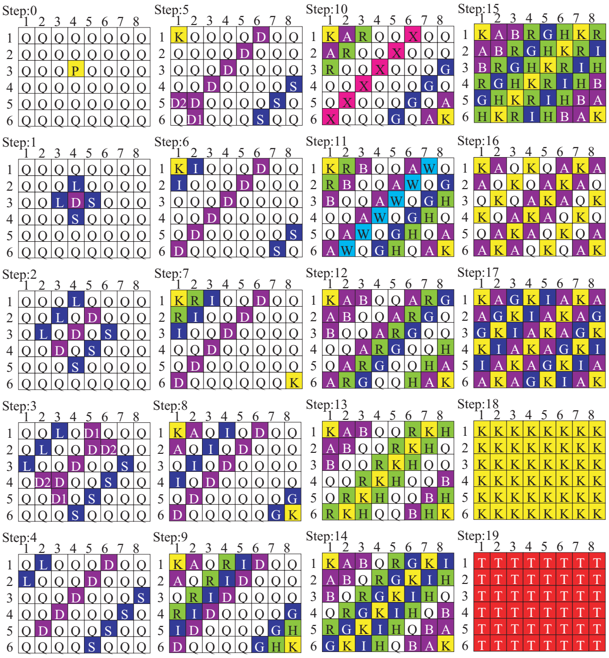


Fig. 7. Snapshots of our 19-state linear-time generalized firing squad synchronization algorithm on rectangular arrays

[Theorem 5] There exists a 6-state 2-D CA that can synchronize any $m \times n$ rectangular array containing isolated rectangular holes in $2(m+n)-4$ steps.

4.2 Multi-dimensional Extension

Theorem 4 can be readily extended to multi-dimensional cases. For example, we can give an extended theorem for three-dimensional (3-D) arrays.
[Theorem 6] There exists a 6-state firing squad synchronization algorithm that can synchronize any 3-D $m \times n \times \ell$ solid arrays in $2(m+n+\ell)-6$ steps.

4.3 Generalized Synchronization Algorithm on Rectangular Arrays

Now we consider a generalized firing squad synchronization problem, in which the general can be initially located at any position on the array. Moore and Langdon [14], Szwerinski [17] and Varshavsky, Marakhovsky and Peschansky [22] developed optimum-time firing algorithms with 17, 10 and 10 internal states, respectively. In these algorithms, n cells in a 1-D array are fired in $n - 2+\max(k, n - k + 1)$ steps, when the general is located at C_k. Our 2-D mapping scheme can be applied to the design of synchronization algorithms even in the generalized case. For any 2-D array M of size $m \times n$ with the general at $C_{r,s}$, where $1 \leq r \leq m$, $1 \leq s \leq n$, there exists a corresponding 1-D cellular array N of length $m + n - 1$ with the general at C_{r+s-1} such that the configuration of N can be mapped on M, and M fires if and only if N fires. Based on the 17-state generalized 1-D algorithm given by Moore and Langdon [14], we obtain the following 2-D generalized synchronization algorithm that fires in $m + n - 1 - 2+\max(r + s - 1, m + n - r - s + 1)= m + n+\max(r + s, m + n - r - s + 2) - 4$ steps. Two additional states are required in our construction (details omitted). Szwerinski [17] also proposed an optimum-time generalized 2-D firing algorithm with 25,600 internal states that fires any $m \times n$ array in $m + n+\max(m, n)-\min(r, m - r + 1)-\min(s, n - s + 1) - 1$ steps, where (r, s) is the general's initial position. Our 2-D generalized synchronization algorithm is $\max(r + s, m + n - r - s + 2)-\max(m, n)+\min(r, m - r + 1)+\min(s, n - s + 1) - 3$ steps larger than the optimum algorithm proposed by Szwerinski [17]. However, the number of internal states required to yield the firing condition is the smallest known at present. Snapshots of our 19-state generalized synchronization algorithm running on a rectangular array of size 6×8 with the general at $C_{3,4}$ are shown in Fig. 7.

[Theorem 7] There exists a 19-state 2-D CA that can synchronize any $m \times n$ rectangular array in $m + n + \max(r + s, m + n - r - s + 2) - 4$ steps with the general at an arbitrary initial position (r, s).

5 Implementation of Optimum-Time Synchronization Algorithm on Square Arrays

In this section, we present a new 9-state synchronization algorithm that runs in the optimum $(2n - 2)$ steps on $n \times n$ square arrays. Our algorithm is the same as that of Shinahr [16] and operates as follows. By dividing the entire square array into n L-shaped 1-D arrays such that the length of the ith L is $2n - 2i + 1$ $(1 \leq i \leq n)$, we treat the square firing as n independent 1-D firings with the general located at the center cell. On the ith L, a general is generated at $C_{i,i}$ at time $t = 2i - 2$, and the general initiates the horizontal and vertical firings on the row and column arrays via an optimum-time synchronization algorithm. The array fires in optimum time $t = 2i - 2 + 2(n - i + 1) - 2 = 2n - 2$. Let Q be a set of internal states for the 1-D optimum-time synchronization algorithm. When we implement the 1-D algorithm on 2-D square arrays based on the scheme above, $2 \mid\mid Q \mid\mid -1$ states are normally required for independent row and column firings when the firing state is shared.

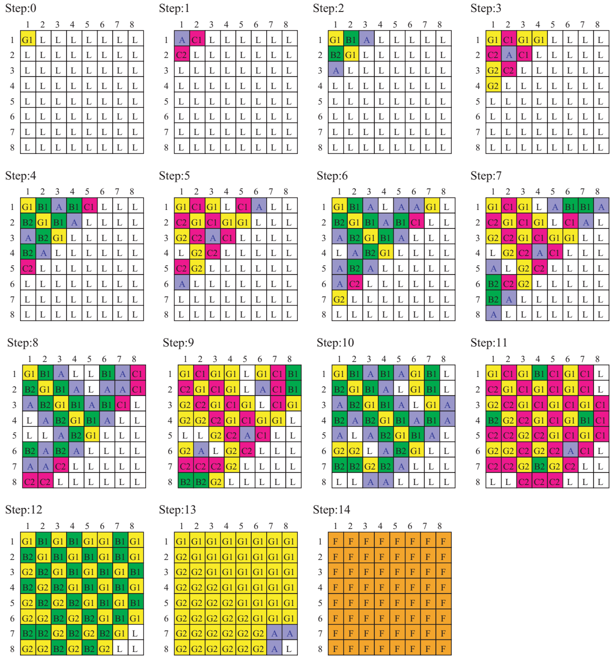


Fig. 8. A configuration of a 9-state implementation of optimum-time firing on square arrays

In our construction, by applying the Mazoyer's 6-state algorithm [10], we find that 2 states can be deleted from the normal construction, rendering 9 states sufficient for optimum-time square firing. We have tested our transition rule set on squares of size 2×2 to 1000×1000. Figure 8 shows snapshots of configurations of our 9-state synchronization algorithm running on a square of size 8×8. Thus we have:

[**Theorem 8**] There exists a 9-state 2-D CA that can synchronize any $n \times n$ square array in $2n - 2$ steps.

6 Conclusions

We have proposed a simple and efficient mapping scheme for embedding any 1-D firing squad synchronization algorithm onto 2-D arrays and presented several new 2-D synchronization algorithms. Although most of the algorithms are slightly slower than the optimum algorithms, the number of internal states is considerably smaller. Our mapping scheme can be readily applied to the design of synchronization algorithms with fault tolerance, algorithms operating on multi-dimensional cellular arrays, and for the generalized case with the general located at an arbitrary position in the array. The mapping scheme provides the first fault-tolerant synchronization algorithm on 2-D arrays. In addition, we have developed a 9-state optimum-time synchronization algorithm on square arrays and a 6-state linear-time algorithm on rectangular arrays, representing the smallest such algorithms presented to date.

References

1. R. Balzer: An 8-state minimal time solution to the firing squad synchronization problem. *Information and Control*, vol. 10(1967), pp. 22-42.
2. W. T. Beyer: Recognition of topological invariants by iterative arrays. Ph.D. Thesis, MIT, (1969), pp. 144.
3. E. Goto: A minimal time solution of the firing squad problem. Dittoed course notes for Applied Mathematics 298, Harvard University, (1962), pp. 52-59, with an illustration in color.
4. A. Grasselli: Synchronization of cellular arrays: The firing squad problem in two dimensions. *Information and Control*, vol. 28(1975), pp. 113-124.
5. J. J. Grefenstette: Network structure and the firing squad synchronization problem. *J. of Computer and System Sciences*, vol.26(1983), pp.139-152.
6. K. Kobayashi: The firing squad synchronization problem for two-dimensional arrays. *Information and Control*, vol. 34(1977), pp. 177-197.
7. M. Kutrib and R. Vollmar: The firing squad synchronization problem in defective cellular automata. *Trans. of IEICE on Inf. and Syst.*, vol. E78-D, No. 7(1995), pp. 895-900.
8. M. Maeda and H. Umeo: A design of two-dimensional firing squad synchronization algorithms and their implementations. *Proc. of 15th Annual Conference of Japanese Society for Artificial Intelligence*, 2C3-05(2001), pp. 1-4.
9. J. Mazoyer: An overview of the firing squad synchronization problem. *Lecture Notes on Computer Science*, Springer-Verlag, vol. 316(1986), pp. 82-93.
10. J. Mazoyer: A six-state minimal time solution to the firing squad synchronization problem. *Theoretical Computer Science*, vol. 50(1987), pp. 183-238.
11. J. Mazoyer: On optimal solutions to the firing squad synchronization problem. *Theoretical Computer Science*, vol. 168(1996), pp. 367-404.
12. M. Minsky: *Computation: Finite and infinite machines.* Prentice Hall, (1967), pp. 28-29.
13. E. F. Moore: The firing squad synchronization problem. in *Sequential Machines, Selected Papers* (E. F. Moore ed.), Addison-Wesley, Reading MA., (1964), pp. 213-214.
14. F. R. Moore and G. G. Langdon: A generalized firing squad problem. *Information and Control*, vol. 12(1968), pp. 212-220.
15. H. B. Nguyen and V. C. Hamacher: Pattern synchronization in two-dimensional cellular space. *Information and Control*, vol. 26(1974), pp, 12-23.
16. I. Shinahr: Two- and three-dimensional firing squad synchronization problems. *Information and Control*, vol. 24(1974), pp. 163-180.
17. H. Szwerinski: Time-optimum solution of the firing-squad-synchronization-problem for n-dimensional rectangles with the general at an arbitrary position. *Theoretical Computer Science*, vol. 19(1982), pp. 305-320.
18. H. Umeo: A fault-tolerant scheme for optimum-time firing squad synchronization. *Parallel Computing: Trends and Applications*, Elsevier Science B.V. 1994, pp. 223-230.

19. H. Umeo, T. Sogabe and Y. Nomura: Correction, optimization and verification of transition rule set for Waksman's firing squad synchronization algorithms. *Proc. of the 4th International Conference on Cellular Automata for Research and Industry*, 2000, pp. 152-160.
20. H. Umeo, M. Maeda and N. Fujiwara: Some implementations on two-dimensional firing squad synchronization algorithms. *Proc. of 2001 Summer Language and Automata Symposium*, held on 23-25, July 2001, 2001, pp. 26:1-2.
21. V. I. Varshavsky: Synchronization of a collection of automata with random pairwise interaction. *Autom. and Remote Control*, vol. 29(1969), pp. 224-228.
22. V. I. Varshavsky, V. B. Marakhovsky and V. A. Peschansky: Synchronization of interacting automata. *Mathematical Systems Theory*, Vol. 4, No. 3(1970), pp. 212-230.
23. R. Vollmar: *Algorithmen in Zellularautomaten.* Teubner, pp. 192, 1979.
24. A. Waksman: An optimum solution to the firing squad synchronization problem. *Information and Control*, vol. 9(1966), pp. 66-78.
25. J. B. Yunes: Seven state solutions to the firing squad synchronization problem. *Theoretical Computer Science*, vol.127(1994), pp. 313-332.

Chaotic Subshifts Generated by One Dimensional Elementary CA. The Role of Transitivity.*

Gianpiero Cattaneo and Alberto Dennunzio

Università degli Studi di Milano–Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione,
Via Bicocca degli Arcimboldi 8, 20126 Milano (Italy)
{cattang, alberto.dennunzio}@disco.unimib.it

Abstract. We present the behavior of simple subshifts generated by 1D Elementary CA (ECA) with respect to some components of chaoticity as transitivity, topological mixing and strong transitivity. A classification of subshifts generated by ECA with respect to transitivity is given. In literature one can find several notions of topological transitivity. We discuss two types of transitivity for discrete time dynamical system: positive and full. The relationships among these notions and properties such as existence of a dense orbit, topological chaos and indecomposability are investigated.

1 Introduction and Basic Definitions

A one–dimensional cellular automaton (CA) is a bi–infinite array of identical elements, called *cells*, located on a straight line and whose position or *site* is labelled by an integer number $i \in \mathbb{Z}$. Each cell can assume a state chosen from a finite set $\mathcal{A}$, the *alphabet* of the CA, and changes its state according to a *local rule*, *homogeneously* applied to all cells of the automaton, in a discrete time evolution. Formally, a one-dimensional bi-infinite CA is described as a triple $\langle \mathcal{A}, r, f \rangle$, where $\mathcal{A}$ is the finite *alphabet of states*, $r \in \mathbb{N}$ is the *radius* of the automaton, and $f : \mathcal{A}^{2r+1} \mapsto \mathcal{A}$ is the *local rule*.

The simplest case is the boolean one of *elementary cellular automata* (ECA) characterized by $r = 1$, defined by local rules of the kind: $f : \{0,1\}^3 \mapsto \{0,1\}$. The different $2^{2^3} = 256$ ECA are classified by the natural number $n_f = f(0,0,0) \cdot 2^0 + f(0,0,1) \cdot 2^1 + \cdots + f(1,1,1) \cdot 2^7$.

A *configuration* (*global state*) is a map $\underline{x} : \mathbb{Z} \mapsto \mathcal{A}$, $i \to x_i$ specifying a state for any site. The set $\mathcal{A}^{\mathbb{Z}}$ of all configurations of a CA can be equipped with the Tychonoff metric, $d(\underline{x}, \underline{y}) = \sum_{i=-\infty}^{+\infty} \frac{1}{4^{|i|}} \delta(x_i, y_i)$ where $\delta : \mathcal{A} \times \mathcal{A} \mapsto \mathbb{R}_+$ is the *Hamming* distance on $\mathcal{A}$. So a CA can be viewed as a (compact, perfect, and complete) *discrete time dynamical system* (DTDS) $\langle \mathcal{A}^{\mathbb{Z}}, F_f \rangle$ where the *global*

* This work has been supported by M.I.U.R. COFIN project "Formal Languages and Automata: Theory and Application"

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 82–91, 2002.

transition map $F_f : \mathcal{A}^{\mathbb{Z}} \mapsto \mathcal{A}^{\mathbb{Z}}$ induced by f, associates with any configuration $\underline{x} \in \mathcal{A}^{\mathbb{Z}}$ the *next time step* configuration $F_f(\underline{x})$ whose i-th component is expressed by the local rule: $[F_f(\underline{x})]_i = f(x_{i-r}, \dots, x_i, \dots, x_{i+r})$.

CA can display a rich and complex time evolution whose exact determination is in general very hard. The empirical observation of 1D CA dynamics leads to realize that many of them share a nontrivial subshift global behavior. A (simple) subshift contained in a CA $\langle \mathcal{A}, r, f \rangle$ is a DTDS $\left\langle S_f, \sigma_{S_f} \right\rangle$, where S_f is the collection of the bi-infinite sequences $\underline{x} \in \mathcal{A}^{\mathbb{Z}}$ on which the global transition map F_f coincides with the shift (formally, $\forall \underline{x} \in S_f : F_f(\underline{x}) = \sigma(\underline{x})$).

Several dynamical properties of CA have been investigated during the last few years. We illustrate the chaotic behavior of simple subshifts generated by ECA. We recall that a discrete time dynamical system (DTDS) $\langle X, g \rangle$, where the *phase space* X is equipped with a distance d and the *next state map* $g : X \mapsto X$ is continuous on X according to the metric d, is *topologically chaotic* according to Devaney [7] (or D-chaotic) iff it is *Regular* (the set of periodic points is a dense subset of the phase space, i.e., for any $x \in X$ and any $\epsilon > 0$ there exists a periodic point p such that $d(p, x) < \epsilon$), *Transitive* (for every pair of nonempty open subsets A and B of the phase space, there exists a positive integer t_0 such that $g^{t_0}(A) \cap B \neq \emptyset$) and *Sensitive to initial conditions* (there exists a constant $\epsilon > 0$ such that for any state $x \in X$ and for any $\delta > 0$ there must be at least one state $y \in X$ and one integer $t_0 \in \mathbb{N}$ such that $d(x, y) < \delta$ and $d(g^{t_0}(x), g^{t_0}(y)) \geq \epsilon$). In [1] it has been shown that if a DTDS with infinite cardinality is regular and transitive, then it is also *sensitive to initial conditions*.

In order to establish the chaoticity of a DTDS, transitivity plays a central role. On an interval of real numbers, transitivity is equal to chaos [12]. In the case of the global DTDS $\left\langle \mathcal{A}^{\mathbb{Z}}, F_f \right\rangle$ induced from a CA local rule f, the simple condition of transitivity implies sensitivity to initial conditions [5]. Again, for a subshift of infinite cardinality, transitivity is a condition equivalent to chaos [4].

Several definitions of "transitivity" can be found in literature. In this paper we rename the above notion given in [7] as *positive transitivity*, since the integer appearing in this definition belongs to $\mathbb{N}$. Differently in [6] one can find a definition in which it is required that the integer t_0 ranges over $\mathbb{Z}$ and that $\langle X, g \rangle$ is a homeomorphic DTDS on a compact space X. In other works ([8], [9] for instance) the notion of transitivity is given in terms of the existence of a dense orbit (that is, of a state $x_0 \in X$ whose positive motion starting from x_0 is dense in X).

We introduce the new notion of *full transitivity* (the integer belongs to $\mathbb{Z}$) and we study the relations among these two types of transitivity and other properties such as existence a dense orbit, indecomposability, and topological chaos. The results are:

- existence of a dense orbit does not imply positive transitivity, not even if X is a compact space, whereas it always implies full transitivity. If X is a perfect set (i.e., without isolated points) the positive transitivity is implied by the existence of a dense orbit.

- if X has infinite cardinality, then full transitivity and regularity together imply sensitivity to initial condition.
- full transitivity implies indecomposability.
- in the case of homeomorphic DTDS on compact space full transitivity is not equivalent to positive transitivity. In other words, the notion of transitivity given in [6] is different from the one considered in [7].

2 Subshifts in 1D CA

In this section we briefly outline the behavior of simple subshifts generated by 1D CA. Precisely we focus our attention to some components of chaotic behavior such as (positive) transitivity and stronger conditions such as topological mixing and strong transitivity.

At first, let us recall the formal notion of subshift.

Definition 1. *A (two-sided)* subshift *over the alphabet* $\mathcal{A} = \{0, 1, \ldots, k-1\}$ *(*$k \geq 2$*) is a DTDS* $\langle S, \sigma_S \rangle$*, where* S *is a closed (*$\overline{S} = S$*) strictly* σ*-invariant (*$\sigma(S) = S$*) subset of* $\mathcal{A}^{\mathbb{Z}}$*, and* σ_S *is the restriction of the shift map* σ *(*$\sigma : \mathcal{A}^{\mathbb{Z}} \mapsto \mathcal{A}^{\mathbb{Z}}$, $\forall i \in \mathbb{Z}\, [\sigma(\underline{x})]_i = x_{i+1}$*) to this subset.*

A subshift $\langle S, \sigma_S \rangle$ distinguishes the words or finite blocks constructed over the alphabet $\mathcal{A}$ in two types: admissible blocks, i.e., blocks appearing in some configuration of S and blocks which are not admissible, called forbidden. We will write $w \prec \underline{x}$ to denote that the $\mathcal{A}$–word $w = (w_1, \ldots, w_n) \in \mathcal{A}^*$ appears in the configuration $\underline{x} \in \mathcal{A}^{\mathbb{Z}}$, formally $\exists i \in \mathbb{Z}$ s.t. $x_i = w_1, \ldots, x_{i+n-1} = w_n$. We will denote by $w \not\prec \underline{x}$ the fact that w does not appear in the configuration $\underline{x}$.

A subshift can be generated by a set of words considered as a set of forbidden blocks. Precisely, let $\mathcal{F}$ be any subset of $\mathcal{A}^*$, and let us construct the set $S(\mathcal{F}) = \{\underline{x} \in \mathcal{A}^{\mathbb{Z}} : \forall w \in \mathcal{F}, w \not\prec \underline{x}\}$. Then $S(\mathcal{F})$ is a subshift, named the subshift generated by $\mathcal{F}$. Now we illustrate a sufficient condition on the local rule in order that the global dynamic turns out to be a subshift. For this goal, a suitable set of forbidden blocks with respect to the local CA rule will be constructed.

Definition 2. *Let* $\langle \mathcal{A}, r, f \rangle$ *be a CA. A finite block* $w_{-r} \ldots w_{-1}\, w_0\, w_1 \ldots w_r \in \mathcal{A}^{2r+1}$ *is said to be left-forbidden (resp., right-forbidden) with respect to the CA local rule* f *iff* $f(w_{-r} \ldots\, w_0\, \ldots w_r) \neq w_1$ *(resp.,* $f(w_{-r} \ldots\, w_0\, \ldots w_r) \neq w_{-1}$*). Conversely, the block is called left-admissible (resp., right-admissible).*

Proposition 1. *Let* $\langle \mathcal{A}, r, f \rangle$ *be a CA and* $\left\langle \mathcal{A}^{\mathbb{Z}}, F_f \right\rangle$ *be the associated DTDS. Let* $\mathcal{F}_{2r+1}(f) = \{w_{-r} \ldots w_0 \ldots w_r \in \mathcal{A}^{2r+1} : f(w_{-r} \ldots w_0 \ldots w_r) \neq w_1\}$ *be the set of all left-forbidden blocks of the local rule* f *and let* $S_f = \{\underline{x} \in \mathcal{A}^{\mathbb{Z}} : \forall w \in \mathcal{F}_{2r+1}(f), w \not\prec \underline{x}\}$ *be the set of all configurations which do not contain any local rule left-forbidden block. Then* $\langle S_f, F_f \rangle$ *is a subshift.*

An analogous result holds considering the right-forbidden blocks of a local rule. From now on we will give some CA left-subshift properties; the analogous ones for CA right-subshift can be obtained in a dual way.

In order to study the dynamical behavior of a CA we introduce the notion of De Bruijn graph.

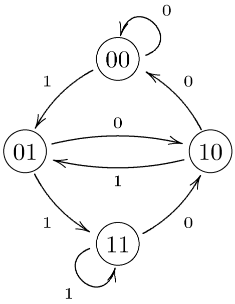
Definition 3. *The De Bruijn graph $B(r, \mathcal{A})$ associated to any $\langle \mathcal{A}, r, f \rangle$ CA is the pair $\langle \mathcal{A}^{2r}, E \rangle$, where $E = \{(x, y) \,|\, x, y \in \mathcal{A}^{2r} \; s.t. \, x_2 = y_1, \dots, x_{2r} = y_{2r-1}\}$.*

This graph is common to any CA, independently from the particular analytical form of the local rule f. But if we consider one particular CA, then the edges of the above De Bruijn graph can be labelled by letters of the alphabet $\mathcal{A}$ using the local rule f of the involved CA. Precisely, the label of each edge $(x, y) \in E$ is defined as $l(x, y) := f(x_1, x_2, \dots, x_{2r}, y_{2r}) \in \mathcal{A}$. This labelled De Bruijn graph will be denoted by $B_f(r, \mathcal{A})$. If we read a configuration $\underline{x}$ as a bi-infinite path along nodes of $B_f(r, \mathcal{A})$, the next time configuration $F_f(\underline{x})$ is the resulting path along the corresponding labels. Therefore we can say that the labelled De Bruijn graph associated to a CA describes its one step forward global dynamics.

Furthermore, in order to represent the CA subshift,we consider the subgraph $\mathcal{AG}_f := \left\langle \mathcal{A}^{2r}, E' \right\rangle$ of its De Bruijn graph, making use of an edge validation function to suppress those edges which represent forbidden blocks of the subshift, i.e., $E' = \{(x, y) \,|\, x, y \in \mathcal{A}^{2r} \; s.t. \, x_2 = y_1, \dots, x_{2r} = y_{2r-1}, \, (x_1, x_2, \dots, x_{2r}, y_{2r}) \notin \mathcal{F}_{2r+1}(f)\}$. Trivially, bi-infinite paths along nodes on the graph $\mathcal{AG}_f$ correspond to bi-infinite strings of the subshift S_f.

In the general case we can minimize the subgraph $\mathcal{AG}_f$ removing all the unnecessary nodes. From now on we will consider only minimized graphs.

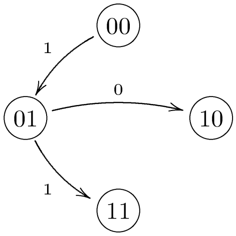
Example 1. The ECA local rule 170. This rule is such that the set of all length 3 left-forbidden blocks is $\mathcal{F}_3(170) = \emptyset$. Thus the $\mathcal{AG}_{170}$ graph coincides with the associated "complete" De Bruijn graph in which each edge is labelled according to the local rule f_{170}.



rule 170

The set S_{170} generated by ECA rule 170 coincides with the whole phase space $\{0, 1\}^{\mathbb{Z}}$. Then the global next state function induced by the ECA local rule 170 is the *full* shift map σ.

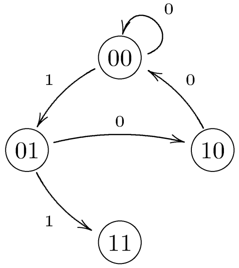
Example 2. The ECA local rule 91. This CA local rule has 000, 100, 101, 110 and 111 as left-forbidden blocks. The $\mathcal{AG}_{91}$ graph of rule 91 is the following:



rule 91

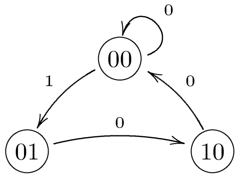
There is no bi-infinite path along nodes of $\mathcal{AG}_{91}$.This means that the set S_{91} generated by ECA rule 91 is empty. All the nodes of the graph $\mathcal{AG}_{91}$ are unnecessary. Thus the minimization leads to completely eliminate the $\mathcal{AG}_{91}$ graph.

Example 3. The ECA local rule 74. This rule is such that the set of all length 3 left-forbidden blocks is $\mathcal{F}_3(74) = \{101, 110, 111\}$. The $\mathcal{AG}_{74}$ graph of rule 74 is the following:



rule 74: $\mathcal{AG}_{74}$ graph

The bi-infinite paths along nodes of $\mathcal{AG}_{74}$ graph give rise to bi-infinite strings in which each 1 is followed and preceded at least by two 0's. We can minimize the $\mathcal{AG}_{74}$ graph removing node 11 and the edge $(01, 11)$.



rule 74: minimized $\mathcal{AG}_{74}$ graph

Now we can express dynamical CA properties over S_f by properties of its associated subgraph $\mathcal{AG}_f$. In particular, the following results hold concerning the existence and the cardinality of a subshift generated by a CA.

Theorem 1. *Suppose that the local rule f of a CA $\langle \mathcal{A}, r, f\rangle$ consists of a positive number n ($0 \neq n \leq |\mathcal{A}|^{2r+1}$) of l-admissible blocks. Then, the following statements are equivalent:*

i) *the corresponding subshift is not trivial ($S_f \neq \emptyset$);*
ii) *there exists a finite block of length $n+2r+1$ consisting of l-admissible blocks;*
iii) *there is at least one cycle in the associated $\mathcal{AG}_f$ graph.*

Theorem 2. *Let $\mathcal{AG}_f$ be the graph associated to the subshift generated by the CA $\langle \mathcal{A}, r, f\rangle$. Then $|S_f| = +\infty$ if at least one of the following conditions holds:*

i) *there exists a spatially aperiodic configuration $\underline{x} \in S_f$;*
ii) *there exist two cycles in the graph $\mathcal{AG}_f$ differently labelled and connected by at least one edge;*

Proposition 2. *If the $\mathcal{AG}_f$ graph consists of n independent cycles of order $k_1, \ldots, k_n$, then $|S_f| = \sum_{i=1}^{n} k_i$.*

For what concerns the dynamical behavior of a CA subshift, we are particularly interesting to the (positive) transitivity and to some stronger conditions such as topological mixing and strong transitivity.

Definition 4. *A DTDS $\langle X, g\rangle$ is said topologically mixing iff for any pair A, B of nonempty open subsets of the phase space X, there exists an integer $n_0 \in \mathbb{N}$ such that $\forall n > n_0$, $g^n(A) \cap B \neq \emptyset$. A DTDS is called M-chaotic iff it is D-chaotic and topologically mixing.*

Definition 5. *A DTDS $\langle X, g\rangle$ is said strongly transitive iff for any nonempty open subsets A of the phase space X, $\bigcup_{n\in\mathbb{N}} g^n(A) = X$. A DTDS is called ST-chaotic iff it is D-chaotic and strongly transitive.*

For a subshift (positive) transitivity implies regularity. Moreover, if the subshift has infinite cardinality, positive transitivity is equivalent to D-chaoticity. Analogous result holds for topological mixing: for a infinite subshift, topological mixing is equivalent to M-chaos. Differently, a strong transitive subshift is necessarily a finite and transitive subshift whose configurations are mutually different periodic points. Therefore, it cannot be D-chaotic.

The study of subshifts generated by an ECA rule leads to the following result: the set of all ECA transitive rules is just the set theoretic union of all mixing and all strong transitive rules. This fact is proved in [4].

We can summarize the subshift behaviour of transitive ECA in the following way:

1. ECA which are *both* mixing *and* strong transitive subshifts are *nonempty trivial*: their $\mathcal{AG}_f$ graphs are composed by a unique node with a loop and so the subshift spaces are the trivial ones, either the singleton $\{\overline{0}\}$ or the singleton $\{\overline{1}\}$.

2. Strong transitive ECA subshifts are characterized by an $\mathcal{AG}_f$ subgraph of the De Bruijn graphs which has a unique strongly connected component with equal number of nodes and edges.
   This kind of subshift is constituted by a finite number of mutually periodic configurations and the cardinality of the subshift is equal to the cycle order of the graph.
3. Mixing (but not strong transitive) ECA subshifts have infinite cardinality.

Theorem 3. *The following two statements are equivalent for ECA:*

i) *it is topological mixing, but not strongly transitive.*
ii) *it is D-chaotic.*

By this theorem, infinite topological mixing subshifts are the only transitive subshifts which are sensitive to initial conditions.

3 Transitivity on DTDS

As we have seen, transitivity is the key to study the chaotic behavior of a subshift. This fact holds also in the case of maps on a interval of real numbers; moreover in the CA case transitivity implies sensitivity. We have renamed transitivity as positive transitivity (see the introduction). We remark that if a DTDS is positively transitive, then for any pair A, B of nonempty open subset of X both the following conditions must hold:

$$\text{PT1)} \qquad \exists n_1 \in \mathbb{N} : g^{n_1}(A) \cap B \neq \emptyset$$
$$\text{PT2)} \qquad \exists n_2 \in \mathbb{N} : g^{n_2}(B) \cap A \neq \emptyset$$

Example 4. A non positively transitive DTDS.
Let $\langle \mathbb{N}, g_s \rangle$ be the DTDS where the phase space $\mathbb{N}$ is equipped with the trivial metric $d_{tr} : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{R}_+$, $d_{tr}(x, y) = 1$ if $x \neq y$ and $d_{tr}(x, y) = 0$ otherwise, and $g_s : \mathbb{N} \mapsto \mathbb{N}$ is the successor function defined as follows: $\forall x \in \mathbb{N}$ $g_s(x) = x + 1$. If $A = \{3\}$ and $B = \{5\}$, there exists $n_1 = 2$, s.t. $g_s^{n_0}(A) \cap B \neq \emptyset$ but there is no integer $n_2 \in \mathbb{N}$ s.t. $g_s^{n_0}(B) \cap A \neq \emptyset$.

In [2] one can find the following notion of transitivity renamed by us as *negatively transitive*.

Definition 6. *A DTDS is said to be* negatively transitive *iff for any pair A, B of nonempty open subsets of the phase space X, there exists an integer $n \in \mathbb{N}$ such that $g^{-n}(A) \cap B \neq \emptyset$.*

It easy to show that the positive transitivity is equivalent to the negative one.

The existence of a dense orbit is not sufficient to guarantee the positive transitivity of a DTDS as shown in the following examples.

Example 5. A non positively transitive DTDS which possesses a dense orbit. Let $\langle \mathbb{N}, g_s \rangle$ be the DTDS of the example 4. The orbit $\{g_s^t(0)\}_{t \in \mathbb{N}}$ of initial state $0 \in \mathbb{N}$ is dense since it coincides with the whole phase space $\mathbb{N}$.

This behavior holds also in the case of compact DTDS.

Example 6. A compact and non positively transitive DTDS with a dense orbit. Let us consider the phase space $X = \{0\} \cup \{\frac{1}{2^n} : n \in \mathbb{N}\}$ equipped with the metric $d(x,y) = |x-y|$ induced by the usual metric of $\mathbb{R}$; in this metric every singleton $\{x\}$ $[x \neq 0]$ is a clopen. The topology induced from d besides the trivial open sets $\{X, \emptyset\}$ contains $\mathcal{P}(\{\frac{1}{2^n} : n \in \mathbb{N}\})$, the power set of $\{\frac{1}{2^n} : n \in \mathbb{N}\}$, and the open balls centered in 0. In particular this topological space is *compact*.

Let $g : X \mapsto X$ be the map defined as: $\forall x \in X$, $g(x) = \frac{1}{2}x$. This map is continuous in the topology induced from d. The unique positive orbit dense in X is the sequence of initial state $x = 1$: $\gamma_1 = \{1, (1/2), (1/2^2), \ldots, (1/2^t), \ldots\}$. But this dynamical system is not positively transitive. Indeed, if we consider the two nonempty open sets $A = \{1/2\}$ and $B = \{1\}$, then we have that $\forall n \in \mathbb{N}$: $g^n(A) \cap B = \emptyset$.

Note that there exist DTDS's which are positively transitive but without dense orbits.

Example 7. A positively transitive DTDS which does not possess any dense orbit. Let $\langle Per(\sigma), \sigma \rangle$ the dynamical subsystem constituted by all periodic points of the full shift $\left\langle \mathcal{A}^{\mathbb{Z}}, \sigma \right\rangle$. It is a positively transitive DTDS which has no dense orbit.

We now introduce a weaker notion of transitivity named *Full Transitivity.*

Definition 7. *A DTDS $\langle X, g \rangle$ is said to be* topologically full transitive *iff for every pair of nonempty open subsets A and B of the phase space, there exists an integer $t_0 \in \mathbb{Z}$ such that $A \cap g^{-t_0}(B) \neq \emptyset$.*

As stressed in the introduction, this is the notion of transitivity one can find in [6] in the case of a homeomorphic DTDS on a compact space. We want to remark that for any pair A, B of nonempty open subset of X the following three conditions are mutually equivalent:

Full Transitivity)	$\exists t_0 \in \mathbb{Z} : A \cap g^{-t_0}(B) \neq \emptyset$
FT1)	$\exists t_1 \in \mathbb{Z} : g^{t_1}(A) \cap B \neq \emptyset$
FT2)	$\exists t_2 \in \mathbb{Z} : g^{t_2}(B) \cap A \neq \emptyset$

Note that the existence of a orbit dense is a sufficient condition to guarantee the full transitivity. It is easy to show that a positively transitive DTDS is full transitive. The converse does not hold, see the example 4 and the following example where a homeomorphic DTDS is involved.

Example 8. Let $X = \mathbb{Z}$ be the set of integer numbers equipped with the trivial metric and $g_p : \mathbb{Z} \mapsto \mathbb{Z} \forall x \in \mathbb{Z}$, $g_p(x) = x - 1$. It is easy to show that the system $\langle \mathbb{Z}, g_p \rangle$ is homeomorphic and full transitive, but it is not positively transitive. Indeed, if $A = \{3\}$ and $B = \{5\}$, there is no integer $n_0 \in \mathbb{N}$ s.t. $g_p^{n_0}(A) \cap B \neq \emptyset$.

In [1], Banks et al. prove that a regular and positively transitive DTDS with infinite cardinality, has sensitive dependence on initial conditions. In [3] we have proved the following.

Proposition 3. *Let $\langle X, g\rangle$ be a regular and full transitive DTDS with infinite cardinality. Then it is sensitive to initial conditions.*

It is an open problem to find a sensitive, regular, full transitive but non positively transitive DTDS. Another topological property of dynamical systems related to the transitivity is the condition of minimality.

Definition 8. *A DTDS $\langle X, g\rangle$ is* indecomposable *iff X is not the union of two nonempty open, disjoint, and positively invariant subsets.*

Of course, if two open subsets decompose X, then they must be also closed. This means that for an indecomposable DTDS the phase space X cannot be split into two (nontrivial) clopen sub-dynamical systems; indecomposability is in a certain sense an *irreducibility* condition [11]. Note that in [10] this property is also called condition of *invariant connection* and X is said to be *invariantly connected*. It easy to show that, in every DTDS $\langle X, g\rangle$, full transitivity implies indecomposability and the existence of a dense orbit implies indecomposability too.

The following result can be found in [3].

Theorem 4. *Let $\langle X, g\rangle$ be a DTDS. If X is perfect and possesses a dense orbit, then it is positively transitive.*

The above notion of full transitivity holds for any metric space X and for any continuous function $g : X \mapsto X$. If we consider a compact metric space and a homeomorphic function, our notion of full transitivity coincides with the definition of topological transitivity given in [6]. We ask whether this "topological transitivity" (equal to our full transitivity) coincides with the positive transitivity. The answer is no, as shown in the following example.

Example 9. A homeomorphic, full transitive and compact DTDS which is not positively transitive.
Let us consider the subshift of finite type $\langle S, \sigma_S\rangle$ on the boolean alphabet generated by the ECA rule 222. The corresponding set of left forbidden blocks is $\mathcal{F}_3(222) = \{010, 100, 101, 110\}$. A configuration $\underline{x} \in \{0,1\}^{\mathbb{Z}}$ belongs to S iff $\underline{x} = \underline{0} = (\dots, 0, 0, 0, \dots)$ or $\underline{x} = \underline{1} = (\dots, 1, 1, 1, \dots)$ or it is of the kind $\underline{x} : \mathbb{Z} \mapsto \{0.1\}$ such that for some $k \in \mathbb{Z}\, x_i = 0$ if $i < k$, $x_i = 1$ otherwise. The subshift $\langle S, \sigma_S\rangle$ is equipped with the Tychonoff metric: $d_T(\underline{x}, \underline{y}) = \sum_{i=-\infty}^{+\infty} \frac{1}{4^{|i|}} |x_i - y_i|$. Since S is a closed subset of $\{0,1\}^{\mathbb{Z}}$ then it is compact. Moreover σ_S is a homeomorphism.

A property of this subshift is that for any configuration $\underline{x} \in S \setminus \{\underline{0}, \underline{1}\}$, the set $\{\underline{x}\}$ is open. It is easy to show that $\langle S, \sigma_S\rangle$ is full transitive but it is not positively transitive.

4 Conclusions

As we have seen, on a suitable set of configurations the global transition map F_f induced by a local rule of a 1D CA coincides with the shift map. In the case of elementary rules, we have presented the behavior of such kind of subshifts with respect to some components of chaoticity such as (positive) transitivity, topological mixing and strong transitivity. The set of all ECA transitive rules turns out to be the set theoretic union of all mixing and all strong transitive rules.

In the more general context of DTDS, transitivity plays a central role in order to establish topological chaoticity. We have renamed the notion of transitivity given in [7] as positive transitivity and we have introduced the new notion of full transitivity. The relations among these notions and other properties of a DTDS can be summarized in the following results: the existence of a dense orbit implies full transitivity and moreover, when the phases space is perfect, the positive transitivity. Full transitivity (and thus also positive transitivity) implies indecomposability. If the phases space has infinite cardinality, full transitivity and regularity together imply sensitivity to initial condition. Finally we have shown an example of a homeomorphic and full transitive DTDS on a compact space which is not positively transitive.

References

[1] J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey, *On Devaney's definition of chaos*, American Mathematical Montly **99** (1992), 332–334.

[2] F. Blanchard, P. Kurka, and A. Maas, *Topological and measure-theoretic properties of one-dimensional cellular automata*, Physica D **103** (1997), 86–99.

[3] G. Cattaneo and A. Dennunzio, *On transitivity of a discrete time dynamical system*, Preprint, 2002.

[4] G. Cattaneo, A. Dennunzio, and L. Margara, *Chaotic subshifts and related languages applications to one-dimensional cellular automata*, To appear in Fundamenta Informaticae, 2002.

[5] B. Codenotti and L. Margara, *Transitive cellular automata are sensitive*, American Mathematical Monthly **103** (1996), 58–62.

[6] M. Denker, C. Grillenberger, and K. Sigmund, *Ergodic theory on compact spaces*, Lecture Notes in Mathematics, vol. 527, Springer-Verlag, 1976.

[7] R. L. Devaney, *An introduction to chaotic dynamical systems*, second ed., Addison-Wesley, 1989.

[8] E. Glasner and B. Weiss, *Sensitive dependence on initial condition*, Nonlinearity **6** (1993), 1067–1075.

[9] A. Kameyama, *Topological transitivity and strong transitivity*, Preprint, 2001.

[10] J. P. LaSalle, *Stability theory for difference equations*, MAA Studies in Math., American Mathematical Society, 1976.

[11] D. Ruelle, *Strange attractors*, Math. Intelligencer **2** (1980), 126–137.

[12] M. Vellekoop and R. Berglund, *On intervals, transitivity = chaos*, American Mathematical Monthly **101** (1994), 353–355.

Stochastic Analysis of Cellular Automata and the Voter Model

Heinz Mühlenbein and Robin Höns

FhG-AiS D-53754 Sankt Augustin

Abstract. We make a stochastic analysis of both deterministic and stochastic cellular automata. The theory uses a mesoscopic view, i.e. it works with probabilities instead of individual configurations used in micro-simulations. We make an exact analysis by using the theory of Markov processes. This can be done for small problems only. For larger problems we approximate the distribution by products of marginal distributions of low order. The approximation use new developments in efficient computation of probabilities based on factorizations of the distribution. We investigate the popular voter model. We show that for one dimension the bifurcation at $\alpha = 1/3$ is an artifact of the mean-field approximation.

1 Introduction

Complex cellular automata are usually analyzed by micro-simulations. A myriad of runs are made with different initial conditions. Then some general patterns are sought for describing the results of the runs. The computer outputs show realizations of complex spatio-temporal stochastic processes. They can be a valuable aid in intuitively defining and characterizing the processes involved and can lead to the discovery of new and interesting phenomena. But one should not infer too much from a few realizations of a stochastic process: it is not the behavior of each individual cell that matters, since the stochasticity will ensure that all realizations are different at least in detail. It is the gross properties of the stochastic process that are likely to be of interest in the long run.

As a first step to solve these problems we propose the *mesoscopic view*. We follow here the approach used in statistical physics. There a *microscopic view,* a *mesoscopic view,* and a *macroscopic view* are distinguished. In the microscopic view a large state space of configurations is defined, together with state transitions by interactions of the elementary units. The mesoscopic view works with stochastic processes. Instead of clearly defined configurations $\mathbf{x}(t)$ it uses probability distributions $p(\mathbf{x}, t)$ on the configuration space. This approach has two advantages: first, it allows a very soft specification of systems, and second, the processes can be modeled with uncertainty. In the macroscopic view differential equations of macroscopic variables are derived. The macroscopic variables can be seen as expectations $\langle A(t)\rangle = \sum_x p(x, t)a(x)$.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 92–103, 2002.

In order to use a mesoscopic view we have to convert existing microscopic or macroscopic theories into a mesoscopic description. This means that we have to find equations describing the evolution of probabilities. It turns out that this problem is much more difficult than we initially thought. But the reward of this research seems great.

2 The Importance of a Stochastic Analysis?

In a recent book about cellular automata [7] we find the following remark: “One should not be surprised that it is precisely that feature which makes CA so appealing as “toy” models of physical phenomena – namely their propensity to exhibit a remarkable wide range of possible behavior– that lies at the heart of the problem what makes CA so difficult to study analytically.”

Wolfram [16] was one of the first to recognize the importance of a stochastic analysis of cellular automata. He proposed as Problem 10: *What is the correspondence between cellular automata and stochastic systems ?* Closely related is Problem 11: *How are cellular automata affected by noise and other imperfections ?*

Wolfram noted [16]: “The problems are intended to be broad in scope, and are probably not easy to solve. To solve any of them completely will probably require a multitude of subsidiary questions to be asked and answered. But when they are solved, substantially progress towards a theory of cellular automata and perhaps of complex systems in general should have been made.”

This remark is absolutely true. The problem we want to solve is very difficult. We approach both problems with the same technique – the approximation of probability distributions by products of low dimensional distributions. The method can be used for deterministic automata given a distribution as input, or stochastic automata, where the transition rules have stochastic components. The approximation of distributions has recently been advanced in such diverse fields as Bayesian networks [8], graphical models in statistics [9], and optimization by search distributions [13].

In [7] we found one approach which is at least similarly in spirit to our approach. It is called the *local structure theory*, developed by Gutowitz [4, 5]. His approximations for 1-D CA are very similar to ours, but he did not come very far with 2-D CA.

Our theory is not restricted to cellular automata. It can also be used for general spatial distributions which arise for instance in ecological problems [2]. The importance of space for evolution was already recognized by Darwin in his ”Origin of Species”. We investigated Darwin’s conjecture by micro-simulations of evolutionary games in [10]. In the very next future we will extend the method presented here to evolutionary games.

3 The Nonlinear Voter Model

We consider a model of two species (or two opinions). For the spatial distribution we assume a one-dimensional stochastic cellular automaton (SCA) defined by a circle of n cells. Each cell c_i is occupied by one individual, thus each cell is characterized by a discrete value $x_i \in \{0, 1\}$. We assume a circular connection. Therefore we set $x_{n+1} := x_1$ and $x_0 := x_n$. The state of cell c_i at time $t + 1$ is defined by the states of cells c_{i-1}, c_i, c_{i+1} at time t. The state transitions of the voter model depend only on $k(t) = x_{i-1}(t) + x_i(t) + x_{i+1}(t)$. This class of automata is called *totalistic*. For the stochastic voter model the transitions are defined as follows.

$k(t)$	$p(x_i(t + 1) = 1\|k(t))$
3	$1 - \epsilon$
2	$1 - \alpha$
1	α
0	ϵ

$p(x_i = 1|k)$ denotes the transition probability given k. ϵ is a small stochastic disturbance parameter. The model is defined by α. If $\alpha < 0.5$ one speaks of positive frequency dependent invasion. This model is also called the *majority vote model*, because the individuals join the opinion of the majority in the neighborhood. For $\alpha > 0.5$ the model is called a negative frequency dependent invasion process. In this case the minority opinion has more weight. The deterministic cellular automata are given by $\epsilon = 0$ and $\alpha = 0, 1$. The voter model has been intensively investigated by micro simulations. The reader is referred to [3].

4 Exact Analysis of CA by Markov Processes

The nonlinear voter model, as any cellular automata, can be seen as a Markov process. We just sketch the Markov process analysis. Let $\mathbf{x} = (x_1, \ldots, x_n)$ denote a binary vector representing the state of a dynamical system at time $t+1$. The vector at time t is denoted by $\mathbf{x}'$. We assume $x_i \in \{0, 1\}$.

Definition 1 *Let $p(\mathbf{x}, t)$ denote the probability of $\mathbf{x}$ in the population at generation t. Then $p_i(x_i, t) = \sum_{\mathbf{x}, X_i = x_i} p(\mathbf{x}, t)$ defines a uni-variate marginal distribution. Bi-variate marginal distributions are defined by $p(x_{i-1}, x_i, t)) = \sum_{\mathbf{x}, X_{i-1} = x_{i-1}, X_i = x_i} p(\mathbf{x}, t)$. Conditional distributions are defined for $p(\mathbf{z}) > 0$ by $p(\mathbf{y}|\mathbf{z}) = p(\mathbf{y}, \mathbf{z})|p(\mathbf{z})$, where $\mathbf{y}$ and $\mathbf{z}$ are disjoint sub-vectors of $\mathbf{x}$.*

The time evolution of the distribution is given for one step by the equation

$$p(\mathbf{x}, t+1) = \sum_{x'} p(\mathbf{x}, t+1|\mathbf{x}', t)p(\mathbf{x}', t) \tag{1}$$

$(p(\mathbf{x}, t+1|\mathbf{x}', t)) := M(t)$ defines the global transition matrix. It is of dimension $2^n * 2^n$. The global transition probabilities can easily be computed from the local transition rules of the automaton.

Definition 2 *The stochastic process is a Markov process if $M(t)$ is independent of t.*

Thus CA are Markov processes. For a Markov process we have

$$p(\mathbf{x}, t) = M^t p(\mathbf{x}, 0) \tag{2}$$

This is the exact solution of our problem. Unfortunately it works only for small n, because otherwise the dimension of the matrix is too large for practical computations. There exist one major result.

Theorem 1 *If all entries of M are greater than 0, then the stochastic CA has a unique limit distribution. It is given by the eigenvector belonging to the eigenvalue $\lambda_1 = 1$. The limit distribution is independent from the initial distribution.*

The proof is based on the *theorem of Frobenius-Perron*.

The relationship between the results of micro-simulations and of the stochastic analysis is very intricate. We will try to explain the problem. The deterministic voter model with $\epsilon = 0$ and $\alpha = 0$ does have a stationary distribution, depending on the initial distribution. It consists mainly of $O = (0, \ldots, 0)$ and $I = (1, \ldots, 1)$. Furthermore there are fixed points consisting of configurations with contiguous blocks of 0's and 1's. In micro-simulations we would observe convergence to one of these configurations, depending on the initial configuration. If we now change both ϵ and α to small values greater than 0, then the assumptions of the above theorem are fulfilled. There exists a unique limit distribution, independent of the initial distribution. The limit distribution now consists of the two configurations O and I only. Thus changing the value of α from 0 to a small positive value has a huge impact. The stationary distributions are very different.

The limit distribution of a a purely stochastic automata with $\epsilon = 0.1$ and $\alpha = 0.2$ consists of all configurations, the largest frequency have the configurations O and I.

The different behavior of the dynamics of the micro-simulations cannot recaptured by just looking at the limit distribution. One needs additional information. The most important one is the *expected passage time*. It gives the number of transitions to get from one configuration to another one.

We recall how the *expected passage time* is computed. We label the configurations $\mathbf{x}$ by integer numbers i. Let $\mathbf{M} = (m_{ij})$ be the transition matrix of a Markov process with K different configurations, labeled with $1, \ldots, K$. The expected first passage times τ_{ij} from configuration i to configuration j can be determined from a set of linear equations:

$$\tau_{ii} = 0\ ,\ 1 \leq i \leq K \tag{3}$$

$$\tau_{ij} = 1 + \sum_{k=1}^{N} m_{ik}\tau_{kj}\ ,\ 1 \leq i, j \leq K,\ i \neq j \tag{4}$$

The right-hand side of equation (4) results from *unfolding* the Markov process one step. In order to compute the expected first passage time from state 1 to state K it is sufficient to consider the vector of passage times $\boldsymbol{\tau}_K = (\tau_{1,K}, \ldots, \tau_{K-1,K})^T$. Let $\hat{\mathbf{M}}$ denote the reduced transition matrix resulting from $\mathbf{M}$ by deleting row K and column K. By $\mathbf{1}_{K-1}$ we denote the vector consisting of $K-1$ ones. Translating the above equations in matrix notation and solving for $\boldsymbol{\tau}_K$ leads to:

Theorem 2 *The vector of expected first passage times from states* $1, \ldots, K-1$ *to state* K *can be computed by the following equation:*

$$\boldsymbol{\tau}_K = (\mathbf{I} - \hat{\mathbf{M}})^{-1} \cdot \mathbf{1}_{K-1} \tag{5}$$

Theorem 2 shows that the computation of passage times is essentially a matrix inversion problem. The matrix $(\mathbf{I} - \hat{\mathbf{M}})^{-1}$ is called the *fundamental matrix*. Several other system characteristics can be expressed as functions of the fundamental matrix or its elements [1]. If we are interested in the passage times of a stationary distribution we have to use M^{∞} instead of M.

We summarize the results: *The Markov process analysis of CA is restricted to small* n*. For many stochastic cellular automata there exist mathematically a limit (stationary) distribution. But the stationary distribution is not sufficient to characterize a Markov process. It characterizes only the long-term behavior. For the short-term dynamics the passage times are also needed. The computation of the passage times is very expensive.*

It is numerically impossible to compute the exact stationary distribution and the matrix of the expected passage times. But we want to mention that a number of new numerical methods have been proposed to compute the limit distribution and the expected first passage times approximately in reasonable time [6].

In this paper we will describe a different method to compute the distributions. Instead of using the full distribution, we will approximate the distribution $p(\mathbf{x}, t)$ by products of marginal distributions using a small number of parameters. Our ultimate goal is to characterize the results of micro-simulations by a probabilistic analysis.

The deterministic automaton with $\alpha = 1$ has very complex behavior. If n cannot be divided by 3 then the automaton has no attractors at all, but only cycles. Thus this automaton belongs to *class III* defined by Wolfram [16].

For $0 < \epsilon, \alpha < 1$ we have stochastic automata fulfilling the assumptions of theorem 1. It has a unique stationary distribution, depending only on ϵ and α. How do the automata behave if we continuously increase α from 0 to 1? What happens on the transition from $\epsilon = 0$ to $\epsilon > 0$?

4.1 Approximations of the Probability Distribution of 1-D SCA

For notational convenience we set $\theta_i := x_i(t + 1)$, and $\sigma_i := x_i(t)$. We will now derive difference equations involving uni-variate, bi-variate, and tri-variate marginal distributions only. We have by definition for the von Neumann neighborhood

$$p(\theta_i) = \sum_{\sigma_{i-1},\sigma_i,\sigma_{i+1}} p(\theta_i|\sigma_{i-1}, \sigma_i, \sigma_{i+1})p(\sigma_{i-1}, \sigma_i, \sigma_{i+1}) \tag{6}$$

The conditional distribution $p(\theta_i|\sigma_{i-1}, \sigma_i, \sigma_{i+1})$ is uniquely defined by the transitions of the cellular automaton, in our case by the voter model with parameters ϵ and α. But on the right hand side tri-variate marginals appear. For these we obtain

$$\begin{aligned} p(\theta_{i-1}, \theta_i, \theta_{i+1}) = \sum_{\sigma_{i-2},\sigma_{i-1},\sigma_i,\sigma_{i+1},\sigma_{i+2}} & p(\theta_{i-1}, \theta_i, \theta_{i+1}|\sigma_{i-2}, \ldots, \sigma_{i+2}) \\ & p(\sigma_{i-2}, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \sigma_{i+2}) \end{aligned} \tag{7}$$

Thus now marginal distribution of size 5 enter. In order to stop this expansion we approximate the marginal distributions of order 5 by marginal distributions of order 3. From the definition of the SCA we obtain

$$\begin{aligned} p(\theta_{i-1}, \theta_i, \theta_{i+1}|\sigma_{i-2}, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \sigma_{i+2}) &= p(\theta_{i-1}|\sigma_{i-2}, \sigma_{i-1}, \sigma_i) \\ & p(\theta_i|\sigma_{i-1}, \sigma_i, \sigma_{i+1})p(\theta_{i+1}|\sigma_i, \sigma_{i+1}, \sigma_{i+2}) \end{aligned} \tag{8}$$

From the theory of graphical models we obtain the approximation

$$p(\sigma_{i-2}, \ldots, \sigma_{i+2}) \approx p(\sigma_{i-1}, \sigma_i, \sigma_{i+1})p(\sigma_{i-2}|\sigma_{i-1}, \sigma_i)p(\sigma_{i+2}|\sigma_i, \sigma_{i+1}) \tag{9}$$

Inserting the last two equations into equation (7) gives the difference equations for the *tri-variate marginal distributions*. The approximations have to fulfill constraints derived from probability theory.

$$\sum_{\sigma_{i-1},\sigma_i,\sigma_{i+1}} p(\sigma_{i-1},\sigma_i,\sigma_{i+1}) = 1$$

$$\sum_{\sigma_{i-1}} p(\sigma_{i-1},\sigma_i,\sigma_{i+1}) = \sum_{\sigma_{i+2}} p(\sigma_i,\sigma_{i+1},\sigma_{i+2})$$

In the same manner approximations of different precision can be obtained. We just discuss the simplest approximation, using *uni-variate marginal distributions*. Here equation (6) is approximated by

$$p(\theta_i) = \sum_{\sigma_{i-1},\sigma_i,\sigma_{i+1}} p(\theta_i|\sigma_{i-1},\sigma_i,\sigma_{i+1}) p(\sigma_{i-1}) p(\sigma_i) p(\sigma_{i+1}) \tag{10}$$

The approximation by uni-variate marginal distributions leads to n difference equations, but these difference equations are nonlinear. It seems very unlikely that analytical solutions of these equations can be obtained. For *spatially homogeneous* problems we have $p(\theta_i) = p(\theta_{i+1})$. In this case the probabilities do not depend on the locus of the cell. This is the *mean-field limit* known from statistical physics [15]. With $x(t) = 1/n \sum_i p(x_i = 1, t)$ we obtain the *mean-field equation*

$$x(t+1) = (1-\epsilon)x(t)^3 + \epsilon(1-x(t))^3 + 3(1-\alpha)x(t)^2(1-x(t)) + 3\alpha x(t)(1-x(t))^2 \tag{11}$$

The mean-field limit is exact, if the *neighbors* of the cellular automata are chosen *randomly* for each step. For $\epsilon = 0$ and $\alpha < 1/3$ the equation has stable attractors at $x \approx 0$ and $x \approx 1$. For $\alpha > 1/3$ the equation has two stable attractors at $x \approx 0.5$. Thus the mean-field limit approximation indicates a *bifurcation* for $\alpha = 1/3$.

It is also possible to derive a bi-variate approximation for the spatial homogeneous case. A fairly simple expression for $p(x = 1, t)$ can be derived if $p(x = 1, t)$ and $p(x_i = 1|x_{i-1} = 1, t)$ are used as free parameters.

4.2 Numerical Analysis of 1-D Voter Model

We now make a comparison of the approximations. The mean-field approximation predicts a bifurcation at $\alpha = 1/3$. We first discuss the case $\epsilon = 0$ and $\alpha = 0.3033$. The mean-field analysis gives $p(x_i = 1) \to 0$ for $p(x_i = 1, 0) < 0.5$ and $p(x_i = 1) \to 1$ for $p(x_i = 1, 0) > 0.5$. The exact Markov chain computation shows a different picture. The results are displayed in table 1.

Table 1. Different approximations, initial distribution binomially distributed with $p(0)$

	α	$p(x_1 = 1, 0)$	$p(x_1 = 1, 50)$	$p(x_1 = 1.\infty)$	$p(x_2 = 1 \vert x_1 = 1)$
Uni	0.3033	0.2	0.0000	0.0000	
Bi	0.3033	0.2	0.1233	0.0000	0.50
Tri	0.3033	0.2	0.1737	0.1734	1.00
Markov	0.3033	0.2	0.1770	0.1770	1.00
Uni	0.3633	0.8	0.5381	0.0000	
Bi	0.3633	0.8	0.6822	0.5000	0.74
Tri	0.3633	0.8	0.7657	0.5400^*	0.93
Markov	0.3633	0.8	0.7756	0.7756	1.00

For $p(x_i = 1, 0) = 0.2$ we obtain $p(x_i = 1) \to 0.1770$. This result can easily be explained. For $\epsilon = 0$ there are two attractors only, $O = (0, 0, \ldots, 0)$ and $I = (1, 1, \ldots, 1)$. The limit distribution consists of these two configurations only. The approximations Uni and Bi give the mean-field result $p(x_i = 1) \to 0$, but the convergence is much slower for the bi-variate approximation. The tri-variate approximation converges to $p(x_i = 1) \to 0.1737$. This is a little less than the exact value. The tri-variate approximation and the exact Markov analysis give the same results for the conditional probabilities, indicating a high correlation between 1's.

For $\alpha = 0.3633$ the mean-field analysis predicts convergence to 0.5 The Markov analysis gives a different result, namely $p(x_i = 1) \to 0.7756$. Unfortunately the tri-variate approximation seems also to converge to 0.5, but it takes a very long time.

We can now characterize the role which the point $\alpha = 1/3$ mathematically plays. For $\alpha = 1/3$ the value of $p(x_i = 1)$ remains constant, for $\alpha < 1/3$ the value of $p(x_i)$ decreases up to a limit value for $p(x_i = 1, 0) < 0.5$. It increases for $p(x_i = 1, 0) > 0.5$. There is not any kind of phase transition to be observed.

We now investigate the difficult case $\epsilon = 0$ and $\alpha \approx 1$. For $\alpha = 1$ there often exist no limit distribution, but cycles. In table 2 the results are displayed. All approximations reproduce the cycle of size 3 between the configurations $((0, 0, 1, 0, 1)$, $(1, 1, 1, 0, 1)$, $(1, 1, 0, 0, 0))$.

But if we change to $\epsilon > 0$ and $\alpha = 1$ we obtain after a very long time an uninteresting limit distribution. In the limit all configurations are equally likely, with the exception of O and I, which are less likely. In this case we can distinguish between a short term dynamics, where one still can observe the cycles, and a long term dynamics, where the probabilities of the cycle configurations decreases. Thus even the smallest disturbance has a dramatic influence for the stationary distribution.

Table 2. Tri-Variate approximation, initial distribution $p = (0, 0, 1, 0, 1)$, $\alpha = 1$

ϵ	t	$p(x_1 = 1)$	$p(x_2 = 1)$	$p(x_5 = 1)$
0	0	0.000	0.000	1.000
0	1	1.000	1.000	1.000
0	2	1.000	1.000	0.000
0	3	0.000	0.000	1.000
0.01	12	0.082	0.082	0.910
0.01	13	0.885	0.885	0.879
0.01	14	0.837	0.837	0.141
0.01	15	0.203	0.203	0.790
0.01	50	0.501	0.501	0.501

Next we have a look at the expected passage times. Here we have the result that nearby $\alpha = 0$ the passage time from configuration $\mathbf{x} = (0, \ldots, 0)$ to $\mathbf{x} = (1, \ldots, 1)$ is about 10^6. It decreases to $6 \cdot 10^3$ nearby $\alpha = 1$. This is still a large number, if one takes into account that the number of configurations is only $2^5 = 32$.

We summarize the results: *For $\epsilon = 0$ the point $\alpha = 1/3$ is no bifurcation point, as the mean-field approximations indicates. The point plays a unique role in so far that at this point the uni-variate marginal frequencies are not changed. For $\epsilon > 0$ $0 < \alpha < 1$ we have a unique stationary distribution because of theorem 1. But the stationary distribution is in many cases very uninteresting.*

5 Approximations of the Probability Distribution of 2-D CA

For two dimensions the approximation problem gets much harder. We restrict ourselves to the von Neumann neighborhood and bivariate approximations. Because of the symmetry assumed the voter model is now defined by three parameters, usually called ϵ, α_1 and α_2. Since the transition depends on the von Neumann neighborhood, the distribution $p(x_{i,j}(t+1), x_{i+1,j}(t+1))$ can be expressed by means of a distribution of the six surrounding variables. For notational convenience, let $\theta_{i,j}$ be a possible value of the variable $x_{i,j}(t+1)$ and $\sigma_{i,j}$ a possible value of $x_{i,j}(t)$, respectively. We start our analysis as before with

$$p(\theta_{i,j}) = \sum_{\sigma} p(\theta_{i,j})|\underline{\sigma})p(\underline{\sigma}) \tag{12}$$

$$\underline{\sigma} = (\sigma_{i,j}, \sigma_{i-1,j}, \sigma_{i+1,j}, \sigma_{i,j-1}, \sigma_{i,j+1}) \tag{13}$$

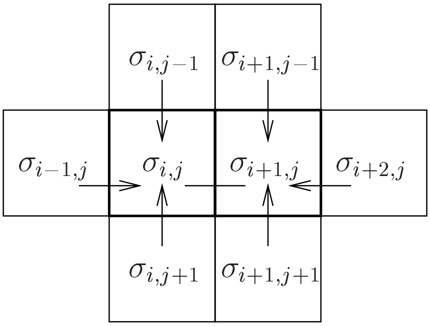


Fig. 1. Factorization of a bivariate approximation. For the approximation eight cells are needed. The arrows indicate directions of conditioning. Dependencies between $\sigma_{i,j-1}$ and $\sigma_{i+1,j-1}$ as well as between $\sigma_{i,j+1}$ and $\sigma_{i+1,j+1}$ are not considered.

We compute the bi-variate approximation as follows (see Fig. 1).

$$p(\theta_{i,j}, \theta_{i+1,j}) = \sum_{\underline{\sigma}} p(\theta_{i,j}, \theta_{i+1,j} | \underline{\sigma})\, p(\underline{\sigma}) \tag{14}$$

where $\underline{\sigma} = (\sigma_{i,j}, \sigma_{i+1,j}, \sigma_{i-1,j}, \sigma_{i,j-1}, \sigma_{i+1,j-1}, \sigma_{i+2,j}, \sigma_{i+1,j+1}, \sigma_{i,j+1})$.

The conditional distribution is uniquely defined by the transitions of the cellular automaton, in our case by the voter model with parameters ε, α_1 and α_2.

$$\begin{aligned} p(\theta_{i,j}, \theta_{i+1,j} | \underline{\sigma}) = {} & p(\theta_{i,j} | \sigma_{i,j}, \sigma_{i-1,j}, \sigma_{i,j-1}, \sigma_{i+1,j}, \sigma_{i,j+1}) \\ & p(\theta_{i+1,j} | \sigma_{i+1,j}, \sigma_{i,j}, \sigma_{i+1,j-1}, \sigma_{i+2,j}, \sigma_{i+1,j+1}) \end{aligned} \tag{15}$$

We approximate the probability $p(\underline{\sigma})$ by the probabilities in the former time step (see Fig. 1).

$$\begin{aligned} p(\underline{\sigma}) \approx {} & p(\sigma_{i,j}, \sigma_{i+1,j}) p(\sigma_{i-1,j} | \sigma_{i,j}) p(\sigma_{i,j-1} | \sigma_{i,j}) p(\sigma_{i,j+1} | \sigma_{i,j}) \\ & p(\sigma_{i+1,j-1} | \sigma_{i+1,j}) p(\sigma_{i+2,j} | \sigma_{i+1,j}) p(\sigma_{i+1,j+1} | \sigma_{i+1,j}) \end{aligned} \tag{16}$$

where

$$p(\sigma_{i-1,j} | \sigma_{i,j}) = \frac{p(\sigma_{i-1,j}, \sigma_{i,j})}{\sum_{\sigma'_{i-1,j}} p(\sigma'_{i-1,j}, \sigma_{i,j})} \tag{17}$$

For the vertical pair, the formula looks completely the same, except that the indices are swapped. (Note that these formulas disregard the connection between $\sigma_{i,j-1}$ and $\sigma_{i+1,j-1}$ or between $\sigma_{i,j+1}$ and $\sigma_{i+1,j+1}$.)

5.1 Consistency of Distributions

In order to justify the approximation of formula (16), we have to check whether it is consistent. There are two conditions which have to be satisfied:

$$\sum_{\theta_{i,j},\theta_{i+1,j}} p(\theta_{i,j},\theta_{i+1,j}) = 1 \tag{18}$$

Furthermore $p(\theta_{i,j}$ has to be the same if computed by different marginalization. There are four ways to calculate $p(\theta_{i,j})$, given by the four neighborhoods in which it takes part:

$$\begin{aligned}
p(\theta_{i,j}) &= \sum_{\theta'_{i-1,j}} p(\theta'_{i-1,j},\theta_{i,j}) \\
p(\theta_{i,j}) &= \sum_{\theta'_{i,j-1}} p(\theta'_{i,j-1},\theta_{i,j}) \\
p(\theta_{i,j}) &= \sum_{\theta'_{i+1,j}} p(\theta_{i,j},\theta'_{i+1,j}) \\
p(\theta_{i,j}) &= \sum_{\theta'_{i,j+1}} p(\theta_{i,j},\theta'_{i,j+1})
\end{aligned}$$

We can show that if these conditions are fulfilled in the initial distributions, then our approximation preserves the consistency. The proof of consistency is much harder for approximations using higher order distributions.

The voter model with five neighbors is currently very popular. Instead of a point bifurcation a "phase separation" in the space generated by α_1 and α_2 is predicted [14]. In this paper we have shown that the "bifurcation" predicted for 1-D automata is an artifact of the approximation. We conjecture that the same result will be true for the 2-D voter model.

6 Conclusion and Outlook

We have reported about our first attempts to make a stochastic analysis of cellular automata. We have shown for the 1-D voter model the observation of a bifurcation is an artifact of the mean-field approximation. The exact Markov process analysis shows a completely different behavior. The stochastic automaton behaves smooth concerning changes of α. Our approximation using tri-variate distributions gives good results for $\alpha < 1/3$, but it also shows a wrong behavior in the limit for $\alpha > 1/3$ and $p(0) > 0.5$. This indicates the problem of our approach. We are sure that a higher order approximation is better than an approximation of lower order, but we do not know how good the approximation is. A formidable task is also to characterize the relation between the outcome of micro-simulations and the results of the stochastic analysis.

We have also presented an approximation using bi-variate distributions for 2-D automata with von Neumann neighborhood. Our approximation consists of $2n^2$ difference equations. These equations can easily be generated on a computer and iteratively solved. Thus our mesoscopic analysis will mainly be based on numerical computations. Analytical expressions can be obtained for homogeneous distributions only, i.e distributions which do not depend on the position of individual cells.

The method we have used for the approximations is now improved in scientific disciplines as different as approximate reasoning, probabilistic logic, graphical models in statistics, and optimization by search distributions [13]. We are confident that by using new results we will derive a practical method for the stochastic analysis of cellular automata and stochastic cellular automata.

References

1. U. N. Bhat *Elements of Applied Stochastic Processes*. Wiley, New York, 1984 .
2. U.Diekmann, R. Law, and J. Metz (eds.). *The Geometry of Ecological Interactions*. Cambridge University Press, Cambridge, 2000.
3. R. Durret. http://www.math.cornell.edu/˜durrett/survey/survhome.html
4. H.A. Gutowitz and J.D. Victor and B.W. Knight. Local structure theory for cellular automata. *Physica D*, 28D:18–48, 1987.
5. H.A. Gutowitz and J.D. Victor. Local structure theory in more than one dimension. *Complex Systems*, 1:57-67, 1987.
6. D. P. Heyman and D. P. OLeary. Overcoming instability in computing the fundamental matrix for a Markov chain. *SIAM J. Matrix Analysis and Applications*, 19:534–540, 1998.
7. A. Ilachinski *Cellular Automata A Discrete Universe*. World Scientific, Singapore, 2001
8. M.I. Jordan. *Learning in Graphical Models*. MIT Press, Cambrigde, 1999
9. S. L. Lauritzen. *Graphical Models*. Oxford:Clarendom Press, 1996.
10. H. Mühlenbein. Darwin's continent cycle theory and its simulation by the Prisoner's Dilemma. *Complex Systems*, 5:459–478, 1991.
11. Heinz Mühlenbein, Thilo Mahnig, and Alberto Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
12. H. Mühlenbein and Th. Mahnig. Evolutionary algorithms: From recombination to search distributions. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, Natural Computing, pages 137–176. Springer Verlag, Berlin, 2000.
13. Heinz Mühlenbein and Thilo Mahnig. Evolutionary optimization and the estimation of search distributions *Journal of Approx. Reason.*, to appear, 2002.
14. N. A. Oomes. Emerging markets and persistent inequality in a nonlinear voter model. In D. Griffeath and C. Moore, editors, *New Constructions in Cellular Automata*, pages 207–229, Oxford University Press, Oxford, 2002.
15. M. Opper and D. Saad, editors. *Advanced Mean Field Methods*, MIT Press, Cambridge, 2001.
16. S. Wolfram. *Cellular Automata and Complexity.* Addison-Wesley, Reading, 1994.

Universality Class of Probabilistic Cellular Automata

Danuta Makowiec and Piotr Gnaciński

Institute of Theoretical Physics and Astrophysics, Gdańsk University,
80-952 Gdańsk, ul.Wita Stwosza 57, Poland
fizdm@univ.gda.pl

Abstract. The Ising-like phase transition is considered in probabilistic cellular automata (CA). The nonequilibrium CA with Toom rule are compared to standard equilibrium lattice systems to verify influence of synchronous vs asynchronous updating. It was observed by Marcq et al. [Phys.Rev.E **55**(1997) 2606] that the mode of updating separates systems of coupled map lattices into two distinct universality classes. The similar partition holds in case of CA. CA with Toom rule and synchronous updating represent the weak universality class of the Ising model, while Toom CA with asynchronous updating fall into the Ising universality class.

1 Introduction

Undoubtedly, the Ising spin system [1] is one of the most fundamental models in statistical mechanics. The simplicity in designing together with richness of cooperative phenomena observed here, generate the fruitful pool for verifying or falsifying theories. One of them is the universality hypothesis. The universality hypothesis claims that many local features of interactions are irrelevant when a thermodynamic system is close to a phase transition [2]. In consequence, at the critical point all equilibrium thermodynamic systems can be grouped in few classes. The classes differ between each other by a set of values, so-called scaling exponents, which describe singularities of basic thermodynamic functions when a system is approaching a critical point.

The Ising model leaves much freedom in the design of the microscopic interactions. Therefore, many stochastic rules have been invented to mimic the dynamics of the model [3] and new ones are still proposed, see , e.g. [4]. In general, any stochastic reversible time evolution such that an elementary step means a single spin flip, is acceptable [5]. The reversibility means that stochastic rule satisfies the detailed balance conditions with respect to the Gibbs measure with the Ising Hamiltonian.

Since the famous Wolfram paper [6] the finite-size Ising model has also been considered as cellular automata (CA). The Q2R cellular automaton has been proposed as an alternative, microcanonical method for studying the Ising model [7]. However, these CA lead to the critical behavior which is unclear. For example,

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 104–113, 2002.

criticality is consistent with the Ising model in some cases and is different in other cases [8]. There are propositions, initiated by Domany and Kinzel [9], to represent the canonical ensemble [10,11]. At the microscopic level such cellular automata and Ising thermodynamic systems are distinct because of the design of stochastic dynamics. The widely applied thermodynamic rule referred to as "Glauber dynamics" uses the following single spin-flip transition rate:

$$P(\sigma_i \to -\sigma_i) = \frac{1}{2}\left[1 - \sigma_i \tanh \beta \sum_{nn} \sigma_{nn}^{(i)}\right] \tag{1}$$

where nn denotes the list of nearest neighbors of σ_i spin and β is proportional to the inverse of temperature. The corresponding probabilistic CA rule is the following:

$$P(\sigma_i \to -\sigma_i) = \frac{1}{2}\left[1 - \varepsilon \sigma_i \mathrm{sign} \sum_{nn} \sigma_{nn}^{(i)}\right] \tag{2}$$

with ε replacing the parameter β. Notice that the stochastic noise perturbs the execution of a given deterministic rule ($\varepsilon = 1$ makes the rule fully deterministic) while temperature β perturbs the influence of the neighboring spins.

In general, the evolution of CA with the probabilistic rule (2) does not satisfy the detailed balance condition [11]. Hence, these CA generate nonequilibrium stochastic systems. The question, whether the stationary state resulting from this evolution is an equilibrium state, is not trivial [12]. It is widely believed that the universality hypothesis spreads to any non-equilibrium stochastic system if the system has the same up-down symmetry as the Ising model [10, 13]. However, Marcq et al. [14] observe that in case of coupled map lattices the synchronous updating leads to the correlation-lenght exponent $\nu = 0.89 \pm 0.02$ which is significantly lower than it is observed in the system with asynchronous updating or in the Ising model for which $\nu_{Ising} = 1$. The relation between coupled map lattices and kinetic Ising models [15] as well as scaling properties of other nonequilibrium stochastic systems, see e.g. [16], are vividly discussed.

The Toom CA is a system of spins on a square lattice where interactions between three spins: North, East and a spin itself, are considered [10,17]. Extensive numerical simulations indicate that the phase transition in these CA is continuous [10,12]. Toom local interactions lead to the important global property of dynamics called eroder property [17]. The eroder property means that any finite island of one phase surrounded by the sea of the other phase will decay in finite number of time steps. Because of this feature investigations are undertaken which examine Gibbsianess of stationary states of Toom systems [18]. Moreover, it is doubtful whether the transition belongs to the Ising universality class [12]. The novel aspect of this paper is that we consider the influence of synchronous and asynchronous updating on critical properties of CA systems. For better understanding of the role of updating we also consider the spin systems with Glauber dynamics.

2 The Models

Toom (TCA) and Glauber (GCA) systems are defined on a square lattice. Each site of the lattice is occupied by a spin σ_i which points either *up* or *down* what is coded by $+1$ or -1, respectively. The future state of a spin σ_i is determined by present states of its three nearest neighbors, named N_i, E_i, C_i chosen as follows:

$$\begin{array}{ccccc} & | & | & | & \\ -\,.\,- & & N_i & -\ \ .\ \ - & . \\ & | & | & | & \\ -\,.\,- & C_i = & \sigma_i & -\ E_i\ - & . \\ & | & | & | & \end{array} \tag{3}$$

Thus, dealing with three-spin interaction on a square lattice we mimic the type of the Ising model on a triangular lattice in which only one type of triangles is considered.

Let $\Sigma_i = N_i + E_i + C_i$. The deterministic dynamic rule is the same in both systems but the stochastic perturbation acts differently, namely:

— in Toom system:

$$\sigma_i(t+1) = \begin{cases} \text{sgn } \Sigma_i \text{ with prob. } & \frac{1}{2}(1+\varepsilon) \\ \\ -\text{sgn } \Sigma_i \text{ with prob. } & \frac{1}{2}(1-\varepsilon) \end{cases} \tag{4}$$

— in Glauber system:

$$\sigma_i(t+1) = \begin{cases} \text{sgn } \Sigma_i \text{ with prob. } & \frac{1}{2}(1+\tanh\varepsilon\ |\Sigma_i|) \\ \\ -\text{sgn } \Sigma_i \text{ with prob. } & \frac{1}{2}(1-\tanh\varepsilon\ |\Sigma_i|) \end{cases} \tag{5}$$

3 Monte Carlo Simulations

We use the standard importance sampling technique to simulate models introduced in the last section. We consider square lattices of linear size L with values of L ranging from $L = 20$ to $L = 100$ and we apply periodic boundary conditions. The computer experiments start with all spins aligned. A new configuration is generated from the old one by the following Markov process: for a given ε the evolution rule either (4) or (5) is employed to each spin in case of synchronous updating, or to a randomly chosen spin when asynchronous updating case is examined. The evolving system is given $100L$ time steps to reach the steady state. Such time interval is sufficient to find all systems studied in stationary ergodic states [19].

When a system is in the stationary state then an expectation value of magnetization m is computed according to a sequence of states $\{\sigma_i(t)\}_{i=1,\ldots,L^2}$:

$$m_L = \frac{1}{T} \sum_{t=1,\ldots,T} \frac{1}{L^2} \sum_{i=1,\ldots,L^2} \sigma_i(t), \tag{6}$$

where t means Monte Carlo steps, i.e., t denotes one simulation step when synchronous updating is performed and L^2 single spin flips in case when the asynchronous updating is examined. $T = 10\,000$ in all experiments. To avoid the possibility that the examined state is attracted by some metastable state, we perform N independent experiments, with N in the range $500, \dots, 5500$.

In case of continuous phase transition on the lattice with a finite size we need to observe the magnitude of the magnetization :

$$|m|_L = \frac{1}{T} \sum_{t=1,\dots,T} \frac{1}{L^2} \left| \sum_{i=1,\dots,L^2} \sigma_i(t) \right| \tag{7}$$

and the n-th moments of magnetization, $n = 2, 4$:

$$m_L^n = \frac{1}{T} \sum_{t=1,\dots,T} \frac{1}{L^2} \left(\sum_{i=1,\dots,L^2} \sigma_i(t) \right)^n \tag{8}$$

due to them we could study properties of the associated finite-size lattice susceptibility:

$$\chi_L = L^2 (m_L^2 - |m|_L^2) \tag{9}$$

and the reduced fourth-order Binder cumulant [20]

$$U_L(\varepsilon) = 1 - \frac{m_L^4}{3(m_L^2)^2} \tag{10}$$

In all systems considered: Toom or Glauber CA with synchronous or asynchronous updating two qualitatively distinct regimes are easily observed: of high magnetization- the ferromagnetic phase, and zero magnetization — the paramagnetic phase, see Fig.1. In each of the system studied the rapid change of magnetization is observed. The huge value of susceptibility is recorded at the same time. One can say that systems pass the ferromagnetic phase transition.

According to the scaling hypothesis, in the thermodynamic systems the singularities of observables: magnetization m, susceptibility χ, correlation length ξ have the power-law form, [2,5]:

$$\begin{aligned} m &\propto (\varepsilon - \varepsilon_{cr})^{\beta} && \text{for} \quad \varepsilon \to \varepsilon_{cr} \quad \text{and} \quad \varepsilon > \varepsilon_{cr}, \\ \chi &\propto (\varepsilon - \varepsilon_{cr})^{-\gamma} && \text{for} \quad \varepsilon \to \varepsilon_{cr}, \\ \xi &\propto (\varepsilon - \varepsilon_{cr})^{-\nu} && \text{for} \quad \varepsilon \to \varepsilon_{cr} \end{aligned} \tag{11}$$

where β, γ and ν are the static critical exponents which determine the universality class of a system. Due to the finite-size lattice theory reliable values for β, γ and ν are accessible [20,5]. For magnetization and susceptibility we obtain the relations:

$$\begin{aligned} |m|_L(\varepsilon_{cr}^*) &\propto L^{-\beta/\nu}, \\ \chi_L(\varepsilon_{cr}^*) &\propto L^{\gamma/\nu}. \end{aligned} \tag{12}$$

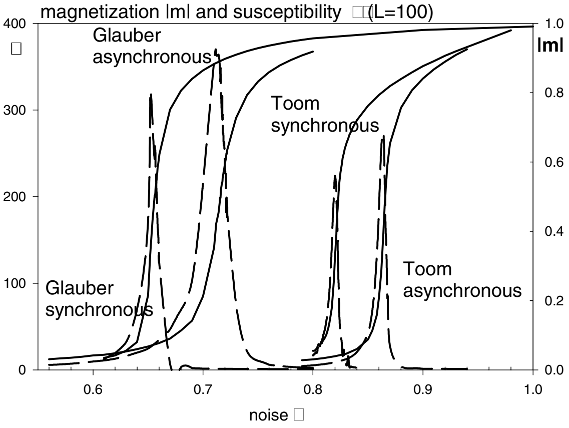


Fig. 1. Noise dependence of the macroscopic observables of TCA and GCA. Magnetization $|m_L|$ (right scale) and susceptibility χ_L (left scale) for all systems considered in case of lattice $L = 100$.

where ε^*_{cr} is the infinite-size transition point. Preliminary identification of ε^*_{cr} can be made by locating the crossing point of the fourth-order cumulants of magnetization (10). The common value of U_L at the crossing point is a universal number determining a universality class also. The Ising kinetic systems are characterized by $U_{Ising} \in (0.610, 0.612)$ [14].

4 Results

4.1 Error Analysis

Since the susceptibility measures the variation of the order parameter, i.e., magnetization, one can draw conclusions about the statistical errors in our simulations from Fig. 1. Standard deviation errors are smaller for other quantities. In the interval where a system is close to the phase transition extra simulations (repeated runs and/or moving ε a little) were performed to assure the validity of the values. The procedure of extracting critical exponents needs estimates for derivatives. Usually, numerical estimates of a derivative consist in approximating the derivative by a finite difference taken between two points neighboring to ε_{cr}. This method, however, is difficult to control in case of noisy data. Moreover, it is sensitive to statistical errors of the values of original data. Instead, we choose to fit experimental data with linear functions. The quality of the fits is estimated by the standard correlation coefficient r^2. In the following, all linear fits together with the corresponding correlation coefficients, calculated by the computer program SigmaPlot2000 by Jandel Scientific, are $r^2 > 0.90$.

4.2 Determination of ε_{cr}

In order to fix the crossing points of cumulants we use 3rd order polynomial fits to U_L data for $U_L \in (0.590, 0.620)$ taken as functions of ε cumulants. These results are given in Fig. 2 .

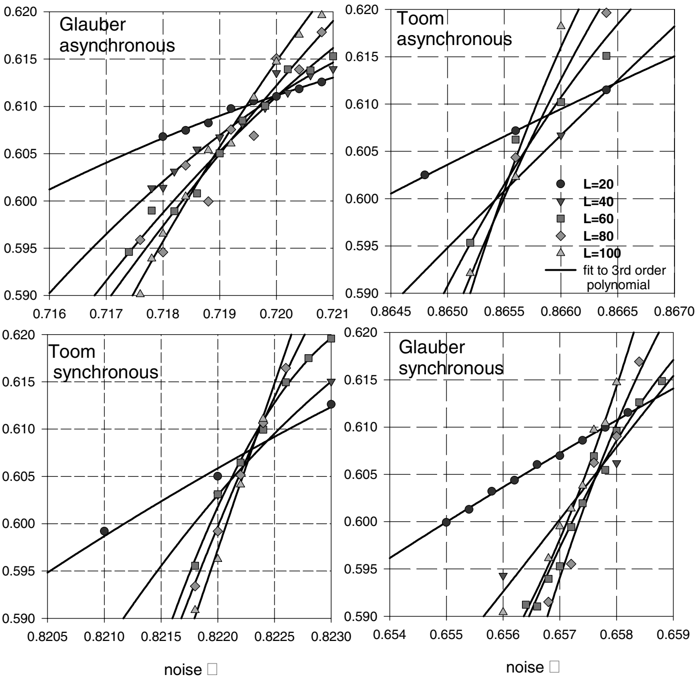


Fig. 2. Estimates of the transition points by Binder's method. Binder's cumulants (10) versus ε are presented for different system sizes $20 \leq L \leq 100$. Symbols correspond to raw data, lines to 3rd order polynomial fits.

Notice that the values of U_L at the crossing points are in remarkable agreement with the ones expected for Ising system.

4.3 Finite-Size Scaling Analysis to Determine ν

The properties of logarithmic derivatives of higher moments of magnetization: $\partial_\varepsilon \log |m|_L$, $\partial_\varepsilon \log m_L^2$, $\partial_\varepsilon \log m_L^4$ at critical point ε_{cr}^* scales with L as $L^{1/\nu}$. Moreover, some related quantities defined as [21]:

$$V_2 = 2[m^2] - [m^4], \qquad V_4 = (4[m] - [m^4])/3, \qquad V_6 = 2[m] - [m^2] \quad (13)$$

where $[m^n] = \ln \partial_\varepsilon m_L^n$ at critical point ε_{cr}^* depend on L like $(1/\nu) \ln L$. This is the typical way to estimate ν. Due to the sufficient density of data we could find linear fits and avoid the problem of the uncertainty of the position of the ε^*_{cr}. Fig. 3 shows the data and linear fits for all functions listed above.

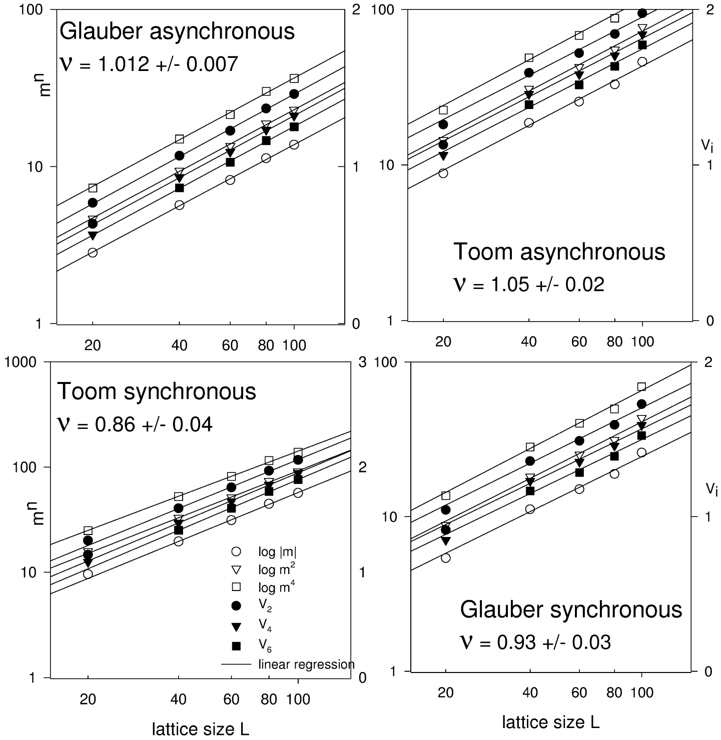


Fig. 3. Estimates for correlation-length exponents ν. Log-log plots of derivatives $\partial_\varepsilon \log |m|_L$, $\partial_\varepsilon \log m^2_L$, $\partial_\varepsilon \log m^4_L$ (left scale) and log-plots of V_2, V_4 and V_6 are presented (right scale). The solid lines correspond to linear fits for L=40,60,80,100.

4.4 Estimates of Other Critical Exponents

Estimates for β and γ are found from relations (12). Again, we work with linear approximations of $|m|_L$, m^2_L, m^4_L and χ_L for ε in critical regimes. We find that values of β/ν and γ/ν are very close to the Ising model ones, i.e., $\beta_{Ising} = 0.125$, $\gamma_{Ising} = 1.75$. Moreover, we could easily calculate ε^*_{cr} by chosing such a value of ε for which the corresponding β/ν and γ/ν take values that are nearest to the Ising ones. This way we can provide the following description of the critical regimes in all considered systems:

	ε^*_{cr}	ν	β	γ	
Glauber CA asynchronous	0.7197	1.012	0.126	1.771	
Glauber CA synchronous	0.6580	0.93	0.116	1.627	(14)
Toom CA asynchronous	0.8658	1.05	0.131	1.837	
Toom CA synchronous	0.8224	0.87	0.109	1.522	

Let us recall that the exponents found for synchronized coupled map lattices are $\nu_{CML} = 0.887$, $\beta_{CML} = 0.111$ and $\gamma_{CML} = 1.55$ [14].

4.5 Data Collapse

Indirect support of the validity of estimates can be provided by observing the collapse of the magnetization and susceptibility data. The following universal functions $\widehat{|m|}$ and $\hat{\chi}$ are expected to emerge:

$$\widehat{|m|}((\varepsilon - \varepsilon^*_{cr})L^{1/\nu}) = L^{\beta/\nu}|m|_L(\varepsilon) \tag{15}$$

$$\hat{\chi}((\varepsilon - \varepsilon^*_{cr})L^{1/\nu}) = L^{-\gamma/\nu}\chi_L(\varepsilon) \tag{16}$$

It turns out that our sets of data lead to the acceptable collapse in all systems, though when the Glauber CA with synchronous updating is considered then some discrepancy is visible, see Fig. 4.

5 Discussion

Ising-like phase transitions studied by probabilistic cellular automata are well described by scaling and finite-size scaling laws valid at equilibrium. Deriving accurate numerical estimates of critical exponents is a difficult task [5]. We believe that the methodology applied here is reliable for three main reasons: (1) the density of data collected in simulations allowed us to use linear approximations of high accuracy (the Pearson coefficient r^2 is always greater than 0.90); (2) since only lattices of sizes greater than 20 were taken into account, it was allowed to neglect corrections to dominant scalings; (3) except estimates for γ all other quantities were derived from two or more functions.

Our simulations have justified the validity of hyperscaling relation which for d-dimensional system takes form:

$$2\beta + \gamma = \nu d$$

and which is known to hold at equilibrium in case of fluctuation-dominated transition, see, e.g., [22]. Moreover, the ratios β/ν and γ/ν follow the Ising behavior. However, the mode of updating divides systems into two classes: $\nu = 1$ and $\nu \approx 0.90$. The weak universality denotes the independence of β/ν and γ/ν on microscopic details [23]. Therefore, we can say that the synchronously updated CA belong to the weak universality class of the Ising model. Furthemore, the same probabilistic cellular automata but updated randomly possess critical properties that are identical with the Ising thermodynamic model.

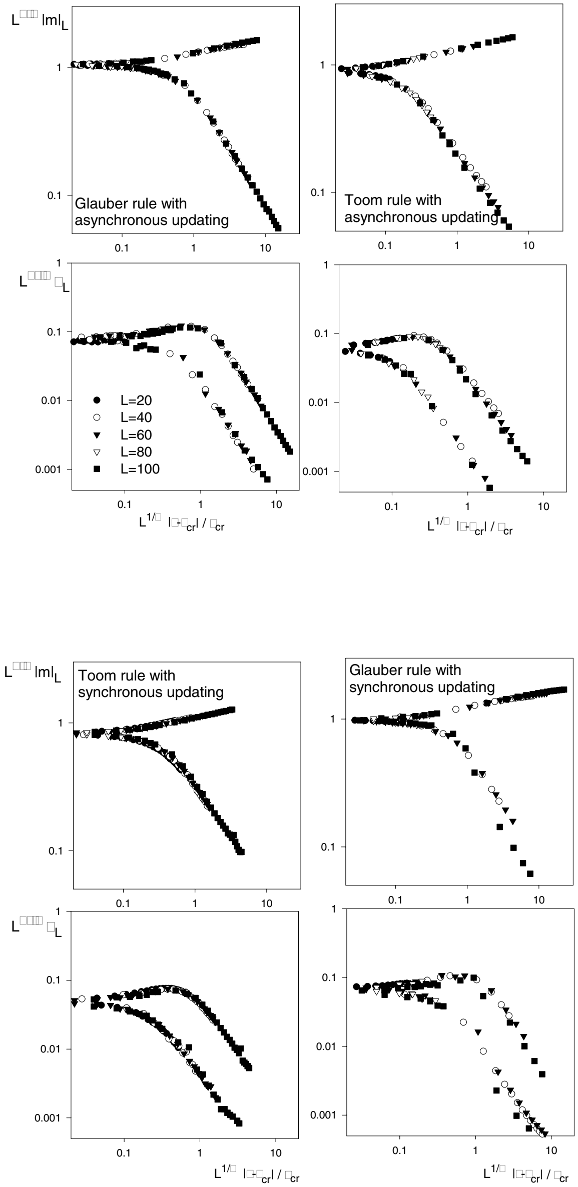


Fig. 4. Data collapses of magnetization -upper line, and susceptibility -lower line, for all systems studied.

Acknowledgments. This work was supported by Gdańsk University Grant, BW-5400-5-0235-2. This work was made possible thanks to a generous allocation of CPU time on supercomputers of Gdansk Academic Computer Center, TASK.

References

1. Lenz, W.: Phys. Zeitschrift **21** (1920) 613
2. Binney, J.J., Dowrick, N.J., Fisher, A.J., Newman, M.E.J.: The Theory of Critical Phenomena (Oxford University Press, Oxford, 1992)
3. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.M., Teller, E.: J. Chem. Phys. **21** (1953) 1087; Glauber, R.J.: J. Math. Phys. **4** (1963) 294; Crutz, M.: Phys. Rev. D **21** (1980) 2308
4. Matsubara, F., Sato, A., Koseki, O., Shirakura, T.: Phys. Rev. E **78** (1997) 3237
5. Landau, D.P., Binder, K.: A Guide to Monte Carlo Simulations in Statistical Physics (Cambridge University Press, Cambridge, 2000)
6. Wolfram, S.: Rev. Mod. Phys. **55(3)** (1984) 601
7. Herrmann, H.J.: J. Stat. Phys. **45** (1986) 145
8. Stauffer, D.: Int. J. Mod. Phys. C **8** (1997) 1263; Stuffer, D.: Commput. Phys. Commun. **127** (2000) 113
9. Domany, E., Phys. Rev. Lett. **52** (1984) 871; Domany, E., Kinzel, W.: Phys. Rev. Lett. **53** (1984) 311
10. Bennett, Ch.H., Grinstein, G.: Phys. Rev. Lett. **55** (1985) 657
11. Lebowitz, J.L., Maes, Ch., Speer, E.R.: J. Stat. Phys. **59** (1990) 117
12. Makowiec, D.: Phys. Rev. E **60** (1999) 3787
13. Grinstein, G.C., Jayaparash, C., Hu, Ye,: Phys. Rev. Lett. **55** (1985) 2527; González-Miranda, J.M., Garrido, P.L., Marro, J., Lebowitz, J.: Phys. Rev. Lett. **59** (1987) 1934; Wang, J-S., Lebowitz, J.: J. Stat. Phys. **51** (1988) 893; Grandi, B.C.S., Figueiredo, W., Phys. Rev. E **53** 5484 (1996)
14. Marcq, D., Chaté, H., Manneville, P.: Phys. Rev. Lett. **77** (1996) 4003; Marcq, D., Chaté, H., Manneville, P.: Phys. Rev. E **55** (1997) 2606
15. Schmuser, F., Just, W., Kantz, H.: Phys. Rev. E **61** (2000) 3675
16. Korneta, W.: Phys. Rev. E **64** (2001) (to appear); Szolnoki, A.: Phys. Rev. E **62** (2000) 7466
17. Toom, A.L., Vasilyev, N.B., Stavskaya, O.N., Mityushin, L.G., Kurdyumov G.L., Prigorov, S.A.: Stochastic Cellular Systems: Ergodicity, Memory, Morphogenesis. Eds Dobrushin, R.L., Kryukov, V.I., Toom, A.L. (Manchester University Press, Manchester, 1990)
18. Maes, Ch., Vande Velde, K.: Physica A **206** (1994) 587; Maes, Ch., Vande Velde, K.: Commun. Math. Phys. **189** (1997) 277; Makowiec, D.: Phys. Rev. E **55** (1997) 6582; Fernandez, R.: Physica A **263** (1999) 117
19. Munkel, Ch., Heermann, D.W., Adler, J., Gofman, M., Stauffer, D.: Physica A **193** (1993) 540
20. Binder, K.: Z. Phys. B **43** (1980) 119
21. Chen, K., Ferrenberg, A.M., Landau, D.P.: Phys. Rev. B **48** (1993) 3249
22. Zinn-Jistin, J.: Quantum Field Theory and Critical Phenomena (Oxford University Press, Oxford, 1989)
23. Suzuki., M.: Prog. Theor. Phys. **51** (1974) 1992

Kinetic Approach to Lattice Quantum Mechanics

Sauro Succi

Istituto Applicazioni Calcolo, viale del Policlinico 137, 00161, Roma, Italy,
succi@iac.rm.cnr.it,
http://www.iac.rm.cnr.it

Abstract. We discuss some lattice discretizations of the Klein-Gordon equation inspired by analogies with discrete kinetic theory.

1 Introduction

Contemporary advances in theoretical physics are clearly indicating that the task of recounciling gravity with quantum physics may require a dramatic swing in our current picture of space-time. In particular, the notion of space-time as a continuum is likely to be untenable at a scale where gravitational and electroweak interactions become comparable in strength [1]. As observed by G.'t Hooft, [2], it is somewhat curious that while the continuum character of space-time is openly questioned in modern quantum field theories, many of these theories still heavily lean on the continuum formalism of partial differential equations.

Discrete lattices are routinely used in computational field theory [3], but, with a few notable exceptions [4], mainly as mere numerical regulators of ultraviolet divergences. Indeed, a major point of renormalization theories is precisely to extract lattice-independent conclusions from the numerical computations.

Given the huge gap between the Planck length (about 10^{-33} cm) and the smallest experimentally probed scales (about 10^{-16} cm) it seems worthwile to investigate the consequences of taking the lattice no longer as mere computational device, but as a bona-fide *discrete* network whose links define the only possible propagation directions for signals carrying the interactions between fields sitting on the nodes of the network. This viewpoint has been addressed before in a series of thorough works exploring quantum mechanics and lattice gravity near the continuum limit [5].

Here we offer a simple view of related issues from a different perspective, namely discrete kinetic theory, in the hope that this additional angle may help shedding further light into this fascinating and difficult topic.

In particular, we shall show that relevant quantum mechanical equations, such as the Schroedinger and Klein-Gordon equations, can be derived within the framework of a *discrete Boltzmann* formalism in which quantum fields are bound to move like classical (although complex) distribution functions along the light-cones of a uniform discrete space-time.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 114–122, 2002.

2 Preliminaries

Let us consider relativistic bosons described by the Klein-Gordon equation (in 1D for simplicity):

$$(\partial_t^2 - c^2\partial_a^2)\phi = -\omega_c^2\phi \tag{1}$$

where

$$\omega_c = mc^2/\hbar$$

is the Compton frequency of the material particle. We describe relativistic bosons by a set of $n = 2D$ complex wavefunctions $\psi_j(x,t)$, such that the j-th component can propagate only along the j-th direction according to the light-cone condition:

$$dx_{ja} = c_{ja}dt \tag{2}$$

where $j = 1, n$ runs over the discrete directions and the latin index a runs over spatial dimensions $1, D$. The discrete wavefunctions are postulated to fulfill the following (complex) Boltzmann equation:

$$\partial_j\psi_j = M_{jk}\psi_k \tag{3}$$

where:

$$\partial_j \equiv c_{i,\mu}\partial_\mu, \quad \mu = 0, D \tag{4}$$

is the covariant derivative along direction j. Here the (anti-symmetric) scattering matrix M_{jk} represents the interaction between the j and k components of the spinorial wavefunction, the analogue of Boltzmann collisions. More precisely each wavefunction has a dual partner propagating along the opposite direction: according to Feynman's picture this dual partner can be identified with an antiparticle propagating backwards in time. The Klein-Gordon equation is obtained by acting upon (3) with the unitary matrix U_{jk}:

$$U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \tag{5}$$

Simple algebra delivers the Klein-Gordon equation for the partner fields $\phi_j = U_{jk}\psi_k$. It is known that that in the adiabatic limit $v/c << 1$, the fast antisymmetric mode (index 2) is enslaved to the slow symmetric one (index 1),

$$|\partial_t\phi_2| << 2m|\phi_2|$$

so that the Klein-Gordon equation reduces to the non-relativistic Schroedinger equation. This limit is formally analogue to the adiabatic relaxation of the kinetic Boltzmann equation to the fluid-dynamic Navier-Stokes equations. This analogy provided the starting point for the quantum Lattice Boltzmann scheme [6] to be described in the next section.

3 Discrete Limit and the Quantum Lattice Boltzmann Equation

Let us now go down to a scale such that space-time discreteness can no longer be ignored. For coinciseness, we shall identify the Planck length and time with the corresponding lattice pitches:

$$t_p = \Delta t, \quad l_p = \Delta x = c\Delta t,$$

The question is to write down a fully discrete equation yielding the Klein-Gordon equation in the large-scale limit. In doing so, we shall require the discrete formulation to comply with fundamental principles of stability, unitarity and space-time locality. These requirements provide a selection rule for the structure of the discrete lattice. Once these are secured, the finite size of the lattice pitch, Δt, should simply affect the quantitative values of the measurable observables, such as the effective mass and group speed of the propagating waves. Along the spirit of taking the lattice for real, we shall try to interpret these departures from the continuum limit as potential trustful physical phenomena at the "Planck" scale.

Let us now proceed to examine a few simple discretizations.

An explicit time-marching along the trajectories would deliver the exact analogue of the lattice Boltzmann equation for classical fluids [7,8,9]:

$$r(x + \Delta x, t + \Delta t) - r(x, t) = ml(x, t) \tag{6}$$

$$l(x - \Delta x, t + \Delta t) - l(x, t) = -mr(x, t) \tag{7}$$

where r, l are shortands for right/left-ward propagation. Note that in physical units m is the ratio of particle Compton frequency $\omega_c = mc^2/\hbar$, to the 'Planck' frequency $\omega_p \equiv 2\pi/\Delta t$:

$$m = \omega_c \Delta t = \omega_c/\omega_p$$

The scheme (6)is immediately shown to be unconditionally unstable for *any* size of Δt, no matter how small. Hence, discrete quantum motion can*not* take place on such type of space-time connectivity.

The typical remedy is the well-known Cranck-Nicolson implicit time-marching:

$$r(x + \Delta x, t + \Delta t) - r(x, t) = \frac{m}{2}[l(x, t) + l(x + \Delta x, t + \Delta t)] \tag{8}$$

$$l(x - \Delta x, t + \Delta t) - l(x, t) = \frac{-m}{2}[r(x, t) + r(x - \Delta x, t + \Delta t)] \tag{9}$$

This corresponds to integrating the right-hand-side (collision term in Boltzmann language) along the characteristic trajectory of the field at the left-hand side. It is immediately seen that, due to the simultaneous dependence introduced by terms at $t+\Delta t$ at the right-hand side, the state of the quantum field at (x, t) depends on *all* sites at the previous time step $t - \Delta t$. From a strict numerical point of view, this implies the solution of a linear algebraic system, a rather heavy task. However, here we shall not deal with numerical issues, but focus instead on the structure of space-time generated by a given numerical discretization. The above scheme is easily shown to be unconditionally stable, for any Δt. Stability comes however at the price of wasting locality, the 'revenge' of space-time for overruling the principle of causality. Thus, again, this type of connectivity cannot correspond to a realizable quantum motion at the lattice space-time scale. Interestingly, *causality can be restored without loosing stability.* Consider the following-apparently minor-variation of the Cranck-Nicolson scheme (see Figure 1):

$$r(x+\Delta x, t+\Delta t) - r(x,t) = \frac{m}{2}[l(x,t) + l(x-\Delta x, t+\Delta t)] \tag{10}$$

$$l(x-\Delta x, t+\Delta t) - l(x,t) = \frac{-m}{2}[r(x,t) + r(x+\Delta x, t+\Delta t)] \tag{11}$$

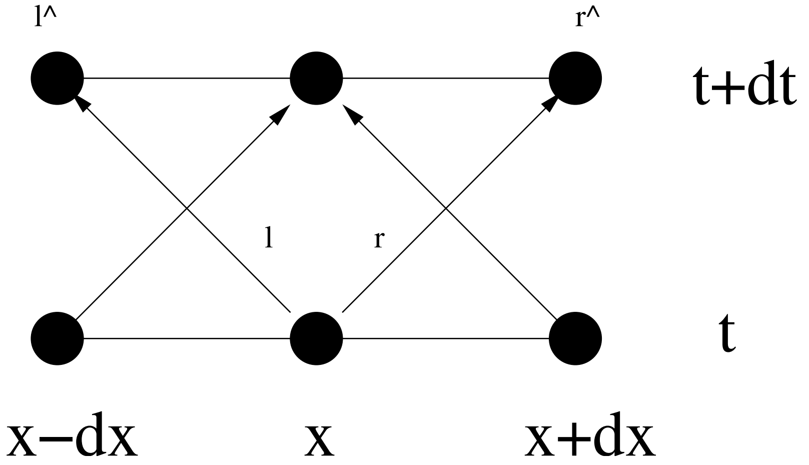


Fig. 1. Space-time diagram of the modified Cranck-Nicolson scheme. The interaction takes place along the arrows labeled r and l

In other words, at the right-hand side, quantum fields are evaluated at the end position of their own flight-path. This is twice non-local, since it adds a non-locality in space on top of the usual time non-locality of the standard Crank-Nicolson scheme. To be noted that *the non-causal link only connects matter-antimatter fields.* These two non-localities annihilate each other in a way to produce a *local* and yet *stable* scheme for any size of Δt. This is easily realized by noting that (10) is a simple 2×2 system which can be solved independently site by site, to deliver the following explicit scheme:

$$r(x+\Delta x, t+\Delta t) = r(x,t)cos\theta - l(x,t)sin\theta \tag{12}$$

$$l(x-\Delta x, t+\Delta t) = r(x,t)sin\theta + l(x,t)cos\theta \tag{13}$$

with

$$cos\theta = (1-m^2/4)/(1+m^2/4)$$

in the role of a particle-antiparticle mixing angle. This mixing angle can also be interpreted as the effective mass, $m' = \theta(m)$, acquired by the particle as a result of the interaction with the discrete lattice. Taylor expansion of both terms in the above expression shows that $m'/m = 1 + O(m^3)$, indicating that the effective mass connects pretty smoothly to its continuum limit (see Figure 2).

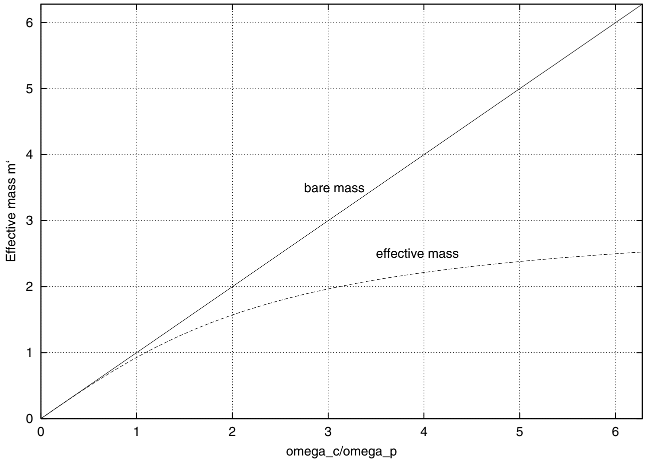


Fig. 2. Lattice renormalization of the bare mass.

In addition, the effective mass stays finite even when the bare mass goes to infinity, a typical signature of renormalization. As to propagating modes, we note that the quantum lattice Boltzmann scheme produces the correct dispersion relation up to second order in the time-step Δt. Standard Fourier analysis of (10) delivers the following dispersion relation:

$$cos\ \omega\Delta t = cos(\theta)\ cos(k\Delta x) \tag{14}$$

It is readily checked that a second order expansion of the cosines of the above dispersion relation yields the well known continuum dispersion relation for relativistic bosons $\omega^2 = c^2k^2 + \omega_c^2$ (encoding Lorentz invariance) up to terms $O(m^3)$. It is tempting to conjecture that (14) is just a specific instance of a more general expression of the form

$$F(\omega/\omega_p) = M(\omega_c/\omega_p)\ F(k/k_p) \tag{15}$$

where the function F and the form-factor M encode the details of the lattice texture. Some general requirements are identified. The consistency requirement:

$$lim_{m\to 0} M(m) = 1,$$

guarantees compatibility of the discrete dispersion relation with its continuum photonic limit $\omega = kc$. Systematic expansions of the form factor $M(m)$ around

$m = 0$ and of the lattice function F around $k = 0$ and $\omega = 0$, deliver corrections to the continuum dispersion relation to all orders in the smallness parameter m. The lowest order (quadratic) terms must reproduce the Klein-Gordon equation.

Renormalizability requires such type of expansions to remain finite even when the bare mass m is sent to infinity, a property which is manifestly fulfilled by the relation (15). More generally, the renormalization condition can be formalized as follows

$$M(\omega_c/\omega_p) = F(\omega_c'/\omega_p) \tag{16}$$

which is:

$$\omega_c'/\omega_p = F^{-1} M(\omega_c/\omega_p)$$

with the further requirement that the function $G \equiv F^{-1}M$ be sublinear ($G(z) < z$, z denoting a generic argument) and remains finite as z is sent to infinity. In view of condition (16), the discrete dispersion relation reads as:

$$F(\omega/\omega_p) = F(\omega_c'/\omega_p)\; F(k/k_p) \tag{17}$$

from which further composition rules in terms of the particle-antiparticle propagators $k/k_p \pm \omega_c'/\omega_p$ can be deduced, depending on the specific form of the lattice function F.

The claim is that, while continuum properties depend on the behaviour of the lattice function $F(z)$ in the vicinity of $z = 0$, the relevant physics at the Planck scale is dictated by the behaviour of the same lattice function in the vicinity of $|z| = 1$.

In order for general relations such as (15), to be interpreted as *more* fundamental than the corresponding continuum limit, one should show that they possess at least the same symmetries (and possibly *more*) than the corresponding continuum partial differential equations. To date, we have not been able to either prove or disprove such a property for the quantum LBE.

In operatorial sense, the discrete space-time partial differential equation associated with (15) reads as follows:

$$F(\frac{\partial_t \psi}{\omega_p}) = M(\frac{\omega_c}{\omega_p}) F(\frac{\partial_x \psi}{k_p})$$

where the functor F (now function of operators) resums an infinite series of partial derivatives associated with the exact representation of discrete derivatives in the finite space and time grid. Finally, it is worth noting that the discrete light-cone condition $\Delta x = c\Delta t$, namely $\omega_p = k_p c$, is essential to all of the above, for it cancels a host of instabilities which would otherwise spoil the entire picture.

4 Numerical Simulations

The scheme (10), named "quantum Lattice Boltzmann scheme" (QLBE) has been demonstrated by actual simulations of some simple quantum mechanical problems [10]. Recently, it has also been extended to the case of non-linear relativistic interactions [11]. Apart from numerical considerations, the message from QLBE seems to be that violation of causality and locality between matter and anti-matter at a lattice scale does not translate into a corresponding violation at larger scales, but it is reabsorbed into an effective mass and group speed of the quantum wavepacket.

This is shown in Figures 3 and 4.

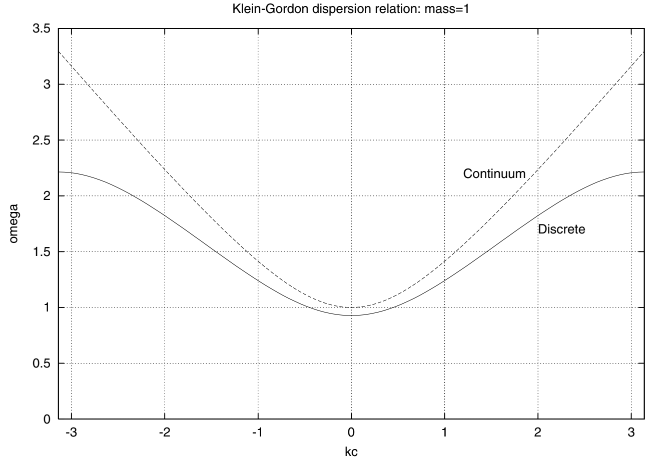


Fig. 3. Continuum and quantum LBE dispersion relations for $m = 1.0$

From Figure 3, we see that in the proximity of the lattice scale ($m = 1$, $kc = \pi$) the discrete scheme exhibits a strong departure from the continuum dispersion relation. In particular, the quantum wave packet is basically stopped by the interaction with the lattice. At just twice larger wavelengths, such an effect is much less dramatic. And much more so at frequencies smaller than the lattice cut-off frequency (Planck frequency) ω_p, as indicated in Figure 4. From this Figure, it is apparent that departures from the continuum dispersion relation remain very negligible up to wavelengths pretty close to the lattice cut-off. As a result the major effect of lattice discreteness is a sensible renormalization of the particle mass and a substantial slowing down of the particle speed. In our opinion, it is by no means obvious that these effects should be regarded as mere “computational artifacts”.

5 Summary

At a scale at which space-time continuity becomes questionable various finite-difference formulations of quantum wave equations become actual statements on the fine-grain structure of space-time. Once this point of view is endorsed,

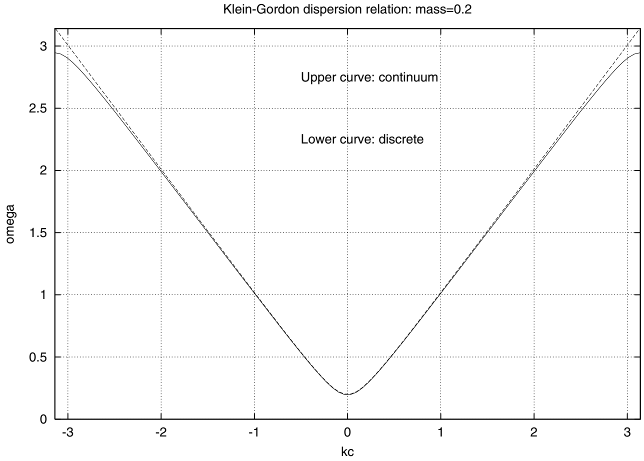


Fig. 4. Continuum and quantum LBE dispersion relations for $m = 0.2$

phenomena that are typically regarded as numerical artifacts or anomalies in the continuum, might be credited as potentially trustful physical effects. Different lattice formulations of a given continuum quantum wave equation can then provide a guideline to identify the admissible structure(s) of space-time around the Planck scale. In this light, the relevant ultraviolet limit is no longer the continuum one, $\omega, k \to \infty$, but rather $\omega, k \to \omega_p, k_p$. If the continuum limit is approached smoothly, say with second order corrections in k/k_p, there is clearly no chance of detecting observable effects of space-time discreteness at any scales but those in the near vicinity of the Planck scale. This conclusion would however change completely in the case of a turbulent space-time with violent bursts generating scales much larger than the original Planck scale. The present work is only scratching the very surface of a difficult subject: quantitative insights require the description of quantum wave-motion with many internal (non-Abelian) degrees of freedom in discrete spacetimes with a self-consistent dynamic curvature. This is a formidable task, but possibly not a completely hopeless one. Random lattices [12], discrete-kinetic versions of Regge calculus [13], cellular automata in dynamic geometries [14], are just a few tantalizing possibilities. As to the lattice Boltzmann approach, a few preliminary efforts in this direction (although confined to classical fluids) have made their appearence in the recent past [15]. Much more work is required to understand whether the discrete kinetic approach can shed any new light into this difficult and fascinating frontier of modern physics.

References

1. Witten, E.: The ultimate fate of space-time, Physics Today, **49** (1996) 24-28.
2. t'Hooft, G.: A confrontation with infinity, Int. J. Mod. Phys. A, **15**, (2000) 4395-4402

3. Creutz, M.: Quarks, gluons and lattices, Cambridge Univ. Press, 1983.
4. Friedberg, R. and Lee, T.D.: Discrete quantum mechanics, Nucl. Phys., B **225** (FS9), (1983) 1-52.
5. Friedberg, R. and Lee, T.D.: Lattice gravity near the continuum limit, Nucl. Phys., B **245** (2), (1983) 343-368
6. Succi, S. and Benzi, R.: Lattice Boltzmann equation for quantum mechanics, Physica D, **69**, 3-4, (1993) 327-332.
7. Benzi, R., Succi, S. and Vergassola, M.: The lattice Boltzmann equation: theory and applications, Phys. Rep. **222**(3), (1992) 145-201.
8. Chen, S., Doolen, G.: Lattice Boltzmann method for fluid flows, Ann. Rev. Fluid Mech., **30**, (1998) 329-363.
9. Succi, S.: The Lattice Boltzmann equation, Oxford Univ. Press, 2001.
10. Succi, S.: Numerical solution of the Schroedinger equation using discrete kinetic theory, Phys. Rev. E, **53**, 2, (1996) 1969-1976.
11. Succi, S.: Lattice Boltzmann equation for relativistic quantum mechanics, Phil. Trans. Roy. Soc. A **360**(1792) (2002) 429-436.
12. Christ, N., Friedberg, R. and Lee, T.D.: Gauge-theory on a random lattice, Nucl. Phys. B **210**, (1982), 310-336 and Random lattice field-theory-general formulation, Nucl. Phys. B**202**, (1982) 89-125.
13. Regge, T.: General relativity without coordinates, Il Nuovo Cimento **19**, (1961), 558-571.
14. Hasslacher, B., Meyer, D. A.: Modeling dynamical geometry with lattice-gas automata, Int. J. Mod. Phys. C, **9**, (1998) 1597-1605
15. Karlin, I.V., Succi, S., Orszag, S.: Lattice Boltzmann method for irregular grids, Phys. Rev. Lett, **82**, (1999) 5245-5248.

Emergence of Self-Replicating Loops in an Interactive, Hardware-Implemented Game-of-Life Environment

André Stauffer[1] and Moshe Sipper[1,2]

[1] Logic Systems Laboratory,
Swiss Federal Institute of Technology in Lausanne,
CH-1015 Lausanne, Switzerland.
andre.stauffer@epfl.ch
[2] Department of Computer Science,
Ben-Gurion University, Beer-Sheva 84105, Israel,
sipper@cs.bgu.ac.il

Abstract. We present the design of an interactive self-replicating loop, wherein the user can physically induce the loop's creation and then control its replication and destruction. After introducing the *BioWall*, a reconfigurable electronic wall for bio-inspired applications, we describe the design of our novel loop and delineate its hardware implementation in the wall.

1 Introduction: Cellular Automata and Self-Replication

The study of self-replicating machines, initiated by von Neumann over fifty years ago, has produced a plethora of results over the years [1,2]. Much of this work is motivated by the desire to understand the fundamental information-processing principles and algorithms involved in self-replication, independent of their physical realization [3,4]. The fabrication of artificial self-replicating machines could have diverse applications, ranging from nanotechnology [5], through space exploration [6], to reconfigurable computing tissues—the latter of which shall be introduced in Section 2.

A major milestone in the history of artificial self-replication is Langton's design of the first self-replicating loop [7]. His 86-cell loop is embedded in a two-dimensional, 8-state, 5-neighbor cellular space; one of the eight states is used for so-called core cells and another state is used to implement a sheath surrounding the replicating structure. Byl [8] proposed a simplified version of Langton's loop, followed by Reggia *et al.* [4] who designed yet simpler loops, the smallest being sheath-less and comprising five cells. More recently, Sayama [9] designed a structurally dissolvable loop, based on Langton's work, which can dissolve its own structure, as well as replicate.

All self-replicating loops presented to date are essentially worlds unto themselves: once the initial loop configuration is embedded within the cellular automaton (CA) universe (at time-step 0), no further user interaction occurs, and the CA chugs along in total oblivion of the observing user.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 123–131, 2002.

In our previous work we described the design of a simple 2×2 self-replicating loop that can be physically activated by the user [10]. In this paper we present the emergence of this interactive self-replicating loop in a Game-of-Life (or, simply, Life) environment [11]. Contrary to the spontaneous emergence demonstrated by Chou and Reggia [12], the user can physically induce the creation of the loop and then control the loop's replication and destruction, two mechanisms which are explained in Section 3. Section 4 presents the detection of the loop in the Life context and the switching to the loop context. Section 5 discusses the hardware implementation of the loop in the interactive reconfigurable computing tissue for bio-inspired applications, the *BioWall.* Finally, we present concluding remarks in Section 6.

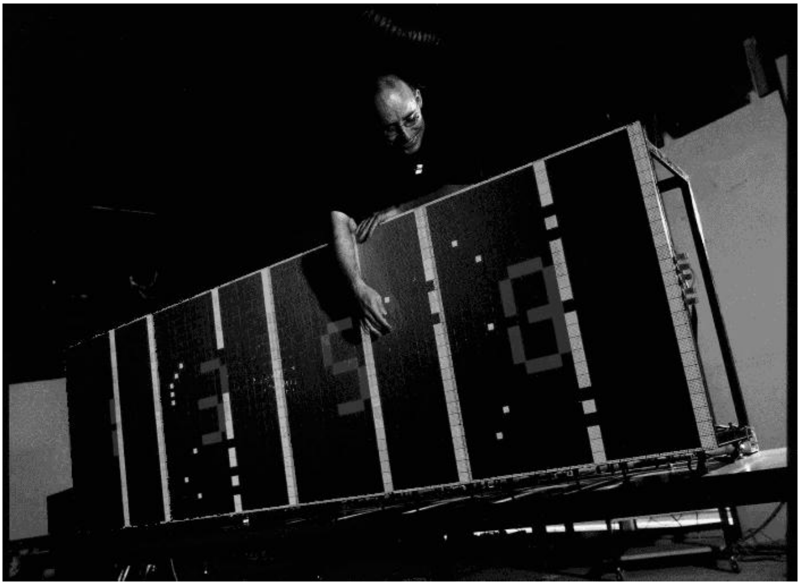


Fig. 1. The BioWall, an interactive reconfigurable computing tissue of $80 \times 25 = 2000$ cells (Photograph by A. Herzog).

2 An Interactive Reconfigurable Computing Tissue

The *BioWall*[1] (bio-inspired electronic wall) is an ongoing project in our laboratory. This wall, first designed to implement an electronic watch with self-repair and self-healing capabilities [13], constitutes a reconfigurable computing tissue

[1] European patent No 01201221.7.

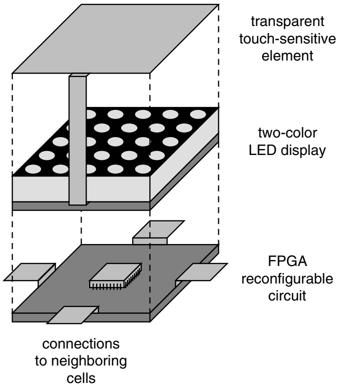


Fig. 2. The basic cell of the interactive reconfigurable computing tissue.

capable of interacting with its environment by means of a large number of touch-sensitive elements coupled with two color light-emitting diode (LED) displays. Figure 1 shows the $80 \times 25 = 2000$ cells of its hardware implementation. Each cell is made up of a transparent touch-sensitive element, a two-color 8×8 dot-matrix LED display, and a reconfigurable Xilinx Spartan XCS10XL FPGA circuit (Figure 2). Within the cell, the transparent touch-sensitive element and the LED display are physically joined by an adhesive film. As each of the cells provides the same connections to its four neighbors, the BioWall is homogeneous and fully scalable.

```
 0: empty component
 1: building component
 2: east-moving signal
 3: north-moving signal
 4: west-moving signal
 5: south-moving signal
 6: left-turn signal
 7: activated left-turn signal
 8: first east-branching signal
 9: second east-branching signal
10: first north-branching signal
11: second north-branching signal
12: first west-branching signal
13: second west-branching signal
14: first south-branching signal
15: second south-branching signal
```

Fig. 3. The seven basic cellular states 0 to 6 used for the idle loop and the nine additional cellular states 8 to 15 involved in the self-replication and self-destruction processes.

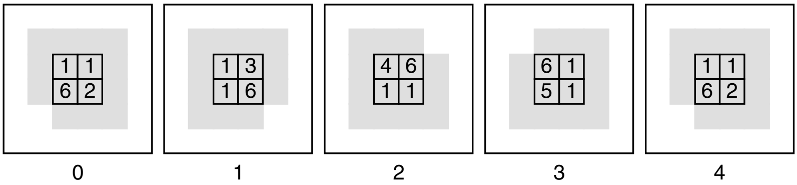


Fig. 4. The four-time-step idle cycle of the inactive loop. The loop context appears in light grey.

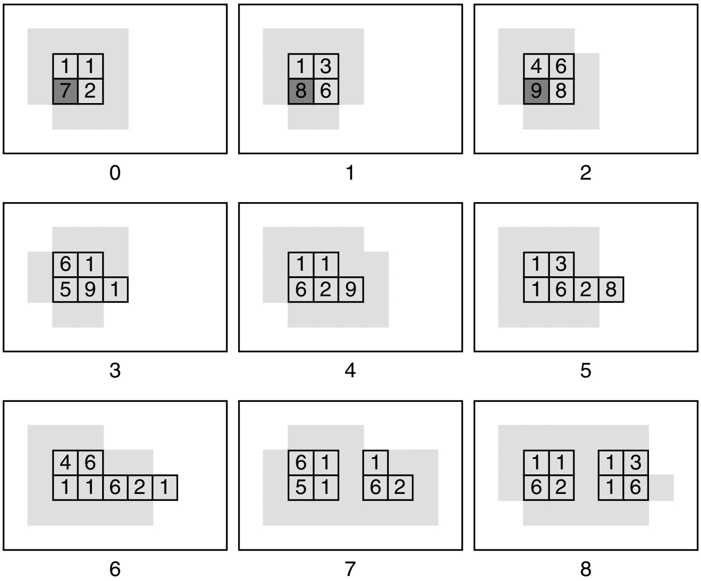


Fig. 5. When the bottom-left cell is activated (touched), the loop self-replicates eastward in an eight-time-step process.

In the electronic watch application (Figure 1), the touch-sensitive element of the BioWall cell acts as a push-button used to render the cell faulty or healthy again. The LED display shows: the boundaries, the spare-cell columns, the current time of each of its four digits, and the faulty or healthy state of the cell.

3 Design of a Life/Loop Automaton Cell

Defined in a two-dimensional cellular space, our automaton cell is made up of a Life state machine, a loop state machine and a pseudo-random number generator.

The Life machine requires a nine-neighbor environment with two states, $Q = 0$ (dead) and $Q = 1$ (living) per cell. The state $Q+$ of the machine at the next

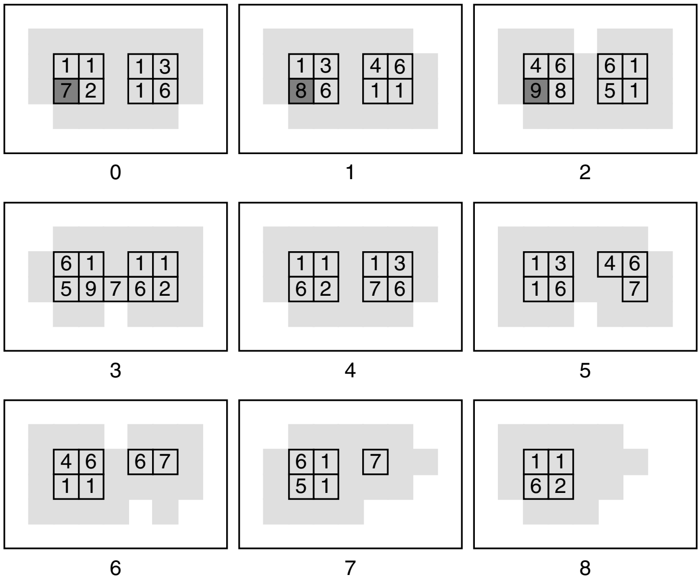


Fig. 6. Activating the bottom-left cell results in an eight-time-step destruction process, since the loop is blocked by another loop and thus cannot self-replicate eastward.

time-step depends on the number of neighboring cells that have a current state of 'living':

- if the number is less than 2, then $Q+ = 0$ (death due to isolation);
- if the number is 2, then $Q+ = Q$ (survival);
- if the number is 3, then $Q+ = 1$ (birth);
- if the number is greater than 3, then $Q+ = 0$ (death due to overpopulation).

The loop machine requires a five-neighbor environment with 16 states per cell (Figure 3; the state-transition rules are given in the Appendix) and defines a minimal 2×2 interactive loop. As long as no physical input is provided, the seven basic states, 0 to 6, lead to the continually undergoing four-time-step cycle of the idle loop (Figure 4). The additional states 7 to 15 take part in the eight-time-step self-replication process (Figure 5) or the eight-time-step self-destruction process (Figure 6), operated when the user activates one of the four cells of the idle loop.

The pseudo-random number generator supplies at each time-step one of the loop machine states 1 to 6. In order to produce randomly these six building states of the idle loop, the generator is implemented as an 8-bit linear feedback shift register working together with a decoder.

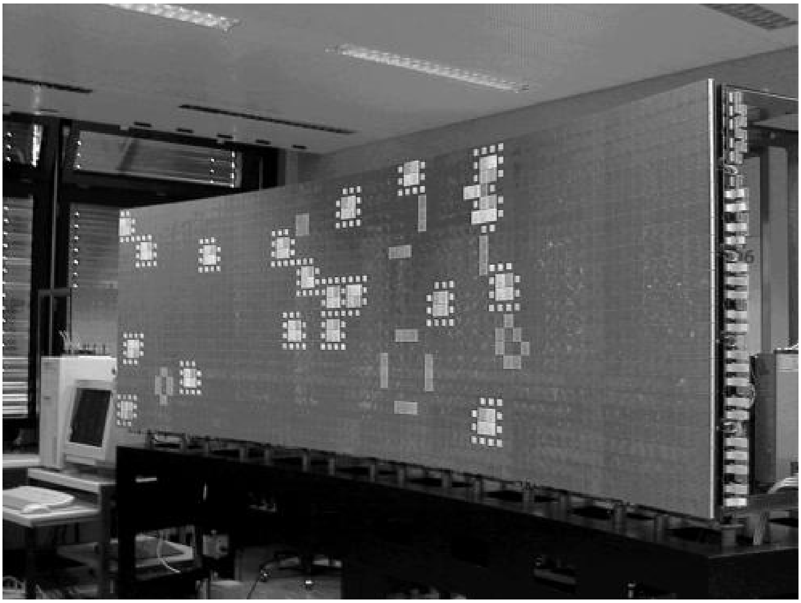


Fig. 7. The BioWall used to physically implement our Life/loop CA. The dots surrounding the loops define visually the loop context (Photograph by A. Badertscher).

4 Emergence of the Loop

The two-dimensional cellular space is initialized as an empty Life environment where the user can induce the birth of cells by physically touching them. In this environment, each living cell presents randomly one of the six idle loop states at each time-step. The context switch between Life and loop happens when a square block of four adjacent cells detects one of the four configurations of the idle loop (Figure 4). Without external activation, this newly created loop then remains idle. Depending on the absence or presence of surrounding loops, the physical activation of the idle loop induces a self-replication or self-destruction process. While performing these processes, the cellular space colonized by replication (Figure 5) shifts to the loop context and the cellular space freed by destruction (Figure 6) returns to the Life context.

5 Hardware Implementation

We have implemented the nine/five environment of the Life/loop CA in our interactive reconfigurable computing tissue as an application of the BioWall (Figure 7). In this hardware implementation, each CA cell corresponds to a cell in the wall. The touch-sensitive element covering the cell's outer surface acts like a digital switch, enabling the user to click on the cell and thereby activate

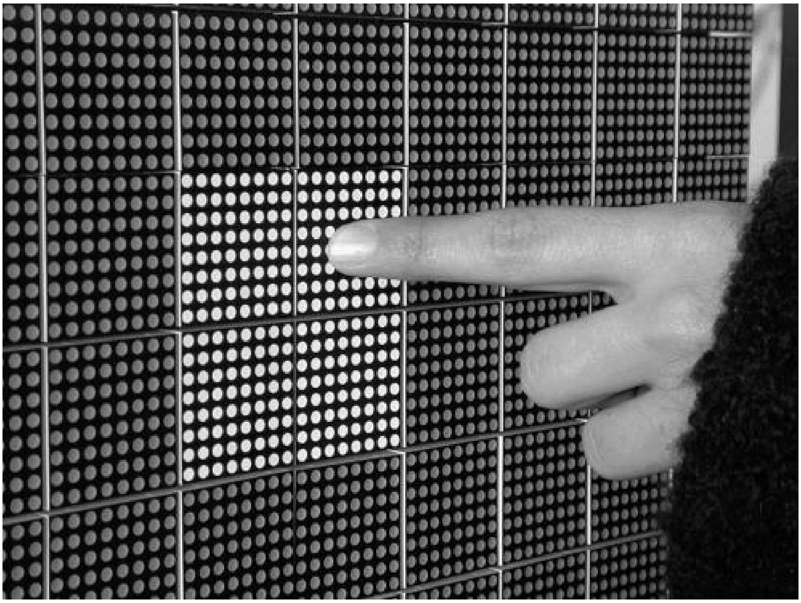


Fig. 8. Touching the BioWall cells in order to physically give birth to a square block in the Life environment (Photograph by A. Badertscher).

either birth in the Life context or self-replication and self-destruction in the loop context (Figure 8).

The field-programmable gate array (FPGA) forming the cell's internal digital circuit is configured so as to implement: (1) the data processing of the external (touch) input, (2) the Life state machine, the loop state machine, the pseudorandom number generator, and the context-switching unit of the CA cell, and (3) the control of the output display. This latter is a two-color LED display that allows the user to view the cell's current state in both contexts and whether the loop cell is in activated or deactivated mode.

6 Concluding Remarks

The ability to interact physically with a CA universe—a little-studied issue—is of fundamental import where cellular devices are concerned: one must be able to enter input and to view the output if any practical application is envisaged [1]. Our ongoing work concerns the design and emergence of interactive cellular replicators implemented in hardware, an issue which we believe will play an important role in the future of such devices.

Acknowledgments. This work was supported in part by the Swiss National Science Foundation under grant 20-63711.00, by the Leenaards Foundation, Lausanne, Switzerland, and by the Villa Reuge, Ste-Croix, Switzerland.

References

1. M. Sipper. Fifty years of research on self-replication: An overview. Artificial Life, 4:237–257, 1998.
2. M. Sipper and J. A. Reggia. Go Forth and Replicate. Scientific American, 285:2:26–35, August 2001.
3. J. von Neumann. Theory of Self-Reproducing Automata. University of Illinois Press, Illinois, 1966. Edited and completed by A. W. Burks.
4. J. A. Reggia, S. L. Armentrout, H.-H. Chou, and Y. Peng. Simple systems that exhibit self-directed replication. Science, 259:1282–1287, February 1993.
5. K. E. Drexler. Nanosystems: Molecular Machinery, Manufacturing and Computation. John Wiley, New York, 1992.
6. R. A. Freitas and W. P. Gilbreath (Eds.). Advanced automation for space missions: Proceedings of the 1980 NASA/ASEE summer study, Chapter 5: Replicating systems concepts: Self-replicating lunar factory and demonstrations, NASA, Scientific and Technical Information Branch (available from U.S. G.P.O., Publication 2255), Washington D.C., 1980.
7. C. Langton. Self-reproduction in cellular automata. Physica D, 10:135–144, 1984.
8. J. Byl. Self-reproduction in small cellular automata. Physica D, 34:295–299, 1989.
9. H. Sayama. Introduction of structural dissolution into Langton's self-reproducing loop. In C. Adami, R. K. Belew, H. Kitano, and C. E. Taylor (Eds.), Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life, pp. 114–122, MIT Press, Boston MA, 1998.
10. A. Stauffer and M. Sipper. Externally controllable and destructible self-replicating loops. In J. Kelemen and P. Sosik (Eds.), Advances in Artificial Life: Proceedings of the 6th European Conference on Artificial Life (ECAL 2001). Lecture Notes in Artificial Intelligence, 2159:282–291, Springer-Verlag, Heidelberg, 2001.
11. M. Gardner. The fantastic combinations of John Conway's new solitaire game "life". Scientific American, 223:4:120–123, October 1970.
12. H. Chou and J. Reggia. Emergence of Self-Replicating Structures in a Cellular Automata Space. Physica D, 110:3–4:252–272, December 1997.
13. A. Stauffer, D. Mange, G. Tempesti, and C. Teuscher. BioWatch: A giant electronic bio-inspired watch. In D. Keymeulen, A. Stoica, J. Lohn and R. S. Zebulum (Eds.), Proceedings of the Third NASA/DOD Workshop on Evolvable Hardware (EH-2001), pp. 185–192, IEEE Computer Society, Pasadena CA, 2001.

Appendix: Specification of the Loop Machine State-Transition Rules

The state-transition rules of the machine implementing the physically controllable self-replicating and self-destructing loops are given in Figure 9. In this figure, where only the rules implying a state change are shown, C, N, E, S, and W correspond to the current states of the cell and of its neighbors to the north, east, south, and west, respectively; $C+$ is the state of the cell at the next time-step. The symbol '$-$' represents a don't-care condition for a binary variable. The binary variable A denotes whether the cell is activated ($A = 1$) or not ($A = 0$). Depending on the value of this variable, there are thus three sets

C,N,E,S,W	->C+	C,N,E,S,W	->C+	C,N,E,S,W	->C+	C,N,E,S,W	->C+
A=-							
0,0,0,2,0	-> 1	1,0,3,8,0	-> 4	5,6,0,1,0	-> 6	7,3,0,0,1	->10
0,0,0,2,1	-> 3	1,4,1,0,0	-> 5	5,6,9,0,0	-> 6	7,0,0,1,4	->12
0,0,3,0,0	-> 1	1,4,10,0,0	-> 5	5,6,11,0,0	-> 6	7,0,1,5,0	->14
0,0,3,1,0	-> 4	1,1,0,0,5	-> 2	5,6,0,14,0	-> 6	8,1,6,0,0	-> 9
0,4,0,0,0	-> 1	1,12,0,0,5	-> 2	5,6,1,15,0	-> 6	8,6,0,0,9	-> 9
0,4,1,0,0	-> 5	2,0,0,0,6	-> 6	5,7,1,0,0	-> 6	8,0,0,0,2	-> 2
0,0,0,0,5	-> 1	2,0,1,0,6	-> 6	6,1,2,0,0	-> 1	9,4,8,0,0	-> 5
0,1,0,0,5	-> 2	2,0,8,0,6	-> 6	6,3,0,0,1	-> 1	9,1,1,0,5	-> 2
0,0,0,0,8	-> 1	2,1,9,0,6	-> 6	6,0,0,1,4	-> 1	9,0,0,0,2	-> 2
0,0,0,0,9	-> 8	2,11,0,0,6	-> 6	6,0,1,5,0	-> 1	10,6,0,0,1	->11
0,0,0,10,0	-> 1	2,13,0,0,6	-> 6	6,3,2,0,1	-> 1	10,0,0,11,6	->11
0,0,0,11,0	->10	2,1,0,0,7	-> 6	6,3,0,1,4	-> 1	10,0,0,3,0	-> 3
0,0,12,0,0	-> 1	3,0,0,6,0	-> 6	6,0,1,5,4	-> 1	11,10,0,0,5	-> 2
0,0,13,0,0	->12	3,1,0,6,0	-> 6	6,1,2,5,0	-> 1	11,1,0,2,1	-> 3
0,14,0,0,0	-> 1	3,10,0,6,0	-> 6	6,12,2,0,0	-> 1	11,0,0,3,0	-> 3
0,15,0,0,0	->14	3,11,0,6,1	-> 6	6,3,0,0,14	-> 1	12,0,0,1,6	->13
1,0,0,0,2	-> 2	3,0,0,6,13	-> 6	6,0,0,8,4	-> 1	12,0,13,6,0	->13
1,0,0,3,0	-> 3	3,0,0,6,15	-> 6	6,0,10,5,0	-> 1	12,0,4,0,0	-> 4
1,0,4,0,0	-> 4	3,0,0,7,1	-> 6	6,3,0,0,8	-> 8	13,0,0,2,12	-> 3
1,5,0,0,0	-> 5	4,0,6,0,0	-> 6	6,0,0,10,4	->10	13,0,3,1,1	-> 4
1,0,0,0,9	-> 9	4,0,6,0,1	-> 6	6,0,12,5,0	->12	13,0,4,0,0	-> 4
1,0,0,11,0	->11	4,0,6,9,0	-> 6	6,14,2,0,0	->14	14,0,1,6,0	->15
1,0,13,0,0	->13	4,0,6,0,12	-> 6	6,0,2,0,1	-> 0	14,15,6,0,0	->15
1,15,0,0,0	->15	4,0,6,1,13	-> 6	6,3,0,1,0	-> 0	14,5,0,0,0	-> 5
1,0,0,2,1	-> 3	4,0,6,15,0	-> 6	6,0,1,0,4	-> 0	15,0,3,14,0	-> 4
1,0,0,2,14	-> 3	4,0,7,1,0	-> 6	6,1,0,5,0	-> 0	15,4,1,1,0	-> 5
1,0,3,1,0	-> 4	5,6,0,0,0	-> 6	7,1,2,0,0	-> 8	15,5,0,0,0	-> 5
A=0							
2,1,0,0,6	-> 6	3,0,0,6,1	-> 6	4,0,6,1,0	-> 6	5,6,1,0,0	-> 6
A=1							
2,1,0,0,6	-> 7	3,0,0,6,1	-> 7	4,0,6,1,0	-> 7	5,6,1,0,0	-> 7

Fig. 9. Loop machine state-transition rules.

of state-transition rules in Figure 9: (1) 108 rules that are performed independently of the activation of the cell ($A = -$), (2) 4 rules performed when the cell is inactive ($A = 0$), and (3) 4 rules performed when the cell is active ($A = 1$).

Spontaneous Emergence of Robust Cellular Replicators

Iker Azpeitia and Jesús Ibáñez*

Computer Languages and Systems Department, University of the Basque Country.
649 Posta Kutxa, 20080 Donostia, Spain
jipibmaj@si.ehu.es

Abstract. An experimental method is developed to evaluate the possibility of spontaneous emergence of self-reproducing patterns from a quasi-random configuration. In order to achieve viability to the emerging patterns and to their components a robust transition table is designed. Genetic reproduction is compared with self-inspection based reproduction in order to conclude that the latter is better adapted to achieve the specified goal.

1 Introduction

One of the first applications, if not the first, of Cellular Automata was the study of the logic of reproduction. John von Neumann [1] regarded reproduction as the ability that living systems exhibit to produce entities and structures as complex as themselves, and opposed to artificial construction processes, in which there is generally a degradation in complexity between the constructing agent and the constructed object. Von Neumann postulated that this ability was the minimum requirement to allow the possibility of open ended evolution, or, to use his own terminology, the possibility of unbounded increase of complexity.

After his awesome work and that of his immediate collaborators [2,3], in which an actual reproducing two-dimensional huge pattern was designed, several authors have developed alternative models to achieve reproduction in simpler conditions than those envisaged by von Neumann, the most notable of those being Chris Langton's [4]. Besides the main goal of achieving reproduction, the different cellular patterns were intended to serve a variety of purposes: to test the role of universality [4,5], to find a minimal model as simple as possible [6,7], to allow reproducers to perform additional tasks [8,9,10] or to try different strategies in the reproducing process [11,12,13,14].

In all the cases the designed patterns are able to reproduce provided they are placed in a *quiescent* cellular space and are not intended to work in an unfriendly environment. This scenario has been proved to be appropriate to test the existence of reproducers under different hypotheses, but not to achieve von Neumann's original goal. If reproduction is to be the basis for any kind of evolution, it can be supposed that variation and selection mechanisms are going to take place. The consequence of both will be a cellular space occupied not only by struggling patterns who try to fill a

* This work has supported in part by the Research Department of the University of the Basque Country under grant 9/UPV 00003230-13707/2001. The authors want to acknowledge the useful comments and suggestions made by the anonymous referees.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 132-143, 2002.

portion of the available cellular space before the others, but also an environment in which the wrecks of unsuccessful previous tries will be common. Sayama [14] has faced this problem with a strategy based on the mechanism of structural dissolution: in his model, the reproducing loops tend to melt into the quiescent environment once they aren't able to reproduce.

Another common feature in the contributions mentioned in the previous paragraphs is that the patterns are placed in the cellular space as an engineering product. Since the question of the origin of life is crucial in understanding life itself [15], we think that it is essential for the study of artificial reproduction (and for the insight that this study could provide in the area of Artificial Life) to pose the question of the minimal conditions for reproducing patterns to appear from a non-reproducing environment.

In this paper we are concerned with the construction of cellular patterns capable of reproduction and robust against external perturbations (that is to say, that can reasonably operate in the presence of nonquiescent environments). The purpose of our design is mainly to explore the possibility of spontaneous emergence of reproducing agents from a unstructured initial distribution of cell states.

Besides that, we have benefited from the fact that the models that have been designed up to now offered different approaches to the problem of reproduction in cellular spaces. In particular, we have compared the relative performance in pattern emerging according to two different reproduction strategies: the genetic one, based on the existence of a self-description of the pattern that is used as a blueprint in reproduction, and the self-inspecting one, in which the reproducing patterns take themselves as the model to guide the replication process.

2 Motivation and Approach

In [12] we proposed a model of a reproducing cellular pattern whose reproduction strategy was based on self-inspection. Then we argued that self-inspection based reproduction was a good choice for testing different properties that were relevant for the role that reproduction plays in the biological world, and in particular, that it could be adequate to study:

- the spontaneous emergence of reproducers
- the self-maintenance of reproducers in unfriendly environments
- the possibility for reproducers to incorporate traits from random variations necessary for evolutionary processes

All three goals share an essential precondition concerning the robustness of the reproducing patterns in nonquiescent environments. On the one hand, the emergence of reproducers cannot originate in a quiescent space, so it seems reasonable to expect that the emerging structures will have to struggle in some kind of primeval soup of cell-states, surrounded by simple components that are candidate to self-organize into reproducing entities. So, if the reproducers were intended to work only in quiescent environments they would never emerge, at least as a result of a continuous growing process. In any case, should any pattern appear, it would by no means be able to reproduce in a "dirty" environment. Thus, in order to be able to produce structure in a non structured environment we need a degree of stability that minimizes the interferences between the emerging patterns and the surrounding matter.

The second goal alludes to the problem of determining to which extent should the reproducing activity of the cellular patterns share its resources with its self-maintenance needs. Even the most complex proposed cellular reproducing patterns cannot be compared with intricate nets of components as are autopoietic systems or autocatalytic nets, but we think that, for the logic of reproduction, the replicating entities must exhibit some degree of identity invariant that requires in its turn a logic of self-production.

Finally, if we have to consider an evolutionary scenario it can be supposed that there will be some random variation source that can be potentially harmful for the normal operation of the reproducers. We also have to take into account that the selective pressure on the patterns can produce "deceases" in the population. This twofold reason determines also that these conditions under which the surviving patterns will have to operate will be far from the blank quiescent space in which most of the proposed reproducing models do operate.

The different models of cellular reproducers have been analyzed and classified into three main groups [16], according to the role that a self-description of the pattern plays in each of them. So, we can distinguish three different reproduction strategies:

- Universal Construction Strategy (UCS)
- Dynamic Description Strategy (DDS)
- Self-Inspection Strategy (SIS)

UCS is best illustrated by von Neumann's automaton [2], in which reproduction is achieved as a fixpoint in the more general feature of universal constructibility. First a mechanism to decode a stored description of an arbitrary cellular pattern is defined. Then a pattern able to construct itself by means of its own description is designed. Langton's loops [4] represent the first and most known instance of DDS, in which the need of decoding the blueprint of the reproducing pattern is removed. This goal is achieved using an executable description, which only need to be transported to the construction area. Finally, as opposed to the previous two, that we can call genetic strategies because of their use of the genotype/phenotype duality, SIS was proposed in our earlier work [12]. Morita and Imai's reversible cellular reproducers [13] have also self-inspecting capabilities. These models don't need to store any kind of self-description because they are able to produce it by self-inspection.

In [16] it is argued that SIS has an intrinsic complexity factor due to the endosemiotic interpretation of all the information handled during the construction process. Moreover, it is suggested that SIS is better adapted to reproduction, and specifically to the emergence of reproducing entities, than genetic strategies.

In this paper we have tried to design an experimental framework in which we can test this hypothesis. Obviously we have been forced to leave apart universal construction-based reproduction: the technical complexity of the models proposed for this paradigm barely allow their implementation, so it doesn't seem feasible to extend their transition tables furthermore to achieve robustness of the patterns. A simple estimation of the cost of such operation can be done extrapolating the data obtained for the costs of our system (see section 3).

On the other hand, the models proposed for reproduction which use dynamic description or self-inspection can be very suitable to be tested together, since they have similar technical complexity (in number of states and transitions or size of the patterns) and an important set of common features. Nonetheless, since robustness is a key characteristic to be achieved, any comparison of these models would be biased if we took them as they have been proposed by their authors. Consequently, we have developed a common scenario to test both reproduction strategies, so that the robustness properties are exactly the same in all different experiments.

Earlier attempts were done by James Reggia *et al.* [17] to achieve the spontaneous emergency of cellular reproducers from a random soup of cell-states. In 99% of the cases emergence of self-replicating loops (2x2-cell patterns) is achieved, and the interaction among these minimal replicators originate the construction of bigger reproducing structures. Robustness of emerging patterns is obtained through a variety of mechanisms. Due to the small size of the reproducers (all their cells are neighbor to each other) the local rules are able to detect whenever one of these small replicating agents is produced. Therefore its cohesion is protected through a shift on the state set (from unbound to bound) and, accordingly, of the rules that guide the dynamic behavior of the system. For instance, the newborn reproducers are able to clean the surrounding area to allow their reproduction dynamics, and there are special failure-detection mechanisms to recycle the cells belonging to an unsuccessful reproducer.

To summarize, our aim is to design a cellular model in which reproducing patterns can be defined and operate in an uncertain environment, and to test the possibility of emergence of such patterns from a random distribution of cell-states. We will define a class of such reproducers and don't want to favor the appearance of any special or canonical reproducing pattern. Moreover, we want to preserve the rule set along all the phases of the system (emergence, reproduction and interaction between patterns), so that the conditions that allow the appearance of replicators are the same as those in which these replicators will operate. Finally, we want to design a cellular transition function that encompasses the features of both self-inspection and genetic models in order to test their relative suitability for the emergence of patterns.

3 Experimental Framework

To implement the approach sketched in the previous section we need to meet three consecutive goals. First of all, a cellular model capable of encompassing two different modes of reproduction. Once a unified framework is designed, it is necessary to enhance the rule set, typically defined just for a quiescent environment, with additional rules that allow a robust behavior of the reproducing patterns in presence of a nonquiescent environment [18]. Finally, the experimental benchmark should be designed, in which the method of generation of the initial configurations (primeval soups) to test the emergence of replicators will be the clue.

3.1 Pattern Choice

Our unified model of replicating patterns is constituted by closed paths through which construction signals can circulate, with a small constructing arm that serves to execute the signals outside the pattern to produce structure. Each constructing signal codes a particular direction (ahead, right or left), and when it arrives to the open end of a path, it makes it one cell longer, following the appropriate direction. This general schema can be found in many of the models introduced by the previous works quoted above.

Some features that are not present in these models have been introduced, however. The paths are constituted of three layers: two sheaths and a “fluid” along which the signals can flow. The two sheaths are different: one of them is intended to serve as the external perimeter of the entire pattern (hard boundary) while the other one faces its internal part (soft boundary). The asymmetry provides a sense for the signals to flow. Besides, the sheaths are intended to protect the pattern from the external perturbations, but the hard one is a stronger barrier than the soft one. The latter shall allow the penetration of any incoming constructing arm (perhaps the own one, or perhaps belonging to another pattern) so that the new-born loop can be closed during reproduction.

As usual the signals duplicate when arriving to the junction of the loop with its construction arm, so that one copy is sent to the outside, while another one remains in the cyclic part of the path.

The shape and size of the reproducing patterns can be arbitrary (see Fig. 1). This feature, which was present in our previous self-inspecting model [12], has been extended also to the genetic version due to the asymmetry of the sheaths. Since the signals get their orientation from polarity of the borders of the path, they can be coded in just one cell. The immediate consequence is that any pattern has enough room to incorporate a detailed self-description. In any case, it is still possible to use more compact descriptions for symmetric patterns.

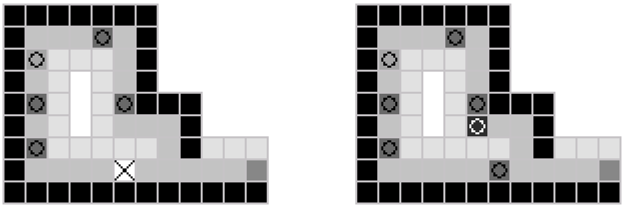


Fig. 1. Examples of reproducing loops belonging to both models. Both carry signals that self-describe the loop by means of the appropriate *construction orders* (with a cercle), but the self-inspecting one (left) includes also a *self-inspecting signal* (with a cross), which is precisely who is generating the signal array.

The number of states is 16. Three of them code the mentioned structural components of the loops, while 12 serve to represent the different constructing, self-inspecting and control signals. A Moore neighborhood is used with radius 1 (8 neighbors are significant for the transition function).

The way reproduction is carried out varies according to the reproduction strategy. In the genetic model the pattern needs a combination of constructing signals that act as its self-description. When they are executed in the outside cellular space, an exact copy of the pattern is generated. On the other side, the self-inspecting model needs only one instance of the self-inspecting signal inside the pattern. This signal runs along the path that defines the loop and reads its shape at the same time. As a consequence, appropriate constructing signals are generated, exported and executed to produce a copy of the pattern.

When the new pattern is constructed it receives the appropriate signals to be able to reproduce in its turn (an array of constructing signals in the genetic model and a single self-inspection signal in the self-inspection model).

In the absence of self-inspection signals both models have exactly the same behavior. Thus, the experiments of the genetic model are characterized by their initial configurations, in which there won't be any self-inspection signal. Since the latter cannot be generated from the other ones, these configurations will only be able to include and/or produce patterns that use the genetic machinery. Conversely, the experiments of the self-inspection model will include self-inspecting signals in their initial configurations, and so it will be possible to include and/or produce patterns that use the self-inspecting machinery.

3.2 Robust Rules

As it has been depicted in previous sections, once defined the basic rules to assure self-reproduction it is necessary to enhance the rule set in order to achieve robustness of the patterns. This task supposes an enormous increase in the size of the active transition table.

Once determined the total number of states *n* and the number of neighbors *k* to be considered in transitions, the total amount of individual transition rules is determined by n^{k+1} (in our case, this would suppose more than 68 billion). Nevertheless, most of these are useless for practical reasons, since the combinations of states that they represent are not going to be considered. Usually the designer chooses the local configurations that are relevant to the problem at hand and defines the corresponding rules, while the other ones are left under a generic "otherwise" condition and produce a default behavior that is particular of each state. The former are those which we have called active rules or explicitly defined rules. When designing the basic rules, that is to say, the ones who assure reproduction in the most favorable conditions provided by a quiescent environment, the active rules include only need to foresee a quite predictable and simple interaction between patterns and the surrounding cellular space. On the contrary, when considering more complex interaction due to the possibility of encountering scattered unstructured (but not necessarily quiescent) in the environment of the patterns, the number of situations to be previewed increases dramatically. This problem has been addressed by Bünzli and Capcarriere [18] using information-theoretical techniques to design fault-tolerant transition tables in a systematic way.

Our approach is closer to the logical design of the automaton than to its empirical behavior. We have used some cellular automata table-editing facilities that help in the construction of these huge tables. One of these is the possibility of defining and using variables in the transitions, which allow a small degree of modular abstraction in the design of the cellular model. A variable is a symbolic object that serves to group related states in order to program jointly their behavior. For instance, if we define the CONST variable as the set {4,5,6} to include all three constructing signal states, and the variable SIGNAL as the set {4,5,6,7,10} to include all transmittable signal states, the we can define the generic transition (or *varsition*):

1 1 1 SIGNAL 2 2 2 CONST -> SIGNAL

which stands for all the 15 transitions that would result from its expansion, that is to say, from the systematic substitution of the variables by their possible values. Then, we define a number of varsitions that are expanded by the system into individual transitions. This process typically produces an important number of repeated transitions (and occasionally conflicts due to cellular programming errors), that are sieved in order to produce the final active transition set.

According to this, we have used 950 different varsitions in the basic rules for reproduction, that have expanded to 1.502.197 individual transitions. After eliminating all the repeated ones, the final number of active transitions was 636.067.

When the transition set has been modified to include robust behavior, the number of relevant local configurations has raised as much as to produce over 30.000 varsitions, that rendered an expanded set of more than 120 million transitions. Nevertheless, since a great amount of varsitions are generated automatically the redundancy degree is higher, and so the final transition state is reduced to about 12 million. The size of the table has heavily conditioned the experimental work: in a Pentium III at 450 MHz with 256 MB RAM we need more than half an hour to load the table (and this process has to be done each time a modified table is compiled). The execution itself is reasonably fast, but it needs 1.5 GB of virtual memory.

The robust set of rules does not only include appropriate transitions to make the patterns more viable in a hazardous environment. Some reconstruction and maintenance rules are also included to improve the stability of components that could organize in order to form a reproducing pattern. An example of the action of these rules is shown in Figure 2.

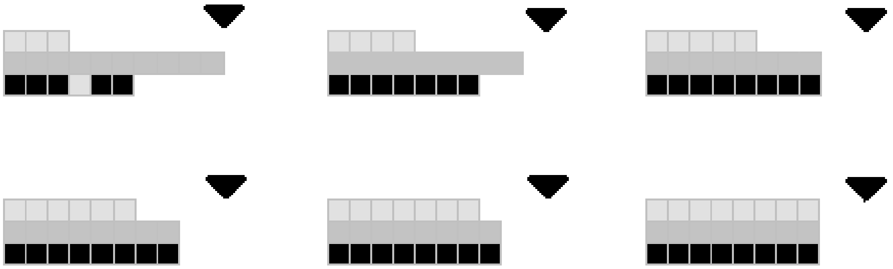


Fig. 2. Example of the maintenance and reconstruction rules. At early stages the structural components of the patterns present in the cellular space (as are small pieces of conducting paths) usually are incomplete and contain errors. The rules can recover some of these small errors. The *black triangle* serves as an absolute reference to see how a piece of nude path (fluid without any sheath) is partially lost and partially recovered.

It is important to note that the maintenance and reconstruction rules do favor the emergence of patterns, but can by no means assure it by themselves. Instead, emergence only occurs as the result the interaction of the small components due to the reproduction mechanisms that are present in the basic rule set. On the other hand, the maintenance and reconstruction rules are operative during all the phases of the cellular experiments, and also have influence in the way stable reproducers interact among them and with the environment.

3.3 Initial Configurations

The very concept of emergence [19] asks for a separation of levels with different entities and relationships, but according to the same laws. In our case it is essential to define which is the situation that corresponds to the pre-replication level and what are the entities that constitute it. Ultimately we have to define a class of cellular landscapes from which replicators are intended to emerge.

A sort of random distribution of cell-states accommodates well with this idea of a previous level in which there are no traces of organized patterns. However, this choice is not practical due to time limitations. So, we have adopted similar criteria as other structure-emergence testing systems like Venus [20], in the sense that the initial configuration state distribution will be biased to favor the affinity of structural states.

To do so we consider a scale of distances from 1 to 5. For each pair of states *(i,j)* we define five affinity factors *A(i,j,d)*, one for each distance *d*, that will be used in determining the probability of a particular arrangement of states in the pseudorandom initial configuration. The value *A(i,j,d)* indicates how a cell in state *i* can influence the probability that a cell at distance *d* is in state *j*. It can be positive or negative, indicating affinity or repulsion between states. The interpretation of the distance scale can be seen in the grid of table 1.

Table 1. Distance grid for a particular cell.

5	4	3	4	5
4	2	1	2	4
3	1	0	1	3
4	2	1	2	4
5	4	3	4	5

To generate an initial configuration the following process is carried on until all the cells are defined (have an initial state):

1. Choose randomly an undefined cell *c*
2. Select the 24 cells which are in the Moore neighborhood of *c* with radius 3.
3. For each state *s* calculate $p(s)= \sum_{c' \in N} A(state(c'),s,distance(c,c'))$
4. Choose the state *t* for cell *c* in a random way so that the probability that *t=s* is proportional to *p(s)* (if *p(s)* is negative consider it as 0)

Note that in step 3 some cells in *N* will be undefined. This problem is overcome defining affinity factors *A(u,s,d)* for the "undefined state" *u*. These special values allow to modulate the global percentage of quiescent states in all the configuration, a parameter that appears to be significant in the experimental results.

With this method some cell states tend to aggregate, and the structural ones have a polymerizing effect (normally in chains no longer than 4 cells), which settle minimal conditions for pattern emergence. An example of initial configuration is shown in Figure 3.

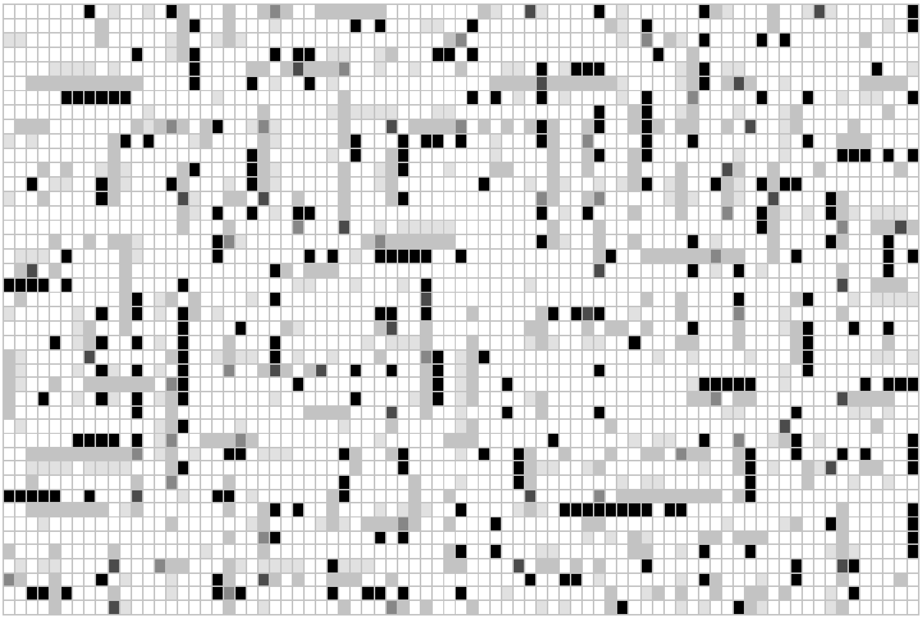


Fig. 3. An example of initial configuration (primeval soup).

4 Experimental Results

First of all we have made some empirical tests to set the optimum quiescent state concentration for the experiments, and we have fixed a value of 70%. Then we have performed 3 sets of 100 batch experiments each. The size of the cellular grid has been 256×256 in all the cases, and the simulations have run up to 1000 steps. In the first set the affinity factors of the initial configurations have included self-inspection signals but no construction signals. They are referred as self-inspecting model experiments. In the second one the design has been the opposite, and we call them genetic model experiments. Finally we have used a third combined set to monitor possible interferences or synergies between the different types of signals.

In the experiments of the genetic model the configuration stabilizes in a space without construction signals, that are wasted away in a short number of steps without generating complete patterns.

In the experiments of the self-inspecting model a variety of structures is produced:

- linear paths without any signal (initially the most common one)
- inverted loops (with the soft sheath towards the outside)
- standard sterile loops (without signal and/or arms)
- structures where the orientation of the path changes
- loops with more than one arm
- standard reproducers

Among the latter there are true reproducers, and after a number of steps these are the principal structures, as can be seen in Figure 4. Some reproducers suffer variation in shape or viability due to their interaction with other structures, which indicates that the robustness induced by the set of rules has a limit.

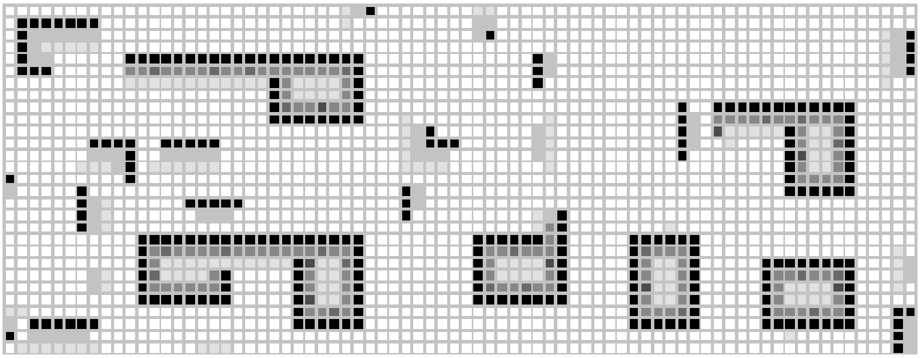


Fig. 4. Detail of the result of a successful experiment of the self-inspecting model. Different copies of true replicators are the main remaining structures after 500 steps.

When self-inspection is present, the rate of success in emerging self-replicating units is fairly high. Out of the 100 experiments of the self-inspecting model, only in 6 there was no emergence, while in 100% of the combined experiments some kind of reproducing pattern did appear. This rates are obtained for the mentioned 70% for the density of quiescent states. Making an emptier or a more crowded environment reduces significantly these rates, though quantitative measures have not been taken.

A very interesting phenomenon is produced in three experiments. Sometimes a self-inspecting loop looses its reading signal due to its interaction with surrounding structures, and so a potential genetic loop is produced, with only construction signals inside. In these cases the self-inspecting path closes losing its self-inspection signal, but it doesn't manage to retain its arm an so it cannot further reproduce.

5 Conclusions

We have tried to contribute with experimental evidence to our earlier hypothesis about the self-inspection strategy as a better adapted one. We have defined a neutral model in which no bias could be attributable in favor of any of the paradigms. Additionally, we have advocated for a uniform design, in which the conditions for emergence would be substantially the same as those in which the emerged patterns would operate, with the only exception of the changes directly produced in the cellular space by the new reproducing units.

Our efforts have been directed also toward the design of a model in which the reproducing entities would be embedded in an unfriendly environment, as opposed to most reproducing cellular automaton models, which are intended to operate in laboratory conditions.

The behaviors observed can be summarized as follows:

- Primeval soups in which no replicating loop is placed are generated by means of a random distribution pattern biased towards the formation of small path components.
- Self-inspection and construction signals start a dynamic behavior in which some path components degrade while some other are reconstructed and combined together.
- In genetic model experiments the dynamics induced by the construction signals do not yield the formation of reproducing loops.
- In self-inspecting model experiments the dynamics induced by the self-inspecting signals originate the spontaneous emergence of replicators with a high probability.
- Newborn replicating loops experiment phenotypic variation due to the interaction with other units or with the environment.
- Newborn replicating loops do not experiment genotypic variation because the self-inspecting signal recovers the encoding of the loop's shape.

Our experimental results suggest that self-inspection based reproduction takes advantage of the environmental irregularities far better than genetic reproduction. The emergence of reproducing patterns seems to occur without difficulty in the former. This is not surprising, since self-inspection based reproduction is a higher order operation than pattern construction, and has fewer operating requirements than genetic reproduction.

It can be interesting to note that a small amount of experiments suggest that self-inspection reproduction could be a precursory stage for the genetic one. Although the lack of additional experimental data doesn't allow to test further this hypothesis, it seems conceivable an ulterior stage in which genetic reproducers would substitute the earlier and more primitive self-inspecting entities. Thus, despite having a better chance to trigger the reproduction dynamics, self-inspection would be taken over by the evolutionary advantages of genetic reproduction.

References

1. von Neumann J.: Theory of reproducing autómata, edited and completed by A.W. Burks, University of Illinois Press(1966).
2. Burks A.W.: Von Neumann's reproducing automata. In *Essays on cellular automata*, edited by A.W. Burks. University of Illinois Press (1970).
3. Thatcher J.W.: Universality in the von Neumann cellular model. In *Essays on cellular automata*, edited by A.W. Burks. University of Illinois Press (1970).
4. Langton C.G.: Self-reproduction in cellular automata. Physica **10** D 135-144 (1984).
5. Perrier J.-Y., Sipper M. and Zahnd J.: Toward a viable, self-reproducing universal computer. Physica **97** D, 335-352 (1996)
6. Byl J.: Self-reproduction in small cellular automata. Physica **34** D (1989) 295-299.
7. Reggia J.A., Armentrout S.L., Chou H.-H. and Peng Y.: Simple systems that exhibit self-directed replication. Science **259**, 1282-1287 (1993).
8. Tempesti G.: A new self-reproducing cellular automaton capable of construction and computation. In *Advances in Artificial Life*, *Proceedings of the ECAL'95* edited by F. Morán, A. Moreno, J.J. Merelo and P. Chacón. Springer (1995).

9. Petraglio E., Henry J.-M. and Tempesti G.: Arithmetic operations on self-replicating cellular automata. In *Advances in Artificial Life, Proceedings of the ECAL'99*, edited by D. Floreano, J.-D. Nicoud and F. Mondada. Springer (1999)
10. Stauffer A. and Sipper M.: Externally controllable and destructible self-replicating loops. In *Advances in Artificial Life*, *Proceedings of the ECAL'01*, edited by J. Kelemen and P. Sosik. Springer (2001).
11. Codd E.F.: Cellular automata. Academic Press (1968).
12. Ibáñez J. Anabitarte D., Azpeitia I., Barrera O., Barrutieta A., Blanco H. and Echarte F.: Self-inspection based reproduction in cellular automata. In *Advances in Artificial Life, Proceedings of the ECAL'95*, edited by F. Morán, A. Moreno, J.J. Merelo and P. Chacón. Springer (1995).
13. Morita K., and Imai K. : A simple self-reproducing cellular automaton with shape-encoding mechanism. In *Artificial life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, edited by C.G. Langton and T. Shimohara, MIT Press (1997).
14. Sayama H.: A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. Artificial Life **5**:4, 343-365 (1999).
15. Dyson F.: Origins of life, Cambridge University Press (1985).
16. Etxeberria A. and Ibáñez J.: Semiotics of the artificial: the "self" of self-reproducing systems in cellular automata. Semiotica, **127**:1/4, 295-320 (1999)
17. Reggia J.A., Lohn J.D. and Chou H.-H.: Self-replicating and self-repairing multicellular automata. Artificial Life, **4**:3, 283-302 (1998)
18. Bünzli D.C. and Capcarrere M.S.: Fault-tolerant structures: towards robust self-replication in a probabilistic environment. In *Advances in Artificial Life*, *Proceedings of the ECAL'01*, edited by J. Kelemen and P. Sosik. Springer (2001).
19. Emmeche C., Køppe S. and Stjernfelt F.: Explaining emergence: towards an ontology of levels. Journal for general philosophy or science **28**, 83-119 (1997).
20. Rasmussen S.: Dynamics of programmable matter. In *Artificial Life II*, edited by C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen. Addison-Wesley (1992).

Emergence of Macro Spatial Structures in Dissipative Cellular Automata

Andrea Roli[1] and Franco Zambonelli[2]

[1] DEIS
Università degli Studi di Bologna
aroli@deis.unibo.it
[2] DISMI
Università di Modena e Reggio Emilia
franco.zambonelli@unimo.it

Abstract. This paper describes the peculiar behavior observed in a class of cellular automata that we have defined as *dissipative*, i.e., cellular automata that are *open* and makes it possible for the environment to influence their evolution. Peculiar in the dynamic evolution of this class of cellular automata is that stable macro-level spatial structures emerge from local interactions among cells, a behavior that does not emerge when the cellular automaton is *closed*, i.e., when the state of a cell is not influenced by the external world. Moreover, we observed that Dissipative Cellular Automata (DCA) exhibit a behavior very similar to that of dissipative structures, as macro-level spatial structures emerge as soon as the external perturbation exceeds a threshold value and it stays below the "turbulence" limit. Finally, we discuss possible relations of the performed experiments with the area of open distributed computing, and in particular of agent-based distributed computing.

1 Introduction

In this paper, we present and discuss a set of experiments that we have performed on a new class of cellular automata that we have defined as *Dissipative Cellular Automata* (DCA). DCA differ from "traditional" cellular automata in two characteristics: while "traditional" cellular automata are composed of cells that interact with each other in a synchronous way and that are influenced in their evolution only by the internal state of the automata themselves, dissipative ones are *asynchronous* and *open*. One the one hand, cells update their status independently of each other, in an "autonomous" way. On the other hand, the automata live dipped in an environment that can directly influence the internal behavior of the automata, as in open systems.

The reported experiments show that DCA exhibit peculiar interesting behaviors. In particular, during the evolution of the DCA, and despite the out-of-equilibrium situation induced by the external environment, stable macro-level spatial structures emerge from local interactions among cells, a behavior that does not emerge when the cellular automaton is synchronous and closed (i.e.,

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 144–155, 2002.

when the state of a cell is not influenced by the environment). Furthermore, ordered patterns emerge, like in dissipative systems [11], when the external perturbation is higher than a critical value and they are present for a specific perturbation strength range.

On this basis, the paper argues that similar sort of macro-level behaviors are likely to emerge as soon as multiagent systems (or likes) will start populating the Internet and our physical spaces, both characterized by intrinsic and unpredictable dynamics. Such behaviors are likely to dramatically influence the overall behavior of our networks at a very large scale. This may require new models, methodologies, and tools, explicitly taking into account the environmental dynamics, and exploiting it during software design and development either defensively, to control its effects on the system, or constructively, as an additional design dimension.

This paper is organized as follows. Sect. 2 defines DCA as CA characterized by asynchronous dynamics and openness. In Sect. 3 we describe experiments and we discuss the results obtained. In Sect. 4 the relation between DCA and dissipative systems is further investigated, by showing the typical system behavior as a function of the external perturbation. We conclude with Sect. 5 outlining potential applications and future work.

2 Dissipative Cellular Automata

In this section we first briefly recall the definition of Cellular Automata (CA) and introduce the terminology that will be used in the following. Then, we define Dissipative Cellular Automata (DCA) as CA characterized by *asynchronous* dynamics and *openness*.

A CA is defined by a quadruple $\mathcal{A} = (S, d, V, f)$, where S is the finite set of possible states a cell can assume, d is the dimension of the automaton, V is the neighborhood structure, and f is the local transition rule. In this work we assume what follows:

- The automaton structure is a 2-dimensional discrete grid closed to a 2-dimensional torus (namely, $N \times N$ square grids with wraparound borders).
- The neighborhood structure is regular and isotropic, i.e., V has the same definition for every cell.
- f is the same for each cell (uniform CA).

The quadruple $\mathcal{A}$ specifies the static characteristics of an automaton. The complete description of a CA requires the definition of its dynamics, i.e., of the dynamics ruling the update of the state of CA cells. In general, the dynamics of a CA assumes a discrete time. The usual definition of CA is with synchronous dynamics: cells update their state in parallel at each time step.

Synchronous CA of this kind have been deeply studied [19,1] and have also an interesting biological/systemic interpretation: cells can be interpreted as alive/dead, or system elements active/inactive depending on their state.

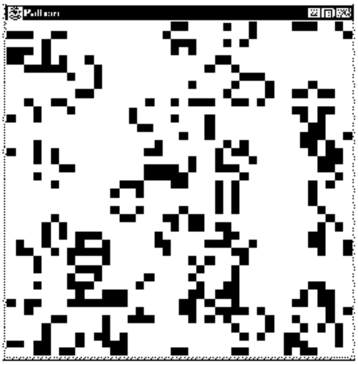


Fig. 1. A synchronous CA having reached a cyclic attractor.

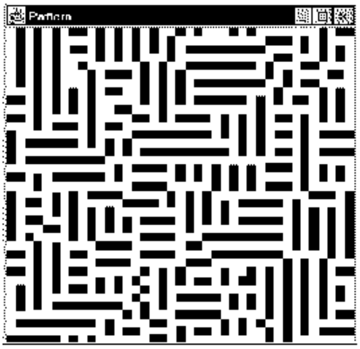


Fig. 2. A fixed point reached by an asynchronous CA. The initial state is the same of the synchronous one.

2.1 Asynchronous Dynamics

Accordingly to the most accepted terminology [6,10,13], a CA is *asynchronous* if cells can update their state independently from each other, rather than all together in parallel, according to a dynamics that can be either *step-driven* or *time-driven*.

In *step-driven* dynamics, a kind of global daemon is introduced, whose job is to choose at each time step one (and only one) cell to update. In *time-driven* dynamics, each cell is assumed to have an "internal clock" which wakes up the cell and makes it update. Also, time-driven dynamics provides for a more continuous notion of time. The updating signal for a cell can be either deterministic (e.g., every time steps) or probabilistic (e.g., the probability that the cell update its state is uniformly distributed), and the next state of a cell is selected on the basis of the current state of neighboring cells.

In the experiments presented in this paper, CA have an asynchronous time-driven dynamics: at each time step, a cell has a probability λ_a to wake up and update its state. The update of a cell has been implemented as atomic and mutually exclusive among neighbors, without preventing non-neighbor cells to update their state concurrently.

In general, it has been observed that the asynchronous CA exhibits behaviors which are very different from the ones of their synchronous counterparts, both in terms of transient and final attractor. Both the dynamics have the same fixed points [13], i.e., attractors that are fixed points under synchronous dynamics are fixed points also under asynchronous dynamics and vice versa. Nevertheless, trajectories in the state space and basins of attraction can be very different and some of the final attractors reached under asynchronous dynamics may be reached with lower probability under synchronous one.

As an example, Fig. 1 and Fig. 2 show the steady states reached by a synchronous and an asynchronous CA, starting from the same initial (random) state. These are characterized by a Moore neighborhood structure (the neighbors of a cell are the 8 one defining a 3×3 square around the cell itself) and the following transition rule:

$f = \{$ a dead cell gets alive iff it has 2 neighbors alive; a living cells lives iff it has 1 or 2 neighbors alive$\}$.

Under asynchronous regime, CA usually reaches a fixed point that its synchronous counterpart has never been observed to be able to reach in all the experiments we performed.

2.2 Openness

Most of CA studied so far are closed systems, as they do not take into account the interaction between the CA and an environment. Instead, the class of CA that we have studied is, in addition to asynchronous, *open*: the dynamic behavior of the CA can be influenced by the external environment.

From an operative point of view, the openness of the CA implies that some cell can be forced from the external to change its state, independently of the cell having evaluated its state and independently of the transition function (see Fig. 3).

From a thermodynamic perspective, one can consider this manifestation of the external environment in terms of energy flows: forcing a cell to change its state can be considered as a manifestation of energy flowing into the system and influencing it [11]. This similarity, together with the fact that the activities of the cells are intrinsically asynchronous and that the externally forced changes in the state of cells perturb the CA in an irreversible way, made us call this kind of CA as *Dissipative Cellular Automata* (DCA).

From a more formal point of view, a DCA can be defined as follows:

- $\mathcal{A} = (S, d, V, f)$,
- asynchronous time-driven dynamics (with probabilityλ_a),
- a perturbation action $\varphi(\alpha, \mathcal{D}, \lambda_e)$.

where $\mathcal{A}$ is the quadruple defining the CA, the dynamics is the one already discussed in Subsect. 2.1, and the perturbation action φ is a transition function which acts concurrently with f and can change the state of any of the CA cells to a given state $\alpha \in V$ depending on some probabilistic distribution $\mathcal{D}$, independently of the current state of the cells and of their neighbors. Specifically, in our experiments $\alpha = 1$ (i.e., the cell if forced to be "alive") and $\mathcal{D}$ is a distribution such that each cell has probability λ_e to be perturbed.

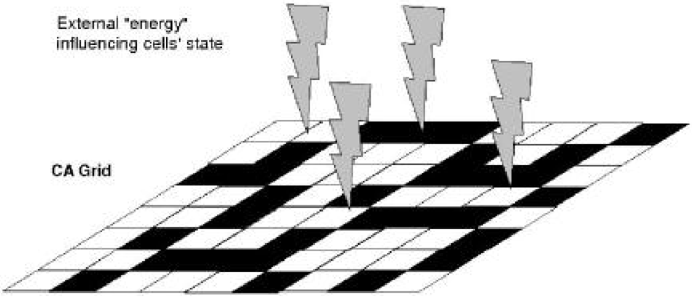


Fig. 3. The basic structure of a dissipative cellular automaton: the environment influences the state of cells by injecting "energy".

3 Emergent Behaviors

The behavior exhibited by DCA is dramatically different from both their synchronous and closed asynchronous counterparts.

In general, when the degree of perturbation (determined by λ_e) is high enough to effectively perturb the internal dynamic of the DCA (determined by the rate of cell updates λ_a) but it is still not prevailing over it so as to make the behavior of the DCA almost random (which happens when λ_e becomes comparable λ_a), peculiar patterns emerge. Since the external perturbation strength is relative to the cells update rate, i.e., the amount of perturbation is given by the ratio between external and internal update rate, the actual control parameter is the ratio λ_e/λ_a.

We have observed that the perturbation on the cells induced by the external – while keeping the system out of equilibrium and making impossible for it to reach any equilibrium situation – makes DCA develop large-scale regular spatial structures. Such structures show long-range correlation between the state of the cells, emerged despite the strictly local and asynchronous transition rules, and break the spatial symmetry of the final state. In addition, such structures are stable, despite the continuous perturbing effects of the external environment.

Our experiments involved many combinations of rules for cells to live/die and get alive and number of neighbors. We tested most of the rules which generate local patterns, excluding those leading to trivial attractors (i.e., all cells alive/dead). For each combination of rules and neighborhood structures we simulated the CA dynamics starting from 20 different random initial states. The interested reader can refer to the Web page: http://polaris.ing.unimo.it/DCA/ to access our simulation files in the form of applets, and appreciate the dynamic evolution of these DCA by reproducing our experiments. In the following we will discuss and show some among the typical results of the general phenomenon we observed.

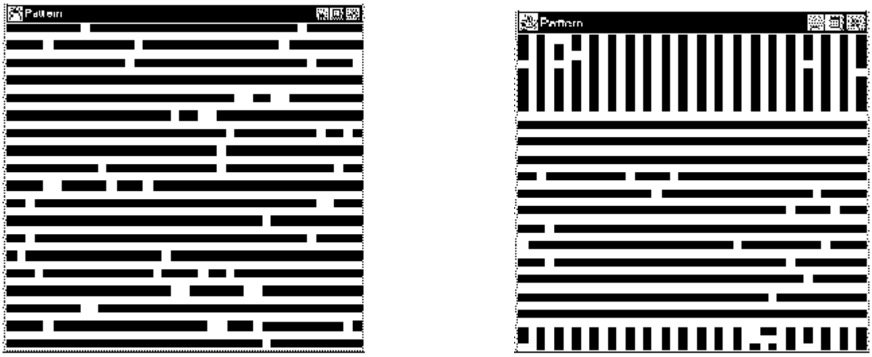


Fig. 4. Two Behaviors Evolved in a Dissipative Cellular Automata. Despite the out-of-equilibrium situation forced by the external environment, stable large-scale and symmetry-breaking patterns emerge.

For example, Fig. 4 shows two different patterns emerged from a DCA, both exhibiting stable macro-level spatial structures. For this DCA, the transition rules and the neighborhood structure are the one described in Subsect. 2.1. In both cases, the presence of global scale patterns – breaking the rotational symmetry of the automata – is apparent. By comparing these patterns with the ones observed in the same CA under asynchronous but close dynamics, one can see that openness has provided for making small scale patterns, emerged from local transition rules, enlarge to the whole CA size. Once this global states has emerged, they are able to restabilize autonomously, despite the fact that the perturbing effects tends to modify them.

As another example, Fig. 5 shows two typical patterns emerged for a DCA with a neighborhood structure made up of 12 neighbors (the neighbors of a cell are all cells having a maximum distance of 2 from the cell itself) and with the following transition rule:

$$f = \{\text{a dead cell gets alive iff it has 6 neighbors alive; a living cells lives iff it has 3,4,5, or 6 neighbors alive}\}$$

Again, it is possible to see large symmetry-breaking patterns emerge, extending to a global scale the local patterns that tends to emerge under asynchronous but closed regime (Fig. 6). The patterns are stable despite the continuous perturbing effect of the environment. Moreover, the pattern shown on the left of Fig. 5 is dynamic. First, the long diagonal stripes change continuously in their microlevel shape, while maintaining the same global structure. Second, all this stripes translate horizontally at a constant speed in the DCA lattice.

DCA share common characteristics with *Stochastic Cellular Automata* [2, 14]). Stochastic Cellular Automata (SCA) are synchronous CA with a transition function characterized by an *outgoing probability distribution*, which biases the choice of the next cell's state. The main difference between DCA and SCA

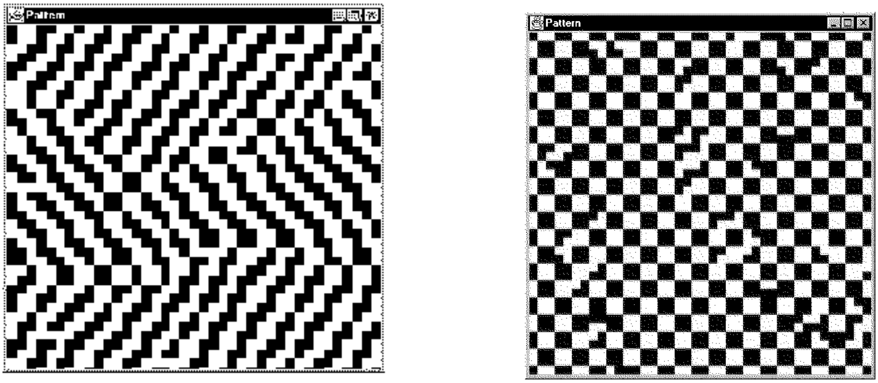


Fig. 5. Two different behaviors evolved in a Dissipative Cellular Automaton, large-scale patterns emerge. The left picture shows a step of a dynamic pattern, with horizontally translating diagonal stripes.

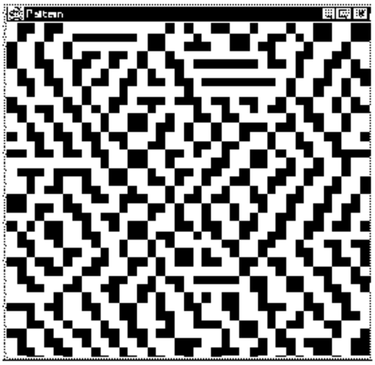


Fig. 6. A stabilized situation in an asynchronous close cellular automaton following the same rules of the DCA in Fig. 5: no large-scale patterns emerge.

is that DCA's transition function is deterministic and the non-determinism is introduced by external perturbations. The DCA model enables us to describe systems composed of several independent interacting entities (asynchronous and concurrently acting) which can be affected by external perturbations.

Behaviors similar to the ones we observed in DCA have also been obtained for synchronous CA in [20], where long-range patterns are generated by means of peculiar transition functions which explicitly introduce symmetry-breaking rules. Therefore, even though the observed behavior is similar, the emergence of macro-level spatial structures has a different origin: in our case, no symmetry-breaking rules are introduced and the regular patterns are generated by the combination of "symmetric" transition functions, asynchrony and external perturbation.

4 Explaining DCA Dynamics

DCA behavior exhibits a strong analogy with the behavior of dissipative systems [11], e.g., Benard's cells. A fluid between two plates is in thermodynamic equilibrium if no thermal energy flows from the external to perturb the equilibrium. In presence of small differences between the temperature of the two plates, the thermal energy is still not enough to perturb the fluid, and energy flows between the two plates in the form of thermal diffusion. However, as soon as the temperature gradient reaches a critical point, thermal flow in the fluid starts occurring via convection. This motion does not occur in a disordered way: regular spatial patterns of movement emerge, with wide-range and symmetry breaking correlation among cell movements. This behavior is maintained until the temperature gradient between the two plates become too high, in which case the regular patterns disappear and the fluid motion becomes turbulent.

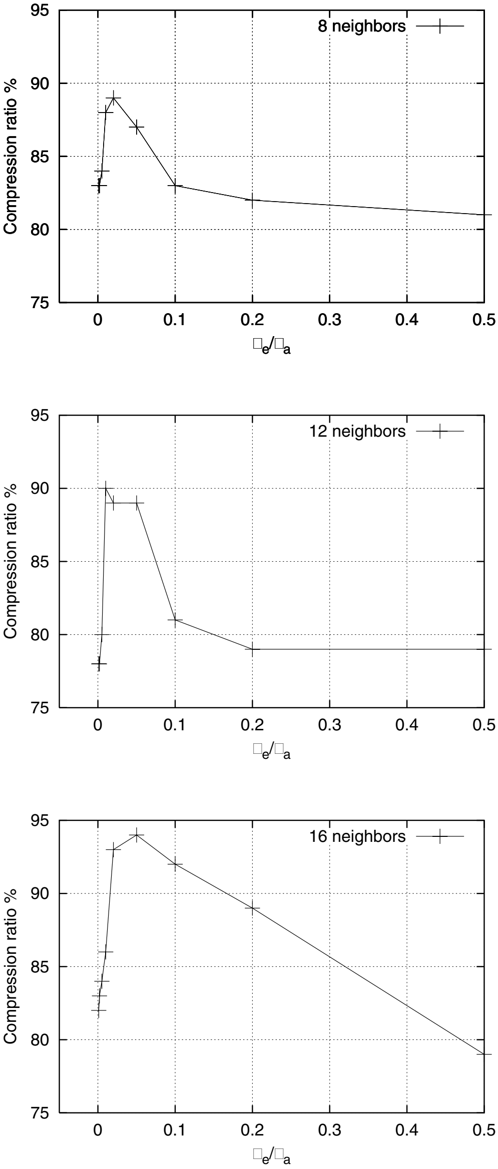
By analogy, we conjecture that the behavior of DCA might be subject to the same phenomenon, where the temperature gradient between the two plates is substituted by the ratio λ_e/λ_a. When this ratio is 0, the system is in equilibrium, and no perturbation from the external occur. For very small perturbation, the dynamic behavior of the DCA does not substantially change. As soon as the ratio becomes high enough, the DCA dynamics changes and regular spatial patterns appears. For very high ratio, spatial patterns disappear and the DCA dynamics becomes highly disordered.

A rough measure of the emergence of macro-level structures can be provided by the compression percentage achieved by compression algorithms. The higher the compression factor, the lower the randomness of the CA configuration. Although this measure does not directly evidence long-range correlations, it nevertheless provides meaningful information about the amount of structure of a CA state. Fig. 7 shows typical results obtained with DCA with different number of neighbors and transition function. The states of DCA have been measured once the equilibrium have been reached[1]. As we can observe, when the ratio λ_e/λ_a approaches a critical value θ_1 the compression ratio *cr* abruptly increases. This corresponds to the onset of structure in the system. *cr* reaches a maximum approximately located at $\lambda_e/\lambda_a \approx 0.05$. Then it decreases till reaching again the initial values, indicating the disappearance of macro-level structures in DCA states. We observe that, for the first two DCA in Fig. 7, *cr* quickly decreases, on the opposite of the last one, for which it seems that structures are still present for higher values of the ration λ_e/λ_a. In general, for all the experiments performed, we observed that the critical value for the onset of structured patterns is approximately $\lambda_e/\lambda_a \approx 0.05$.

The above similarity suggests that the same causes that determine the behavior of Benard cells also determine the behavior of DCA.

Without any perturbation, or in the presence of small one, each autonomous component (a molecule or a DCA cell), acting asynchronously accordingly to

[1] The compression algorithm used is that provided by usual compression utilities like *gzip*, at maximum compression level.



8 neighbors
A dead cell gets alive iff it has 2 neighbors alive; a living cells lives iff it has 1 or 2 neighbors alive.

12 neighbors
A dead cell gets alive iff it has 6 neighbors alive; a living cells lives iff it has 3-6 neighbors alive.

16 neighbors
A dead cell gets alive iff it has 6 neighbors alive; a living cells lives iff it has 3-6 neighbors alive.

Fig. 7. Amount of structure as a function of the ratio λ_e/λ_a. The compression ratio is evaluated by a usual compression algorithm (*gzip*). Observe that macro-level structures appear only in a specific range, as in dissipative structures.

strictly local rules, tend to reach a local equilibrium (or a strictly local dynamics), which produce a global uniform equilibrium of the whole system.

When the system is kept in a substantial out-of-equilibrium situation, the locally reached equilibrium situations are continuously perturbed, resulting in continuous attempt to locally reestablish equilibrium. This typically ends up with cell groups having found new equilibrium states more robust with regard to the perturbation (or compatible with it). Such stable local patterns start soon dominating and influencing the surrounding, in a sort of positive feedback, until a globally coordinated (i.e., with large-scale spatial patterns) and stable situation emerges.

When the degree of perturbation is high enough to avoid local stable situations to persist for enough time, they can no longer influence the whole systems, and the situation becomes turbulent.

5 Conclusion and Future Work

This paper has reported the outcomes of a set of experiments performed on a new class of cellular automata, DCA, which are open to the environment and can be perturbed by its dynamics. These experiments have shown that the perturbation makes large-scale symmetry breaking spatial structures, not observed under closed regime, emerge. By introducing a measure of the randomness of DCA states we have shown that structures emerge when the external perturbation is higher than a critical value and below the turbulence limit.

The experiments reported in this paper are indeed preliminary, and further work is in progress:

- we are currently exploring different measures for evaluating the emergence of large-scale patterns. For example, we may consider techniques analogous to the ones presented in [4,3,5], where structure is measured by evaluating the complexity of the probabilistic automaton reconstructed from the data series representing the CA evolution. Other possibilities rely on the application of techniques derived from image analysis (for example, we may use spatial correlation measures);
- we are extending our DCA simulation framework so as to study the behavior of network structures other than the regular ones of DCA, such as small-world graphs [18] and boolean networks [9], as well as networks with mobile nodes;
- we intend to perform further experiments to evaluate the behavior of DCA under different perturbation regimes and to experiment with more complex DCA, i.e., DCA with large set of states and/or with non-uniform transition functions [17,16].

The results presented in this paper promise to have several potential implications in the area of distributed computing. In fact, DCA exhibit characteristics (i.e., autonomy of components, locality in interactions, openness to the environment) that are typical of modern distributed computing environments, e.g., sensor networks and multi-agent systems.

Agents are autonomous entities [7], as their execution is not subject to a global flow of control. Indeed, the execution of an agent in a multiagent system may proceed asynchronously, and the agent's state transition occur according to local internal timings. This is actually what happens in DCA, because of the adopted time-driven dynamics. Moreover, agents are situated entities that live dipped in an environment, whether a computational one, e.g., a Web site, or a physical one, e.g., a room or a manufacturing unit to be controlled. The agent is typically influenced in its execution (i.e., in its state transitions) by what it senses in the environment. In this sense, agents and multi-agent systems are "open systems": the global evolution of a multi-agent system may be influenced by the environment in which it lives. And, in most of the cases, the environment possesses a dynamics which is not controllable or foreseeable. For instance, computational resources, data, services, as well as the other agents to be found on a given Web site cannot be predicted and they are likely to change in time. This sort of openness is the same that we can find in DCA, where the perturbation of the environment, changing the internal state of a cell, can make us consider the cell as situated in an environment whose characteristics dynamically change in an unpredictable way.

Given the above similarities, we argue that similar sort of macro-level behaviors are likely to make their appearance also in such systems, raising the need for models, methodologies, and tools, explicitly taking into account the autonomy and environmental dynamics and exploiting them either constructively, to achieve globally coordinated behaviors, or defensively, to control the behavior of the system. On the one hand, one could think at exploiting the environmental dynamics to control and influence a multi-agent system from "outside the loop" [15], that is, without intervening on the system itself. In a world of continuous computations, where decentralized software systems are always running and cannot be stopped (this is already the case for Internet services and for embedded sensors) changing, maintaining and updating systems by stopping and re-installing them is not the best solution, and it could not be always feasible. On the other hand, the reported experiments open up the possibility that a software system immersed in a dynamic environment may exhibit behaviors very different from the ones it was programmed for. Obviously, this is not desirable and may cause highly damaging effects.

Of course, we are not the first discussing the possibility of emergence of complex self-organizing behaviors in multi-agent systems. However, most of the studies (apart from a few exceptions [12]) have focused on "closed" agent systems, in which the internal dynamics of the systems totally drive its behavior. Instead, we have shown, via a very simple and "minimal" multi-agent system, as a DCA can be considered, that complex non-local behaviors can emerge due to the influence of the environmental dynamics. The impact of this observation in the modeling, engineering, and maintaining of distributed agent systems may be dramatic [8,21].

References

1. Y. Bar-Yam. *Dynamics of Complex systems*. Addison Wesley, 1997.
2. T.D. Barfoot and G.M.T. D'Eleuterio. Multiagent Coordination by Stochastic Cellular Automata. Proceeding of IJCAI 2001, Seattle, 2001.
3. J.P. Crutchfield. Discovering Coherent Structures in Nonlinear Spatial Systems. In *Nonlinear Dynamics of Ocean Waves*, 1992.
4. J.P. Crutchfield and J.E. Hanson. Turbulent Pattern Bases for Cellular Automata. Physica D **69**:279–301, 1993.
5. J.E. Hanson and J.P. Crutchfield. The Attractor-Basin Portrait of a Cellular Automaton. J. Statistical Physics **66**:1415–1462, 1992.
6. T.E. Ingerson and R.L. Buvel. Structure in asynchronous cellular automata. Physica D, 10:59–68, 1984.
7. N.R. Jennings. On Agent-Based Software Engineering. Artificial Intelligence, 117(2), 2000.
8. T. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, 3rd Edition, Nov. 1996.
9. S.A. Kauffman. *The origins of order.* Oxford University Press, New York, 1993.
10. E.D. Lumer and G. Nicolis. Synchronous versus asynchronous dynamics in spatially distributed systems. Physica D, 71:440–452, 1994.
11. G. Nicolis and I. Prigogine. *Exploring Complexity: an Introduction* W. H. Freeman (NY), 1989.
12. V. Parunak and S. Bruekner. Entropy and Self-Organization in Agent Systems. 5th International Conference on Autonomous Agents, ACM Press, May 2001.
13. B. Schönfisch and A. de Roos. Synchronous and asynchronous updating in cellular automata. BioSystems, 51(3):123–143, 1999.
14. B. Schönfisch and M.O. Vlad. Physical approach to the ergodic behavior of stochastic cellular automata with generalization to random processes with infinite memory. Physica A **229**:273–294, 1996.
15. D. Tennenhouse. Proactive Computing. Communications of the ACM, May 2000.
16. M. Sipper. The Emergence of Cellular Computing. IEEE Computer, 37(7):18–26, July 1999.
17. M. Sipper and M. Tomassini. Computation in artificially evolved, non-uniform cellular automata. Theoretical Computer Science, 217(1):81–98, March 1999.
18. D. Watts. *Small-Worlds.* Princeton University Press, 1999.
19. S. Wolfram. *Cellular Automata and Complexity.* Addison–Wesley, 1994.
20. S. Wolfram. *A New Kind of Science.* Wolfram Media Inc., 2002.
21. F. Zambonelli and H.V. Parunak. From Design to Intention: Signs of a Revolution. Proceedings of the 1st ACM Joint Conference on Autonomous Agents and Multi-Agent Systems, 2002.

Enhancing Cellular Spaces by Multilayered Multi Agent Situated Systems

Stefania Bandini, Sara Manzoni, and Carla Simone

Department of Informatics, Systems and Communication
University of Milano-Bicocca
Via Bicocca degli Arcimboldi 8 20126 Milan - Italy
tel +39 02 64487835 fax + 39 02 64487839
{bandini, manzoni, simone}@disco.unimib.it

Abstract. This paper presents the formal description of the Multilayered Multi–Agent Situated System (MMASS)[1] which can be seen as a generalization of cellular spaces since it relaxes some constraints on uniformity, locality and closure. MMASS allows the description, representation and simulation of complex systems that explicitly require to consider spatial features. Different forms of interaction are possible within a MMASS: synchronous reaction between spatially adjacent agents and asynchronous and at–a–distance interaction through a field emission–propagation–perception mechanism.

1 Introduction

Cellular Automata (CA) can be seen as a kind of Multi Agent System (MAS), where spatial structure of agent environment is explicit and structured in a grid, agents are immobile, homogeneous and dense (all the cells of the CA are identical and include agent representation) and their behavior is synchronous. This is not the general case in MAS–based models, since heterogeneous and asynchronous agents might live in the same, possibly not structured, environment. Thus, as clearly stated in [1], "CA can be considered either as 'degenerate' MAS in which agents have become fixed or, more positively, as good environmental models". A MAS consists of a number of agents that are defined in terms of their behaviors and characteristic parameters and are located in an environment that makes their interactions possible. The behavior of a MAS is defined as the global effects of local interactions among autonomous agents populating the environment.

Accordingly, Multi Agent Based Simulation (MABS [2]) is based on the idea that it is possible to represent the global behavior of a dynamic system as the result of interactions occurring among an assembly of agents with their own operational autonomy. MAS have been used to simulate artificial worlds [3] as well as natural or social phenomena [4]. In MABS, agents might represent animals

[1] The work presented in this paper has been partially funded by the Italian Ministry of University and Research within the project 'Cofinanziamento Programmi di Ricerca di Interesse Nazionale'

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 156–167, 2002.

in ecosystems, vehicles in traffic, people in crowds, or autonomous characters in animation and games. Unlike CA–based simulation, which is based on a dense and uniform dissection of the space where the execution control is centralized and the simulation inner loop proceeds cell by cell [5], in MABS applications the system simulation is based on autonomous and distributed agents (i.e. it proceeds agent by agent each one with its own thread of control).

Recent results in complexity science [6] suggest that the topology of agent interaction is critical to the nature of the emergent behavior of the MAS. With the exception of some preliminary proposals [7], none of the MAS models presented in the literature explicitly takes into account the spatial structure (i.e. topology) of the environment where agents are located. This happens despite of the fact that a large class of problems is characterized by unavoidable spatial features requiring the related modelling approach to incorporate an explicit or implicit model of the space. In fact, several domains deal with space itself (e.g., geographical location) or a model of it (e.g., information flow in an organizational structure) or both of them (e.g., to conquer some favorable selling location in a region or to play a new role implying a move in the physical and organizational structure of a company). In traditional MAS approach, agents can be associated with a location in their environment, but no explicit structure of the environment is given. For instance, mobile information agents that are located on MAS models of networked computers do not refer to the network structure as an explicitly defined geometrical space [8].

On the contrary, CAs have offered a very interesting framework to model and simulate natural and artificial phenomena involving space, due to their basic definition and structure [9]. CA have been profitably used in various cases of simulation where space has a crucial role. In this respect, we can distinguish between CA models that intrinsically allow parametric spatial conditions to be represented (e.g. fluid–dynamics in porous media, cellular geography and so on), and CA models that allow a spatial representation to be created (e.g., in competitive behavior, population dynamics, financial data clustering). CA modelling is designed to simulate the dynamics of spatial interaction and, as a consequence, CA have been employed in the exploration of various urban phenomena (e.g. traffic simulation, regional urbanization, land use dynamics) explicitly dealing with the spatial relation and interaction among locations [10]. Moreover, in some cases, CA offered the possibility to conceptualize and visualize abstract and intrinsically not spatial problems [11].

Interesting results have been shown by the combination of MAS and CA [12]. The approach basically consists in the positioning of a MAS on a cellular space and has been mainly applied to analyze urban system dynamics and pedestrian activity [13]. Two sets of example scenarios where agents have been combined with cellular spaces to model environmental and urban systems can be found in [10] and [14]. In these examples, the cellular space simulates the dynamics of the urban infrastructure (e.g. land–use transition, real estate development and redevelopment, urban growth) while a MAS simulates the dynamic of the interacting entities that populate this infrastructure (e.g. residential location, pedestrian movement, traffic simulation).

The main aim of this paper is to present a modelling approach that is based on the notion of Multilayered Multi Agent Situated System (MMASS). The MMASS offers a formal and computational framework where to describe, represent and simulate complex systems which explicitly require spatial features to be considered and integrates different forms of interaction. The MMASS can be seen as a generalization of CA since it relaxes constraints on uniformity, locality and closure. The MMASS model defines a system of agents situated in an environment that is not homogeneous neither in its properties nor in its structure (as a consequence, neighborhoods are not uniform across the space). The MAS is composed by heterogeneous agents that is, different agent types can be defined. Moreover different forms of interaction are possible within a MMASS: synchronous reaction between spatially adjacent agents and asynchronous and at–a–distance interaction through a field emission–propagation–perception mechanism. Fields are emitted by agents according to their type and state, and propagate throughout the spatial structure of the environment according to their diffusion function, reaching and being eventually perceived by other spatially distant agents. In fact, different sensitivity to fields, capabilities and behaviors characterize agents of different types. Finally, the MMASS can model open systems by being influenced by external fields.

The MMASS approach has been applied in the MABS domain to the simulation of localization problem and for the design of a technology promoting awareness in the Computer Supported Cooperative Work domain. A brief description of these example applications of the MMASS model is reported in Section 4 and more details can be found in [15] and [16].

2 The Multilayered Multi Agent Situated System

A *Multilayered Multi Agent Situated System (MMASS)* can be defined as a constellation of interacting *Situated MAS*'s (MASS) through the primitive:

$$\textbf{Construct}(MASS_1 \ldots MASS_n)$$

Thus, a MMASS is defined by the set of its constituting *Situated MAS*'s ($\{MASS_1 \ldots MASS_n\}$). Each MASS of a Multilayered MASS is defined by the triple

$$< Space, F, A >$$

where *Space* models the environment where the set A of agents is situated, acts autonomously and interacts via the propagation of the set F of fields. Thus a MMASS can be also denoted by:

$$\Big\langle < Space_1, F_1, A_1 > \ldots < Space_n, F_n, A_n > \Big\rangle$$

where,

- $\left(\bigcup_{i=1..n} Space_i\right)$ is the set of *Spaces* defining the multilayered spatial structure of the MMASS;
- $\left(\bigcup_{i=1..|F|} F_i\right)$ denotes the set of *fields* acting in the MMASS;
- $\left(\bigcup_{i=1..|A|} A_i\right)$ denotes the set of *agents* situated in it.

Field emission–propagation–perception is the mechanism defined for asynchronous interaction among agents situated in the same or different MASS's: an agent emits a field that propagates throughout the *Space* and can be perceived by other agents. In order to allow MASS interaction the MMASS model introduces the notion of *interface*. The interface of a MASS specifies fields imported from and exported into each MASS, and takes the following format:

$$\mathbf{Interface}(MASS_i, \mathbf{export} : \mathbf{E}; \mathbf{import} : \mathbf{I})$$

where $E \subset F_i$ and $I \subset F_i$ are respectively the set of fields exported into and imported from $MASS_i$. Imported fields are used in agent actions as internal fields do. As will be better explained in section 2.3 and 2.4, they can be used in the definition of agent perception function and mentioned in the specification of agent actions. By definition, the value of an external field at any site of the local *Space* of a MASS is the value specified at its emission. Moreover, the receiving MASS has to define if and how this value has to be internally propagated by means of local fields defined for this purpose. In fact, their distribution function (see section 2.2) is highly dependent on the structure of the local *Space* which is completely hidden to external MASS's. In sections 2.1, 2.2, 2.3 each component of a MASS will be described in more detail.

2.1 Spaces

The environment where the agents of a MASS are situated is named *Space* and is defined as made up of a set P of sites arranged in a network. Each *site* $p \in P$ can contain at most one agent and is defined by the 3–tuple

$$< a_p, F_p, P_p >$$

where:

- $a_p \in A \cup \{\perp\}$ is the agent situated in p ($a_p = \perp$ when no agent is situated in p, that is p is empty);
- $F_p \subset F$ is the set of fields active in p ($F_p = \emptyset$ when no field is active in p);
- $P_p \subset P$ is the set of sites adjacent to p.

In this way the *Space* can be considered as an undirect graph of sites.

2.2 Fields

Fields acting within a MASS can be generated by agents of the MASS or have a source outside the local *Space* in the case of fields imported from another MASS or from outside the MMASS in the case of open systems. Each field of a MASS is characterized by the set of values that the field can assume during its propagation throughout the *Space*. Propagation occurs according to the diffusion function that characterizes the field and that specifies how its values propagate throughout the space according to its spatial structure. Moreover, field comparison and field composition functions are defined in order to allow field manipulation.

A field $f \in F$ is defined by the 4–tuple

$$< W_f, Diffusion_f, Compare_f, Compose_f >$$

where:

- W_f denotes the set of values that the field can assume;
- $Diffusion_f : P \times W_f \times P \to (W_f)^+$ is the diffusion function of the field computing the value of a field on a given site taking into account in which site and with which value it has been generated. Since the structure of a *Space* is generally not regular and paths of different length can connect each pair of sites, $Diffusion_f$ returns a number of values depending on the number of paths connecting the source site with each other site. Hence, each site can receive different values of the same field along different paths.
- $Compose_f : (W_f)^+ \to W_f$ expresses how field values have to be combined (for instance, in order to obtain the unique value of the field at a site).
- $Compare_f : W_f \times W_f \to \{True, False\}$ is the function that compares field values. For instance, in order to verify whether an agent can perceive a field value the value of a field at a site and agent sensitivity threshold are compared by this function (see the definition of agent perception in the following subsection).

2.3 Agents

The *Space* of each MASS is populated by a set A of individuals called *agents*. An agent $a \in A$ is defined by the 3–tuple

$$< s, p, \tau >$$

where:

- $s \in \Sigma_\tau$ denotes the *agent state* and can assume one of the values specified by its type;
- $p \in P$ is the site of the *Space* where the agent is situated;
- τ is the *agent type* describing the set of states the agent can assume, a function to express agent sensitivity to fields, and the set of actions that the agent can perform.

An agent type τ is defined by the 3–tuple

$$< \Sigma_\tau, Perception_\tau, Action_\tau >$$

where:

- Σ_τ defines the set of states that agents of type τ can assume;
- $Perception_\tau : \Sigma_\tau \to [\mathbf{N} \times W_{f_1}] \ldots [\mathbf{N} \times \mathbf{W}_{\mathbf{f}_{|\mathbf{F}|}}]$ is a function associating to each agent state the vector of pairs

$$\left(c_\tau^1(s), t_\tau^1(s)\right), \left(c_\tau^2(s), t_\tau^2(s)\right), \ldots, \left(c_\tau^{|F|}(s), t_\tau^{|F|}(s)\right)$$

  where for each i ($i = 1 \ldots |F|$), $c_\tau^i(s)$ and $t_\tau^i(s)$ express respectively a coefficient to be applied to the field value f_i and the agent sensibility threshold to f_i in the given state s. This means that an agent of type τ in state $s \in \Sigma_\tau$ can perceive a field f_i only when it is verified

$$Compare_{f_i}(c_\tau^i(s) \cdot w_{f_i}, t_\tau^i(s))$$

  that is, when the first component of the i–th pair of the perception function (i.e. $c_\tau^i(s)$) multiplied for the received field value w_{f_i} is greater than the second component of the pair (i.e. $t_\tau^i(s)$).
- $Actions_\tau$ denotes the set of actions, described in the following subsection, that agents of type τ can perform.

2.4 Actions

$Actions_\tau$ specifies whether and how agents change their state and/or position, how they interact with other agents, and how neighboring and at–a–distance agents can influence them. In general, actions have two possible purposes: they can be undertaken by an agent in order to modify its state or position (i.e. *intra–agent actions*), or to interact with other agents in both synchronous or asynchronous ways (i.e. *inter–agent actions*). Specifically, *trigger* and *transport* are the intra–agent actions. *trigger* defines how the perception of a field causes a change of state in the receiving agent, while *transport* defines how the perception of a field causes a change of position in the receiving agent. *emit* and *react* are the inter–agent actions that respectively are asynchronous and at–a–distance and synchronous and local.

As previously introduced, *asynchronous interaction* among agents takes place through a field emission–propagation–perception mechanism. An agent emits a field, that is, it generates a field defining its parameters (i.e. intensity values, diffusion function, and so on), when its state is such that it can be source for it. This constitutes one side of the asynchronous interaction among agents: that is, the sending of broadcast messages by an agent. Field values propagate throughout the space according to the diffusion function of the field. Field diffusion along the space allows the other agents to perceive it. $Perception_\tau$ function, characterizing each agent type, defines the second side of an asynchronous interaction among agents: that is, the possible reception of broadcast messages conveyed through a field, if the sensitivity of the agent to the field is such that it can perceive it. This means that a field can be neglected by an agent of type τ if its value at the site where the agent is situated is less than the sensitivity threshold computed by the second component of the $Perception_\tau$ function. This is the very essence of the broadcast interaction pattern, in which messages are not addressed to specific receivers but potentially to all agents populating the space.

Reaction defines the *synchronous interaction* among a set of agents characterized by given states and types and pair–wise situated in adjacent sites (that is, *adjacent agents*). Synchronous interaction is a two–steps process. Reaction among a set of agents takes place through the execution of a protocol introduced in order to synchronize the set of autonomous agents. The synchronization protocol involves a set of adjacent agents. When an agent wants to react with the set of its adjacent agents since their types satisfy some required condition, it starts an *agreement* process whose output is the subset of its adjacent agents that have agreed to react. An agent agreement occurs when the agent is not involved in other actions or reactions and when its state is such that this specific reaction could take place. The agreement process is followed by the synchronous reaction of the set of agents that have agreed to it.

In order to define agent action set ($Actions_{\tau}$) let us consider an agent a of type τ whose current position is site p and whose current state is s (i.e. $a =< s, p, \tau >$). Moreover, $p =< a, F_p, P_p >$ where F_p is the set of fields active in p and P_p is the set of sites adjacent to p. To define the four action above outlined, we will use operators of the form *action–condit–effect*. *action* is an expression of the form $f(x_1 \ldots x_n)$ where f specifies the action name and x_i are variables which can appear in *condit* and *effect* expressions. *condit* and *effect* express respectively the set of conditions that must be verified in order to let the agent execute the action, and the set of effects deriving from the execution of the action. They are sets of atomic formula $p(a_1 \ldots a_k)$ where p is a predicate of arity k and a_i are either constants or variables. According to this syntax, the four basic agent actions are:

1. `action:` `trigger`(s, f_i, s')
   `condit:`$state(s), perceive(f_i)$
   `effect:`$state(s')$
   where $state(s)$ and $perceive(f_i)$ are verified when the agent state is s, and $f_i \in F_p$ and $Compare_{f_i}\big(c^i_{\tau}(s) \cdot w_{f_i}, t^i_{\tau}(s)\big) = True$ (that is, the field f_i is active in p and agents of type τ in state s can perceive it). The effect of a trigger action is a change in state of the agent according to the third parameter.
2. `action:`$transport(p, f_i, q)$
   `condit:`$position(p), empty(q), near(p, q), perceive(f_i)$
   `effect:`$position(q), empty(p)$
   where $perceive(f_i)$ has the same meaning as in $trigger()$, while $position(p)$, $empty(q)$ and $near(p, q)$ are verified when the agent position is p, $q \in P_p$ and $q =< \perp, F_q, P_q >$ (that is, q is a site adjacent to p and no agent is situated in it). The effect of the execution of a transport action is the change in position of the agent undertaking the action and, as a consequence, the change of the local space where the agent is situated. Let us suppose that agent a has executed a $transport(p, f_i, p')$. This means that a has changed its position from p to p' (i.e. $a =< s, p', \tau >$) and as a consequences sites p and p' have changed respectively to

$p =< \perp, F_p, P_p >$
$p' =< a, F_{p'}, P_{p'} >$

3. `action:`$emit(s, f, p)$
`condit:`$state(s), position(p)$
`effect:`$added(f, p)$
where $state(s)$ and $position(p)$ are verified when the agent state and position are s and p. The effect of a emit action (i.e. $added(f, p)$) is a change at each site of the space according to $Diffusion_f$. In particular the set of fields in p where the emitting agent is situated changes to

$$p =< a, F_p \diamond f, P_p >$$

where the operator $F \diamond f$ simply adds a field f to a field set F if the field does not already belong to the set. Otherwise it composes the already present field value(s) and the new one. Let w_f^{old} and w_f^{new} be their names, $F \diamond f = F \cup f$ and $w_f = w_f^{new}$ when $f \notin F$ (before the execution of the emit action) and $w_f = Compose(w_f^{old}, w_f^{new})$ if $f \in F$.
4. `action:`$reaction(s, a_{p_1}, a_{p_2}, \ldots, a_{p_n}, s')$
`condit:`$state(s), agreed(a_{p_1}, a_{p_2}, \ldots, a_{p_n})$
`effect:`$state(s')$
where $state(s)$ and $agreed(a_{p_1}, a_{p_2}, \ldots, a_{p_n})$ are verified when the agent state is s and agents situated in sites $\{p_1, p_2, \ldots, p_n\} \subset P_p$ have previously agreed to undertake a synchronous reaction. The effect of a reaction is the synchronous change in state of the involved agents; in particular, agent a changes its state to s'.

3 Adding Features to the Basic MMASS Model

The basic MMASS model presented above has been extended to incorporate relevant features of the target systems. Typically, the application of the MMASS model to specific problems and domains required introducing the possibility to dynamically modify both the spatial structure of environment and the number of agents constituting each MASS. To this aim agent types can be characterized by the following additional actions:

- *AddSite(p)*: the agent adds a site p to its neighborhood, i.e., the set of sites adjacent to the site in which it is situated. Since agents have only a local visibility of their environment (i.e. they ignore the existence of sites not adjacent to the one hosting them) this action can be performed only after that the agent has received from other agents the needed information about a spatially not adjacent site (typically through a reaction or the perception of a field).
- *DeleteSite(p)*: the agent deletes a site p from its neighborhood (obviously, this action can be performed only if the site is not occupied by any agent).
- *Clone(a)*: the agent creates an agent, specifying all its parameters: its type (only agents of the same type can be created), its initial state (one of the

states specifiable for its type), and its position (one of the site of the neighborhood of the creating agent);
- *Clear(self)*: the agent deletes itself. This action has been introduced for sake of garbage collecting agents that are no more operative. This fact is denoted by a specific state (i.e. 'dead'). The agent that executes this action simply clears itself and makes free its site.

4 MMASS Applications to MABS

The MMASS approach has been applied in the MABS domain to the localization problem of suitable sites for extra–urban shopping centers. Simulations can be useful when suitable space is available and a good location for a new shopping center has to be chosen, or when the possibility of success in locating a new shopping center in an area already served by other retailers has to be evaluated. Finding a "good" location for a shopping center, that is, close enough to widely populated areas and roads, and far away from other centers, is a problem that has been widely studied by urbanists and economists, and many different approaches have been proposed. In particular, the idea we implemented with agents is based on the gravitational model, where shopping centers are seen as masses that emit gravitational fields that attract consumers and repel other installations. Geographical factors, such as the proximity of other centers and residential areas, are essential for the choice of a suitable location for a shopping center. Moreover, once some centers have settled in a given area, other factors like competition should be taken into account. For example, one center could start low price special sales in order to attract consumers, or invest heavily on advertisement. This kind of model has been previously implemented with Cellular Automata [17, 5], but in the most significant examples the strict CA uniformity requirements of CA formalism have been removed. In [15] it has been shown how the MAS approach fits more naturally to this problem and domain, and how the design of a MMASS composed by two MASS's allows the simulation of the two aspects involved in this problem (Figure 1 shows an example of the system and the user interface that has been developed. The area shown is the city of Bergamo and its surroundings). Two MASS's have been defined: the territorial MASS and the strategic MASS. In the former, the formation or the disappearance of a shopping center is modelled by considering only geographical factors. In the latter, already existing centers compete with one another trying to attract consumers. Each shopping center is represented in both the territorial and the strategic MASS. Interaction between the two MASS's is performed through field emission by agents belonging to a MASS and their perception as external fields by agents of the other MASS. Further investigation of the application of the MMASS approach to socio–economic analysis in urban simulation will be continued within a collaboration with the Austrian Research Center Seibersdorf (ARCS).

The MMASS model has been derived by the previously introduced Multilayered Reaction–Diffusion Machine (MRDM). In [18], it has been shown that a special class of MRDM can simulate a deterministic Turing Machine and that, by

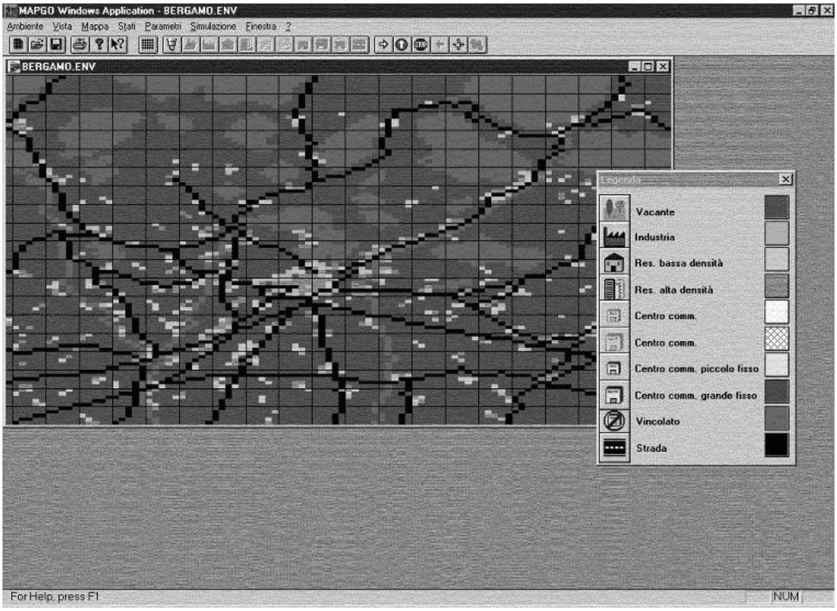

MAPGO Windows Application - BERGAMO.ENV
Ambiente Vista Mappa Stati Parametri Simulazione Finestra ?
BERGAMO.ENV
Legenda
Vacante
Industria
Res. bassa densità
Res. alta densità
Centro comm.
Centro comm.
Centro comm. piccolo fisso
Centro comm. grande fisso
Vincolato
Strada
For Help, press F1
NUM

Fig. 1. An example of the user interface of the system that has been developed to simulate the localization problem of shopping centers in the area of Bergamo (Italy)

introducing some constraint, a MRDM collapses to a standard CA. The MRDM has been implemented on two different computer architectures. The first implementation realizes a CA based model described in terms of MRDM has been developed on a Cray T3E and Origin S2000 parallel computers. It provides an environment suitable to simulate complex systems of physical and chemical nature: specifically, the domain concerns transportation phenomena occurring in the percolation processes of pesticides through soils. The parallel implementation has been chosen because of the high computational demand of the model [19]. The second implementation, realized in Java(TM), provides an environment suitable for cooperative applications supporting awareness. The main feature of this implementation concerns the possibility to model different forms of interactions occurring on different layers and on a single layer, allowing significant aspects of the awareness model to be represented and computed [16]. An additional application of the MMASS approach to model and simulate complex system dynamics in the theoretical immunology domain is currently under design. Within a project funded by the Italian Ministry for University and Research ('Cofinanziamento Programmi di Ricerca di Interesse Nazionale'), the analysis of a previously proposed modelling and simulation approach based on Cellular Automata [20] will be continued by exploring the possible benefits deriving from the adoption of the MMASS approach.

Finally, it is under design a support tool for designing, developing and running applications based on the MMASS model. The aim of the project is to provide designer with a language and the related infrastructure for the development of systems of agents that are characterized and influenced by their spatial position and where the spatial relationship among agents is considered in agent interaction. The preliminary language specification and a discussion on the mechanisms needed develop a framework that implements the model can be found in [21] and [22] respectively.

5 Conclusions

This paper has presented the description of the Multilayered Multi Agent Situated System model, a spatially explicit and distributed model incorporating synchronous and asynchronous forms of interaction. Agent behavior occurring, for instance, as response to perception of fields generated by other agents and propagating throughout the environment, is strongly influenced by the spatial structure of the latter. Since field values can decrease during field propagation, agents perceive them depending on the site of the space where the agent is situated and the site in which the field has been generated. Moreover, an explicit definition of the spatial structure of agent environment allows the definition of distance and adjacency among situated agents. The MMASS model provides two types of interaction among agents constituting the MAS (i.e. reaction and field emission–propagation–perception), and both types take into account the spatial relationship between agents. In fact, fields influence agents depending on their spatial distance, and reactions occur only when agents are spatially adjacent. Finally, since in some cases the environment is a shared resource for agents, its spatial structure constraints also agent actions, like agent change of position in the environment.

As stated in Section 4, future work will concern the development of a language and a tool to design, develop and execute applications according to the MMASS approach. Moreover, the MMASS approach application will contemporaneously proceed within the MABS and CSCW domains.

References

1. Ferber, J.: Multi-Agents Systems. Addison-Wesley, Harlow (UK) (1999)
2. Moss, S., Davidsson, P., eds.: Multi Agent Based Simulation, 2nd International Workshop, MABS 2000, Boston, MA, USA, July, 2000, Revised and Additional Papers. Volume 1979 of LNCS. Springer (2001)
3. Epstein, J.M., Axtell, R.: Growing Artificial Societies. MIT Press, Boston (1996)
4. Drogoul, A., Ferber, J.: Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In Castelfranchi, C., Werner, E., eds.: Artificial Social Systems, MAAMAW '92, Selected Papers. Volume 830 of LNCS, Springer (1994) 3–23
5. Couclelis, H.: From cellular automata to urban models, new principles for model development and implementation. Urban systems as Cellular Automata **24** (1997)

6. Parunak, H.V.D., Brueckner, S., Sauter, J., Matthews, R.: Distinguishing environmental and agent dynamics: A case study in abstraction and alternate modeling technologies. In: Engineering Societies in the Agents' World (ESAW'00) at ECAI 2000, Proceedings. (2000) 1–14
7. Parunak, H.V.D.: The Process-Interface-Topology model: Overlooked issues in modeling social systems. In: Modelling Artificial Societies and Hybrid Organizations (MASHO'00) at ECAI 2000, Proceedings. (2000)
8. Cremonini, M., Omicini, A., Zambonelli, F.: The explorable topology: Supporting agent autonomy on the internet. In: 4th Workshop on Distributed Systems: Algorithms, Architectures, and Languages (WSDAAL99), Proceedings. (1999) 24–28
9. Goles, E., Martinez, S.: Neural and automata networks, dynamical behavior and applications. Mathematics and Its Applications (1990)
10. Torrens, P., O'Sullivan, D.: Cellular automata and urban simulation: Where do we go from here? Environment and Planning B **28** (2001) 163–168
11. Jacob, G., Barak, L., Eitan, M.: Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. Journal of the Academy of Marketing Science (2001)
12. Batty, M., Jiang, B.: Multi-agent simulation: Computational dynamics within GIS. Innovation in GIS VII: Geocomputation (2000) 55–71
13. Dijkstra, J., Timmermans, H., Jessurun, A.: A multi-agent cellular automata system for visualising simulated pedestrian activity. In Bandini, S., Worsch, T., eds.: Theoretical and Practical Issues on Cellular Automata, (ACRI2000) Proceedings, Springer Verlag (2001) 29–36
14. Jiang, B.: Agent-based approach to modelling urban and environmental systems within GIS. In: 9th International Symposium on Spatial Data Handling, Beijing, Proceedings. (2000)
15. Bandini, S., Manzoni, S., Pavesi, G., Simone, C.: Location of extra-urban shopping centres: A multi-agent based approach. In Concilio, G., Monno, V., eds.: Proceedings 2nd National Conference on Information Technology and Spatial Planning: Democracy and Technologies (INPUT 2001), Bari (2001)
16. Simone, C., Bandini, S.: Integrating awareness in cooperative applications through the reaction-diffusion metaphor. CSCW, The international Journal of Collaborative Computing **11** (2002) to appear.
17. Engelen, G., White, R., Uljii, I., Drazan., P.: Using cellular automata for integrated modelling of socio-environmental systems. Environmental Monitoring and Assessment **34** (1995) 203–214
18. Bandini, S., Simone, C.: Integrating forms of interaction in a distributed coordination model. Fundamenta Informaticae (in press)
19. Bandini, S., Mauri, G., Pavesi, G., Simone, C.: Parallel simulation of reaction-diffusion phenomena in percolation processes: a model based on cellular automata. Future Generation Computer Systems **17** (2001) 679–688
20. Bandini, S., Mauri, G.: Multilayered cellular automata. Teoretical Computer Science **217** (1999) 99–113
21. Bandini, S., Manzoni, S., Pavesi, G., Simone, C.: L*MASS: A language for situated multi-agent systems. In Esposito, F., ed.: AI*IA 2001: Advances in Artificial Intelligence, Proceedings. Volume 2175 of LNCS, Berlin, Springer (2001) 249–254
22. Bandini, S., DePaoli, F., Manzoni, S., Simone, C.: Mechanisms to support situated agent systems. Proceedings of 7th IEEE Symposium on Computers and Communications (ISCC2002), to appear (2002)

Perturbing the Regular Topology of Cellular Automata: Implications for the Dynamics

Roberto Serra and Marco Villani

Centro Ricerche Ambientali Montecatini, via Ciro Menotti 48
I-48023 Marina di Ravenna (RA)

Abstract. The topology of Cellular Automata (CA) is that of regular graphs with high clustering coefficients and long characteristic path lengths. The introduction of some long range connections modifies the topology, and it may give rise to small world networks, with high clustering and short path lengths, modifying also the system dynamical properties (attractors, basins of attraction, transient duration). In order to investigate the effects on the dynamics of the introduction of long range connections it is appropriate to keep the number of connections per node constant, while the existing algorithms give rise to nodes with different connectivities. Here we present an algorithm able to re-direct the links without changing the connectivity degree of the nodes. We then analyze the effects of small topological perturbations of a regular lattice upon the system dynamical properties in the case where the transition function is the majority rule; we show that these effects are indeed important and discuss their characteristics.

1 Introduction

Major features of cellular automata (CA) include the locality of the interactions and the regularity of the topology, which make them well suited to deal e.g. with spatially extended systems. The CA approach has been used also to model systems (like e.g. interacting companies in an economy [1] and genetic networks in a cell [2]) where the topological constraints appear to be somewhat artificial. In these cases, an alternative approach is that of random networks, where each element (node) is influenced by a number of other elements chosen at random [3][4][5][6].

Recent researches [7][8][9][10][11] highlighted that several networks of interest (ranging from biochemical interaction networks to the Internet, from friendship relations to the World Wide Web, from scientist collaboration to electric power distribution) show an odd combination of the features of the above systems, i.e. they have short characteristic path lengths (like random graphs) yet their clustering coefficients are much higher than those of the random graphs. Theoretical models of these "small world" networks have also been proposed and their properties have been investigated. In particular, the Watts-Strogatz (WS) [12] and the scale-free (SF) [13] models have been the focus of many researches.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 168-177, 2002.

Since these networks represent the backbone of dynamical interacting systems, it is very important to investigate how the topology affects the dynamics (for a review see [14]). It is worth stressing, however, that until now the algorithms for generating a small world network give rise to a distribution of connectivities.

The presence of long range connections and that of a distribution of connectivities may both influence the dynamical properties; therefore, in order to disentangle their roles, in this paper we introduce an algorithm for generating a small world network from a regular lattice leaving the number of connections for each node unchanged; we can then leave also the transition function of each node unchanged, and test the effects of the modification of the topology on the dynamical properties.

We consider here a well-known CA rule, the majority rule (although in this case an obvious generalization exists when the number of connections changes, this is not so in general). This rule has been chosen because it has been extensively studied in the literature on cellular automata and genetic algorithms [15, 16, 17] therefore allowing comparison with known results for regular lattices.

Among the interesting aspects, let us remark that the small world character becomes apparent when just a small fraction of long range interactions has been added, and that the introduction of such interactions affects the number of attractors and the size of their basins. Moreover, it also affects the duration of the transients, which may be very important in some open systems.

The outline of the paper is as follows. In Section 2 a short review of definitions and properties of graphs is given. In Section 3 our algorithm for perturbing a regular lattice is described, and it is shown that it gives indeed rise to a small world network. The dynamical properties of the majority rule are analyzed in section 4 as a function of the number of redirected links. This is only the beginning of a wide research program, yet it already provides some interesting observations. We do not know the degree of generality of these observations, which will be the subject of further work. A brief comment on this is the subject of section 5.

2 Graphs and Cellular Automata: Characteristic Path Length and Clustering Coefficient

A graph G={V,E} is defined by a set of labelled nodes V={v| v=1 … n} and a set of edges E⊆{(a,b)|a=1…n, b= 1…n, b#a}, where an edge is defined as a pair of nodes (a,b). If the pair is ordered, then the graph is ordered. In the following we will consider unordered graphs, most generalizations to the case of ordered graphs being straightforward. An edge (a,b) is said to connect the nodes a and b. Synonyms are often used, i.e. "vertex" instead of node and "link" instead of edge. Two vertices are adjacent if there is a link connecting them.

For a recent review on regular, random and small-world graphs, we refer the reader to [9]. In order to provide a gross characterisation of the topological properties of different graphs, three quantities play a major role: the average connectivity per node (k), the characteristic path length (L) and the clustering coefficient (C). Although this represents a strong compression of the information, they may prove very useful and effective. They are defined as follows.

Let k_v be the connectivity of v, i.e. the number of links which connect node v (to other nodes in the graph) in graph G; the average of k_v over all the nodes is the average degree of connectivity k of graph G.

A path between node a and b is defined as an ordered set of links which start from node a and end in node b (or viceversa) such that any node is traversed only once; the length of a given path is the number of links it contains. The distance between a pair of nodes a and b, d(a,b), is the length of a shortest path between the two nodes. For each vertex v let d_v be the average of d(v,a) taken over all the other n-1 nodes. The characteristic path length L of graph G is then the median of d_v taken over all the vertices [7].

The *neighbourhood* Γ(v) of node v is the subgraph consisting of the nodes adjacent to v (excluding v itself). Let ε_v be the number of edges in Γ(v), i.e. the number of links which connect nodes which are both adjacent to v. The clustering coefficient of node v is the ratio between this number and the maximum number of possible edges in Γ(v). i.e.:

$$\gamma_v = \frac{\varepsilon_v}{\binom{k_v}{2}} \qquad \textbf{(1)}$$

The *clustering coefficient* C of graph G is then defined as the average of γ_v, taken over all the nodes of the graph.

A graph is

- simple if multiple edges between the same pair of nodes are forbidden.
- sparse, if k << n (the value corresponding to a "fully connected" graph)
- connected, if any node can be reached from any other node by means of a path consisting of a set of edges (there are no separate islands).

The topology of cellular automata is that of simple, sparse connected graphs that are spatially regular, i.e. they are d-dimensional lattices. A one-dimensional CA ring with degree of connectivity k is characterised by [7]:

$$L = \frac{n(n+k-2)}{2k(n-1)} \approx \frac{n}{2k} \qquad C = \frac{3}{4}\frac{k-2}{k-1}. \qquad \textbf{(2)}$$

If the lattice were d-dimensional, L would scale as $n^{1/d}$ for large n, while C would be independent from n.

The d-dimensional lattices constitute an important limiting case, and their natural counterparts are the random graphs, which are defined by the fact that any pair of vertices is connected by a link with a fixed probability p. In sparse random graphs (p<<1) both *L* and *C* take small values for large n (for a thorough discussion, see [9]); it turns out that the distribution of the number of connections per node is approximately Poissonian and that $L \cong \log(n)/\log(k)$, and $C \cong p = k/n$.

3 The Algorithm of Minimal Perturbation

The best known models which account for the small world phenomenon are those of Watts and Strogatz (WS) [12] and the scale free (SF) [13] model. These different algorithms have nodes with different connectivities: the WS model perturbs a regular lattice, giving rise to a distribution of connectivities which is approximately Poissonian, while the SF model is built from the very beginning with a distribution of connectivitities (which turns out to follow a power law). Besides providing some long-range connections, the introduction into the graph of some nodes whose connectivity degree is higher (or smaller) than the average may have two further effects:

- the previous condition of homogeneity among the nodes no longer holds;
- the dynamical rule of the node whose connectivity has changed must be changed, too.

It is impossible to tell in advance whether the consequences of these changes are more or less important than those due to the introduction of some random long-range connections in a regular topology like that of a CA. We present here an algorithm which introduces at random some long range connections, but which doesn't change the connectivity of any node of the system, and doesn't change the connectedness of the system. In this way we are able to introduce a topological perturbation into the system without any other change of the system parameters (number of connections, transition function, etc.): in this sense, we refer to this algorithm as a "minimal perturbation algorithm". We will show that it displays the small world phenomenon for a range of parameter values.

The algorithm, for an arbitrary undirected graph, involves the following steps:

1. select two nodes (A_1 and A_2) that are not directly connected to each other;
2. for each node, select one of the neighbouring ones (B_1 and B_2 nodes);
3. check the existence of a path that connects B_1 with A_2, and another path which connects B_2 with $A_{1;}$ these paths don't pass through the other already selected nodes; if these paths don't exist, return to point 1;
4. add an edge between A_1 and A_2 and an edge between B_1 and B_2;
5. delete the edges between A_1 and B_1, and between A_2 and B_2.

Let us remark that each node maintains exactly the same number of links as before, and that the algorithm can be used for any kind of connected graph, either regular or not (if the graph were directed, a similar algorithm with three cutting points instead of two could be applied, with the same results, but here we consider only undirected graphs).

In order to verify the small world properties, we performed a series of simulations based upon the perturbation of a 1-lattice with a fixed connectivity degree for each node, measuring both L and C as a function of the fraction f of links which have been subject to the redirection algorithm described above. The number of nodes was 1000 and the tests were run for connectivities ranging between 4 and 10. Typical results are shown in fig. 1 (note that f may exceed 1 as the algorithm can be applied indefinitely).

It is interesting to observe that already for very small f values, where the clustering remains high (and where most of the links are regular) the introduction of a few random connections leads to a drastic decrease of the characteristic path length L. The small world region corresponds to the interval of values of f where C is still high while L is small with respect to the values they take in case of large f values (which corresponds to a high degree of randomness).

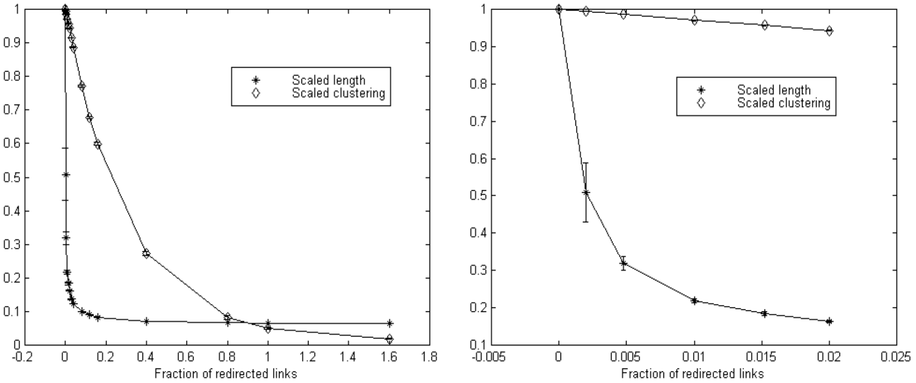


Fig. 1. The characteristic path length *L* and the clustering coefficient *C* vs. the fraction of redirected links f (on the right, a magnification of the small world region). n=1000; k=10; each point is the average of the values obtained from 10 different networks

4 The Dynamical Properties

In order to study the effects of the topological properties upon the system dynamics we considered a particular case, i.e. boolean automata whose transition function obeys the majority rule. The state of cell i at time t+1 equals the state of the majority of its neighbours at the previous time step. If the number of neighbours is even, then the state is left unchanged in case of parity. This rule or similar ones have been extensively studied in the cellular automata literature [15]. It is also related to the so-called majority problem, which stimulated many researches on the effectiveness of genetic algorithms [16] and on computational mechanics [17].

It should be made precise that in the following simulations

- cell i itself is excluded from the counting, i.e. there is no self-coupling in the graph
- asynchronous updating is always adopted, with a random choice at each time step of the cell to be updated
- the regular topology we start from is that of a 1D ring with k/2 connections on each side of cell i

At this stage we cannot claim any generality of the results which follow, and we limit to report the results of the perturbation of the topology on this specific rule.

It is known that, in regular CA, the majority rule with asynchronous updating admits several fixed point attractors, which can be loosely described as arising from a division of the cellular space into homogeneous domains [15]. As k is increased, the size of the domains expands, and their number is reduced. Starting from an initial condition with a high prevalence of either 1 or 0, the system is likely to reach a uniform state of "all 1" or "all 0". We will give a special name to these uniform attractors, U1 and U0 respectively.

Let us consider first the case where the topology is regular and no link has been redirected (i.e. f=0), and let r be the fraction of "1" in the initial conditions (for obvious symmetry reasons, we can limit our discussion to the interval $r\in[0,1/2]$). By performing experiments with 1000 different initial conditions, one observes that, if r is very small, all the initial states tend to the U0 attractor. By increasing r, one observes that also other attractors are reached and, in experiments with r=0.2, it may happen that some hundreds of attractors are actually reached (table 1). An interesting feature is that all these attractors share a large common part with U0, i.e. most of their nodes (>90%) take the value 0, and the Hamming distances among these attractors are very small. They can then be considered as "perturbations" of the uniform U0 attractor, with some "regions of disorder". Let $\phi_0(v)$ be defined, for each node v, as the fraction of initial conditions that leads to a final state in which node v takes the value 0. For each network, ϕ_0^{min} and ϕ_0^{max} are defined respectively as the minimum and maximum value of $\phi_0(v)$. We can see that in the f=0 case, while $\phi_0(v)$ is close to 1 for most nodes, ϕ_0^{min} can take also small values.

When r=0.5, the initial state has no prevalence of either 0 or 1: here every initial condition out of 1000 almost always leads to a different fixed point, but now these are more different from each other, as it can be seen by examining their Hamming distances, which take large values (table 1) and show a distribution around the value 500 (which is the average distance between two random boolean vectors of 1000 elements each). Note also that ϕ_0^{min} and ϕ_0^{max} are close to 0.5.

We then modified the fraction of redirected links. It is interesting to notice that the introduction of long-range connections may modify the dynamical behaviour. As it can be seen from table 1 when r=0.2 but f=0.8 one finds that only the U0 attractor is reached: the presence of long range connections seems to eliminate the cloud of attractors similar to U0 which were reached in the case of a regular topology.

This guess is confirmed by considering the r=0.5 case. When r=0.5 the average number of attractors which are reached from 1000 random initial conditions shows a large decrease between f=0.4 and f=0.8 (fig. 2). By considering the distribution of Hamming distances, one observes that in the f=0.8 case there are the uniform attractors U0 and U1 (with fairly large basins of attraction, see table 1), and a number of other attractors whose Hamming distances are distributed around 500. The uniform attractors are not surrounded by a cloud of slightly different fixed points, but there are still a large number of attractors which are markedly different from both.

It is interesting to consider also the case of a graph which is closer to the regular one, e.g. the f=0.08 case (table 1), where one can see that the dynamical behaviour is already affected by the introduction of a few random long range connections. At r=0.2 the number of reached fixed points is reduced, with respect to the regular case, and the basin of attraction of U0 is larger. The cloud of similar attractors is shrunk, but has not yet disappeared as in the f=0.8 case. In the r=0.5 case one observes that 1000 random initial conditions are mapped into 1000 different fixed points, as in the regular graph. An indication that shows that the dynamical properties are modified comes from the values of ϕ_0^{min} and ϕ_0^{max}, which are intermediate between the regular and the disordered case. Moreover, the average number of domains of this zone is already close to the random graph value (fig.2).

Table 1. Data about the dynamical properties of the systems at different degree of randomness f (fraction of redirected links). Data concerning simulations performed on 10 networks of 1000 nodes with 1000 initial conditions each are shown. A_{1000}≡Number of final states reached from 1000 random initial conditions. B_{1000}≡number of initial conditions (out of 1000) which belong to the largest basin of attraction. S_0≡number of nodes (out of 1000) which always take the value 0 in the final state, in a given network. H_{med}≡ average Hamming distance between two different fixed points. ϕ_0^{min} and ϕ_0^{max} are defined in the text

r=0.5	**f=0**		**f=0.08**		**f=0.8**	
	Average	**St. Dev.**	**Average**	**St.Dev.**	**Average**	**St.Dev.**
A_{1000}	1000	0	1000	0	298	14
B_{1000}	1	0	1	0	365	13
S_0	500	2	499	3	500	4
H_{med}	501	1.0	499.6	1.6	501	4.0
ϕ_0^{min}	0.46	0.005	0.29	0.01	0.33	0.01
ϕ_0^{max}	0.55	0.01	0.72	0.013	1	0
r=0.2	**f=0**		**f=0.08**		**f=0.8**	
	Average	**St.Dev.**	**Average**	**St.Dev.**	**Average**	**St.Dev.**
A_{1000}	196	176	37	58	1	0
B_{1000}	313	303	721	230	1000	0
S_0	955	24	966	10	1000	0
H_{med}	14	4	12	3	…	…
ϕ_0^{min}	0.19	0.13	0.98	0.01	1	0
ϕ_0^{max}	1	0	1	0	1	0

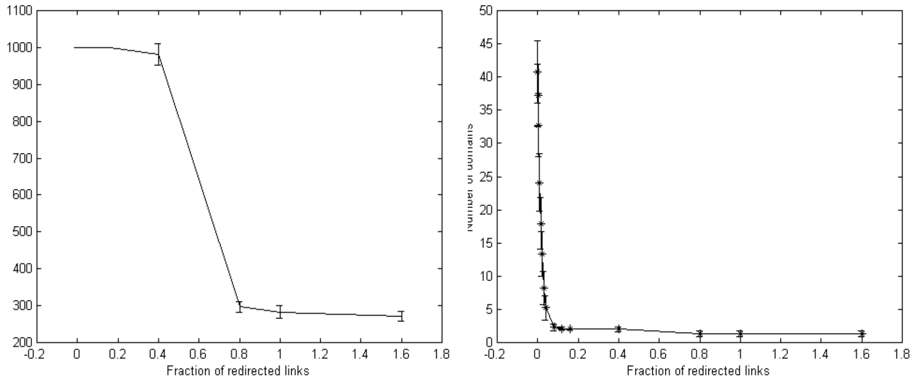


Fig. 2. Majority rule: average number of attractors (left) and of homogeneous domains (right) versus the fraction f of redirected links in the case r=1/2; Each point is the average of the values obtained from 10 different networks; n=1000, 1000 different initial conditions for each net

An interesting case is also that of a highly randomized lattice; Fig. 3 shows that, in the case f=1.6, the number of different attractors which are reached from 1000 random initial conditions starts to grow at a much higher r value (beyond 0.4) than in the regular case, and that the peak reached around r=0.5 corresponds to a number of attractors which is much smaller than in the regular case. In the case r=0.5 there are two major attractors, with equal basins of attraction, which together cover about 80% of the initial conditions.

As far as transients are concerned, there is a nearly linear increase as r grows; this is similar to what is observed also in the regular case, but the maximum transient duration is much higher (transient duration also displays a very high variance in the region near 0.5).

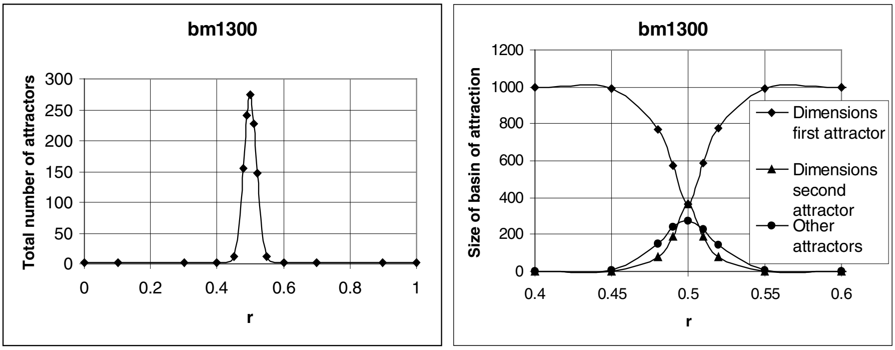


Fig. 3. Majority rule on a randomized graph (f=1.6): average number of attractors reached starting from 1000 random initial conditions (left); fraction of initial conditions which are attracted by the largest, second largest and "all the other" stable points (right). All the variables are shown as a function of r (n=1000; k=10)

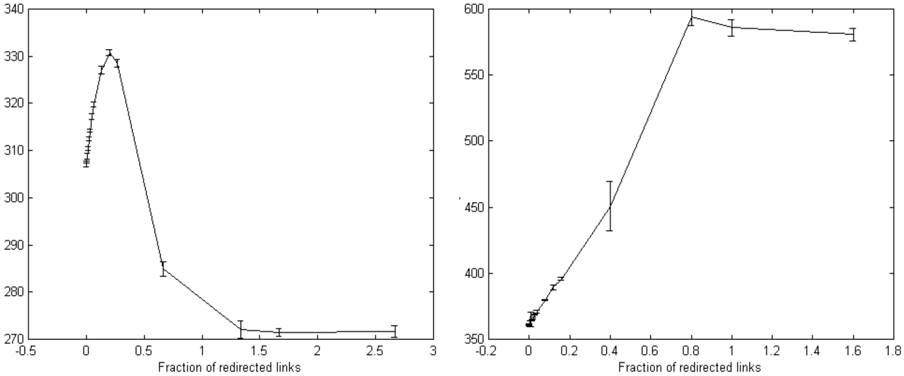


Fig. 4. Duration of transients (time stpes needed to reach the fixed point) as a function of the degree of randomness at k=6 (left) and k=10 (right). Each point is the average of the values obtained from 10 different networks; n=1000, 1000 different initial conditions for each net

Finally, it is also interesting to observe the behaviour of the transients as a function of the degree of randomness. At high k values their duration increases with increasing f (as it might be intuitively expected, since the introduction of many long-range connections interferes with the establishment of local domains largely independent from the configurations which take place far away). However, at smaller k values the transient duration displays a maximum for an intermediate value of k, and then declines, as shown in fig. 4.

5 Conclusions

Further work is needed to explore the effects of topological perturbations in the case where different transition functions are used, like e.g. in boolean models of genetic circuits [3,4,6], as well as in the case of different updating methods, like synchronous updating.

However, the results reported here already demonstrate that the introduction of long range connections (without changing the number of connections of every node) can have a profound effect on the dynamics. In the case of a small world network, it has also been shown that major dynamical features are affected at a fairly small fraction of redirected links. While the literature on this subject is still limited, by taking into account other results (see e.g. [14] and [18]) one is led to guess that the influence of topological modifications of this kind on the dynamics is a generic property.

In the particular case considered here, it has been observed that these changes lead, with respect to the regular case, to a decrease in the number of attractors that are reached, which can be observed even at small f values. Qualitatively, one sees that a "cloud" of attractors that correspond to minor perturbations of the major, uniform attractors disappear. This is a very nice property that holds in this case, and the possibility of finding similar features in different dynamical rules has yet to be examined.

References

1. Gilbert,N., Troitzsch, K.G.: Simulation for the social scientist. Buckingham (UK): Open University press (1999)
2. Serra, R., Villani, M., Salvemini, A.: Continuous genetic networks. Parallel Computing **27**, (2001) 663-683
3. Kauffman, S.A.: Behavior of randomly constructed nets: binary element nets. In: C.H. Waddington (ed): Towards a theoretical biology. Vol.3. Edinburgh University Press (1970)
4. Kauffman, S.A.: The origins of order. Oxford University Press (1993)
5. Serra, R., Zanarini, G.: Complex systems and cognitive properties. Springer-Verlag (1990)
6. Serra, R., Villani, M.: Modelling bacterial degradation of organic compounds with genetic networks. J. Theor. Biol. **189** (1) (1997) 107-119
7. Watts, D.J.: Small worlds: the dynamics of networks between order and randomness. Princeton University press (1999)

8. Amaral, L.A.N., Scala, A., Barthelemy, M, Stanley, H.E.: Classes of small world networks. PNAS **97**, (2000) 11149-11152
9. Reka, A., Barabasi, A.L.: Statistical mechanics of complex networks. arXiv:cond-mat/0106096v1 (2001)
10. Wagner, A., Fell, D.: The small world inside large metabolic networks. Tech. Rep. 00-07-041, Santa Fe Institute (2000)
11. Jeong, H. et al.: The large scale organization of metabolic networks. Nature **407**, (2000) 651-654
12. Watts, D.J, Strogatz, S.H.: Collective dynamics of small world networks. Nature **393** (1998) 440
13. Barabasi, A.L., Albert, R.: Science **286**, (1999) 509
14. Strogatz, S.H.: Exploring complex networks. Nature **410**, (2001) 268-276
15. Toffoli T., Margolus N.: *Cellular automata machines*. MIT Press (1987)
16. Mitchell,M.: An introduction to genetic algorithms. MIT Press (1996)
17. Crutchfield, J.P., Mitchell, M.: The evolution of emergent computation. Proc. Natl. Acad. Sci. USA 92 (1995) 10742-10746
18. Wang, X.F., Chen, G.: Synchronization in scale-free dynamical networks: robustness and fragility. IEEE Trans. Circuits and Systems **49** (1) (2002) 54-62

A Path-Planner for Mobile Robots of Generic Shape with Multilayered Cellular Automata

Fabio M. Marchese

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano - Bicocca
Via Bicocca degli Arcimboldi 8, I-20126, Milano, Italy
Marchese@DISCo.UniMIB.IT

Abstract. In this paper we present a Path-Planning Algorithm for a non-holonomic mobile robot. The robot we considered has to move using smoothed trajectories, without stopping and turning in place, and with a minimum steering radius. We have studied an algorithm based on a directional (anisotropic) propagation of attracting potential values on a Multilayered Cellular Automata model. The algorithm finds all the optimal collision-free trajectories following the minimum valley of a 3D potential hypersurface embedded in a 4D space, built respecting the imposed constraints. Our approach turn out to be distributed and reactive to environmental dynamics. It is also flexible, because it is applicable on a wide class of vehicles with generic shapes and with different kinematics.

1 Introduction

In this paper we describe a very fast, safe and complete path-planning method for robots based on Multilayered Cellular Automata. Our aim is to design and construct a robot navigation system that interacts with the environment and reacts as fast as possible to its dynamical events. Many authors have proposed different solutions during the last twenty years, based, for example, on a geometrical description of the environment (e.g. [7,8]). The path-planners working on these models generate very precise optimal trajectories and can solve really difficult problems, also taking into account non-holonomic constraints, but they are very time consuming, too. In our opinion, to face a real dynamical world, a robot must constantly sense the world and re-plan accordingly to the new information.

Other authors have developed alternative approaches less precise, but more efficient: the Artificial Potential Fields Methods. In the eighties, Khatib [5] first proposed this method for the real-time collision avoidance problem in a continuous space. Jahanbin and Fallside first introduced a wave propagation algorithm in the Configuration Space on discrete maps (*Distance Transform* [4]). In [2], the authors used the Numerical Potential Field Technique on the Configuration Space to build a generalized Voronoi Diagram. Zelinsky extended the *Distance Transform* to the *Path Transform* [12]. Tzionas et al. in [11] described an algorithm for a diamond-shaped holonomic robot in a static environment, where

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 178–189, 2002.

they let a CA to build a Voronoi Diagram. In this paper, we have used CA as a formalism for merging a Grid Model of the world (Occupancy Grid) with the Configuration Space ($\mathcal{C}$*-Space*) of a robot and Numerical (Artificial) Potential Field Methods, with the aim to give a simple and fast, even if sub-optimal, solution for the path-planning problem for a non-holonomic mobile robot. This method uses a directional (anisotropic) propagation of distance values between neighbor automata to build a potential hypersurface embedded in 4D space. Using a constrained version of the descending gradient is possible to find out all the admissible, equivalent and shortest (for a given metric of the space) trajectories that connect two points of the robot $\mathcal{C}$*-Space*.

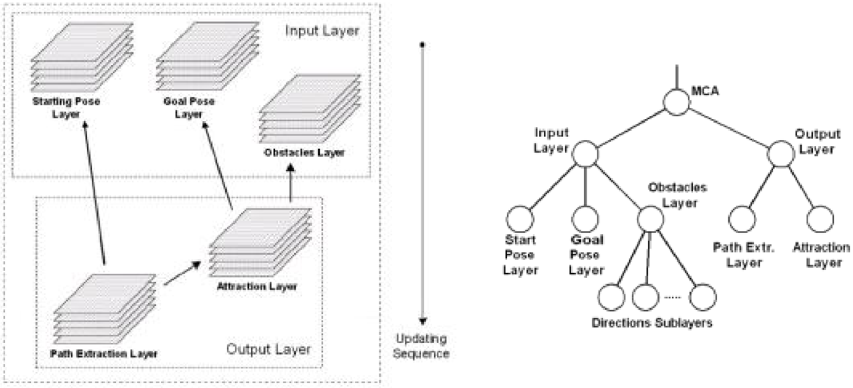
2 Problem Statements

A wide variety of world models can be used to describe the interaction between an autonomous agent and its environment. One of the most important is the Configuration Space ($\mathcal{C}-$*Space*) [6,8]. The $\mathcal{C}-$*Space* of a rigid body is the set of all its configurations. If the robot can translate and rotate on a 2D surface, its $\mathcal{C}-$*Space* is a 3D manifold $\mathbb{R}^2 \times \mathbf{SO}(2)$. It can be modelled as a 3D Bitmap $\mathcal{GC}$ ($\mathcal{C}$*-Space Bitmap*), represented by the application $\mathcal{GC} : \mathcal{W} \rightarrow \{0,1\}$, where 0s represent non admissible configurations. The $\mathcal{C}-$*Potential* is a function $\mathbf{U}(\mathbf{q})$ defined over the $\mathcal{C}-$*Space* that "drives" the robot through the sequence of configuration points to reach the goal pose [2]. Let us introduce some other assumptions: 1) space topology is finite and planar; 2) the robot has a lower bound on the steering radius (non-holonomic vehicle). The latter assumption introduces important restrictions on the types of trajectories to be found.

Cellular Automata are automata defined on a Cellular Space $\mathbb{Z}^n$ with transition functions invariant under translation [3]: $\mathbf{f}_c(\cdot) = \mathbf{f}(\cdot), \forall c \in \mathbb{Z}^n$, $\mathbf{f}(\cdot) : \mathbf{Q}^{|A_0|} \rightarrow \mathbf{Q}$, where $\mathbf{c}$ is the coordinate vector identifying a cell, $\mathbf{Q}$ is the set of states of an automaton and $\mathbf{A}_0$ is the set of arcs outgoing from a cell to the neighbors. The mapping between the Robot Path-Planning Problem and CA is quite simple: every cell of the $\mathcal{C}-$*Space Bitmap* $\mathcal{GC}$ is an automaton of a CA. The state of every cell contributes to build the $\mathcal{C}-$*Potential* $\mathbf{U}(\mathbf{q})$ through a diffusion mechanism between neighbors. The way the diffusion is performed (in this case with an anisotropic mechanism), contributes to respect the given constraints. The trajectories are found following the minimum valley of the surface $\mathbf{U}(\mathbf{q})$. In this work, we use a simple extension of the CA model: we associate a vector of attributes (state vector) to every cell. Each state vector depends on the state vectors of the cells in the neighborhood. There is a second point of view: this is a Multilayered Cellular Automaton [1], where each layer corresponds to a subset of the state vector components. Each subset is evaluated in a single layer and depends on the same attribute of the neighbor cells in the same layer and depends also on the states of the corresponding cell and its neighbors in other layers. In the following section, we describe each layer and the transition functions implemented in its cells.

3 Multilayered CA Architecture

The system is composed of five CA layers, each one has three or more dimensions. They are conceptually grouped in two major subsystems: an Input Layer and an Output Layer (Fig. 1.a). The Input Subsystem is thought as an interface with the outside environment. Its layers have to react as fast as possible to the "external" changes: the robot starting pose, the robot goal pose, and, the more important, the changes of the environment, i.e. the movements of the obstacles in a dynamical world. Through a sensorial system (not described here), these changes are detected and the information is given to the planner via the Input Subsystem, even during the updating phase, thus having a reactive planner. The first two layers (Start and Goal Pose L.) are considered statical, because they are update only from the outside of the system and with a frequency lower than the internal updating frequency; the third one (Obstacles L.), instead, is updated externally and also evolves to search all the admissible movements as described in section 3.2. The Output Subsystem returns the results of the planning, that is an entire trajectory (a sequence of passing points) from the Path Extraction L., or a single motion step from the Attraction L. In the following subsections, we briefly describe each layer and the transition functions implemented in its cells.



a) b)

Fig. 1. Layers Architecture: a) dependency graph; b) layers hierarchical structure

3.1 Goal_Position Layer and Starting_Position Layer

The *Goal Position Layer* specifies the goal pose of the robot. The attribute in each cell is a boolean: $GPos_c \in \{True, False\}^8$. Each cell c contains a vector of eight boolean variables, one for each possible outgoing direction, where the robot orientation θ has been sampled at eight values $\mathcal{D} = \{N, NE, E, SE, S, SW, W, NW\}$. All the attributes of the cells are normally set to $False$; only the final orientation in the goal cells is set to $True$. It represents the desired goal configuration $q_g = (x_g, y_g, \theta_g)$. Similarly, the *Starting Position Layer* represents the starting pose of the robot. It is initialized with the starting configuration $q_s = (x_s, y_s, \theta_s)$. Both layers are statical and do not evolve during the computation.

3.2 Obstacles Layer

The *Obstacles Layer* is used to map an extended representation of the obstacles and of the environment boundary. In Regular Decomposition world models, the obstacles are decomposed in cells full or free (Occupancy Grids, e.g. Fig. 3.a) and the robot is often represented as a material point (the robot cinematic center) moving from one free cell to a neighbor free one. To take into account of its real extension, the well-known technique of enlarging the obstacles by a given quantity (the robot major radius) can be used [7]. A simple isotropic enlargement with a constant radius would have the same result as with an equivalent cylindrical robot. The consequence is a great loss of space around the obstacles, ad a loss of executable trajectories. An anisotropic enlargement [7], i.e. a different enlargement for each robot orientation, would solve the problem also for asymmetric robots (wrt the cinematic center), but only partially: counter-examples can be found where the robot still collides with obstacles due to the sweeping of its silhouette between two consecutive poses, as shown in Fig. 2.c. The problem rises when a coarse orientation discretization is used: the robot cinematic center (cross in Fig. 2.a-b) does not overlap the expanded obstacle (hatched), i.e. the dual obstacle, generated by the circle next to it. Unfortunately, during the rotation, the robot still collides with it (Fig. 2.c). To face also these situations, a more accurate representation is needed. This layer mainly defines the topology of a space in which both the $\mathcal{C}-Space$ and the movements set are represented.

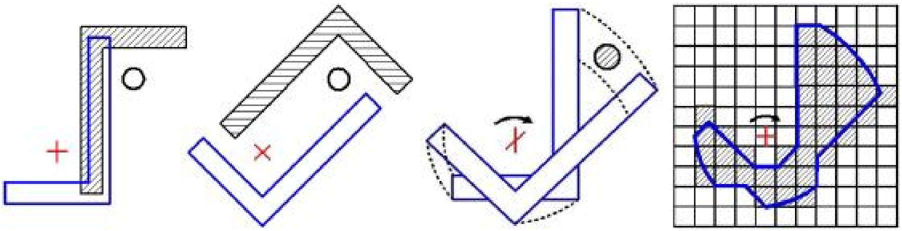


Fig. 2. Silhouette sweeping: a-b) expanded obstacle (hatched) for two robot orientations (white); c) a counter-example due to a coarse discretization of the orientation; d) a motion silhouette/cell neighborhood (hatched cells), obtained sweeping the robot silhouette between the two poses

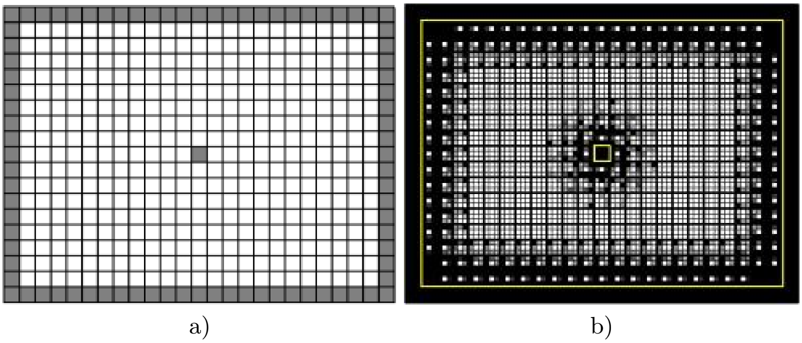


Fig. 3. An example of a) a world map (Occupancy Grid), b) the result of the evolution of the Obstacles Layer for the "L-Robot" of Fig. 2.a

The latter is the set of all the admissible movements as defined by the specific kinematics of the robot. The admissibility of a move is also influenced by the vicinity of the obstacles: an obstacle too close to the robot has the effect to inhibit some of its movements. The attribute evaluated in this layer (*Obstacles Attribute*) is a boolean: $Obst_c(\theta, \mu) \in \{True, False\}$, and represents the admissibility of each move in each robot pose, on the base of the obstacles distribution also. Therefore, this is a 4D CA and its transition function is the application: $\mathbb{R}^2 \times \mathbf{SO}(2) \times \{Move\} \rightarrow \{True, False\}$.

$$Obst_c(\theta, \mu, t+1) = \begin{cases} False & if\ \exists a \in A_0 :\ isObstacle(c+a) \wedge (c+a) \in \mathcal{GC} \\ Obst_c(\theta, \mu, t) & otherwise \end{cases}$$

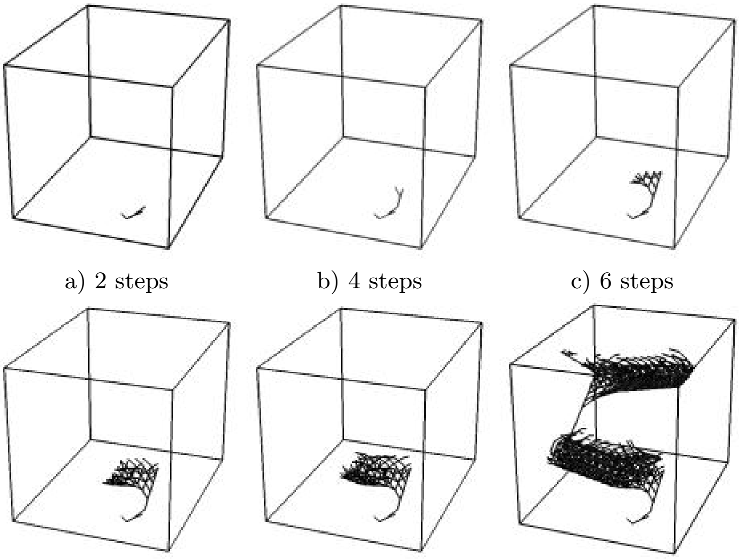
$\forall t > 0, \forall c \in \mathcal{GC}, \forall \theta \in \mathcal{D}, \forall \mu \in \{Move\}$

As a matter of fact, this layer is composed with sublayers, one for each robot orientation and move: it is itself a Multilayered CA on a 2D domain (the space $\mathbb{R}^2$ of positions). It is a particular CA also for another reason: each sublayer has a non-standard fixed architecture [10], i.e. the neighborhood does not have the standard square shape. The neighborhood shape reflects the motion silhouette of the robot during a movement (sweeping) as in the example of Fig. 2.d. In fact, the above transition function simply realizes a collision test: given a robot pose (x, y, θ) and an admissible (just for the kinematics) move μ, it will test if, during the movement, the silhouette overlaps an obstacle. In this case, the cell (x, y, θ, μ) is marked $False$, with the meaning that the move μ, normally admissible, is not admissible for that particular pose because of the presence of the obstacle. In other words, the presence of an obstacle cell inhibits a number of movements in the surrounding cells. The test is implemented very easily: each automaton searches in its (non-standard) neighborhood if there is an obstacle cell. If there is, the robot cannot execute the move without to collide with an obstacle. The test is very fast because all the neighborhood shapes are automatically precalculated off-line starting from the basic robot silhouette. In Fig. 3.b is shown the result of the evolution of the layer for a "L-Robot" from the Occupancy Map (Fig. 3.a).

The cells are subdivided in 9 subcells, each one representing one of the eight robot orientations and movement directions. Each subcell has a grey level, ranging from white (all moves admissible) to black (no move). The middle cell is black iff all the moves are not admissible, white otherwise. The planner will try to find the shortest trajectories that connect the starting cell to the goal cell exploiting exclusively the admissible moves in each passing cell.

3.3 Attraction_to_Goal Layer

This is the core of the entire Path-Planning Algorithm. The *Attraction_to_Goal Layer* is defined as: $Attr_c(d_{out}) \in \mathbb{N}^8$. It is a vector of eight values corresponding to the eight main directions. It represents the integer distance of the cell c from the goal cell if the robot moves to the $d_{out} \in D$ direction along a collision-free path. It is a digitalized representation of the $\mathcal{C}-$*Potential* function $\mathbf{U}(\mathbf{q})$ defined on the $\mathcal{C}-$*Space Bitmap.* To evaluate the path length, we have introduced a set of costs (weights) for each robot movement: (*forward, forward_diagonal, direction_change, stop, rotation, backward, backward_diagonal*). It is not a metric in the mathematical sense: it has been called in such a way because it defines a method to evaluate the trajectory length. The robot is subjected to non-holonomic constraints, therefore not every movement can be done in every robot pose. We have introduced a subset of admissible moving directions $D'(c, d) \subseteq D$ depending on the robot position (cell c) and orientation d which satisfy the



a) 2 steps b) 4 steps c) 6 steps

d) 8 steps e) 10 steps f) 40 steps

Fig. 4. Attraction Potentials Skeletons at different evolution steps

maximum curvature constraint. From a configuration (c, d), the robot can achieve a set of configurations $(\gamma, \delta), \forall\gamma \in I'_c(d), \forall\delta \in D'(\gamma, d)$, where $I'_c(d) \subseteq I_c$ is the subset of neighbor cells reachable from the configuration (c, d), and $D'(\gamma, d) \subseteq D$ is the subset of admissible orientations at the destination cell γ compatible with the leaving direction d. The transition function is defined as follows:

$$Attr_c\,(d, t+1) = \begin{cases} 1 & if c = goal\_cell \wedge d = final\_direction \\ minAttr_c\,(d, t) & if\ \neg Obst_c\,(d, t) \wedge Attr_c\,(d, t) \neq minAttr_c\,(d, t) \\ Attr_c\,(d, t) & otherwise \end{cases}$$

$\forall t > 0, \forall c \in \mathcal{GC}, \forall d \in D$
where: $minAttr_c(d, t) = \min_{\substack{\forall\gamma\in I'_c(d) \\ \forall\delta\in D'(\gamma,d)}} \{Attr_\gamma\,(\delta, t) + Cost\,(c, d, \gamma, \delta)\}$ and $Cost(\cdot, \cdot, \cdot, \cdot)$ is the length/cost of the move from the current configuration (c, d) to the configuration (γ, δ), and $Obst_c\,(d, t)$ is the admissibility value of the same move evaluated in the Obstacle Layer. If we change the movement costs, we can consider robots with different types of kinematics. For example, the kinematics $(2, 3, 1, 0, High, High, High)$ emulates a car-like kinematics moving only forward (Dubin's car), while the kinematics $(2, 3, 1, 0, High, 2, 3)$ emulates a common car-like kinematics moving also backward (Reed's and Shepp's car). Two main properties can be demonstrated: the termination of the propagation and the absence of local minima (refer to [9]). The later property is very important for our aims: it ensures to achieve the goal (the global minimum) just following the negated gradient vector without stalling in a local minimum. The graphical representation of the potential surface is quite difficult. To be able to infer its aspect, we must reduce the number of dimensions projecting it from the 4D space to a 3D space, and obtaining a skeleton laying on the hypersurface which gives an approximate idea of its topology. This skeleton is a tree (the root is in the goal) composed of the admissible robot movements that connect one cell to another. In Fig. 4 are shown some evolution steps of the growing of the skeleton tree.

3.4 Paths_Extraction Layer

This layer determines all the shortest paths that connect the starting point to the goal point on the base of potential hypersurface $\mathbf{U}(\mathbf{q})$ computed in the previous layer. The method extends the descent gradient methods described in [4,12]. Because of the steering constraint, we have to use a constrained negated gradient, i.e. the gradient vector is evaluated only for the admissible directions. The resulting trajectories (Fig. 5.a) are subsets of the attraction skeleton of Fig. 4.f, i.e. only those parts of the skeleton which connect the starting cell to the goal cell belong to the final trajectories.

4 Algorithm Properties

This algorithm has the advantage of being computable in an asynchronous way: there is no specific evaluation order of the cells in each layer, and every layer can be updated without waiting that the other layers have reached a stationary point.

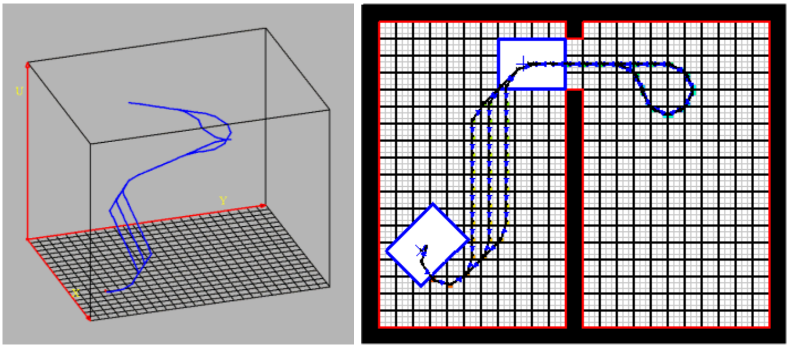


Fig. 5. 3D and 2D Projection of Path Potentials

A second important property is related to the consistency of the solution found. For a given environment (i.e. obstacles distribution) with a given starting and goal cell, the solution found, if it exists, is the set of all optimal paths for the given metric. The CA evolution can be seen as a motion from one point to another point of the global state space until an optimal solution is reached. This is a convergence point for the given problem or a steady global state. If we make some perturbations, such as changing the environment (adding, deleting or moving one or more obstacles) or changing the goal cell, then the point becomes unstable and the CA starts to evolve again towards a new steady state, finding a new set of optimal trajectories. We called this property as *Incremental Updating*. The CA spontaneously evolves to a new steady state from the previous one, without to be reset and re-initialized, therefore realizing a *Reactive Path-Planner*: a path-planner that reacts to external (environmental) changes. The algorithm complexity is strictly related to the obstacles distributions and it is impossible to evaluate it because of the huge number of possible obstacles distributions. We can only make a good estimate of the upper-bound. In the worst cases, the number of free cells is approximatively $N/2$, where N is the total number of cells of the *$\mathcal{C}$-Space Bitmap*, therefore, the longest distance covered is nearly of $N/2$ cells. The longest paths (going farthest and then coming back) cover $2\frac{N}{2}$ cells and require about N updating steps to be computed. Thus, the upper-bound of the complexity is $O(N^2)$. In cluttered environments, the result can be much better because the number of free cells is lower.

5 Experimental Results

In this section, we illustrate only few experimental results (because of the limited space) obtained with the algorithm previously described.

5.1 Robots with Different Kinematics

In Fig. 6 are shown the solutions of a path-planning problem in a more complex environment using two different kinematics. In the first example the robot can move only forward. To reach the goal it has to turn around an obstacle to drive through the narrow passage. In the second, the robot can also move backward and rotate in the same place. Exploiting this kinematics, the first part of the trajectory is followed moving forward until the robot reaches the narrow passage, then in the second part it moves backward up to the goal. The results show that a robot, rotating also in the same place, has a wider set of alternative trajectories. The performance tests, carried out with an Intel Pentium III 1 GHz PC, gives the following results: 78.2 ms and 43.1 ms respectively. These mean times are evaluated over 1,000 experiments for each problem and include the initialization phase of each test.

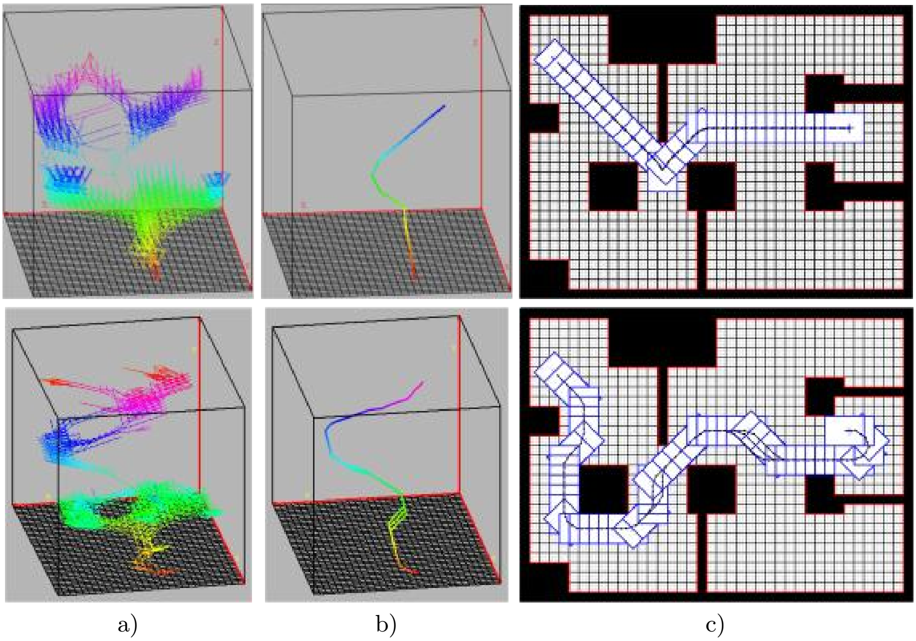


Fig. 6. An example of a more complex world: a) Attraction Potentials Skeleton; b) Paths Skeleton; c) Trajectory, for robots kinematics with and without backward movements respectively

5.2 Maneuvering Examples

In this section are reported just few experiments to highlight the flexibility of this planner. It has been used with robots/objects with different shapes, facing different situations and it has generated interesting solutions. In Fig. 7, two examples of robots maneuvers are shown: a) a classic car-like robot parking; b) the motion of an L-Robot in a real cluttered word. These are only the solutions found with a set of costs for the movements: if we change the relative weight of the movements, without changing the type of kinematics, other solutions can be found. In Fig. 8, other examples involving the motion of objects around a peg are shown. It is noteworthy that the obstacles are never overlapped by the robot silhouette, thanks to the algorithm used in the repulsive layer (Obstacles L.).

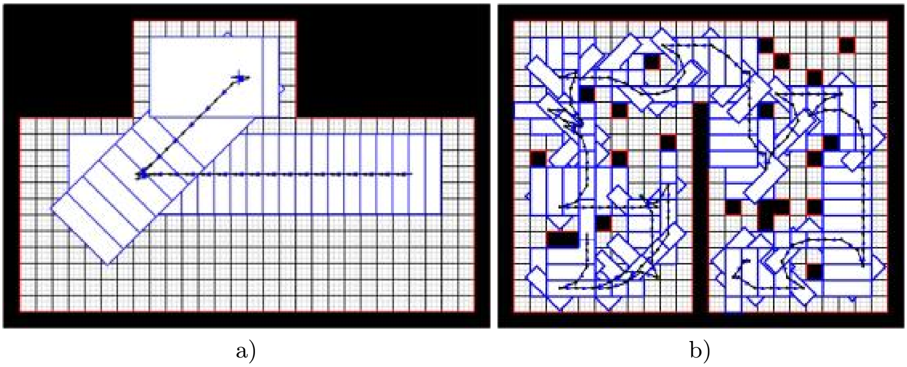


Fig. 7. Maneuvering examples: a) parking a rectangular Robot; b) a complex path of a L-Robot in a cluttered world

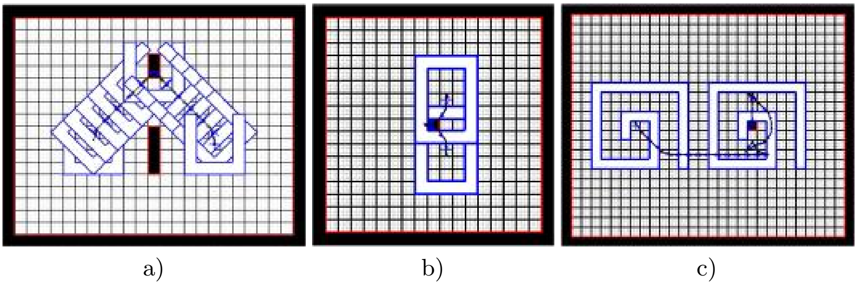


Fig. 8. Maneuvering around a peg: a) U-Robot; b) O-Robot with a peg inside; c) Spiral Robot inserting around a peg;

5.3 Multiple Starting and Goal Positions

Another property of this algorithm is to compute in parallel the trajectories from more than one starting position. In Fig. 9.a, a new starting point is added in the first room of the previous problem. The new paths found, from a certain point on, merge into the old paths, as expected when we have optimal trajectories. Another property is the selection of the nearest goal in a set of goals. Setting more goals, the algorithm finds the paths that reaches the nearest of them along an admissible path. We can also address mixed situations where there are more than one starting and/or goal points: in this particular problem, the results are the shortest paths connecting each starting points to the nearest goal as in Fig. 9.b, in which a goal in the second room has been added.

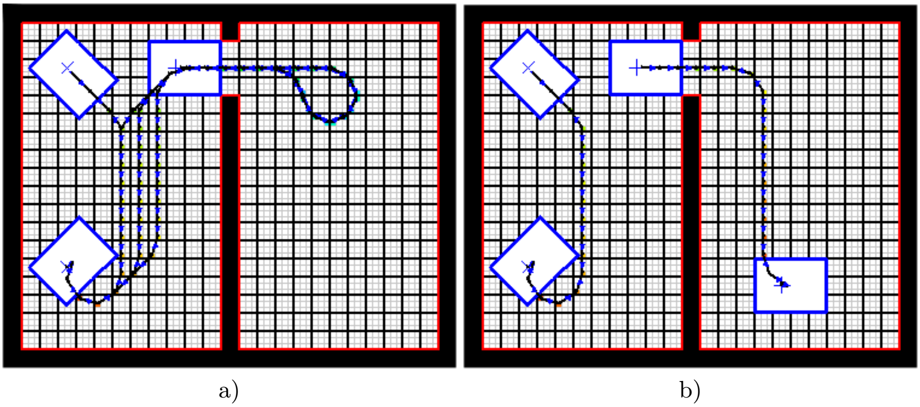


Fig. 9. An example of a two starting points problem (a) and an example with two goal poses (b)

6 Conclusions

In this paper we have described a solution to the Path-Planning Problem for a non-holonomic Mobile Robot with a Cellular Automata approach. Some results and properties of this work can be stated: 1) CA is a good formalism when Decomposition Models are used to represent the environment (Occupancy Grid); 2) the algorithm is flexible, it can be used with different types of kinematics, just changing the set of weights, and with robot with different shapes; 3) it is quite simple to implement this algorithm directly on a SIMD machine; 4) it generates all the collision-free trajectories, also with cluttered obstacles distributions, even allowing to pass more than once in the same position; 5) trajectories found are smoothed, while respecting the imposed kinematics constraints; 6) it is a Reactive Path-Planner: it allows an incremental updating of the trajectories every time a modification of the environment is detected by sensors, or the goal changes (e.g. following other robots).

References

1. Bandini S., Mauri G., Multilayered cellular automata, *Theoretical Computer Science*, 217 (1999), 99-113
2. Barraquand J., Langlois B., Latombe J. C., Numerical Potential Field Techniques for Robot Path Planning, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 22, No. 2 (Mar 1992), 224-241
3. Goles E., Martinez S., Neural and Automata Networks: dynamical behavior and applications, Kluwer Academic Publishers (1990)
4. Jahanbin M. R., Fallside F., Path Planning Using a Wave Simulation Technique in the Configuration Space in Gero J. S., Artificial Intelligence in Engineering: Robotics and Processes, Computational Mechanics Publications (Southampton 1988)
5. Kathib O., Real-time Obstacle Avoidance for Manipulator and Mobile Robots, *Proc. of Int. Conf. on Robotics and Automation* (1985)
6. Latombe J. C., Robot Motion Planning, Kluwer Academic Publishers, Boston, MA (1991)
7. Lozano-Pérez T., Wesley M. A., An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles, *Comm. of the ACM*, Vol. 22, No. 10, (Oct 1979), 560-570
8. Lozano-Pérez T., Spatial Planning: A Configuration Space Approach, *IEEE Trans. on Computers*, Vol. C-32, No. 2 (Feb 1983), 108-120
9. Marchese F. M., Reactive Path-Planning: a Directional Diffusion Algorithm on Multilayered Cellular Automata, *Proc. of the 4th Int. Conf. on CA for Research and Industry (ACRI 2000)*, Karlsruhe (D), (Oct 2000), 81-89
10. Sipper M., Evolution of Parallel Cellular Machines - The Cellular Programming Approach, LNCS 1194, Springer (1997)
11. Tzionas P. G., Thanailakis A., Tsalides P. G., Collision-Free Path Planning for a Diamond-Shaped Robot Using Two-Dimensional Cellular Automata, *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 2 (1997), 237-250
12. Zelinsky A., Using Path Transforms to Guide the Search for Findpath in 2D, *Int. J. of Robotics Research*, Vol. 13, No. 4 (Aug 1994), 315-325

Dynamics of Populations in Extended Systems

Michel Droz[1] and Andrzej Pękalski[2]

[1] Insitut de Physique Théorique, Université de Genève,
quai E. Ansermet 24, 1211 Genève 4, Switzerland,
Michel.Droz@physics.unige.ch,
http://theory.physics.unige.ch/~droz

[2] Institute of Theoretical Physics, University of Wrocław,
pl. M. Borna 9, 50-204 Wrocław, Poland,
apekal@ift.uni.wroc.pl

Abstract. Two models of spatially extended population dynamics are investigated. Model A describes a lattice model of evolution of a predator - prey system. We compare four different strategies involving the problems of food resources, existence of cover against predators and birth. Properties of the steady states reached by the predator-prey system are analyzed. Model B concerns an individual-based model of a population which lives in a changing environment. The individuals forming the population are subject to mutations and selection pressure. We show that, depending on the values of the mutation rate and selection, the population may reach either an active phase (it will survive) or an absorbing phase (it will become extinct). The dependence of the mean time to extinction on the rate of mutations will also be discussed. These two problems illustrate the fact that cellular automata or Monte-Carlo simulations, which take completely the spatial fluctuations into account, are very useful tools to study population dynamics.

1 Introduction

The dynamics of interacting species has attracted a lot of attention since the pioneering works of Lotka [1] and Volterra [2]. In their independent studies, they showed that simple prey-predator models may exhibit limit cycles during which the populations of both species have periodic oscillations in time. However, this behavior depends strongly on the initial state, and is not robust to the addition of more general non-linearities or to the presence of more than two interacting species [3]. In many cases the system reaches a simple steady-state.

In such mean-field type models, it is assumed that the populations evolve homogeneously, which is obviously an oversimplification. An important question in modeling population dynamics consists in understanding the role played by the local environment on the dynamics [4]. There are many examples in equilibrium and nonequilibrium statistical physics showing that, in low enough dimensions, the local aspects (fluctuations) play a crucial role and have some dramatic effects on the dynamics of the system.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 190–201, 2002.

A lot of activities have been devoted during the past years to the study of extended prey-predator models (see [5,6] and references therein). It was shown that the introduction of stochastic dynamics plays an important role [7], as well as the use of discrete variables, which prevent the population to become vanishingly small. These ingredients are included in the so called individual agents based lattice models and it is now recognized that these models give a better description of population dynamics than simple mean-field like models. Both synchronous [8] and asynchronous dynamics may be be considered.

Generally speaking, there are two goals in studying the dynamics of the predator-prey systems. One is the explanation of the possible oscillations in the temporal evolution of the densities of prey and predators, as well as of the correlations between them. This is the classic problem in the field [9,7,10]. The second problem is the derivation and discussion of the long time, steady states at which a predator-prey system finally arrives.

In this contribution we shall discuss two differents models of population dynamics, illustrating the importance of considering spatially extended systems.

The first one [11] (Model A), is a lattice model of evolution of a predator - prey system. Both species in order to survive must eat, at least once during a given time period. The prey eats the grass growing in unlimited quantities on most of the cells of the lattice. On the remaining cells the prey find cover against predators, but not food. We compare four active strategies - the prey move in the direction of grassy cells, they move away from the predators, the predators move in the direction of the prey and both species move in the direction of their respective food resources. These strategies are compared with random wandering of both species. We show that the character of the asymptotic stationary state depends on the initial concentrations of predators and prey, density of the shelter for the prey and on the strategy adopted. The predators are more vulnerable to the proper choice of the strategy. The strategies which lead in most cases to the extinction of predators and prey are those in which the predators are searching the prey.

The second model [12] (Model B) is an individual-based model of a population which lives in a changing environment. The individuals forming the population are subject to mutations and selection pressure. Using Monte Carlo simulations we have shown that, depending on the values of the mutation rate and selection, the population may reach either an active phase (it will survive) or an absorbing phase (it will become extinct). We have determined that the transition between the two states (phases) is continuous. We have shown that when the selection is weaker the population lives in all available space, while if the selection is stronger, it will move to the regions where the living conditions are better, avoiding those with more difficult conditions. The dependence of the mean time to extinction on the rate of mutations has been determined and discussed.

In the following sections, we shall define the models and discuss their properties.

2 Model A

2.1 Definition

We consider square lattice $L \times L$, with periodic boundary conditions. On each cell of the lattice there may be a predator (e.g. a wolf), a prey (e.g. a rabbit), both of them, or it may be empty. It is forbidden for two or more animals of the same species to occupy the same cell at the same time. The animals are moving from one cell to another following the rules specific for a given strategy. A part of the cells serves as a shelter for rabbits against wolves. On such cells (*rabbit-holes*) a wolf cannot eat a rabbit. The remaining cells are covered with grass which is eaten by rabbits. If a wolf and a rabbit are at a same time at the same grassy cell, the wolf eats the rabbit. The distribution of rabbit-holes is random and remains unchanged in time. The grass is always available for the rabbits on grassy cells.

Each animal has to feed at least once every k Monte Carlo steps (MCS). This is realized in our model by attributing to each animal a counter containing k "food rations". The counter is increased by one after each "meal" (eating grass by a rabbit or eating rabbit by a wolf) and decreased after completing one MCS. For simplicity we assume the same value of k for predators and prey. An animal which did not eat for k MCS dies.

If an animal has at least one nearest neighbor of the same species, the pair produce, at most, M offspring. M is the physiological birth rate. In order to breed, apart from finding a partner in its neighborhood, the animal must be strong enough, i.e. it must have at least k_{min} food rations. Each offspring receives at birth k_{of} food rations. The offspring are located, using the blind ant rule (i.e. only one search is made for an empty cell for each progeny), within the Moore neighborhood [8], which in the case of the square lattice contains 8 cells. If the density is high, there is room for only a fraction of M, hence less progeny is born. This procedure takes care of the unrealistic unlimited growth of a population found in the classic Lotka-Volterra models and replaces, in a natural way, the phenomenological Verhulst factor.

If an animal moves into a cell with a rabbit-hole - nothing happens. A rabbit cannot neither breed nor eat, but it cannot also be eaten by a wolf.

We shall investigate the role of the five strategies (four active and one passive) for the animals.

1. Passive strategy. It is simply no strategy at all, denoted in the following as *NO*. This is the case when the animals move randomly through the lattice.
2. Food for wolves (*W*). A wolf moves into a cell on which there is a rabbit but not a rabbit-hole. If there is more than one such cell, or there is none, the choice is made at random. The rabbits are moving randomly.
3. Food for rabbits (*R*). A rabbit moves into a cell on which there is grass (with or without a wolf). As above, if there is more than one such site the choice is random. Wolves have no strategy.
4. Food for both species (*RW*). Wolves and rabbits search for their food. Simultaneous application of the rules 2 and 3.

5. Escape for rabbits (*ESC*). A rabbit prefers a move into a cell without a wolf. It does not matter whether there is a rabbit-hole or not. Wolves move randomly.

The model has the following parameters - linear size of the lattice L, initial densities of rabbits $C_r(0)$, wolves $C_w(0)$, concentration of rabbit-holes C_h, maximum period of time (MCS) k an animal may survive without eating, the amount of food, k_{min}, needed to be fit for breeding, the amount of food, k_{of}, received at birth, and finally M, the physiological birth rate. The role of the most of the parameters has been estimated in [6], hence we have decided on fixing the parameters at the following values: $L = 50$, $k = 6$, $k_{min} = 2$, $k_{of} = 2$ and $M = 4$. We shall be interested in the role played by initial concentrations of predators and prey and the density of shelter.

Typically we have averaged over 100 independent runs, although as a check in some cases we averaged over 300 runs.

2.2 Properties of Model A

A detailed description of the properties of model A can be found in [11]. Accordingly, we shall only give here a summary of the results.

We have found three asymptotic states for the evolution - regardless of the adopted strategy. In the first one the predator and prey coexist, and their concentrations fluctuate mildly around some stationary values. In the second one the predators vanished and the concentration of prey reached a value higher than in the coexisting case, but below the carrying capacity of the system. In the third one the predators ate away the prey and the final state is the empty one.

The effect of the different strategies, was measured as a probability that a population will reach a stationary state.

It turns out that there are two, nearly equivalent, strategies which give a better chance to survive for the wolves population. Both concern the behavior of the rabbits only - escape (ESC) and food for rabbits (R). When the initial concentration of predators is relatively high ($C_w(0) = 0.4$), the remaining strategies give no chance for survival for the wolves. Lowering the initial number of predators permits them to survive also under other strategies, although they are always inferior to ESC and R. Only with very low initial concentration of predators all strategies are equivalent and all lead always to the coexisting state. With initial concentration of predators at the level $C_w(0) = 0.5$, neither strategy is good and the wolves face extinction almost always.

The fate of the prey (rabbits) population is more bright. At $C_w(0) = 0.2$ all strategies guarantee survival in the coexisting state; at $C_w(0) = 0.3$ there are three good strategies (NO, ESC, R) and two bad ones (W, RW). With increasing the initial concentration of wolves, the group of three splits into better ones (NO, ESC) and a worse (R). For still higher $C_w(0)$, the preference of the two becomes more and more pronounced. This is understandable since when there are many predators it is likely that on a grassy cell, for which the rabbit is looking, there is also a wolf. Our rabbits are not intelligent and do not distinguish between a deadly cell (grass and wolf) and a friendly one (grass only).

We have also investigated the role played by the concentration of cover - the rabbit-holes. For the wolves the rabbit-holes are no good at all. Therefore if there is relatively less cover ($C_h = 0.1$) in the system, the three strategies ESC, NO and R are the best for the wolves and give them approximately the same chance of survival. However when the number of rabbit-holes increases, it is better for the wolves if the rabbits follow the ESC or R strategy. The chances of survival are then improved by a factor of at least two with respect to the NO strategy, which directs often the rabbits into cover, where they are inaccessible to the wolves. The gain is even larger if the initial concentration of predators is lower. For higher concentration of shelters, $C_h = 0.3$, the wolves have no chance at all to survive under the passive NO strategy. Their survival chances raise to 27% under the ESC strategy and to 41% under the R one. This is the result of rabbits looking primarily for food, thus avoiding the rabbit-holes. For the same reason the optimal strategy for the rabbits changes, from ESC and R at lower concentrations of shelter, into NO at higher concentrations. The NO strategy is the best for the rabbits (it gives them 100% survival chances) for the density of shelters $C_h = 0.3$. Then the wolves become extinct pretty fast and the rabbit population lives without danger.

We may conclude that the comparison of the four active and one passive strategies (escape for rabbits, food for rabbits, food for wolves, food for wolves and rabbits and finally no strategy at all) shows that

- for both species the worse strategies are W and RW,
- the advantage of one or another of the remaining three strategies depends on the concentration of rabbit-holes, initial concentrations of wolves and rabbits,
- when the number of predators is low, the choice of the strategy is not important. In the extreme conditions the survival of the species may strongly depend on the strategy adopted by the prey,
- wolves are more vulnerable to the choice of the strategy.

The above findings are in good, qualitative of course, agreement with the known biological data and models (see e.g [13]).

3 Model B

3.1 Definition

We consider a population composed at time t of $N(t)$ individuals. The individuals live in a habitat which has the form of a square lattice of dimensions L_x along the X-axis and L_y along the Y-axis. The lattice is divided into three equal parts (denoted regions I, II and III afterwards), perpendicular to the X-axis. A site could be either empty or occupied by only one individual.

An individual is characterized by its position on the lattice, **j**, and by its genotype, g_j, which is a double string (of 32 positions) of 0's and 1's. It could have the form

$$g_{\mathbf{j}} = \begin{cases} 0010110... \\ 0101101... \end{cases}$$

We have decided on using one pair of homologous chromosomes since increasing their number with the same number of loci in the genome and the same recombination rate per chromosome would increase the overall, already very high, recombination rate. It could not produce any effect of linkeage. It should be noted that in the model one recombination per generation equals one recombination per 32 loci.
From the genotype the phenotype, $f_{\mathbf{j}}$, of the individual is constructed, as a single string (vector) of the same length, according to the following rule. For each position (locus) of the genotype the product of the two values is taken and the result is put at the corresponding place of the phenotype. In biological terms it means that 0 corresponds to a dominant, and 1 to a recessive allele:

$$(00),\ (01),\ (10) \to 0, \qquad (11) \to 1. \tag{1}$$

Hence the phenotype corresponding to the genotype presented above would be

$$\mathbf{f_j} = \{0000100...\}. \tag{2}$$

Moreover each individual is also characterized by its age, $a_{\mathbf{j}}$, which at birth is set equal to 1 and is increased during the simulations (see below). This feature makes the model more realistic by diminishing the survival probability of an individual with its age and also by eliminating perfectly adapted individuals, who otherwise could live forever and eventually dominate the population.

In constructing our model we aimed at maximum simplicity, within the class describing population dynamics with recombination and phenotype following from the genotype. Suggestions for this type of models were given in [14]. However in almost all biological papers describing evolution, individuals are characterized simply by their phenotypes.

3.2 Algorithm of Model B

The algorithm governing the dynamics of our system has the following structure:

1. Chose an individual, for example on a site **j**,
2. Calculate its survival probability $p_{\mathbf{j}}$, according to the rule

$$p_{\mathbf{j}} = \exp(-s \cdot a_{\mathbf{j}}/z_{\mathbf{j}}), \tag{3}$$

where s is a parameter characterizing the selection pressure and $z_{\mathbf{j}}$ is the fitness of the **j** individual, defined as the agreement between the optimal phenotype ("climate") **F** and the individual's phenotype

$$z_{\mathbf{j}} = \frac{1}{32} \sum_{i=1}^{32} \left(1 - (f_{\mathbf{j}}^i - F^i)^2\right). \tag{4}$$

Hence $z_{\mathbf{j}} \in [0,1]$. Survival probability defined by (3) increases with the individual's fitness but goes down with increasing age and the selection pressure. Since the exact form of the survival probability is not known, exponential or gaussian dependence on the selection is quite often used by biologists (see e.g. [15]). It should be noted that the results do not depend quantitatively on the form of the survival probability.

The notion of the optimal phenotype is known since a long time [14] and it represents here a vector of 32 components taking values equal either 0 or 1.

3. A random number $r_{\mathbf{j}}$ is chosen. If $r_{\mathbf{j}} \leq p$ the individual lives, otherwise it is removed from the system and the procedure goes back to 1.
4. If the individual's age is greater than 1, it may breed. To do so it must go through the steps listed below
5. Find an empty place in the von Neumann neighborhood [8]. On a square lattice used here, it means in the four principal directions - N, E, S, W. Only one such search is made (blind ant rule).
6. After moving to the new place, find a partner in the von Neumann neighborhood of the new position.
7. The two produce, at most 4 offspring. Each of them receives its own genotype constructed through recombination and formation of two gametes from each parent (genetic shuffling). The process may be described as follows. The two strings of a parent's genotype are cut at a random place. The resulting four pieces are glued across, forming two gametes. The same is done for the second parent. From each parent one gamete is chosen randomly, thus forming two strings for the genotype of the offspring.
8. On each site of the chosen gametes a harmful mutation may take place with probability p_{mut}. This means that if at a given site (locus in biological terms) the value agrees with the value at the corresponding site of the optimal phenotype, it may be changed with that probability. If the value is different from that of the optimal phenotype, no action is taken. Each progeny receives its genotype independently. Only harmful mutations are considered because their frequency is much larger than the one for positive mutations.
9. From the genotype a phenotype is produced along the lines described above.
10. For each offspring a search for an empty place is made in the Moore neighborhood [8] of the first parent. On a square lattice it means 8 sites - $N, NE, E, SE, S, SW, W, NW$. If the place is found, the offspring is put there, if not, it lost its chance and is not born.
11. After choosing as many first parents as there are individuals in the population at that time, one Monte Carlo step (MCS) has been made and the age of all individuals is increased by one.

At the beginning of simulations the optimal phenotype is the same in all three regions and is the most favorable, i.e. it is a string of zeros only (zero is a dominant allele !). The initial population is random, i.e. it has random genotypes and the spatial distribution is random too, but restricted to the first region only, with a given initial concentration. We have checked that if that concentration is above a certain level (about 0.05) its precise value is unimportant, since the population very quickly (about 50 MCS) reaches the same value of about 0.6 and then it follows its evolution, depending on the values of the parameters. Regions II and III are empty. The values of the selection pressure, s, and mutation rate, p_{mut}, are fixed. We let the population adapt to the existing conditions and then we change the optimal phenotype (climate) as follows. At the first change, after 200 MCS after the adaptation, the values at two, randomly chosen, sites in the optimal phenotype in the first region are changed from 0 to 1. Since a zero in the individual's phenotype can be obtained from three combinations of alleles in the genotype, i.e., (00), (01) and (10), a change from 0 to 1 in the optimal phenotype means that there are less sites in the genotypes that satisfy this condition. One may therefore say that the living conditions turned to worse. At the second change, which also occurred after 200 MCS, the optimal phenotype in the first region is again changed in the same way, i.e. at two, randomly chosen sites, zeros were switched for ones. Now however, also the optimal phenotype in the second region has been changed. We simply adopted here the previous (changed) optimum phenotype of the region I. One may see it as a gradual moving, from the left, of a colder climate. Finally, once again after 200 MCS the climate is changed. Two more sites in the optimal phenotype in the region I are switched from 0 to 1, thus it contains now six sites with value one and twenty six sites with value 0. In region II the optimal phenotype contains four sites with value 1, and in the region III it contains just two sites with values equal 1. Since the average age of an individual in our simulations oscillates around 3.5 MCS, the changes are made after about 100 generations. We consider here a system with overlapping generations.

Although in our model there is no force pushing the individuals into the regions II and III, it is not a simple colonization of empty territories. An individual adapted to the climate in the first region finds, after the conditions changed, a better chance of survival if it follows the climate, hence if it moves into the II or III regions. In such a way the region I becomes depopulated. A simple invasion process occurs in our model when the climatic changes are negligible. Then the density in the first region remains approximately the same as in the other two.

For simulations we have used a lattice of the following dimensions - $L_x = 150$, hence each region was 50 lattice sites long, $L_y = 1000$. We have checked that increasing the value of L_x did not change the results, while increasing L_y lead to very long simulations with only slightly better statistics. Typically we let a population evolve for 5×10^4 to 10^5 MCS.

3.3 Properties of Model B

A detailed description of the properties of model B can be found in [12]. Accordingly, we shall only give here a summary of the results.

The time evolution of the densities in the three regions, for a given value of the selection pressure and mutation rate, has been investigated.

There are three possible outcomes for a fixed value of $s = 0.065$. When the mutation rate is low ($p_{mut} = 0.001$), the population survives and colonizes

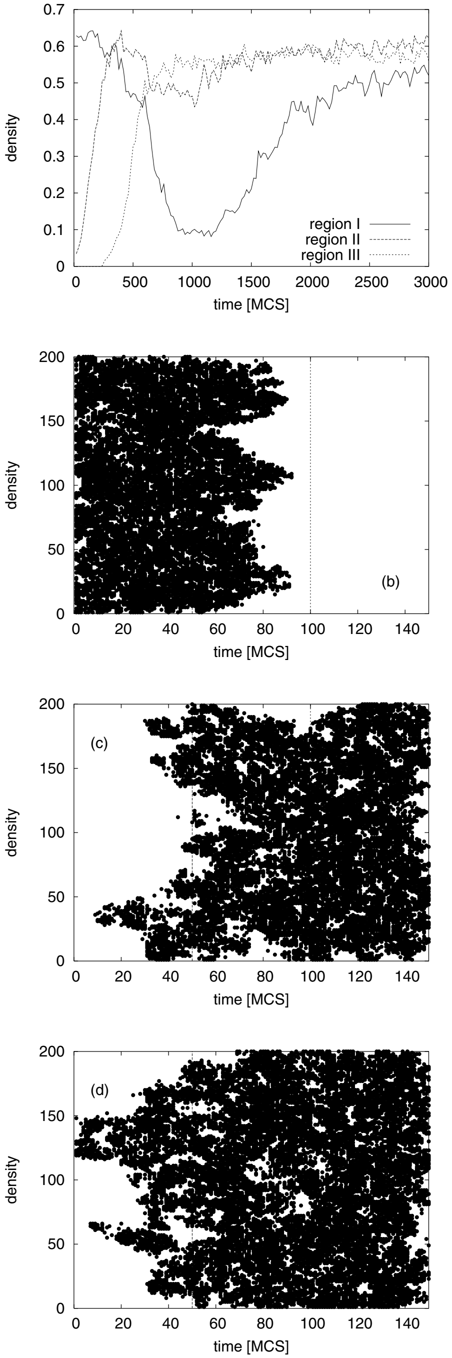


Fig. 1. Density dependence on time (a), Spatial distribution of the population after 160 MCS (b), 1000 MCS (c), 15000 MCS (d).

all three regions with similar densities. When the mutation rate increases to $p_{mut} = 0.003$, although the population also survives, the mutation rate is so high, that life in the I region became difficult. Hence the population moves to more friendly regions II and III, with only very few individuals remaining in the region I. Both cases correspond to the active state. When $p_{mut} = 0.004$, the population becomes, after some time, extinct. This is the absorbing state. The fitness of the individuals in the three regions raises very quickly to about 0.8 and remains at that level.

The cases in which the mutation rate is kept constant and the selection pressure is varied have also been studied. If the living conditions in the region I are not too difficult, the population will recolonize it and the density grows there to about 0.5. This is the case illustrated in Figures 1, where the density dependence on time and some "snapshots", showing the spatial location of the population, are presented.

We have performed a series of simulations keeping either the selection pressure or mutation rate fixed and changing the other parameter. We have found that in each case, for some values of the varying parameter, the population arrived at an active state, while for others it became extinct. Far away from the critical point, i.e. the lowest value of the parameter which was changed (selection or mutation rate) and for which the populations died, average density stabilized pretty soon. The closer the system was to the critical point, the more the density fluctuated. The fact is well known to biologists as a demographic stochasticity, affecting populations of small sizes [16].

Examining the final, quasi-stationary, states obtained from the time dependencies of the density of the populations, we can construct a phase diagram in the *(selection, mutation rate)* plane, presented in Figure 2. The diagram shows a critical line separating the alive (active) and extinct (absorbing) phases. The line is not symmetric, since the role played by the two parameters is not equal. Selection acts in the same way in all three regions, while mutation rate affects individuals differently. First of all, if a mutation (there are only harmful mutations) occurred for an individual in the region II, then if it immigrates either to the region I or III, the mutation may turn beneficial, because of different optimal phenotypes there. Next, a mutation changes only one allele in the genotype, and for having an effect on the phenotype, also the second allele at the same locus (site on the genotype) has to be 'wrong'. In biological terms it means that if a harmful mutation is to affect the phenotype it has to occur in a heterozygote (different alleles in the same site) at that locus. Third, there is a genetic shuffling, meaning that progeny receives genotypes of the parents changed in the process of recombination.

In many biological studies, see e.g. [17,16,18], the average time to extinction is considered. In most cases the parameters are the reproduction rate (number of offspring) and the size of the carrying capacity of the habitat (maximum number of individuals who can live there at the same time). Here we have found

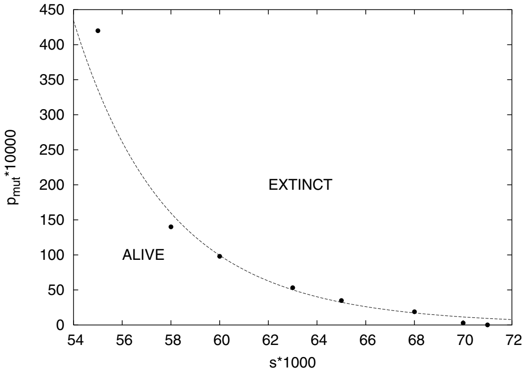


Fig. 2. Phase diagram in the selection-mutation plane.

that the dependence of the average time of extinction on the mutation rate can be fitted into a dependence $< t_{ext} >\approx p_{mut}^{\alpha}.$ where α varies with the selection pressure s. Since however the data covers rather narrow range, it would be difficult to claim that we have here a true power law.

Our results [12] can be interpreted as follows. For strong selection the population has really no chance to develop and changing the mutation rate has no marked effect. Any individual which is not well fit has a very low chance of survival. Therefore the time to extinction increases with growing mutation rate in more or less the same way. If the selection is weaker, then it matters whether the mutation rate is low or high. If it is low, the population has enough time to grow, since at the beginning there are few mutations. Afterwards the mutations accumulate and the population dies. This behavior is reflected as the upper part of the t_{ext} versus p_{mut} curve. If the mutation rate is high, then the population very quickly acquires enough mutations and behaves similarly as in the case of strong selection, and this is the lower part of the t_{ext} versus p_{mut} dependence. Similar kind of dependence of the average extinction time on the selection and mutation could be deduced from the data presented in Ref. [16]. It should be noted that the role of the selection pressure is realized in our model via the survival probability of an individual, and since it depends explicitly on its fitness, it it impossible to describe the model in terms of global variables.
Because in order to produce offspring an individual has to move, find a partner and find the place for the progeny, there is no need to introduce outside restrictions on the population growth, like the Verhulst factor, see e.g. Ref.[19].

Summarizing, we have presented a microscopic model describing the behavior of a population under selection pressure and mutational load living in a changing environment. The population is confronted with a choice - either adapt to new conditions or follow the climate and colonize new territories. We have found that if the selection pressure is not high, the population will be spread with similar densities all over the three regions. With increasing selection pressure and rate of mutations the first region will become depleted and the regions II and III will contain most of the population, with a slight preference for the region II, since there will be immigrants from regions I and III. Finally, if the selection pressure is very strong, the population will die out. In the model the role of the selection pressure seems to be more important because of its global character. We have constructed the phase diagram of the final states of the evolving population - an active state (living population) or an absorbing state (extinct population).

4 Conclusions

The two models treated above have shown that several important aspects of the dynamics of population cannot be understood at the level of simple mean-field like equations, overlooking the local fluctuations present in any spatially extended systems. However, simple stochastic models using discrete variables evolving with sequential or parallel dynamics can be very powerful in investigating the properties of these complexes systems.

Acknowledgement. This work was partially supported by the Swiss National Science Foundation.

References

1. A.J. Lotka, Proc. Natl. Acad. Sci. U.S.A. **6**, 410 (1920).
2. V. Volterra, “Leçons sur la théorie mathématique de la lutte pour la vie”, (Gauthier-Villars, Paris 1931).
3. N.S. Goel, S.C. Maitra and E.W. Montroll, “Nonlinear models of interacting populations”, (Academic Press, New-York, 1971); J. Hofbauer, K. Sigmund, “Evolutionary Games and Population Dynamics”, (Cambridge Univ. Press, 1998).
4. P. Rohani, R.M. May and M.P. Hassell, J. Theor. Biol. **181**, 97 (1996)
5. T. Antal and M. Droz, Phys. Rev **6.E**, 1111 (2000).
6. M. Droz and A. Pękalski, Phys. Rev **63E**, 051909 (2001).
7. A.T. Bradshaw, L.L. Moseley, Physica A **261** 107 (1998).
8. B. Chopard and M. Droz *Cellular Automata Modeling of Physical Systems*, Cambridge University Press (1998).
9. A. Lipowski and D. Lipowska, Physica **A 276**, 456 (2000).
10. H. Taitelbaum, Z. Koza, Y. Yanir and G.H. Weiss, Physica **A 266**, 280 (1999)
11. M. Droz and A. Pękalski, Physica **A298**, 545 (2001).
12. M. Droz and A. Pękalski, Phys. Rev **E**, to appear (May 2002),
13. M. Begon, M. Mortimer and D.J. Thompson *Population Ecology*, Blackwell Science Ltd, Oxford 1996.
14. A. Fraser and D. Burnell, Computer Models in Genetics, Mc Graw-Hill, New York 1970.
15. R. Bürger and M. Lynch, Evolution, **49**, 151 (1995).
16. M. Lynch, J. Connery and R. Bürger, Evolution, **49**, 1067 (1995).
17. R. Lande, Evolution, **48**, 1460 (1994).
18. R. Gomulkiewicz and R.D. Holt, Evolution, **49**, 201 (1995).
19. J.D. Murray, Mathematical Biology, Springer, Berlin 1993.

Simulation of Vegetable Populations Dynamics Based on Cellular Automata

Stefania Bandini and Giulio Pavesi

Dept. of Computer Science, Systems, and Communication
University of Milan–Bicocca
Milan, Italy
{bandini,pavesi}@disco.unimib.it

Abstract. Modeling the dynamics of vegetable populations is an extremely challenging problem. The evolution of a vegetable population, that is, of all the weeds, plants and trees that grow in a given area, is mainly influenced by the resources available on the territory (i.e. sunlight, water, substances present in the soil), and how the single individuals compete for them. Traditional models for this case study are continuous and based on differential equations. However, most of the data needed to provide reliable parameters for these models are usually scarce and difficult to obtain. The model we present is instead based on two–dimensional Cellular Automata, whose cells, arranged on a square grid, represent portions of a given area. Some resources are present on the area, divided among the cells. A cell can host a tree, represented in the model by a set of parameters defining its species, its size (that is, the size of its parts such as limbs, trunk, and roots), the amount of each resource it needs to survive, to grow, and/or reproduce itself (that is, produce fruits). The model has been applied to the simulation of populations consisting of robiniae (black locust), oak, and pine trees on the foothills of the italian alps, with encouraging results reproducing real conditions.

1 Introduction

Modeling the dynamics of vegetable populations, that is, of all the weeds, plants and trees living in a given area, is an extremely challenging problem [1]. The main difficulty lies in the acquisition of data for the definition of the parameters of the models, that must cover very long time periods, especially in the case of perennial plants. Such data must include the resources available on the territory, and those needed by plants to sprout, survive, grow, and reproduce themselves. In fact, the evolution of a vegetable population is mainly influenced by the resources available (i.e. sunlight, water, substances present in the soil), and how the different individuals compete for them. Traditional models are continuous and based on differential equations [2,3,4], and usually model the evolution of the system with global parameters such as the total number of trees and their overall biomass. More recently, Cellular Automata have been introduced to study this

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 202–209, 2002.

problem [5,6], but usually their application was limited to the evolution of single infesting species.

In this paper, we present a discrete model based on two–dimensional Cellular Automata, that allows to model and simulate the evolution of heterogeneous vegetable populations composed by different perennial species as in real woods and forests. The evolution of the system is thus modeled in a bottom–up fashion, that is, is the result of the interactions among single individuals and their competition for the resources available on the territory [7].

The cells of the CA represent portions of a given area. Each cell contains some resources, and if conditions are favorable, can host a tree. A tree is represented in the model by a set of parameters, defining its species, its size (that is, the size of its parts such as limbs, trunk, and roots), the amount of each resource it needs to survive, to grow, and reproduce itself (that is, produce fruits). A single tree has been "decomposed" in different parts in order to reproduce the effect of environmental influences. In fact, the environment and the resources available determine how the overall biomass of the tree is divided among the different parts composing it [8].

When a tree produces fruits, some seeds are scattered in the neighboring cells. A seedling can sprout in a cell when the latter contains a seed, no other tree, and a sufficient amount of each resource. In this case, a tree is born, and the state of the cell now comprises also all the parameters defining the tree present in it (otherwise set to zero, if no tree is present). Then, the cell has also to contain enough resources to sustain the growth of the plant. The quantity of resources needed varies according to the species of the tree and its size. When a tree starts growing, its increasing mass begins to need a larger amount of resources, that can also be taken from the neighbors of the cell where it is located. Thus, the sprouting or the growth of other trees in its proximity is negatively influenced, that is, the tree starts competing for resources with the others. Whenever a tree cannot find enough resources to survive, it dies.

The model has been applied to the simulation of populations consisting of robiniae (black locust), oak, and pine trees on the foothills of the italian alps, with encouraging results reproducing real conditions.

2 The Cellular Automaton

We now give a more formal description of the model. The state of each cell of the CA is defined by a flag denoting whether or not it contains a tree, the amount of each resource present in the cell, and a set of variables defining the features of the tree (possibly) growing in it. The update rule of the automaton mainly depends on the presence of a tree in a cell. In case a tree is present, part of the resources present in it (and in the neighboring ones, if the tree is large enough) are absorbed by the tree. Every cell also produces at each update step a given amount of each resource (that in any case cannot exceed a maximum threshold value). The production of resources in the cells is determined by a set of global parameters, and reproduces environmental factors such as rain, presence of animals in the area, and so on. The effect of the presence of a tree in a cell on the neighboring ones has been modeled by making resources flow from richer cells to poorer ones (that possess less resources since a part of them is consumed by the tree). The resources we explicitly included in the model are water, light, nitrogen, and potassium. Both von Neumann and Moore neighborhoods have been considered in the simulations.

The CA can be thus defined as:

$$\mathbf{CA} = \langle R, N, Q, f, I \rangle$$

where:

1. $R = \{(i,j) | 1 \leq i \leq N, 1 \leq j \leq M\}$ is a two–dimensional $N \times M$ lattice;
2. H is the neighborhood, that can be either the von Neumann or Moore neighborhood;
3. Q is the finite set of cell state values;
4. $f : Q \times Q^{|H|} \rightarrow Q$ is the state transition function;
5. $I : R \rightarrow Q$ is the initialization function.

2.1 The Cells

Each cell of the automaton reproduces a square portion of terrain with a side ranging from three to five meters. As mentioned before, each cell contains some resources, and can host a tree. Thus, the possible states of a cell must define:

1. The type of terrain the cell reproduces;
2. The resources present in the cell;
3. The amount of resources the cell produces at each update step, and the maximum amount of resources it can contain, according to its type;
4. Whether a tree is present in the cell, or not;
5. If a tree is present:
   a) the size of the tree;
   b) the amount of each resource it needs at each update step to survive and grow;
   c) the amount of each resource stored by the tree at previous update steps;
6. Seeds scattered by trees living in the area.

If we assume that k types of resource and l different tree species are present in the area, the finite set of states Q can be defined as follows:

$$Q = \{\mathbf{R}, \mathbf{M}, \mathbf{P}, T, \mathbf{Z}_T, \mathbf{N}_T, \mathbf{U}_T^G, \mathbf{U}_T^S, \mathbf{R}_T, \mathbf{M}_T, \mathbf{G}_T, \mathbf{S}\}$$

where:

1. $\mathbf{R} = \{r_1, \ldots, r_k\}$ is a vector defining the amount of each resource present in the cell;
2. $\mathbf{M} = \{m_1, \ldots, m_k\}$ is the maximum amount of each resource that can be contained by the cell;

3. $\mathbf{P} = \{p_1, \ldots, p_k\}$ is the amount of each resource produced by the cell at each update step;
4. T is a flag indicating whether a tree is present in the cell or not;
5. $\mathbf{Z}_T = \{z_T^r, z_T^t, z_T^l, z_T^f\}$ is a vector defining the size of the different parts of the tree (in our model, roots, trunk, leaves, and fruits);
6. $\mathbf{N}_T = \{n_T[1], \ldots, n_T[k]\}$ are the amounts of each resource the tree takes from the cell at each update step (depending on its size);
7. $\mathbf{U}_T^G = \{u_T^G[1], \ldots, u_T^G[k]\}$ is the vector defining the amount of each resource needed at each update step by the tree to *grow*;
8. $\mathbf{U}_T^S = \{u_T^S[1], \ldots, u_T^S[k]\}$ is a vector defining the minimum amount of each resource the tree needs at each update step to *survive*; for each i, $1 \leq i \leq k$, we have $u_T^S[i] < u_T^G[i] < n_T[i]$;
9. $\mathbf{R}_T = \{r_T[1], \ldots, r_T[k]\}$ is the amount of each resource stored by the tree at previous update steps;
10. $\mathbf{M}_T$ is a vector of threshold values for different parameters defining the tree, such as maximum size, maximum age, minimum age for reproduction, maximum number of seeds produced for each mass unity of fruits, and so on. These threshold values can be fixed or picked at random in a given range when a new tree is created;
11. $\mathbf{G}_T = \{g_T^r, g_T^t, g_T^l, g_T^f\}$ is a vector defining the *growth rate* of each of the parts of the tree, that is, how much each part of the tree grows when enough resources are available;
12. $\mathbf{S} = \{s_1, \ldots, s_l\}$ is a vector defining the number of seeds present in the cell for each of the l species growing in the territory.

2.2 The Update Rule

At each update step of the automaton, the tree present in each cell (if any) takes the resources it needs from the cell itself and uses them to survive, grow (if enough resources are available), and produce seeds. If the resources available in the cell exceed its needs, the tree stores some resources. Conversely, if the resources available in the cell are not sufficient, the tree uses resources stored at previous update steps. If also the resources stored are not sufficient for the tree to survive, the tree dies. A newborn plant can sprout in a vacant cell, if the latter contains a seed of its species, and again enough resources.

Moreover, we defined the update rule in order to reproduce the increasing influence that a growing tree can have on neighboring cells. For example, its roots can extend beyond the limits of the cell hosting it. Or, when it gets taller, it shades an increasingly wider area around itself, thus having a negative influence on the growth of other trees in its neighborhood. We modeled the impact of a tree in a given position on its neighborhood by making resources flow from richer cells to poorer ones. In other words, a cell hosting a large tree is poor on resources, since the tree at each update step takes most (or all) of them. If the neighboring cells are vacant, their resources remain unused, and thus are richer than the one hosting the tree. Therefore, if we let resources flow from richer cells to poorer neighbors, the effect is that in practice a large tree starts to collect resources also from neighboring cells. Notice that if we include sunlight among the resources contained by a cell, we can model in this way also the “shade” effect. Seeds are also introduced in the model as a resource that moves from cell to cell. Thus, a tree can scatter its seeds in the neighboring cells.

Now, let $C(i,j)$ be the cell located at position (i,j) in the lattice. With $\mathbf{R}(i,j)$ we will denote the resource vector of cell $C(i,j)$, with $\mathbf{M}(i,j)$ the maximum resource values, and so on. The transition function can be divided in four sub–steps, defined as follows.

Tree sustenance. If a tree is present in cell $C(i,j)$, it takes from it a given quantity (defined by $\mathbf{N}_T(i,j)$) of each available resource $\mathbf{R}(i,j)$. If, for some resource i, the amount available $r_i(i,j)$ is lower than the corresponding value in $\mathbf{N}_T(i,j)$, then the tree takes the whole quantity $r_i(i,j)$. The amount of resources taken depends on the size of the tree $\mathbf{Z}_T(i,j)$. Then, if enough resources (those taken at this step, plus the resources stored at previous steps), are available, as defined by vector $\mathbf{U}_T^G(i,j)$, the tree grows, that is, each part grows according to the growth rate vector $\mathbf{G}_T(i,j)$ associated with the tree. Else, the resources might be just sufficient for the tree to survive (vector $\mathbf{U}_T^S(i,j)$). In this case, the tree parameters are left unchanged. In both cases, the tree “burns” an amount of each resource, as defined by vector $\mathbf{U}_T^G(i,j)$ or $\mathbf{U}_T^S(i,j)$. All the unused resources collected at this step are stored and added to vector $\mathbf{R}_T(i,j)$. Otherwise, if the overall amount (stored plus collected) of at least one resource is under the “survival threshold” of the tree, the latter dies. The tree also dies when it reaches its maximum age defined in vector $\mathbf{M}_T(i,j)$.

Tree reproduction. We have two cases to consider: a tree is present in the cell, or the cell is empty. In the former case, the tree may produce some seeds (if it is old enough, and according to the size of its fruits $z_T^f(i,j)$), that are used to update the corresponding variable in the seed vector $\mathbf{S}(i,j)$. Also, new trees cannot sprout from the seeds contained in the cell if a tree is already present. Instead, the cell can be vacant and contain some seeds. If the resources present in the cell are sufficient (quantities defined as global parameters for each tree species) a new tree is born. If seeds from different species are present in the cell, the winning species is chosen at random, with probability proportional to the number of its seeds.

Resource production. In the third sub–step, each cell produces a given amount of resources, according to its production vector $\mathbf{P}(i,j)$. In any case, the amount of each resource contained in the cell cannot exceed the corresponding maximum value defined by vector $\mathbf{M}(i,j)$.

Resource flow. In this step, resources are balanced among neighboring cells, in order to let resources flow from richer to poorer cells. Let $r_h(i,j)$ be the amount of resource h contained by cell $C(i,j)$, and assume that we are using the von Neumann neighborhood. $r'_h(i,j)$, the amount of resource i after this update sub–step, is defined as:

$$r'_h(i,j) = \frac{r_h(i,j) + \frac{r_h(i+1,j)+r_h(i-1,j)+r_h(i,j+1)+r_h(i,j-1)}{4}}{2}$$

In other words, we can see each cell as divided in four parts, each one containing the amount $r_h(i,j)/4$ of resource h, and corresponding to one of the neighbors. The amount of resource h contained in each part is balanced with the corresponding part of the neighbors. In case we adopt the Moore neighborhood, we can imagine the cells as split into eight portions. The effect is that, if cell $C(i,j)$ is richer on resource h than its neighbors, part of its content will spill into them. As mentioned before, $r'_h(i,j)$ cannot exceed the corresponding maximum value defined for the cell $(m_h(i,j))$. In this case, we set $r'_h(i,j) = m_h(i,j)$. The same rule is applied to each of the components of the seeds vector $\mathbf{S}(i,j)$.

2.3 The Initial Configuration

The initial configuration of the CA can be defined by the user, by setting appropriate resource parameters for each cell. Also, some trees might be already present on the territory, with all the variables defining them set. Or, the territory might be empty, with some seeds scattered here and there (clearly, if no tree and no seeds are present, nothing happens when the automaton is started).

3 The User Interface

The model has been implemented in C++ under Windows NT. The user interface permits to define explicitly:

1. Different types of cell, according to the maximum amount of resources the cell can contain and the amount of resources it produces, in order to resemble the features of different types of terrain. Moreover, it also possible to reproduce rivers (by setting high values for water content and production, and zero maximum content values for other resources), rocky terrain (with very low values for all the resources), roads (zero values for all the resources), and so on;
2. Different tree species according to the amount of resources needed at each update step, to the growth rate of the different parts, that is, how resources are distributed among the different parts, the quantity of seeds produced;
3. The initial configuration of the automaton.

The interface shows step–by–step the evolution of the system, giving a straightforward image of the growth of the trees. Moreover, it is possible to show the distribution of the resources on the territory at each step, and the overall results of the simulation (total number of trees, trees for each species, total biomass, biomass of each single species and single tree, and so on), as shown in Fig. 1, 2, and 3.

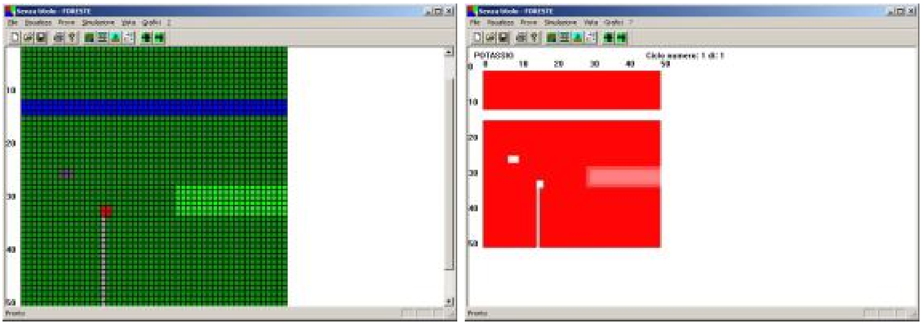


Fig. 1. An initial configuration of the automaton (left). The dark strip represents a river. The image to the right shows the initial distribution of potassium in the cells. Darker areas are richer on potassium.

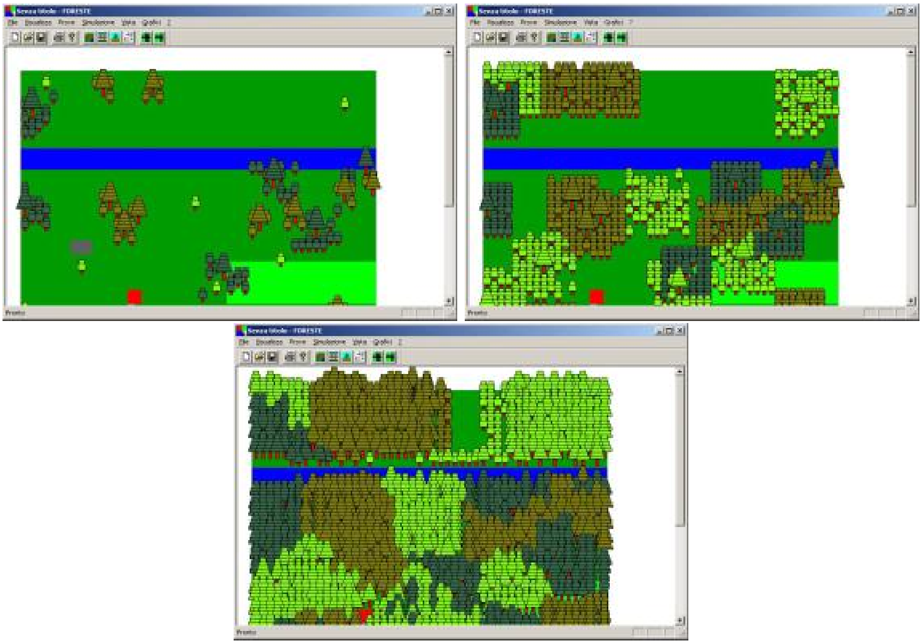

Senza titolo - FORESTE
File Visualizza Prove Simulazione Vista Grafici ?
Pronto
Senza titolo - FORESTE
File Visualizza Prove Simulazione Vista Grafici ?
Pronto
Senza titolo - FORESTE
File Visualizza Prove Simulazione Vista Grafici ?
Pronto

Fig. 2. Example of the user interface, showing three different stages of the evolution of a vegetable population composed by black locusts, oaks, and pine trees, starting from the initial configuration of Fig 1.

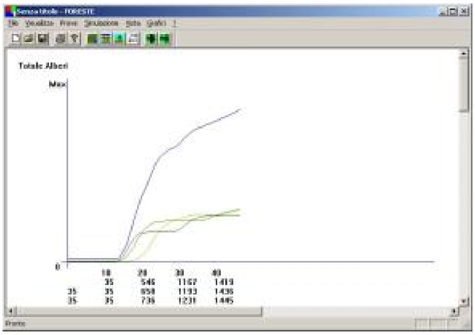


Fig. 3. The user interface showing the total number of trees, and the number of trees of each species present in the area for the example shown in Fig. 2.

4 Conclusions

In this paper we presented a model based on CA for the simulation of the dynamics of vegetable populations. Our simulations, reproducing populations of robiniae, oaks and pine trees living on the foothills of the italian alps have shown results qualitatively similar to real case studies. We believe that the flexibility of the model, that allows the user to define explicitly different types of terrain and tree species can provide an useful tool, not only for the simulation of real case studies, but also for a better comprehension of the main factors influencing the dynamics of vegetable populations.

References

1. M.G. Barbour, J.H. Burk, W.D. Pitts, F.S. Gilliam, M.W. Schwartz, Terrestrial Plant Ecology, Benjamin Cummings, 1998.
2. J.L. Uso–Domenech, Y. Villacampa–Esteve, G. Stübing–Martinez, T. Karjalainen, M.P. Ramo. MARIOLA: A Model for Calculating the Response of Mediterranean Bush Ecosystem to Climatic Variations. *Ecological Modelling*, **80**(1995), pp. 113–129.
3. Q. Zeng, X. Zeng, An Analytical Dynamic Model of Grass Field Ecosystem with Two Variables. *Ecological Modelling*, **85**(1996), pp. 187–196.
4. Q. Zeng, X. Zeng, Two Variables Dynamic Model of Grass Field Ecosystem with Seasonal Variation. *Ecological Modelling*, **85**(1996), pp. 197–202.
5. H. Baltzer, W.P. Braun, W. Köhler. Cellular Automata Models for Vegetation Dynamics. *Ecological Modelling*, **107**(1998), pp. 113–125.
6. R.L. Colasanti, J.P. Grime. Resource dynamics and vegetation processes: a deterministic model using two–dimensional cellular automata. *Functional Ecology*, **7**(1993), pp. 169–176.
7. D. Tilman. Competition and biodiversity in spatially structured habitats. *Ecology*, **75**(1994), pp. 2–16.
8. D. Tilman. Dynamics and Structures of Plant Communities. Princeton University Press, 1988.

A Fish Migration Model

Birgitt Schönfisch[1] and Michael Kinder[2]

[1] Biomathematics, University of Tübingen, Auf der Morgenstelle 10,
72076 Tübingen, Germany,
birgitt.schoenfisch@uni-tuebingen.de,
http://www.uni-tuebingen.de/uni/bcm/schoenfe.html
[2] University of applied sciences Koblenz,
RheinAhrCampus, Südallee 2, 53424 Remagen, Germany
kinder@rheinahrcampus.de,
http://www.rheinahrcampus.de/fachbereiche/fb2/mut02.html

Abstract. Today most german rivers are flow regulated by dams. They form a barrier for migrating fish species like salmon and are one cause for sustainable disturbances of the ecosystem even to extinction of several fish species. Our question is how to distribute financial resources to improve fish passage systems to obtain a maximal effect. We present a concept to model up- and downstream fish migration with an individual based approach. It allows us to test different strategies of resource distribution. We give an example modelling a part of the river Moselle in Germany and discuss further extensions.

1 Introduction

Several fish species migrate during their life cycle between fresh- and seawater. Two groups are distinguished: anadromous fish hatch in freshwater, spend most of their life in the sea and return to freshwater to spawn. Common examples are salmon, smelt, shad, striped bass, and sturgeon. Catadromous species, like most eels, live in freshwater and spawn in seawater. Dams built today for shipping traffic, hydropower usage and flow regulation at most rivers all over the world hinder migration of these fish. They form a barrier for upstream migration, although often for example fish ladders are installed. And even though downstream passage is usually less problematic, especially if young small fish travel, populations suffer substantial loss [2]. Consequently populations of migrating fish decrease and in the river Moselle for example several species including salmon are even extinct.

Our model is motivated by plans to increase inland navigation on river Moselle, a large and shippable stream flowing from France to the river Rhine in Germany. Beside an enlargement of weirs and sluices an improvement of fish passage systems is discussed. The interest in this river is high since at its headwaters in Luxembourg there are suitable spawn habitats for salmon (*salmo salar*). Also the water quality in rivers Rhine and Moselle would allow a return of the salmon. 180 adult salmon have been observed in the river Rhine and its tributaries Sieg und Saynbachs between 1990 and 1999 and salmons have been caught

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 210–219, 2002.

near Koblenz [4]. There are fish ladders associated with the weirs in river Moselle but up to now no adult salmon migrating upstream through the fish ladders has been observed [5]. Some few adult individuals, found in fact in river Moselle, probably passed through the sluices. An improvement of the fish passage systems to reintroduce salmon in this area is an important objective. The question arises how to invest the given financial resources yielding the best effect. Our concept is to optimise not a single fish passage system but migration on the whole river, from estuary to headwater.

On first view this task seems straightforward and even solvable analytically. However the aim is not to bring most fish to the river's source. Salmon for example needs suitable gravel beds to lay spawn. These can be found in tributary rivers and will have different capacities. To meet these demands we develop a simulation tool that can be adjusted to the specific habitat setting. It allows to test different strategies of fish passage improvement.

We will first discuss the optimisation problem. Then the basic model is introduced and a simulation example of migrating salmon through river Moselle is shown. Finally we discuss further applications and extensions of the model.

1.1 Optimisation of Fish Migration

Most models on optimisation of fish migration known from literature are quite technically oriented. They aim for example on the improvement of fish ladders for upstream migration [1] or optimisation of a certain turbine type for downstream migration (for a review see [2]). In our setting however the problem is how achieve best fish migration on the *whole* river system. Often optimisation problems investigate how to get a certain effect with minimal resources. In our case the resources are more or less fixed. Therefore it is more appropriate to ask how to invest the given resources for maximal effect. So basically we have a number of river dams and a certain amount of money. The question is how the given money should be distributed on the dams. To describe solutions a 'permeability' coefficient is given to each dam. It characterises how well fish can migrate. The value is interpreted as the probability that one fish migrates successfully over the respective dam in one try. A solution for the whole river can be formulated by a vector or tuple of such permeability coefficients.

For this optimisation problem quite general analytic solutions can be obtained [13]. Consider for example a river without tributaries and a finite number of dams. Fish migrate upstream from the estuary, i.e. we aim for most fish arriving at the river's source. Assume that effort depends at least linearly on effect, i.e. if we want to double the permeability, at least the double amount of resources have to be invested. Then it can be shown that the best strategy is to invest such that all dams have equal permeability.

In real applications however the setting will be more complex. Suitable spawning habitats for anadromous fish for example will be found to a different extent in the tributaries of the flow regulated river. Since habitat suitability of the spawning sites depends also on the density of individuals it may be advantageous for one population not to be concentrated at one optimal spawning site but distributed over several sites. This implies that the goal function will not be as simple as getting as much individuals as possible to one location. Also upstream migrating salmon for example will more and more exhausted and perhaps die before reaching suitable gravel beds for reproduction. However in at least some of these settings analytic solutions can be found.

The structure of optimal fish migration is related to problems from quite different fields, for example to optimal vaccination strategy problems [10], and results may be transferred to our question. An individual based model approach however offers a very flexible method to test different strategies and include various parameters. Also the model is simple and transparent for the users. A fish passage model enables to test different strategies of fish passage facilities extensions, however it does not solve the optimisation problem directly. One idea is to add a part that forecasts permeabilities on the base of resources invested in a certain dam – a cost-effect function has to specified for every barrier. The best way to distribute resources can be found with stochastic hill-climbing algorithms for example, the goal function being the number of successfully migrating fish.

2 Fish Passage Model

For simplicity we will first describe an *upstream* migration model for one fish species. The fragmentation of rivers by dams suggests discrete units in space. Every cell will represent an impoundment which is the segment between two successive dams. A river without tributary rivers for example can be seen as a one-dimensional grid $G \subset \mathbb{Z}$. Since we look only at upstream migration, in this simple case every cell will have only one neighbour cell, except the last cell at the river's source. Please note that cells are not their own neighbours. To describe more complex river systems, like for example the river Moselle with its tributary system, we have to include branching. Then some cells will have two neighbours or even more – although every cell is only neighbour of one other cell. This gives us a grid G of cells x with 'branches' embedded in two-dimensional space.

Every cell can take states from the set $E = 0, 1, ..., n$ where n is the total number of simulated individuals. The state of a cell is interpreted as the number of fish present. The state of the system is then a function $z : G \mapsto E$, $z : x \mapsto z(x)$.

We first explain the dynamics in an individual based formulation. To every cell a number $p_i \in [0, 1]$ is assigned. We interpret it as probability that one fish successfully migrates *into* this impoundment in one try. For upstream migration it therefore describes the permeability of the dam located downstream of the cell. At confluences, were tributary rivers join, p_i can be seen also as the preference for fish to migrate into tributary rivers versus following the main stream. At these branching points cells have more than one neighbour. Then we have to be more careful with the interpretation of p_i. Consider for example a cell with two neighbours with permeabilities p_1 and p_2. Suppose that fish try to migrate in one of the neighbours with equal probability. Then the probability that a fish successfully migrates into the first or second neighbouring cell is $p_1/2$ respectively $p_2/2$. Since in the individual based formulation we take the view of individuals, i.e. fish, we will choose a fish, locate its position and decide whether it will migrate in contrast to choosing a cell to be iterated. So the rules read:

1. Choose a fish. Let x_i be the cell where it is.
2. Choose a neighbouring cell of x_i, let this be x_j.
3. With probability p_j the fish migrates from x_i to x_j, otherwise it stays at x_i.

These rules imply asynchronous dynamics. The first rule has to be specified more detailed, i.e. we have to select a method to choose the 'next' fish. The most satisfying approximation of continuous time from theoretical point of view is to assign exponentially distributed waiting times to each fish. For many applications the easier choice with uniform distributed probability will be sufficient. If we want to formulate the first rule such that we choose a *cell* instead of choosing fish, then the probability a certain cell is selected should depend on the state of the cell. The more fish it contains, the more likely it should be selected. Otherwise fish in crowded cells would migrate slower than fish in cells with low density. In the second rule the simplest possibility is to chose the neighbour cell with equal probability from all neighbours of the cell. Certainly neighbours can also be chosen with different probabilities q_i, with $\sum_i q_i = 1$. In many cases a cell will anyway have only one neighbour.

Our model fits in the framework of dimer automata [12]. If we want to formulate this model as a cellular automaton, the grid will be G as before and also the states of the cells will be in $E = 0, 1, ..., n$. The problem is to enable migration of fish. To ensure mass conservation, i.e. a constant total fish number, we have to change the states of *two* cells. For this we have to introduce some kind of handshake between the cell the fish migrates from and the cell it migrates to. Also at confluences we have to decide to which neighbour cell the fish moves. This means that here stochasticity has to be involved. Therefore this automaton can not be formulated as a classical deterministic cellular automaton. Also the different values of p_i and the structure of the grid gives a spatial inhomogeneous model.

This model can be also formulated as a flow on a directed graph where every vertex represents the segments between two dams. The parameter p_i will then be assigned to every edge – an example is shown in Figure 1c. Such a graph is a subset of a binary tree (if we ignore river 'crossings' and circuits). This formulation shows that fish migration can be seen as a percolation problem.

So far we only described how fish migrate on a river with given dam permeabilities. Our problem however is to invest our resources and change these permeabilities such that most fish will arrive in suitable spawning habitats for example. Therefore we may not specify p_i but the amount of resources r_i used on a certain dam. To every dam (or cell) we have to give a function that gives p_i for any value of r_i. Usually this function will be nonlinear, i.e. if we want to double the permeability, the resources used at this dam have to be more than doubled. Let p_{i_0} be the actual estimated permeability of dam i. Then $p_i = p_{i_0} + f_i(r_i)$ where f_i is the cost function of dam i.

2.1 Example: Migration of (Atlantic) Salmon in the River Moselle

We give an example to illustrate our model concept. We simulate upstream migrating adult salmon coming from river Rhine and passing river Moselle for reaching spawning sites in its tributaries. The question is how to invest resources for the improvement of fish passages to enable and optimise salmon migration.

In our simplification we consider a section of river Moselle with seven dams and only two larger tributaries. This gives ten river fragments, i.e. ten cells as shown in Figure 1b. To each of these cells a probability p_i is assigned, a measure for the permeability of the dam, respectively the preference of tributary rivers. Figure 1c shows a more formal representation as a directed graph with an example setting of p_i values.

We look only at the number of fish that are able to migrate upstream to the headwaters, i.e. to the most left cell (Fig. 1c). We start with a finite number of fish at the rightmost cell, i.e. fish entering the River Moselle. The number arriving at the headwaters will increase over time and finally, if all fish travelled up, reach a constant level. It is shown in Figure 2 for two choices of p_i: upstream increasing permeability – these are the values drawn in Figure 1c – and constant permeabilities (with the same mean) on the main stream.

Figure 3 shows also simulation snapshots. With constant p_i fish arrive faster at the headwater; this recovers the analytic result that equal permeabilities lead to optimal migration in the case of a river without tributaries. In our example with increasing permeabilities finally more individuals reach the headwater. The p_i values of the three leftmost cells are about 1.0, 0.86 and 0.71 while with constant permeabilities we have $p_i = 0.57$. The branches to the tributary rivers both have $p_i = 0.2$, i.e. in the case of constant permeabilities more fish will swim into the tributary and at the river's headwater finally less fish will arrive. We see also that the variability between different realisations is a little bit larger with constant permeabilities.

These results may be obtained also analytically. In more realistic settings however, as described in the next section, this may be at least laborious.

2.2 Model Extentions

The model formulation allows to include almost arbitrary details of the fish life cycle. Through upstream migration for example some fish will die by various causes. This can be included by an additional process – once a fish is chosen it will decided with certain probability whether it will migrate or die. This probability may even depend on space, i.e. the position of the fish. It is also reasonable that fish get tired the more often they try to migrate through a given dam. We can give each fish a counter – the more times it makes a try, the smaller the probability it succeeds. As well fish become more and more exhausted the further they travel. This can be incorporated likewise giving each fish a 'fitness' value (eventually according to a certain distribution) that decreases with time - if it drops below a given threshold, then the fish will die.

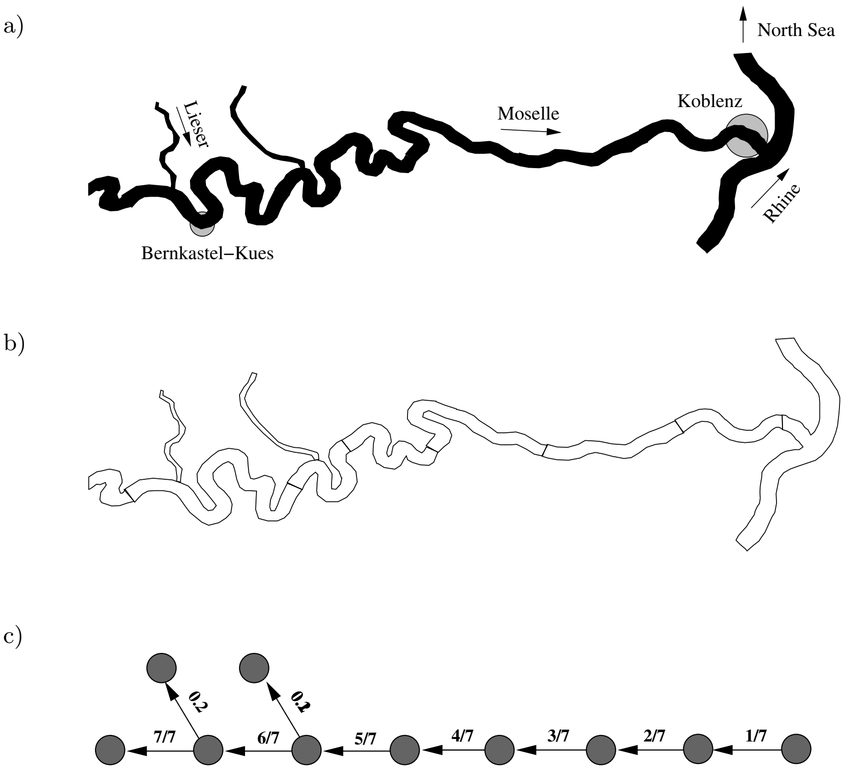


Fig. 1. River Moselle example: a) map of part of the river, b) fragmentation by dams and tributary rivers, c) representation as directed graph.

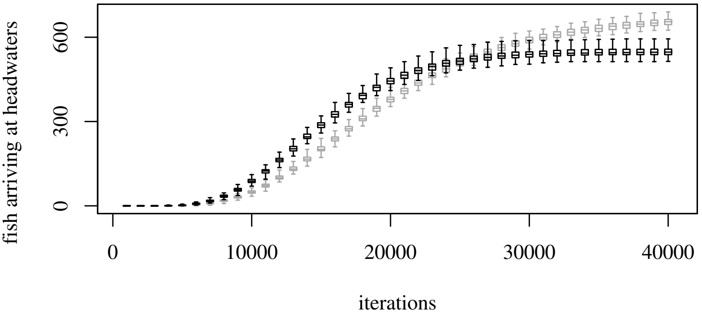


Fig. 2. Numbers of fish arriving at river's headwaters. Simulation of upstream migration of 1000 fish with increasing permeabilities (grey) and equal permeabilities on mainstream (black). Shown are box-and-whisker plots from 100 runs each.

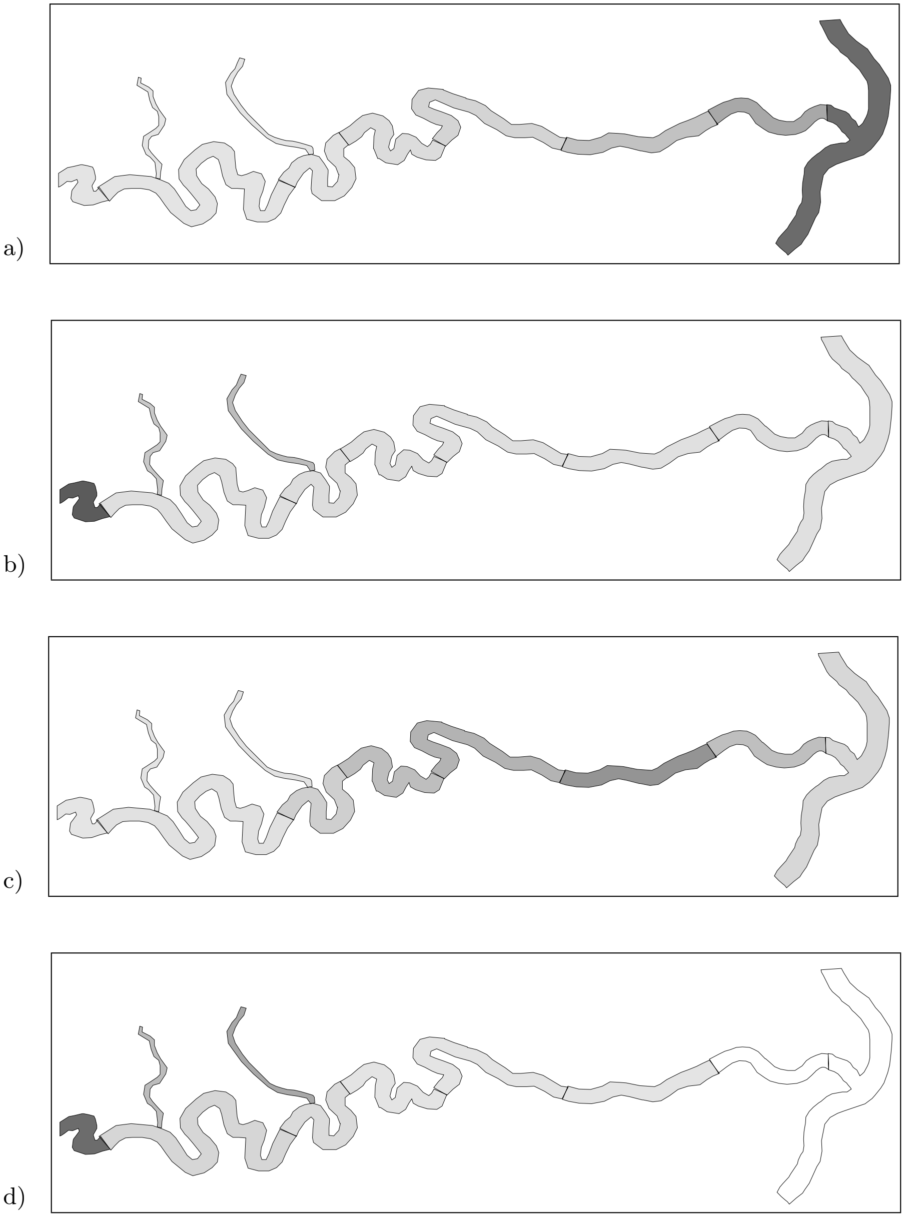


Fig. 3. River Moselle example: fish numbers on river with increasing permeabilities (a and b) after 5000 and 20000 iterations and with equal permeabilities on mainstream (c and d) also after 5000 and 20000 iterations. Fish numbers are increasing from white, grey to black.

Salmon usually returns to the site in the tributary where it was born once. But some individuals will go into other rivers. This can be incorporated by giving the simulated fish appropriate preferences with a certain variability.

So far only upstream migration was discussed. Downstream migration will be very much the same, only the graph will be different, every cell has at most one neighbour, but some cells are neighbours of two or more cells. We can connect down- and upstream migration. Since usually the permeabilities in both directions will be different, the simplest solution is to model this with two graphs connected by one cell – symbolising the estuary for example. We can follow fish from spawn places to the sea and back. Here, the effect of releasing hatched salmon fingerlings at headwaters can be observed.

During the high season of downstream migration of eel temporally variation of the regime of the turbines at electric power plants even their stop is discussed [3]. The date of migration varies from year to year, therefore power plant regime depends on observations or capture of fish. This aspect can be also an important part of the reintroduction program for atlantic salmon.

For catadromous fish the graphs will be connected via the river's source. Finally it is possible to model different fish species which compete for example.

3 Discussion

While in 1885 from the river Rhine 250.000 salmon had been caught today it is virtually extinct there. Beside water pollution the fragmentation of rivers by dams has been a relevant cause. Today the water quality is much better, but dams are still a barrier for migrating fish.

We introduce a concept for modelling migrating fish. A river fragmented by dams is represented by a grid of cells. Individuals are selected by random, their migration to a neighbour cell is successful with probability p_i. This value is specific to each cell and describes the dam's permeability. The permeability parameters of the existing dams can be estimated from fish counts: at many dams upstream migrating fish are monitored when passing through fish ladders for example. Values can also be obtained from other comparable dams or turbine types. With appropriate resources even catch and release experiments before and after a dam are possible.

Our model allows to test different strategies of fish migration improvement. It can be part of an application oriented decision support system. Results are obtained fast and last not least easy visualisation facilitates communicating them to policymakers. With the model different permeability settings can be tested. If we built in a part that computes these permeabilities on base of an cost-effect function specified for each barrier we can use a stochastic hill-climbing method for example to get an optimal strategy for distribution of resources. Also strategies obtained from analytic results with simplifying assumptions can be tested.

Space is represented discrete in our model. Every cell represents one impoundment. This implies that the time it takes for a fish travels from one end of the impoundment to the other is short in comparison to the time it needs to cross a dam. This will be appropriate in situations where fish passage is low, i.e. if dams have low fish permeabilities and certainly regarding salmon migration on the river Moselle for example. If more details in modelling are necessary, as one advantage, many habitat modelling approaches or ecomorphological assessment procedures can be easily connected to our model [7,8,9].

Generally the benefits of individual based models is that they are easy to formulate and variables and rules can be interpreted directly in biological terms. There are few discrete freshwater fish models known from literature and in their review on habitat modelling Parasiewicz and Dunbar [11] also discuss discrete models. Another example is the work of Jager et al. [6] who study effects of river fragmentation on sturgeon populations with an individual based model - they investigate also the effect of fragmentation on fish migration. However one drawback of individual based models is that often no analytic results are available and we are have to rely on simulations.

A continuous description corresponding to our approach would be a system of coupled ordinary differential equations. Such models are known for metapopulations or optimal vaccination strategies for example. Each equation would describe one impoundment. For some of these formulations analytic solutions can be obtained. And like in discrete models, if the time fish spend *in* the impoundment is important, this can be incorporated with using delay equations, or even a system of reaction-diffusion equations. Related to our model is the work of Zabel and Anderson [14]: With a partial differential equation formulation they investigate the migration of juvenile salmon through segments of the Columbia and Snake Rivers. Their model is connected with the CPiSP project and program [15] to predict downstream migration and survival of juvenile fish at the tributaries and dams of the Columbia and Snake rivers. Yet, as often, the differential equations can not be solved analytically. Solutions are approximated using numerical methods. It has been argued that then it would be better to formulate the model in an discrete, individual based view from the beginning.

An individual based migration model is very flexible and different dam improvement strategies can be tested easily and fast. Using analytical methods from optimisation theory however can lead to more general results. Here, results from established models may be translated to fish migration, even if such an interpretation is not near at hand. An example is a age-structured model for optimisation of vaccination strategies [10]: age classes correspond to impoundments, individuals to fish, investing resources in vaccinations at a certain age class is equivalent to improving the permeability of a dam.

The formulation and investigation of different models – optimisation models and individual based models – will show us different aspects of fish migration. It ensures that the results obtained are not artefacts of the modelling techniques but solely derived from the specified assumptions. Therefore our work will proceed in two directions: adapting the model to the situation of river Moselle situation in very detail with estimating permeability and cost-effect function for each dam. On the other hand we are working on analytic results.

References

1. Clay, C.H.: Design of fishways and other fish facilities. Lewis Publishers (1995)
2. Countant, C.C.: Fish behaviour in relation to passage through hydropower turbines: A review. Trans. of the Am. Fish. Soc. **129** (2000) 351–380
3. Eckmann, R., Jürgensen, L: Personal communication (2001/2002)
4. ICPR: Salmon 2000 — The Rhine a salmon river again? Report of the International Commission for the Protection of the Rhine (ICPR) (1999), see also http://www.iksr.org/22ge.htm
5. ICPR: personal communication with the International Commission for the Protection of the Rhine (ICPR) (2002)
6. Jager, H.I., Chandler, J.A., Lepla, K.B., van Winkle, W.: A theoretical study of river fragmentation by dams and its effects on white sturgeon populations. Envir. Biol. of Fishes **60** (2001) 437-361.
7. Kinder, M., Schönfisch, B.: Individual based models for integrating population dynamics in physical habitat modelling of large rivers, 2nd Symposium for European Freshwater Sciences, Toulouse (2001)
8. Kinder, M., Schönfisch, B, von Landwüst, C., Wurzel, M.: Individual based modelling in the ecology of large rivers. in preparation
9. Kern, K., Fleischhacker,T., Sommer, M., Kinder, M.: Ecomorphological Survey of Large Rivers - Monitoring and Assessment of Physical Habitat Conditions and its relevance to biodiversity. Large Rivers **13** (2002) 1-28
10. Müller, J.: Optimal vaccination strategies for age structured populations. SIAM J. Appl. Math. **59** (1999) 222-241.
11. Parasiewicz, P., Dunbar, M.J.: Physical habitat modelling for fish – a developing approach. Large rivers **12** (2001) 239–268, Archiv für Hydrobiologie Supplement. 135/2-4:1-30.
12. Schönfisch, B., Hadeler, K.P.: Dimer Automata and Cellular Automata. Physica D **94** (1996) 188-204
13. Schönfisch, B., Kinder, M.: Optimisation of fish migration. in preparation.
14. Zabel, R.W., Anderson, J.J.: A Model of the Travel Time of Migrating Juvenile Salmon, with an Application to Snake River Spring Chinook Salmon. North American Journal of Fisheries Management **17** (1997) 93-100.
15. see http://www.cqs.washington.edu/crisp/crisp.html

A Parallel Cellular Ant Colony Algorithm for Clustering and Sorting

Paul Albuquerque[1,2] and Alexandre Dupuis[1]

[1] Computer Science Department, University of Geneva, 1211 Geneva 4, Switzerland
[2] LII, Ecole d'ingénieurs de Genève, HES-SO, 1202 Geneva, Switzerland

Abstract. Some ant species are known to gather and sort corpses in an auto-organized way. In this contribution, we propose a new cellular ant colony algorithm for sorting and clustering. This algorithm mimics the behavior of real ants. The cellular automata nature of our algorithm implies a straightforward parallelization. The rule consists of a pick-up, a deposition and a diffusion. Our probabilistic pick-up rule is based on some spatial neighborhood information. We observe that probabilistic pick-up yields compact clusters and also speeds up the clustering process. In the long run, a single cluster emerges. Moreover, in the presence of several corpse species, our algorithm sorts the corpses into distinct clusters. Thus our model reproduces realistic results, but however we do not observe any collective effect.

1 Introduction

Some ant species are capable of clustering corpses or sorting their brood without any of the working ants having a global perception of the overall task [1,2]. It thus seems that the building of a single cluster emerges as the result of a collaboration between the ants. Ant colony algorithms try to mimic this behavior.

The famous forerunner model proposed by Deneubourg [3] for clustering and sorting is the following. Ants move randomly in some arena containing corpses, which they can load and deposit later. They are endowed with some form of intelligence: (i) picking up a corpse occurs preferentially in a small cluster of dead bodies and (ii) the probability to deposit the corpse in a cluster increases with its size. With both these hypotheses, it is clear that smaller clusters should disappear to the benefit of larger ones. The ants estimate a cluster size from a memory of their past steps, and act probabilistically according to this estimate.

In this contribution, we define a new cellular automaton (CA) which models an ant colony. Our CA model performs clustering and sorting similarly to Deneubourg's model. However, we use a deterministic deposition rule and a probabilistic pick-up rule based on the ants' local spatial perception of their environment. The motion of the ants is related to a CA diffusion rule [5]. To our knowledge, the effect of a parallel synchronous updating scheme has not yet been explored in much detail. Here, we only address the parallelization and not the performance issue, even though we give some timing indications.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 220–230, 2002.

In some sense, it is more realistic to use a CA, since the problem of clustering in ant colonies is parallel by nature. Indeed, the ants perform their task simultaneously and may actually compete for a given corpse. Furthermore, ant colony algorithms achieve a global task in a distributed way without the need of central control. Hence, they are local by essence. Current applications include robust distributed sorting and graph partitioning [1,2].

In this work, we show that the clustering process resulting from our CA, produces a single cluster in the long run. We observe the effect on the density of this cluster of probabilistic versus deterministic pick-up. If there are different types of corpses, our algorithm sorts them into distinct clusters, one per type. In addition to observing clustering and sorting from a qualitative point of view, we study the number of clusters and their density as a function of time. It seems that both functions are invariant under a time rescaling by a factor equal to the number of ants. This indicates that the clustering process does not result from a collaboration between the ants.

2 The Deneubourg Model

The model of Deneubourg [3] defines a discrete dynamics on a square lattice, whose sites are occupied by ants and corpses. The motion of the ants is sequential in the sense that ants are visited according to an initial, arbitrary numbering. At each time step, ants move randomly from their present position to one of the four neighboring sites (left, right, above or below), provided it is not occupied by another ant. The displacement of the corpses is totally dependent on the ants.

An ant encountering a corpse on its path, picks it up with probability p_p unless it is already loaded with a corpse. Then the ant moves again randomly and, at each step, may deposit the body with probability p_d on the new site it stands on. The probabilities p_p and p_d are computed as

$$p_p = \left(\frac{k_1}{k_1 + f}\right)^2 \qquad p_d = \left(\frac{f}{k_2 + f}\right)^2 \tag{1}$$

where k_1, k_2 are parameters and f is the proportion of corpses encountered by the ant during its last T steps. This quantity f represents the ant's memory.

If f is small compared to k_1 (i.e. the ant is in the vicinity of a small cluster), p_p is close to one and it is very likely that the ant will take the corpse. If f is large with respect to k_2 (i.e. the ant has come across many dead bodies), then so is the probability p_d of depositing the corpse. Clearly, this rule biases the dynamics towards emptying small existing clusters and augmenting larger ones, thus favoring the formation of a single cluster of corpses.

Note that the current occupation state of the cell on which the ant lies is taken into account only after the pick-up or drop down decision is made.

In this implementation, ants are authorized to move onto sites occupied by corpses. This allows the memory update as well as the picking up of corpses to take place. Likewise, ants drop the corpse they carry on the site they stand on.

3 A Cellular Automata Model

In Deneubourg's model, ants are updated *sequentially* in a given order specified by some arbitrary (but fixed) numbering of the ants. Here, we propose a *parallel* synchronous updating scheme. We thus introduce a cellular automaton (CA) model whose dynamics is based on the diffusion rule of [5].

In our CA model, ants have a local perception of their environment: they can access any information located in their direct neighborhood. Deneubourg's ants have an intelligence based on a memory (temporal information), while our ants behave according to the knowledge of their surroundings (spatial information).

The dynamics consists in alternating a pick-up, deposition and marking rule with a random walk rule. The CA approach implies dealing with any potential conflict which arises when several ants simultaneously enter a site.

3.1 The Underlying Space

The cells of our CA are arranged on a regular two-dimensional square lattice. The local topology around a cell is given by the Moore neighborhood, which consists of the 8-neighboring cells (above, below, right and left, plus diagonals). The state of a cell consists of: (1) ants moving in any of the 8 directions, and (2) a corpse. In our model all ants are identical, but there can be several types of corpses. A cell can hold at most eight ants, each one traveling in a different direction, and at most one corpse. If we think of a cell as being connected to its neighbors by links, then the ants can be thought as traveling along these links.

3.2 The Evolution Rule

The CA rules are described in table 3.1. Let us list some differences with Deneubourg's model. First, our parallel updating scheme implies some kind of arbitration to handle conflicts. Second, the pick-up probability is now based on local spatial information. Third, the deposition phase, which is deterministic, requires a supplementary stage: the marking of loaded ants. Fourth, the ants move according to a CA diffusion rule.

1. **Pick-up** : If there is a corpse, a free ant picks it up with a probability based on neighborhood information (see below). The many-ant situation is resolved by selecting the ant on the direction of smallest index.
2. **Marking** : A loaded ant is marked with a flag (hasToDepose), if it encounters a corpse of the same species it is carrying.
3. **Deposition** : A marked loaded ant deposits its corpse if the site is unoccupied. The many-ant situation is resolved by selecting the ant on the direction of smallest index.
4. **Diffusion** : Ants are rotated with angle $k\pi/4$ ($k \in [0;7]$) with probability r_k.
5. **Propagation** : Ants moving in direction k are moved to their neighbor in direction k.

Algorithm 3.1: The cellular ant colony algorithm.

Handling Corpses. As in Deneubourg's implementation of his model, our ants are allowed to enter the site of a body. Many ants are allowed onto the same site as long as they are traveling along different directions. The probability of picking up a corpse is given by

$$\frac{1}{1+\exp(\alpha_p \cdot (f-\beta_p))} \tag{2}$$

where f is the proportion of bodies in the neighborhood and α_p, β_p parameters. The marking is a kind of memory, which tells the loaded ant to deposit as soon as possible after being marked

Random Walk. The ant motion rule considered by Deneubourg also differs from ours. Each step of our random walk consists in successively applying a diffusion followed by a propagation. The ants move one cell at a time. We use the diffusion rule of [5]. At each site, an angle $k\pi/4$ ($k \in [0;7]$) is chosen with probability r_k. All ants on that site are then rotated by this angle. Since r_0 is the zero angle probability, ants entering a given site are deflected from their trajectory with probability $1-r_0$. An ant moving in direction k is transfered to the corresponding neighbor. This step does obviously not alter its direction of motion. The propagation achieves the update of the cells. In the current implementation, the mean free path is given by $r_0/(1-r_0)$ independently for each ant [5]. The quantity r_0 determines the diffusion coefficient of the random walk. For example, $r_0 = 0.98$ yields a mean free path of 50. This yields a diffusion coefficient much larger than in Deneubourg's original implementation. At each time step, the ants move to the nearest cell along their direction of motion and check for the possibility to pick up, deposit a corpse or mark themselves. The ant motion is decoupled from the handling of the corpses.

3.3 Parallelization

A common way for speeding up a computation is to use more than one processor to achieve the given job. This procedure is called a parallelization. It implies decomposing a task into several sub-tasks which are equitably assigned to the available processors.

In the present context, the lattice is partitioned into several domains, one per processor. An extra layer, called the boundary, is actually added to each domain. It contains the cell-neighborhood information that is not local to the processor. At each time step, boundaries are exchanged between processors managing adjacent domains, so as to maintain the cell-neighborhood information up to date. Note that the partitioning of the lattice is optimized in order to minimize communications. These are overlapped with the propagation phase to save time.

Our CA algorithm was implemented using PELABS (Parallel Environment for LAttice Based Simulations) [6]. PELABS is a library developed at the University of Geneva, which offers many functionalities for designing parallel CAs.

4 Simulation Results

In this section, we comment on some simulations. We shall first compare the formation of a unique cluster for a single type of corpses, in the cases of probabilistic and deterministic pick-up. We then observe how our CA, in the presence of several types of corpses, sorts them into distinct clusters. Finally, we study the number of clusters and the density of the largest cluster as a function of time. This provides a measurement of the efficiency of the clustering process.

We consider a 290×200 lattice, with periodic boundary conditions, containing $N_0 = 1500$ corpses. This is fixed for all simulations. During a given run, N_0 and the number of ants remain constant (no birth or death). Throughout the dynamics, clusters of corpses appear, gathered as a result of the pick-up and deposition activity of the moving ants. The mean-free path of an ant is set to 50 via the probability $r_0 = 0.98$ of not being deflected at a site.

Simulations were run on a farm of PCs using eight 1 GHz Pentium nodes under Linux. As our code is not optimized, we only give an idea of the simulation time. Typically, $10'000$ iterations with a 1000 ants took 320[s] on 8 nodes and 1600[s] on a single one. Note that these timings depends little on the number of ants.

4.1 Clustering

In figure 1, we show three stages of the clustering process with a 1000 ants in a space without obstacles. Ants and corpses are initially randomly distributed. The pick-up probability parameters are $\alpha_p = 10, \beta_p = 0.35$. Deterministic pick-up ($\alpha_p = \infty, \beta_p > 1$) means a corpse is loaded with probability 1. For probabilistic pick-up, after about 3.6×10^5 iterations (approximately 190 minutes), a single compact cluster emerges. At the same stage, in the deterministic case, there are two loosely bound clusters scattered with empty sites. This results from the ants being allowed to pick-up indiscriminately from any cluster. Similar sparse clusters were already observed in [4]. In both cases, the final cluster is not static, since the ants can still withdraw corpses from it and re-deposit them elsewhere about its perimeter. However, in the deterministic case, the final cluster keeps on assembling and disassembling, while in the probabilistic case, it is almost inactive. The bias between pick-up and deposition, related to the probability given by (2), is responsible for cluster compactness. In Deneubourg's model, this bias results from the intelligence of the ants (their memory) which locally favors the growth of large clusters at the expense of smaller ones.

Let us now define what we mean by cluster. This definition depends on the local topology. A cluster is a set of corpses. Two corpses with overlapping neighborhoods belong to the same cluster. The area covered by a cluster is defined as the number of sites occupied by its corpses and their neighborhoods. The density of a cluster is then the number of corpses in it divided by its area. From our simulations, we observe that $\rho \approx 0.88$. Note that the density of corpses in clusters is mainly constant after the start-up stage. The area covered by a cluster always includes sites without corpses on its border. Hence, our density

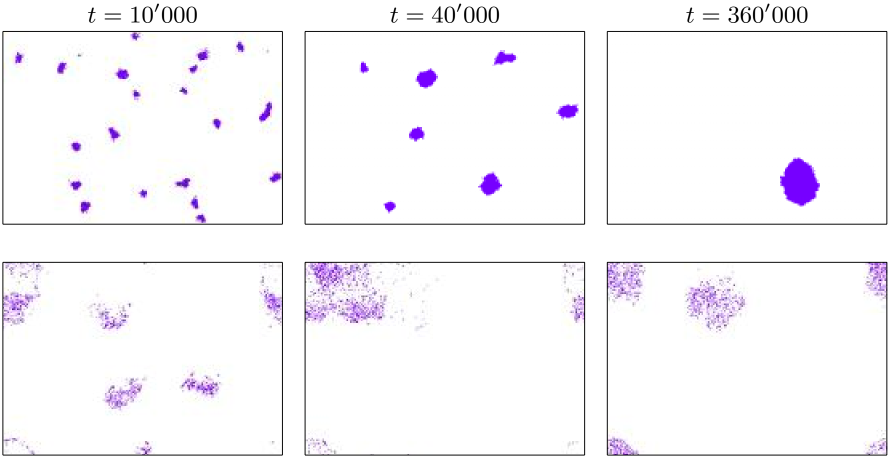


Fig. 1. Snapshots at various iteration times. The domain size is 290 × 200. There are 1500 corpses and 1000 ants. Results reported on the upper and on the lower row are obtained using probabilistic, respectively deterministic, pick-up.

measurement never yields the value 1 and its maximum value depends on N_0. The maximum density is approximately $1/(1+2(\pi/N_0)^{1/2}) \approx 0.92$ which corresponds to the case of a disk.

Recall that in our CA model, ants have no idea of the size of the cluster from which they remove a corpse or to which they add one. Hence on average in the deterministic case, each cluster is fed and depleted at the same rate, because ants pick up and deposit bodies symmetrically. However, statistical fluctuations may cause any cluster to vanish, with a probability inverse to its size. Therefore, the dynamics is biased towards forming larger clusters until only one remains. Probabilistic pick-up favors the depletion of non-compact clusters, and therefore the growth of compact ones. The clustering process appears to be accelerated towards forming few large compact clusters. However, these big size compact clusters are very robust. Hence in the end, the algorithm spends a lot of time to form a single cluster from a few similar size compact clusters.

Real ants are known to preferentially cluster bodies against obstacles. We thus decided to observe the effect of obstacles in the ant environment. In our model, we had to adapt the motion rule: an ant colliding into an obstacle simply reverses its direction. The presence of an obstacle in the simulation arena biases further the pick-up rule and breaks the spatial symmetry of the ant motion. In particular, a cluster located against an obstacle is difficult to erode from that side. Thus, once a cluster builds up against an obstacle, due to statistical fluctuations, it is likely to stay there forever. However, if during the primary stage of the clustering process, no cluster is formed against the obstacle, then the final cluster ends up elsewhere. This occurred in several of our simulations.

Note that we had to adapt the pick-up probabilities on the sites adjacent to an obstacle. Neighbors belonging to an obstacle, are counted as if holding a corpse. However, no modification was needed as concerns deposition.

4.2 Sorting

In the presence of many types of corpses, our algorithm sorts the corpses into different clusters, eventually forming one cluster per type. There is no rule that forbids corpses of different type from being deposited next to each other. Only the marking discriminates between types of corpses.

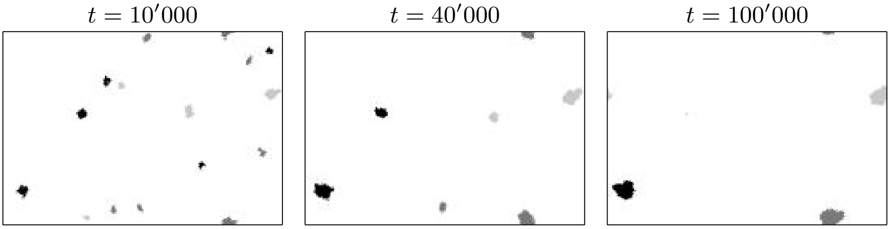


Fig. 2. Snapshots at various iteration times. The domain size is 290 × 200. There are 1500 = 3 × 500 corpses of three types, and 1500 ants. Probabilistic pick-up is used.

Snapshots of a simulation run with three types of corpses $N_0 = 3 \times 500$ and 1500 ants are shown in figure 2. Probabilistic pick-up was used with the same parameters as above. Ants carrying a corpse of a given type are blind to other types. The dynamics looks like three different clustering algorithms evolving simultaneously in the same arena with one third of the corpses and ants.

4.3 Clustering Dynamics

In this section, we measure the efficiency of the clustering process as well as cluster compactness. To this end, we measure how fast the number $N(t)$ of clusters goes to 1 and we monitor the density of the largest cluster. In section 4.1, we gave a definition of a cluster and how to compute its area and density.

We first compare in figure 3, the density of the largest cluster for probabilistic and deterministic pick-up. We observe in both cases a rapid increase of density during the first few thousand iterations. Throughout this first stage, small clusters appear. Then, in the probabilistic case, the density slowly increases from around 0.75 to approximately 0.85, while in the deterministic case, it fluctuates around 0.3. These fluctuations are due to the brittleness of the clusters. The main observation is that added intelligence yields more compact clusters.

Next, we look at the formation speed as a function of the number M of ants. We ask the following question: is the spontaneous emergence of a single cluster due to some elaborate collaboration between the ants?

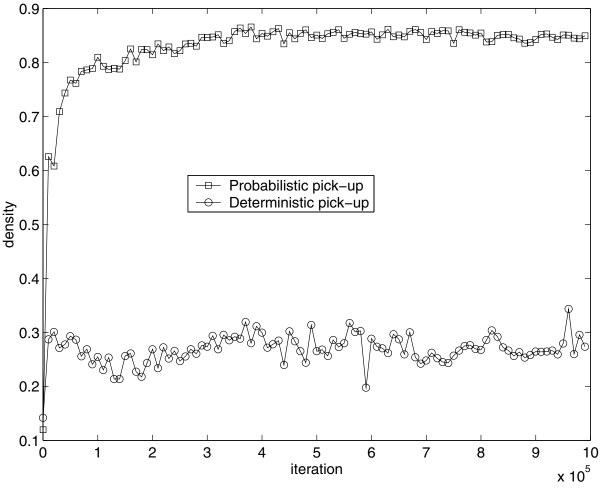


Fig. 3. Time evolution of the density of the largest cluster. Squares (resp. circles) correspond to the model with probabilistic (resp. deterministic) pick-up.

Figure 4 shows, for probabilistic pick-up, the number $N(t)$ of clusters as a function of time (in Log-Log scale) and the density of the largest cluster for different numbers M of ants. Here M takes the values 100, 200, 400, 800 and 1600. At $t = 0$, $N(t)$ is of the order of N_0. After between $400'000$ and $2'000'000$ iterations, depending on M, there remains only one large cluster, hence $N(t) = 1$. With respect to performance, we wish to point out that the CPU-time of an iteration depends little on M. Indeed, most of the simulation time is devoted to the domain traversal, which occurs at each update of the CA. Therefore, there should be an optimal M, depending on the other fixed parameters, which produces the best performances.

In figure 4, $N(t)$ appears as a stair function. The increasing length steps indicate a slowing down of the clustering process. Indeed, large compact clusters are more robust, requiring thus on average more time to vanish.

After an initial start-up of a few thousand iterations in which small clusters appear, $N(t)$ enters a power-law regime. Here, the clustering activity has reached its normal level and $N(t)$ gradually decreases towards 1. The power-law exponent is approximately $-3/4$. It is interesting to compare this exponent with the exponents $-1/2$ of a diffusion process and -1 of a global pick up and deposit anywhere algorithm. An exponent of 3 was measured in [4] for Deneubourg's model, but however for a relatively short intermediate power-law regime. The final regime leading from a dozen of clusters to a single one, was very noisy and rather slow. In our CA model, the power-law regime survives until the end. In the deterministic case, the exponent is $-1/2$. Therefore, adding in some intelligence, via probabilistic pick-up, shows that our algorithm actually does better

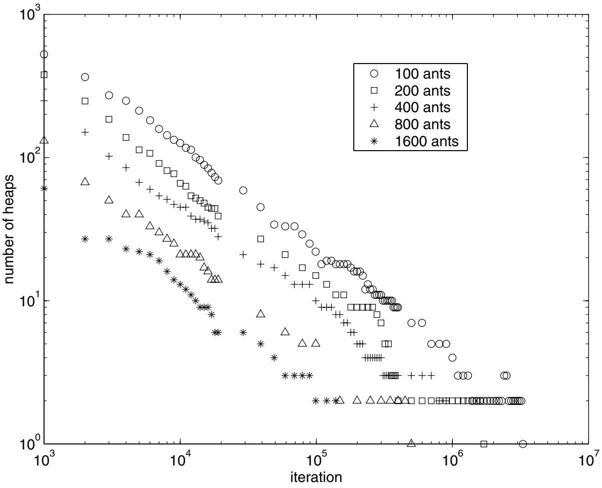


Fig. 4. Time evolution of the numbers of clusters.The domain size is 290 × 200 and there are 1500 corpses. The algorithm uses probabilistic pick-up.

than a simple diffusion process. We also want to emphasize that our CA model, as opposed to [4], is not plagued by noise at the end of the clustering process.

Finally, we remark that the curves of figure 4 can be approximately superimposed by linearly rescaling time by a factor M. This result indicates that a single ant can also produce the single cluster alone. Adding more ants just improves linearly the formation speed. This is the most basic form of cooperation: more workers do more work in the same amount of time. The global task is thus equally divided among the ants which perform their job independently. There is no collective effect requiring, for example, a minimal number of workers to achieve the global task. Similarly, we also observed that the density $d(t)$ of the largest cluster follows some kind of universal behavior under time rescaling. This can be summarized by the formulae

$$N(t) = C \cdot (Mt)^{-0.75} \qquad d(t) = 0.92 \cdot \big(1 - \exp(-\nu(Mt)^{\gamma})\big) \tag{3}$$

where C, ν, γ are constants depending on the details of the model, 0.75 the power-law exponent and 0.92 the maximum cluster density. As a matter-of-fact, these results are not really surprising, since the ants in our model have no added-in intelligence. However, it was not completely obvious that some ants would not undo the job done by others, thus yielding a factor smaller than M for the formation speed.

5 Conclusion and Further Work

We proposed a new parallel CA ant colony algorithm which performs clustering and sorting of corpses, initially randomly distributed in a two dimensional arena.

We compared, for the clustering process, a probabilistic with a deterministic pick-up rule. Our CA naturally produced a single cluster which was loosely bound, resp. compact, in the deterministic, resp. probabilistic case. A comparison of the density of both largest clusters provided a quantitative measurement of this fact. Compactness results from the pick-up rule being biased towards depleting sparse clusters. The underlying process in the deterministic case is just to take a corpse from one cluster and drop it into another at a rate specified by the ants' random walk. This rate is improved with probabilistic pick-up. Our pick-up rule differs from Deneubourg's as it is based on the ants' spatial perception of their environment rather than on their memory. However, our deposition rule relies in a certain sense on a very simple memory. In the presence of different types of corpses, our algorithm sorted the corpses into distinct clusters, one per type. For probabilistic pick-up, we then considered the dynamics of clustering from a more analytical point of view. We observed that the number of clusters follows a power-law with exponent greater than that of the underlying diffusion process, while the largest cluster density converges exponentially to the maximum value.

It appears that the curves for the number of clusters and the density exhibit a universal behavior as regards the number M of ants. Indeed, rescaling time by a factor M yields the same results. This hints strongly towards asserting that our clustering process does not stem from some kind of swarm intelligence. Nevertheless, these results were to be expected, since the ants in our model have no added-in intelligence. However, it was not completely obvious that some ants would not undo the job done by others. The important fact is to identify clearly what level of intelligence produces what result. From that point of view, our model is a base model over which one could add one or many layers of intelligence and observe the outcome. Indeed, it would be interesting to exhibit some kind of intelligence hierarchy.

Concerning performance, the CPU-time of an iteration depends little on M. Thus, it is more advantageous to use more ants. However, there is an optimal M, depending on the other fixed parameters, which produces the best performances. Furthermore, a CA is naturally parallelizable. Hence, we ran our simulations on a farm of PCs and thus benefited from a speedup.

We conclude by mentioning some directions for future work. The present paper does not include a systematic study of the probabilistic pick-up parameters in relation to the clustering process. The influence of the neighborhood on these probabilities is also of interest. Our CA model does not include a termination criterion. This should imply some kind of learning, which would allow the ants to adapt their pick-up probability during the simulation. To add in more intelligence, we could also endow our ants with a memory or have them deposit pheromones along their trail. Hence, the ants would react to their environment on a larger scale. The idea of combining spatial and temporal perception of the environment, as is sketched in this work, also seems interesting. Finally, the main purpose still remains to unearth collaborative effects between artificial ants.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York (1999).
2. Bonabeau, E., et al. Special issue on stigmergy. *Artificial Life*, **5**(2) (1999).
3. Deneubourg, J.-L., et al. The dynamics of collective sorting: robot-like ant and ant-like robot. In J.A. Meyer and S.W. Wilson, eds., *Proc. 1st European Conf. on Simulation of Adaptative Behavior: From Animal to Animats*, MIT Press, Cambridge (1991) 356–365.
4. Martin, M., Chopard, B., Albuquerque, P. *A Minimal Model for the Formation of an Ant Cemetery.* Proc. 4th Int'l Conf. ACRI, Karlsruhe, Germany (2000).
5. Chopard, B., Droz, M. *Cellular Automata Modeling of Physical Systems.* Cambridge University Press (1998).
6. Dupuis, A., Chopard, B. *An object oriented approach to lattice gas modeling..* Future Generation Computer Systems, vol. 16(5), (2000) 523–532.

A Multiparticle Lattice Gas Automata Model for a Crowd

Stefan Marconi and Bastien Chopard

Computer Science Department
University of Geneva
1211 Geneva 4, Switzerland marconi@cui.unige.ch and
Bastien.Chopard@cui.unige.ch

Abstract. We propose to study the complex motion of a crowd with a mesoscopic model inspired by the lattice gas method. The main idea of the model is to relax the exclusion principle by which individuals are not allowed to physically occupy the same location. The dynamics is a simple collision-propagation scheme where the collision term contains the rules which describe the motion of every single individual. At present, these rules contain a friction with other individuals at the same site, a search for mobility at neighboring sites, coupled to the capacity of exploring neighboring sites. The model is then used to study three experiments: lane formation, oscillations at a door and room evacuation.

1 Introduction

The problem of crowd movement has been studied in the past using various approaches ranging from fluid dynamics [1] to coupled Langevin equations [2]. These approaches are based on the resolution of partial differential equations which model the interaction between each pair of individuals in crowd. Lately, however, the use of cellular automata have been introduced as a simple and intuitive way of simulation [3]. Such an approach deals with a local set of rules which describes the motion of individuals on a discrete representation of space. The local rules are usually simple to understand while still allowing a complex collective behavior to emerge. We propose a new model using such a mesoscopic approach inspired by the so-called lattice gas techniques [4] which are in the same stream of thought as cellular automatas. The main novelty of our model consists in relaxing the exclusion principle, by which individuals in a crowd cannot occupy the same physical location, to a probabilistic view where individuals are in fact allowed to superpose albeit an influence on their movement. With complete analogy to the statistical approach of transport phenomena in fluids, more specifically the Boltzmann equation, the dynamics of the model consists then of a succession of collisions and propagations where the emerging behavior of the system is essentially governed by the collision term. The main motivation for this approach arises from the fact that the discretization of space on a lattice does not yield any specific scale. A lattice site may just as well represent 1 m^2 or 10 m^2 and may, consequently, contain a varying number of individuals. The

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 231–238, 2002.

movement of the crowd is by definition only the flow of individuals from site to site. The main advantage of such a point of view is to completely suppress the need to resolve any conflict between individuals competing for the same physical location and thus simplify the algorithm without suppressing the richness of the phenomena.

In the following sections, we formally introduce the model and then illustrate three common experiments namely the formation of lanes in two crowds moving in opposite directions, the alternating flow at a door of a two crowds moving in opposite directions and finally the evacuation from a room of a single crowd through a door [3]. The purpose of these experiments is not serve as much to validate our choice of rules nor to outperform other approaches than to show that the exclusion principle and the conflict management it engenders can be avoided when simulating complex behavior of a crowd.

2 Model Description

The crowd is modeled by the collection of a number N of individuals distributed on a 2-dimensional regular lattice with $z+1$ directions $\boldsymbol{c}_i$, $i = 0..z$. The total number of individuals $\rho = \sum_{i=0}^{z} n_i(\boldsymbol{r}, t)$ at each lattice site $\boldsymbol{r}$ is arbitrary. In agreement with lattice gas formalism, $n_i(\boldsymbol{r}, t)$ denotes the number of individuals entering site $\boldsymbol{r}$ at time t along direction $\boldsymbol{c}_i$. Each individual is locally characterized by its favorite direction of motion $\boldsymbol{c}_F$. The movement of an individual is considered to be at $v_{\max} = |\boldsymbol{c}_i|$ towards the site pointed at by the lattice direction $\boldsymbol{c}_i$.

The direction labelled $\boldsymbol{c}_0 = 0$ points onto the site itself. It is used to model the rest direction i.e the case when there is no actual movement.

As is the case in lattice gas systems, the dynamics of the crowd movement is described by two steps: collision and propagation. The propagation consists in nothing more than moving an individual to the site pointed at by the lattice direction determined during the collision process.

The collision consists in determining the direction imposed on each individual due to the interaction with other individuals occupying the same site. Depending on the strength of the interaction, this imposed direction may or may not correspond to the favorite direction $\boldsymbol{c}_F$ of the the individual. Note that the favorite direction is typically a constant or the lattice direction which best corresponds to the shortest path to a given final position.

In a real crowd, there is no first principle which dictates how the individuals are going to interact when they meet. In what follows we propose a new set of rules which are based on common observations. First we assume that the crowd motion is subject to some friction which occurs when density is too high. This will slow down the local average velocity of the crowd by reducing the number of individuals allowed to move to a nearest lattice site. Second, the individuals are confronted with the choices of moving in their favorite direction or to move in the direction where flow exists. Finally, each individual can consider only a

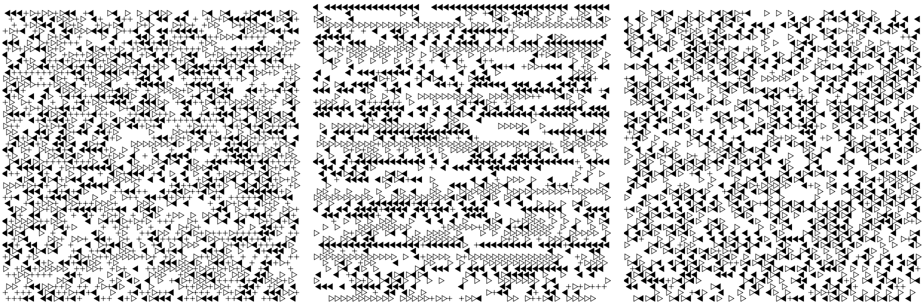


Fig. 1. Lanes formation in a crowd. Three different situations are depicted, which correspond to different model parameters. The left picture shows a crowd with no lanes: movement is essentially a sequence of collision with no deviation. The center picture shows the presence of dense lanes where individuals strongly follow each other. The right picture shows a sparse lane where individuals spread over the whole space. Black triangles indicate cells in which the mobility points to the left, whereas white ones indicate that the mobility points to the right. A cross shows a cell with zero mobility.

reduced number ξ of optional directions $\boldsymbol{c}_i$ around its favorite choice $\boldsymbol{c}_F$ when selecting its actual way out of the cell.

Thus, the factors taken into account in the collision are: the favorite direction $\boldsymbol{c}_F$ of each individual, the local density ρ, i.e. the number of individuals per site, and a quantity termed the mobility $\boldsymbol{\mu}(\boldsymbol{r}+\boldsymbol{c}_k,t)$ at all the neighboring cells $\boldsymbol{r}+\boldsymbol{c}_k$. The mobility is a post-collision normalized measure of the local flow and is simply defined as

$$\boldsymbol{\mu} = \frac{1}{\rho}\sum_{i=0}^{z} n_i \boldsymbol{c}_i \tag{1}$$

Parameters of the model are: (1) the critical density ρ_0, i.e. the number of individuals per site after which free movement is hindered; (2) a disorder term $\xi \in \{0, z/2\}$; and (3) $\eta \in [0,1]$ a term describing the will of individuals to prefer cells with high mobility. Given these definitions, the collision algorithm is

1. move to target cell $\boldsymbol{r}+\boldsymbol{c}_t$ with probability

$$P = \begin{cases} 1 & \text{if } \rho \leq \rho_0 \\ 1-\rho_0/\rho & \text{if } \rho > \rho_0 \end{cases} \tag{2}$$

   otherwise stay in current cell $\boldsymbol{r}$ (i.e. pick direction $\boldsymbol{c}_0$).

2. choose the target cell $\boldsymbol{r}+\boldsymbol{c}_t$ among the neighboring cells $\boldsymbol{r}+\boldsymbol{c}_j$ which maximize the quantity

$$2\eta \boldsymbol{c}_j \cdot \boldsymbol{\mu}(\boldsymbol{r}+\boldsymbol{c}_j,t) + 2(1-\eta)\boldsymbol{c}_j \cdot \boldsymbol{c}_F \tag{3}$$

where F labels the favorite direction and $j = F, F \pm 1, .., F \pm \xi$. Note that directions $F + i$ and $F - i$ are considered in a random order to avoid a systematic bias in case of an equal score.

Lets first consider the probability of moving: it is constructed such that the number of people allowed to move is on average ρ_0 or less. A possible interpretation for this rule is that only people at the boundary of the area of cell are able to move.

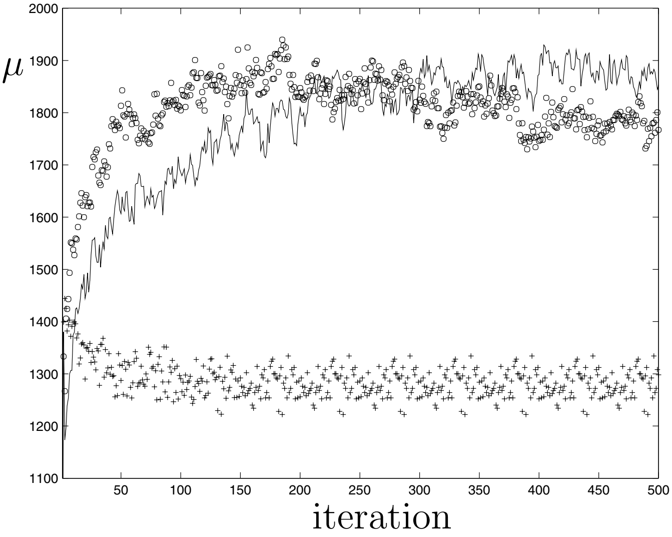


Fig. 2. A measure of total mobility in a crowd for the three cases shown in fig. 1:(crosses) no lanes $\lambda \approx 0, \eta \approx \frac{1}{2}$, (circles) dense lanes $\lambda \approx 1, \eta \approx \frac{1}{2}$,(line) sparse lanes $\lambda \approx 0, \eta \approx 1$. Note that the maximum value of the total mobility is 2500, i.e. the total number of individuals.

Once an individual is allowed to move, the choice of the destination cell is mainly governed by the agreement between the mobility at a given cell $\boldsymbol{\mu}(\boldsymbol{r}+\boldsymbol{c}_j, t)$ and the direction $\boldsymbol{c}_j$ needed to reached the cell which is given by the first term of eq. (3). This reflects the need to find mobility in a dense crowd. On the other hand, the mobility found at $\boldsymbol{r} + \boldsymbol{c}_j$ should not completely outweigh the fact that an individual possesses a favorite direction $\boldsymbol{c}_F$. Hence the presence of the second term in eq. (3) which measures the agreement between the considered direction $\boldsymbol{c}_j$ and the favorite direction $\boldsymbol{c}_F$. Therefore, movement towards a cell which opposes $\boldsymbol{c}_F$ is only possible if high mobility compensates backtracking.

The η factor is a parameter which determines whether the crowd will generally prefer mobility or their final destination. A value of $\eta = 1/2$ means both scalar products have equal weight. In the present model this is a free parameter.

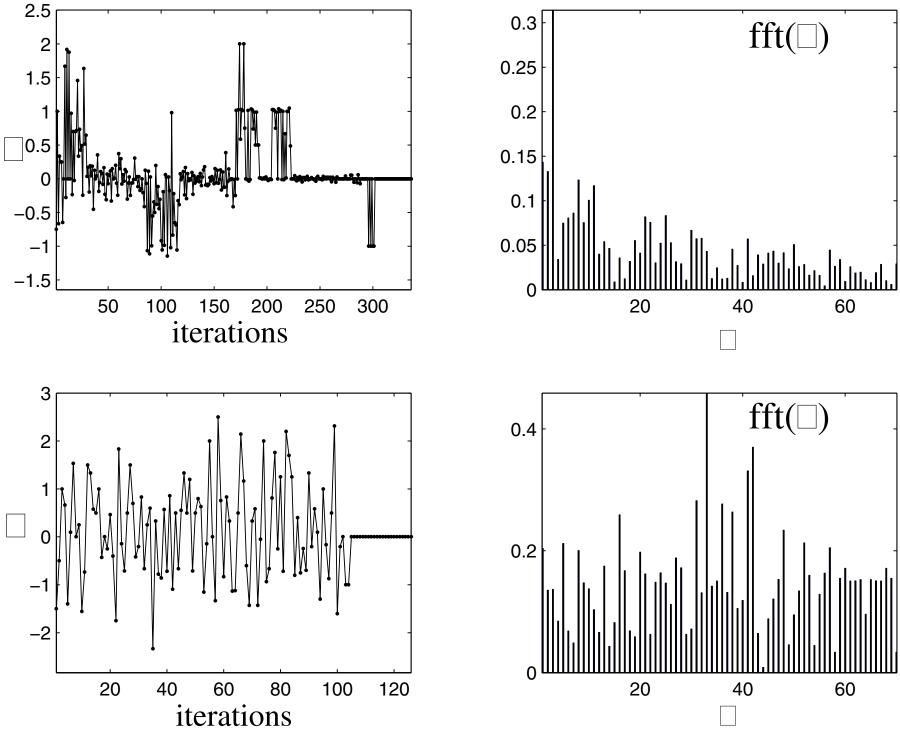


Fig. 3. Total mobility at a door of width 3 in a 200 × 50 hexagonal lattice with a wall at $x = 100$. The bottom plots are the signal and its Fourier transform fft(μ) obtained with individuals who do not interact. This measure serves as a test of the noise at the door due to the initial random configuration of the crowd. The top plots are the ones with an interacting crowd: we observe higher amplitude peaks at low frequency showing that the two crowd gain access through the door with oscillations.

However, it is more probable that in reality this should dynamically depend on the situation of an individuals e.g. even in a dense crowd backtracking just before reaching an exit door is never considered while it might be an issue before.

The disorder parameter ξ represents the ability to keep focused on the favorite direction($\xi = 0$) or to consider neighboring cells i.e. directions($\xi > 0$). Therefore the effect of ξ can be viewed as the ability to explore the environment with the side effect of diffusing the individual. We consider this as a way of modeling panic. The definition of panic is however situation dependent: a stressed individual with a clear destination will not consider any other direction but his favorite one, namely $\xi = 0$; on the contrary if there is no clear destination, one might choose to explore all directions, namely $\xi = z/2$, in the hope of finding a hidden way out.

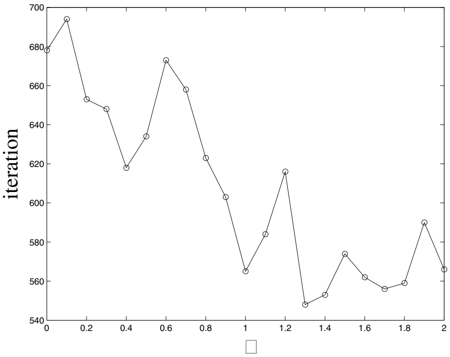


Fig. 4. The time needed to evacuate a room as a function of λ. Low λ means people do not consider other directions than the favorite one. The plot shows that by increasing the directions considered and thus slowing the mean velocity to the door, higher throughput is achieved. The plot is the result of single runs with all individuals on one side of wall with a door of size 3.

To create a sufficiently diverse crowd, the parameter ξ is initially randomly chosen for each individual. For the sake of convenience, we would like to generate this diversity using a continuous parameter $\lambda \in [0, \infty[$ which describes the mean behavior of the crowd. We therefore use a discretized power-law distribution, namely we compute $\xi = [(4x^{\frac{1}{\lambda}})]$, where $x \in [0, 1]$ is a random variable uniformly distributed.

With this method, ξ will take integer values between 0 and 3. The probability that $\xi = l$ is given by

$$\begin{aligned} P(\xi = \ell) &= P(\ell \leq 4x^{1/\lambda} < \ell + 1) \\ &= P\left(\left(\frac{\ell}{4}\right)^{\lambda} \leq x < \left(\frac{\ell+1}{4}\right)^{\lambda}\right) \\ &= \left(\frac{\ell+1}{4}\right)^{\lambda} - \left(\frac{\ell}{4}\right)^{\lambda} \end{aligned} \tag{4}$$

These probabilities result in a mean value of

$$\langle \xi \rangle = 3 - \frac{3^{\lambda} + 2^{\lambda} + 1}{4^{\lambda}}$$

When $\lambda \to 0$, ξ is always equal to 0 which means no individual ever considers any other cell but its favorite direction; with a value of $\lambda = 1.0$, values of ξ are equiprobable with $\langle \xi \rangle = 3/2$. For higher values of λ the density of individuals with $\xi = 3$ increases. Again the actual value of ξ is probably situation dependent and should vary dynamically.

3 Numerical Results

All simulations in this section were obtained using a hexagonal lattice with 2500 individuals and $\rho_0 = 2$ with an object oriented code C++ code[5,6] running on a single processor with CPU clock at 700MHz under Linux.

3.1 Lane Formation

The formation of lanes in two crowds moving in opposite directions is one of the macroscopic observations of interests to validate our approach. Fig. 1 shows the three types of states the model is able to produce. The first is a state with no lanes, individuals do not consider deviation form their favorite direction ($\lambda \approx 0$) and create high density cells where movement is reduced. The second is a dense lane formation with individuals strongly following each other($\lambda \approx 1, \eta \approx \frac{1}{2}$) even when no obstacles are present. The third is a sparse configuration of individuals where individuals have optimized the space occupation, ($\lambda \approx 0, \eta \approx 1$). The ability of the crowd to optimize its total mobility $\boldsymbol{\mu} = \sum_{\boldsymbol{r}} \mu(\boldsymbol{r}, t)$ under the three above conditions is shown in fig. 2. We see that the global behavior of the crowd is highly dependent of η and $\langle \xi \rangle$ which leads us to conclude that a realistic simulation must most probably include a mechanism which dynamically sets the values of η and ξ with regards to circumstances. However, without this sophistication, the system qualitatively behaves in a correct way.

3.2 Door Oscillations

We now look at the oscillations at a door between two crowds moving in opposite directions. The door serves as a bottleneck and jamming naturally ensues at the door. Oscillations appear when, for noise fluctuations reasons, a population on one side of the door wins access through the door thus increasing mobility for people behind. This results in a burst of one population through the door until fluctuations will inverse the situation.

In order to shows the bursts, we measure the total mobility at the door and compare it with a test simulation where no interaction ($\rho_0 > \rho, \eta = 0$) occurs between individuals. In order to outline the oscillations, we look at the Fourier transform of the signal, see fig. 3. We thus observe that the Fourier transform of the signal for interacting crowd possess higher amplitudes at low frequencies. This shows that the throughput across the door oscillates over periods of time significantly higher than an iteration.

3.3 Evacuation Problem

In the third experiment we look at the influence of the disorder parameter ξ on the time needed for a crowd to evacuate a room. The ξ parameter statistically determines how much an individual will search in other directions to find mobility. By doing so however we increase the diffusion of individuals and should consequently increase the time needed to reach the door, thus increase the total time to evacuate the room. Due to the threshold on the density, however, we observe a "freezing-by-heating" effect, see fig. 4: the average effect of increasing $\langle\xi\rangle$ is to lower evacuation time. Indeed, as λ increases, we observe a decrease of the number of iterations needed to evacuate all the individuals initially placed in the room.

The somehow oscillating structure of the plot is not yet understood; among possible explanations are the effect of the finite values of scalar product in equation (3) or an effect of sub-lattices. Such effects exist between individuals who move in opposite way through the same link at the same time and thus do not interact because they do not meet on a lattice site.

4 Conclusion

The model presented in this paper is a first attempt to consider a different approach at modeling crowd motion at a mesoscopic level, the main idea being the relaxation of the exclusion principle. This in turn allows us to define an interaction between individuals thereby assuming that what prevents individuals from moving are not the conditions at the target location but the conditions at the initial location. The non-local aspect of movement is introduced by considering mobility at neighbor locations. The main ingredients of the rules for motion are then: a simple propagation-collision scheme where the collision consists of a local friction, a search for mobility and a capacity to explore alternative routes. In this paper, we therefore show that although the exclusion principle is relaxed most of the macroscopical phenomena occurring in a crowd are simulated. The gain achieved is a simpler algorithm where no conflicts between individuals have to be resolved while still qualitatively achieving the expected complex behavior of the crowds motion. Future work should thus definitively involve a quantitative comparison with real data.

References

1. D.Helbing. A fluid-dynamic model for the movement of pedestrians. *Complex Systems*, 6(391):391 – 415, 1992.
2. D.Helbing and P.Molnar. Social force model for pedestrian dynamics. *Phys. Rev E*, 51(5):4282, 1995.
3. C.Burstedde, K.Klauck, A.Schadschneider, and J.Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A*, (295):506–525, 2001.
4. Bastien Chopard and Michel Droz. *Cellular Automata Modeling of Physical Systems.* Cambridge University Press, 1998.
5. Alexandre Dupuis and Bastien Chopard. An object oriented approach to lattice gas modeling. *Future Generation Computer Systems*, 16(5):523–532, March 2000. Best selected papers from HPCN'99.
6. PELABS site: `http://cuiwww.unige.ch/~dupuis/PELABS.html`.

CA Approach to Collective Phenomena in Pedestrian Dynamics

Andreas Schadschneider[1], Ansgar Kirchner[1], and Katsuhiro Nishinari[1,2]

[1] Institut für Theoretische Physik, Universität zu Köln, D-50937 Köln, Germany
[2] Department of Applied Mathematics and Informatics, Ryukoku University, Shiga, Japan

Abstract. Pedestrian dynamics exhibits a variety of fascinating and surprising collective phenomena (lane formation, flow oscillations at doors etc.). A 2-dimensional cellular automaton model is presented which is able to reproduce these effects. Inspired by the principles of chemotaxis the interactions between the pedestrians are mediated by a so-called floor field. This field has a similar effect as the chemical trace created e.g. by ants to guide other individuals to food places. Due to its simplicity the model allows for faster than real time simulations of large crowds.

1 Introduction

The investigation of traffic flow using methods from physics has attracted a lot of interest during the last decade [1,2]. Due to their simplicity, especially cellular automata models (CA) have been at the focus of attention. In contrast to highway traffic, pedestrian flow [3] is truely 2-dimensional and effects due to counterflow become important. This gives rise to several self-organization phenomena not observed in vehicular traffic.

The most successful model for pedestrian dynamics so far is the so-called social force model [4]. Here pedestrians are treated as particles subject to long-ranged forces induced by the social behaviour of the individuals. This idea leads to equations of motion similar to Newtonian mechanics.

Most cellular automata models for pedestrian dynamics proposed so far [5, 6,7] and can be considered as generalizations of the BML model for city traffic [8]. However, these models are not able to reproduce all the collective effects observed empirically. The same is true for more sophisticated models [9,10].

In [11,12,13,14] a new kind of CA model has been introduced which – despite its simplicity – is able to reproduce the observed collective effects. It takes its inspiration from the process of chemotaxis as used by some insects. They create a chemical trace to guide other individuals to food places. This is also the central idea of active-walker models [15,16] used for the description of human and animal trails. In the approach of [11] the pedestrians also create a trace which, in contrast to trail formation and chemotaxis, is only virtual although one could assume that it corresponds to some abstract representation of the path in the mind of the pedestrians. Its main purpose is to transform effects of long-ranged interactions

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 239–248, 2002.

(e.g. following people walking some distance ahead) into a local interaction (with the "trace"). This allows for a much more efficient simulation on a computer.

Many interesting collective effects and self-organization phenomena have been observed in pedestrian dynamics (for a review, see [2,3,17]):

Jamming: At large densities various kinds of jamming phenomena occur, e.g. when many people try to leave a large room at the same time. This clogging effect is typical for a bottleneck situation where the flow is limited by a door or narrowing and is important for practical applications, especially evacuation simulations. Other types of jamming occur in the case of counterflow where two groups of pedestrians mutually block each other.

Lane formation: In counterflow, i.e. two groups of people moving in opposite directions, a kind of spontaneous symmetry breaking occurs (see Sec. 3.1). The motion of the pedestrians can self-organize in such a way that (dynamically varying) lanes are formed where people move in just one direction [4]. In this way, strong interactions with oncoming pedestrians are reduced and a higher walking speed is possible.

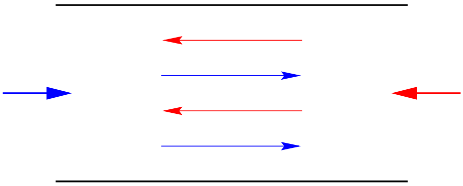


Fig. 1. Illustration of lane formation in counterflow in a narrow corridor.

Oscillations: In counterflow at bottlenecks, e.g. doors, one can observe oscillatory changes of the direction of motion (see Fig. 2). Once a pedestrian is able to pass the bottleneck it becomes easier for others to follow in the same direction until somebody is able to pass the bottleneck in the opposite direction.

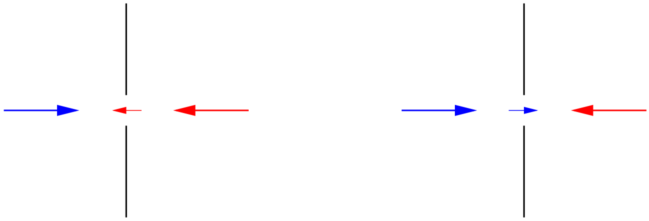


Fig. 2. Illustration of flow oscillations at a door with counterflow.

Panics: In panic situations, many counter-intuitive phenomena can occur. In the faster-is-slower effect [18] a higher desired velocity leads to a slower movement of a large crowd. In the freezing-by-heating effect [19] increasing the fluctuations can lead to a more ordered state. For a thorough discussion we refer to [17,18] and references therein.

2 Definition of the Model

First we discuss some general principles applied in the development of the model [11,12]. To allow an efficient implementation for large-scale computer simulations a discrete model is preferable. Therefore a two-dimensional CA is used with stochastic dynamics taking into account the interactions between the pedestrians. Similar to chemotaxis, we transform long-ranged interactions into local ones. This is achieved by introducting so-called *floor fields*. The transition probabilities for all pedestrians depend on the strength of the floor fields in their neighbourhood such that transitions in the direction of larger fields are preferred.

Interactions between pedestrians are repulsive for short distances ('private sphere'). This is incorporated through hard-core repulsion which prevents multiple occupation of the cells. For longer distances the interaction is often attractive, e.g. in crowded areas it is usually advantageous to walk directly behind the predecessor. Large crowds may be attractive due to curiosity and in panic situation often herding behaviour can be observed [18].

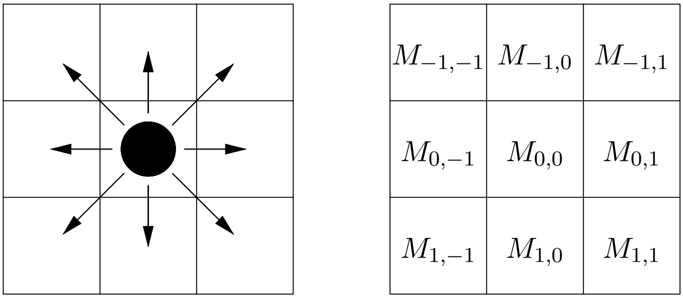
The long-ranged part of the interaction is implemented through the floor fields. We distinguish two kinds, a *static floor field* and a *dynamic floor field.* The latter models the dynamic interactions between the pedestrians, whereas the static field represents the constant properties of the surroundings.

The dynamic floor field corresponds to a virtual trace which is created by the motion of the pedestrians and in turn influences the motion of other individuals. Furthermore it has its own dynamics (diffusion and decay) which leads to a dilution and vanishing of the trace after some time. We assume the dynamic field to be discrete. Therefore the integer field strength D_{xy} can be interpreted as number of bosonic particles located at (x, y).

The static floor field does not change with time since it only takes into account the effects of the surroundings. It allows to model e.g. preferred areas, walls and other obstacles. A typical example can be found in Sec. 3.2 where the evacuation from a room with a single door is examined. Here the strength of the static field decreases with increasing distance from the door.

The introduction of the floor fields allows for a very efficient implementation on a computer since now all interactions are local. We have translated the *long-ranged spatial interaction* into a *local interaction with "memory"*. Therefore the number of interaction terms grows only linearly with the number of particles. Another advantage of local interactions can be seen in the case of complex geometries. Due to the presence of walls not all particles within the interaction range interact with each other. Therefore one needs an algorithm to check whether two particles “see” each other or whether the interaction is blocked by some obstacle. All this is not necessary here.

For some applications it is useful to introduce a *matrix of preference* which encodes the preferred walking direction and speed of each pedestrian. It is a 3 × 3 matrix (see Fig. 3) where the matrix elements M_{ij} can directly be *related to observable quantities*, namely the average velocity and its fluctuations [11].



$M_{-1,-1}$	$M_{-1,0}$	$M_{-1,1}$
$M_{0,-1}$	$M_{0,0}$	$M_{0,1}$
$M_{1,-1}$	$M_{1,0}$	$M_{1,1}$

Fig. 3. A particle, its possible transitions and the associated matrix of preference $M = (M_{ij})$.

The area available for pedestrians is divided into cells of approximately 40 × 40 cm^2 which is the typical space occupied by a pedestrian in a dense crowd [20]. Each cell can either be empty or occupied by exactly one particle (pedestrian). Apart from this simplest variant it is also possible to use a finer discretization, e.g. pedestrians occupying four cells instead of one.

In contrast to vehicular traffic the time needed for acceleration and braking is negligible in pedestrian motion. The velocity distribution of pedestrians is sharply peaked [21]. These facts naturally lead to a model where the pedestrians have a maximal velocity $v_{\max} = 1$, i.e. only transitions to neighbour cells are allowed. Furthermore, a larger $v_{\max}$ would be harder to implement in two dimensions and reduce the computational efficiency.

The stochastic dynamics of the model is defined by specifying the transition probabilities p_{ij} for a motion to a neighbouring cell (von Neumann or Moore neighbourhood) in direction (i,j). The *transition probability* p_{ij} in direction (i,j) is determined by the contributions of the static and dynamic floor fields S_{ij} and D_{ij} and the matrix of preference M_{ij} at the target cell:

$$p_{ij} = N e^{k_D D_{ij}} e^{k_S S_{ij}} M_{ij} (1 - n_{ij}) \xi_{ij} . \tag{1}$$

N is a normalization factor to ensure $\sum_{(i,j)} p_{ij} = 1$ where the sum is over the possible target cells. The factor $1 - n_{ij}$, where n_{ij} is the occupation number of the neighbour cell in direction (i,j), takes into account that transitions to occupied cells are forbidden. ξ_{ij} is a geometry factor (obstacle number) which is 0 for forbidden cells (e.g. walls) and 1 else. The coupling constants k_D and

k_S allow to vary the coupling strengths to each field individually. Their actual values depend on the situation and will be discussed in Sec. 3.2.

The update rules of the full model including the interaction with the floor fields then have the following structure [11,12]:

1. The dynamic floor field D is modified according to its diffusion and decay rules: Each boson of the dynamic field D decays with probability δ and diffuses with probability α to one of the neighbouring cells.
2. From (1), for each pedestrian the transition probabilities p_{ij} are determined.
3. Each pedestrian chooses a target cell based on the probabilities p_{ij}.
4. The conflicts arising by $m > 1$ pedestrians attempting to move to the same target cell are resolved. To avoid multiple occupancies of cells only one particle is allowed to move while the others keep their position. In the simplest case the moving particle is chosen randomly with probability $1/m$ [11].
5. The pedestrians which are allowed to move execute their step.
6. The pedestrians alter the dynamic floor field D_{xy} of the cell (x, y) they occupied before the move. The field D_{xy} at the origin cell is increased by one ($D_{xy} \to D_{xy} + 1$).

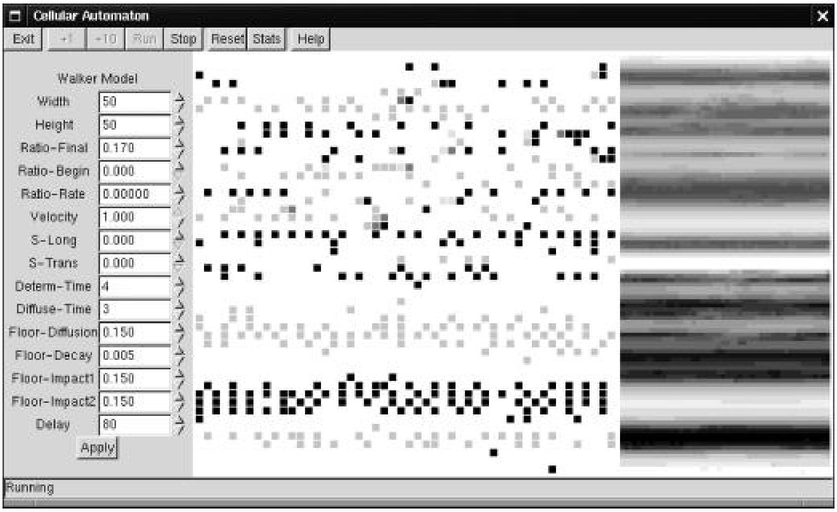
These rules are applied to all pedestrians at the same time (parallel dynamics). This introduces a timescale which corresponds to approximately 0.3 *sec* of real time [11] by identifying the maximal walking speed of 1 cell per timestep with the empirically observed value 1.3 m/s for the average velocity of a pedestrian [20]. The existence of a timescale allows to translate evacuation times measured in computer simulations into real times.

One detail is worth mentioning. If a particle has moved in the previous timestep the boson created then is not taken into account in the determination of the transition probability. This prevents that pedestrians get confused by their own trace. One can even go a step further and introduce 'inertia' [11] which enhances the transition probability in the previous direction of motion. This can be incorporated easily by an additional factor d_{ij} in eq. (1) such that $d_{ij} > 1$ if the pedestrian has moved in the *same* direction in the previous timestep and $d_{ij} = 1$ else.

3 Results

3.1 Collective Phenomena

As most prominent example of self-organization phenomena we discuss lane formation out of a randomly distributed group of pedestrians. Fig. 4 shows a simulation of a corridor which is populated by two species of pedestrians moving in opposite directions. Parallel to the direction of motion the existence of walls is assumed. Both species interact with their own dynamic floor field only. Lanes have already formed in the lower part of the corridor and can be spotted easily, both in the main window showing the positions of the pedestrians and the small windows on the right showing the floor field intensity for the two species. Simulations show that an even as well as an odd number of lanes may be formed,



Walker Model
Width 50
Height 50
Ratio-Final 0.170
Ratio-Begin 0.000
Ratio-Rate 0.00000
Velocity 1.000
S-Long 0.000
S-Trans 0.000
Determ-Time 4
Diffuse-Time 3
Floor-Diffusion 0.150
Floor-Decay 0.005
Floor-Impact1 0.150
Floor-Impact2 0.150
Delay 80

Fig. 4. Snapshot of a simulation of counterflow along a corridor illustrating lane formation. The central window is the corridor and the light and dark squares are right- and left-moving pedestrians, respectively. In the lower part of the corridor lanes have already formed whereas the upper part still disordered. The right part shows the floor fields for right- and left movers in the upper half and lower half, respectively. The field strength is indicated by the greyscale.

the latter corresponding to a spontaneous breaking of the left-right symmetry. In a certain density regime, the lanes are metastable. Spontaneous fluctuations can disrupt the flow in one lane causing the pedestrians to spread and interfere with other lanes. Eventually the system can run into a jam by this mechanism.

Apart from lane formation we have also observed oscillations of the direction of flow at doors and the formation of roundabout-like flow patterns at intersections [11,13]. Therefore the model captures – despite its simplicity – the main phenomena correctly which is important for practical applications, e.g. evacuation simulations or the optimization of escape routes.

3.2 Influence of the Floor Fields

In order to elucidate the influence of the coupling parameters k_S and k_D we investigated an evacuation process in a simple geometry, namely a large room of $L \times L$ cells with one door [14]. In the simulations, N particles are distributed randomly in the beginning, corresponding to a density $\rho = N/L^2$. All information about the location of the exits is obtained from the floor fields. The field values of the static floor field S increase in the direction of the door and are determined by some distance metric [14]. Fig. 5 shows a complex structure and the corresponding static floor field. The field strength is proportional to the distance to the nearest exit measured using a Manhattan metric.

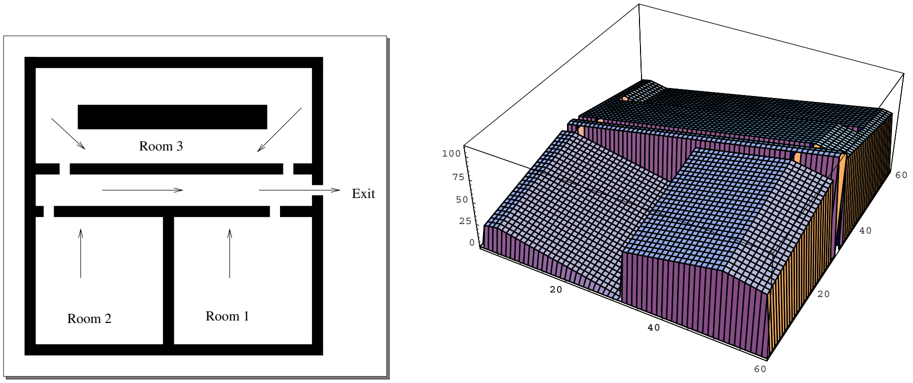


Fig. 5. Static floor field (right) for a rather complex geometry (left).

k_S, the coupling to the static field, can be viewed as a measure of the knowledge about the location of the exit. A large k_S implies an almost deterministic motion to the exit on the shortest possible path. For vanishing k_S, the individuals will perform a random walk and just find the exit by chance. So the case $k_S \ll 1$ is relevant for processes in dark or smoke-filled rooms where people do not have full knowledge about the location of the exit. For fixed sensitivity parameter k_D, the evacuation time decrease monotonically with increasing k_S (see Fig. 6(a)). k_S can be interpreted as some kind of inverse temperature for the degree of information about the inanimate surrounding.

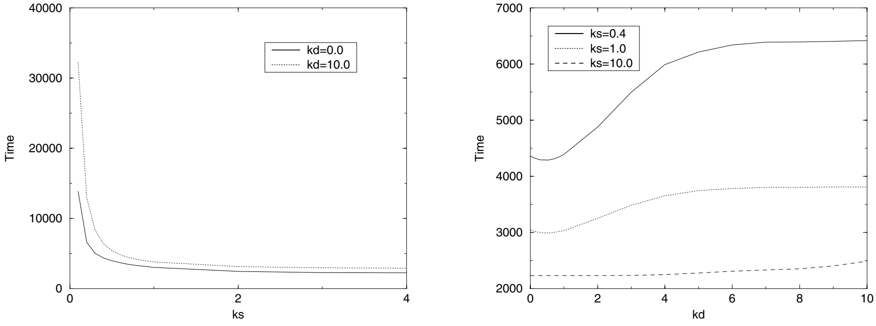


Fig. 6. Averaged evacuation times for a large room with an initial particle density of $\rho = 0.3$ and $\delta = 0.3$, $\alpha = 0.3$ for **(a)** fixed k_D, and **(b)** fixed k_S.

The parameter k_D controls the tendency to follow the lead of others. A large value of k_D implies a strong herding behaviour as observed in panics [18]. For fixed k_S (see Fig. 6(b)), the evacuation times converge to maximal values for growing k_D. The most interesting point is the occurence of minimal evacuation times for non-vanishing small values of the sensitivity parameter k_D of the dynamic field. Therefore a small interaction with the dynamic field, which is proportional to the velocity-density of the particles, is of advantage. It represents some sort of minimal intelligence of the pedestrians. They are able to detect regions of higher local flow and minimize their waiting times.

3.3 Friction Effects

In [22] a friction parameter μ has been introduced to describe clogging effects between the pedestrians. Whenever $m > 1$ pedestrians attempt to move to the same target cell, the movement of all involved particles is denied with the probability μ, i.e. all pedestrians remain at their site. This means that with probability $1 - \mu$ one of the individuals moves to the desired cell. Which particle actually moves is then determined by the rules for the resolution of conflicts described in Sec. 2. If μ is high, the pedestrians handicap each other trying to reach their desired target sites. As we will see, this local effect can have enormous influence on macroscopic quantities like flow and evacuation time [22]. Note that the kind of friction introduced here only influences interacting particles, not the average velocity of a freely moving pedestrian.

Fig. 7(a) shows the influence of the friction parameter on the evacuation time T for the scenario described in Sec. 3.2. As expected, T is monotonically increasing with μ. The strongest effect can be observed in the ordered regime,

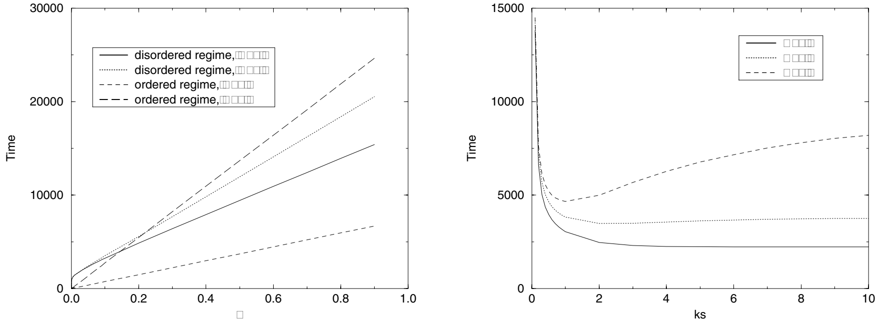


Fig. 7. Dependence of evacuation times on the friction parameter μ (a) in the ordered (k_S large, k_D small) and disordered regimes (k_S small, k_D large) and (b) as a function of k_S for $\rho = 0.3$.

i.e. for strong coupling k_S and weak coupling k_D. Here the evacuation time is mainly determined by the clogging at the door. For large values of μ it increases strongly due to the formation of self-supporting arches. This "arching" effect is well-known from studies of granular materials [23].

For fixed μ and varying coupling strength k_S a surprising result can be observed (Fig. 7(b)). For $\mu = 0$ the evacuation time is monotonically decreasing with increasing k_S since for large coupling to the static field the pedestrians will use the shortest way to the exit. For large μ, $T(k_S)$ shows a minimum at an intermediate coupling strength $k_S \approx 1$. This is similar to the faster-is-slower effect described in Sec. 1: Although a larger k_S leads to a larger effective velocity in the direction of the exit, it does not necessarily imply smaller evacuation times.

4 Conclusions

We have introduced a stochastic cellular automaton to simulate pedestrian behaviour[1]. The general idea in our model is similar to chemotaxis. However, the pedestrians leave a virtual trace rather than a chemical one. This virtual trace has its own dynamics (diffusion and decay) which e.g. restricts the interaction range (in time). It is realized through a dynamic floor field which allows to give the pedestrians only minimal intelligence and to use local interactions. Together with the static floor field it offers the possibility to take different effects into account in a unified way, e.g. the social forces between the pedestrians or the geometry of the building.

The floor fields translate spatial long-ranged interactions into non-local interactions in time. The latter can be implemented much more efficiently on a computer. Another advantage is an easier treatment of complex geometries. We have shown that the approach is able to reproduce the fascinating collective phenomena observed in pedestrian dynamics. Furthermore we have found surprising results in a simple evacuation scenario, e.g. the nonmonotonic dependence of the evacuation time on the coupling to the dynamic floor field. Also friction effects (Sec. 3.3), related to the resolution of conflict situations where several individuals want to occupy the same space, can lead to counterintuitive phenomena.

References

1. Chowdhury, D., Santen, L., Schadschneider, A.: Statistical physics of vehicular traffic and some related systems. Phys. Rep. **329** (2000) 199–329
2. Helbing, D.: Traffic and related self-driven many-particle systems. Rev. Mod. Phys. **73** (2001) 1067–1141
3. Schreckenberg, M., Sharma, S.D. (Ed.): *Pedestrian and Evacuation Dynamics*, Springer 2001
4. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. **E51** (1995) 4282–4286
5. Fukui, M., Ishibashi, Y.: Self-organized phase transitions in cellular automaton models for pedestrians. J. Phys. Soc. Jpn. **68** (1999) 2861–2863
6. Muramatsu, M., Irie, T., Nagatani, T.: Jamming transition in pedestrian counter flow. Physica **A267** (1999) 487–498

[1] Further information and Java applets for the scenarios studied here can be found on the webpage `http://www.thp.uni-koeln.de/~as/as.html`.

7. Klüpfel, H., Meyer-König, T., Wahle, J., Schreckenberg, M.: Microscopic simulation of evacuation processes on passenger ships. in *Theory and Practical Issues on Cellular Automata*, S. Bandini, T. Worsch (Eds.), Springer (2000)
8. Biham, O., Middleton, A.A., Levine, D.: Self-organization and a dynamical transition in traffic-flow models. Phys. Rev. **A46** (1992) R6124–R6127
9. Gipps, P.G., Marksjös, B.: A micro-simulation model for pedestrian flows. Math. and Comp. in Simulation **27** (1985) 95–105
10. Bolay, K.: Nichtlineare Phänomene in einem fluid-dynamischen Verkehrsmodell. Diploma Thesis, Stuttgart University (1998)
11. Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J.: Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. Physica **A295** (2001) 507–525
12. Schadschneider, A.: Cellular automaton approach to pedestrian dynamics - Theory. in [3], p. 75–85
13. Burstedde, C., Kirchner, A., Klauck, K., Schadschneider, A., Zittartz, J.: Cellular automaton approach to pedestrian dynamics - Applications. in [3], p. 87–97
14. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. Physica **A** (in press)
15. Helbing, D., Schweitzer, F., Keltsch, J., Molnar, P.: Active walker model for the formation of human and animal trail systems. Phys. Rev. **E56** (1997) 2527–2539
16. Chowdhury, D., Guttal, V., Schadschneider, A.: Cellular-automata model of ant-trail and vehicular traffic: similarities and differences. cond-mat/0201207
17. Helbing, D., Farkas, I., Molnar, P., Vicsek, T.: Simulation of pedestrian crowds in normal and evacuation situations. in [3], p. 21–58
18. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407** (2000) 487–490
19. Helbing, D., Farkas, I., Vicsek, T.: Freezing by heating in a driven mesoscopic system. Phys. Rev. Lett. **84** (2000) 1240–1243
20. Weidmann, U.: Transporttechnik der Fussgänger. Schriftenreihe des IVT **80**, ETH Zürich (1992)
21. Henderson, L.F.: The statistics of crowd fluids. Nature **229**, 381 (1971)
22. Kirchner, A., Nishinari, K., Schadschneider, A.: Friction effects and clogging in a cellular automaton model for pedestrian dynamics. in preparation.
23. Wolf, D.E., Grassberger, P. (Ed.): *Friction, Arching, Contact Dynamics*, World Scientific 1997

Opinion Formation and Phase Transitions in a Probabilistic Cellular Automaton with Two Absorbing States

Franco Bagnoli[1,4], Fabio Franci[2,4], and Raúl Rechtman[3]

[1] (Dipartimento di Energetica, Università di Firenze,
Via S. Marta, 3 I-50139 Firenze, Italy. bagnoli@dma.unifi.it
[2] Dipartimento di Sistemi e Informatica, Università di Firenze,
Via S. Marta, 3 I-50139 Firenze, Italy. bagnoli@dma.unifi.it
[3] Centro de Investigacíon en Energía, UNAM,
62580 Temixco, Morelos, Mexico. rrs@teotleco.cie.unam.mx
[4] INFM, Sezione di Firenze.

Abstract. We discuss the process of opinion formation in a completely homogeneous, democratic population using a class of probabilistic cellular automata models with two absorbing states. Each individual can have one of two opinions that can change according to that of his neighbors. It is dominated by an overwhelming majority and can disagree against a marginal one. We present the phase diagram in the mean field approximation and from numerical experiments for the simplest nontrivial case. For arbitrarily large neighborhoods we discuss the mean field results for a non-conformist society, where each individual adheres to the overwhelming majority of its neighbors and choses an opposite opinion in other cases. Mean field results and preliminary lattice simulations with long-range connections among individuals show the presence of coherent temporal oscillations of the population.

1 Modeling Social Pressure and Political Transitions

What happens to a society when a large fraction of people switches from a conformist to a non-conformist attitude? Is the transition smooth or revolutionary? These important questions, whose answers can make the difference between two well-known political points of view, is approached using a theoretical model, in the spirit of Latané's social impact theory [1,2].

We assume that one's own inclination towards political choices originates from a mixture of a certain degree of conformism and non-conformism. Conformists tend to agree with the local community majority, that is with the average opinion in a given neighborhood, while non-conformists do the opposite. However, an overwhelming majority in the neighborhood (which includes the subject itself) is always followed.

We shall study here the case of a homogeneous population, i.e. a homogeneous democratic society.[1] It may be considered as the annealed version of a

[1] The case with strong leader was studied in Ref [3].

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 249–258, 2002.

real population, which is supposedly composed by a mixture of conformist and non-conformist people who do not change easily their attitude.

In Sec. 2 we introduce a class of probabilistic cellular automata characterized by the size $2r+1$ of the neighborhood, a majority threshold q, a coupling constant J and an external field H.

We are interested in the two extreme cases: people living on a one-dimensional lattice, interacting only with their nearest neighbors ($r = 1$) and people interacting with a mean-field opinion.[2]

In Sec. 3 we present the simplest case where each individual interacts with his two nearest neighbors ($r = 1$), the mean field phase diagram and the one found from numerical experiments. For this simple case, we find a complex behavior which includes first and second order phase transitions, a chaotic region and the presence of two universality classes [6]. In Sec. 4 we discuss the mean field behavior of the model for arbitrary neighborhoods and majority thresholds when the external field is zero and the coupling constant is negative (non-conformist society). The phase diagram of the model exhibits a large region of coherent temporal oscillations of the whole populations, either chaotic or regular. These oscillations are still present in the lattice version with a sufficient large fraction of long-range connections among individuals, due to the small-world effect [7].

2 The Model

We denote by x_i^t the opinion assumed by individual i at time t. We shall limit to two opinions, denoted by -1 and 1 as usual in spin models. The system is composed by L individuals arranged on a one-dimensional lattice. All operations on spatial indices are assumed to be modulo L (periodic boundary conditions). The time is considered discontinuous (corresponding, for instance, to election events). The state of the whole system at time t is given by $\boldsymbol{x}^t = (x_0^t, \dots, x_{L-1}^t)$ with $x_i^t \in \{-1, 1\}$;

The individual opinion is formed according to a local community "pressure" and a global influence. In order to avoid a tie, we assume that the local community is formed by $2r+1$ individuals, counting on equal ground the opinion of the individual himself at previous time. The average opinion of the local community around site i at time t is denoted by $m_i^t = \sum_{j=-r}^{r} x_{i+j}^t$.

The control parameters are the probabilities p_s of choosing opinion 1 at time $t+1$ if this opinion is shared by s people in the local community, i.e. if the local "field" is $m = 2s - 2r - 1$.

Let J be a parameter controlling the influence of the local field in the opinion formation process and H be the external social pressure. The probability p_s are given by

$$p_s = p_{(m+2r+1)/2} \propto \exp(H + Jm).$$

[2] Related models in one and two dimensions have been studied in Refs [4,5].

One could think to H as the television influence, and J as educational effects. H pushes towards one opinion or the other, and people educated towards conformism will have $J > 0$, while non-conformists will have $J < 0$. In the statistical mechanics lingo, all parameters are rescaled to include the temperature.

The hypothesis of alignment to overwhelming local majority is represented by a parameter q, indicating the critical size of local majority. If $s < q$ ($m < 2q - 2r - 1$), then $x_i^{t+1} = -1$, and if $s > 2r + 1 - q$ ($m > 2r + 1 - 2q$), then $x_i^{t+1} = 1$.

In summary, the local transition probabilities of the model are

$$p_s = \begin{cases} 0 & \text{if } s < q; \\ \dfrac{\mathcal{A}\mathcal{B}^s}{1 + \mathcal{A}\mathcal{B}^s} & \text{if } q \leq s \leq 2r + 1 - q; \\ 1 & \text{if } s > 2r + 1 - q; \end{cases} \tag{1}$$

where $\mathcal{A} = \exp[2H - 2J(2r + 1)]$ and $\mathcal{B} = \exp(4J)$.

For $q = 0$ the model reduces to an Ising spin system. For all $q > 0$ we have two absorbing homogeneous states, $\boldsymbol{x} = \boldsymbol{-1}$ ($c = 0$) and $\boldsymbol{x} = \boldsymbol{1}$ ($c = 1$) corresponding to infinite coupling (or zero temperature) in the statistical mechanical sense. With these assumptions, the model reduces to a one-dimensional, one-component, totalistic cellular automaton with two absorbing states.

The order parameter is the fraction c of people sharing opinion 1. [3] It is zero or one in the two absorbing states, and assumes other values in the active phase. The model is symmetric since the two absorbing states have the same importance.

3 A Simple Case

Let us study the model for the simplest, nontrivial case: one dimensional lattice, $r = 1$ and $q = 1$. We have a probabilistic cellular automaton with three inputs, and two free control parameters, p_1 and p_2, while $p_0 \equiv 0$ and $p_3 \equiv 1$, according to Eq. (1).

One can also invert the relation between (H, J) and (p_1, p_2):

$$H = \frac{1}{4} \log \left(\frac{p_1 p_2}{(1 - p_1)(1 - p_2)} \right), \qquad J = \frac{1}{4} \log \left(\frac{p_2(1 - p_1)}{p_1(1 - p_2)} \right).$$

The diagonal $p_2 = p_1$ corresponds to $J = 0$ and the diagonal $p_2 = 1 - p_1$ to $H = 0$.

At the boundaries of probability intervals, we have four deterministic (elementary) cellular automata, that we denote T3, T23, T13 and T123, where the digits indicate the values of s for which $p_s = 1$ [8].

Rule T3 (T123) brings the system into the $c = 0$ ($c = 1$) absorbing state except for the special case of the initial configuration homogeneously composed by the opposite opinion.

[3] The usual order parameter for magnetic system is the magnetization $M = 2c - 1$.

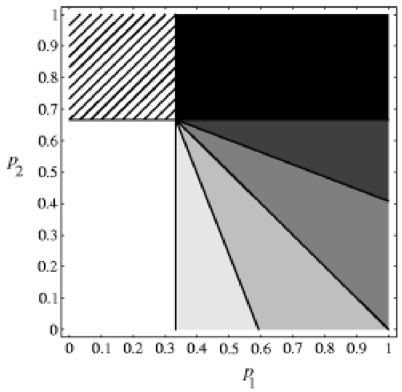


Fig. 1. Mean-field phase diagram for the density c coded as gray levels ranging from white ($c = 0$) to black ($c = 1$). The dashed upper-left region denotes the coexistence phase, in which both states $c = 0$ and $c = 1$ are stable, and the final state depends on the initial density (first-order transition).

Rule T23 is a strict majority rule, whose evolution starting from a random configuration leads to the formation of frozen patches of zeros and ones in a few time steps. A small variation of the probabilities induces fluctuations in the position of the patches. Since the patches disappear when the boundaries collide, the system is equivalent to a model of annihilating random walks, which, in one dimensions, evolves towards one of the two absorbing states according to the asymmetries in the probabilities or in the initial configuration.

Rule T13 on the other hand is a "chaotic" one, [4] leading to irregular pattern for almost all initial conditions (except for the absorbing states). These patterns are microscopically very sensitive to perturbations, but very robust for what concerns global quantities (magnetization).

The role of frustrations (the difficulty of choosing a stable opinion) is evidentiated by the following considerations. Rule T23 corresponds to a ferromagnetic Ising model at zero temperature, so an individual can simply align with the local majority with no frustrations. On the contrary, rule T13 is unable to converge towards an absorbing state (always present), because these states are unstable: a single individual disagreeing with the global opinion triggers a flip in all local community to which he/she belongs to. It is possible to quantify these concepts by defining stability parameters similar to Lyapunov exponents [10].

We start by studying the mean-field approximation for the generic case, with p_1 and p_2 different from zero or one.

Let c and c' denote the density of opinion 1 at times t and $t+1$ respectively. We have

$$c' = 3p_1 c\left(1-c\right)^2 + 3p_2 c^2\left(1-c\right) + c^3.$$

[4] Called rule 150 in Ref. [9]

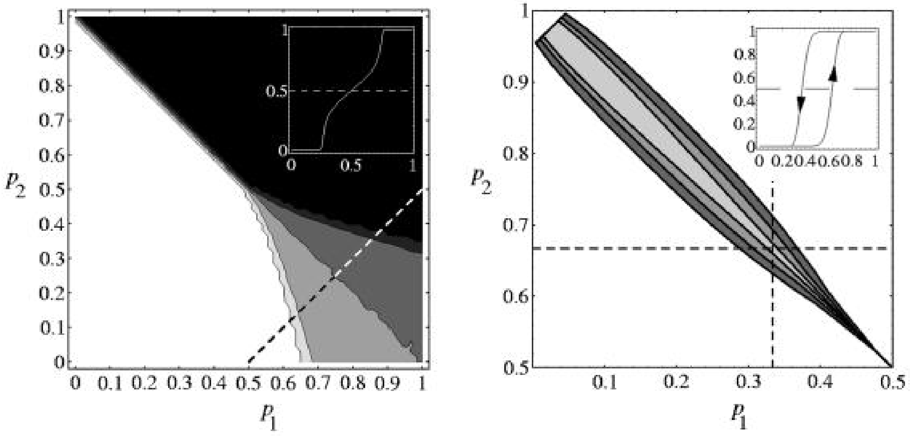


Fig. 2. Numerical phase diagram for the density c (left) and hysteresis region for several values of the noise ϵ and relaxation time T (right). Same color code as in Figure 1.

This map has three fixed points, the state $\boldsymbol{x} = \mathbf{-1}$ ($c = 0$), the state $\boldsymbol{x} = \mathbf{1}$ ($c = 1$) and a microscopically disordered state ($0 \leq c \leq 1$). The model is obviously symmetric under the changes $p_1 \to 1 - p_2$, $p_2 \to 1 - p_1$ and $x \to 1 - x$, implying a fundamental equivalence of political opinions in this model. The stability of fixed points marks the different phases, as shown in Fig. 1.

The stability of the state $c = 1$ ($c = 0$) corresponds to large social pressure towards opinion 1 (-1). The value of J determines if a change in social pressure corresponds to a smooth or abrupt transition.

3.1 Phase Transitions and Damage Spreading in the Lattice Case

The numerical phase diagram of the model starting from a random initial state with $c^0 = 0.5$ is shown in Fig. 2. The scenario is qualitatively the same as predicted by the mean-field analysis. In the upper-left part of the diagram both states $c = 0$ and $c = 1$ are stable. In this region the final fate of the system depends on the initial configuration.

Due to the symmetry of the model, the two second-order phase transition curves meet at a bicritical point $(p_t, 1 - p_t)$ where the first-order phase transition line ends. Crossing the second-order phase boundaries on a line parallel to the diagonal $p_1 = p_2$, the density c exhibits two critical transitions, as shown in the inset of the right panel of Fig. 2. Approaching the bicritical point the critical region becomes smaller, and corrections to scaling increase. Finally, at the bicritical point, the two transitions coalesce into a single discontinuous (first-order) one.

First-order phase transitions are usually associated to a *hysteresis cycle* due to the coexistence of two stable states. To visualize the hysteresis loop (inset of the right panel of Fig. 2) we modify the model slightly by letting $p_0 = 1 - p_3 = \epsilon$

with $\epsilon \ll 1$. In this way the configurations $\boldsymbol{x} = -\mathbf{1}$ and $\boldsymbol{x} = \mathbf{1}$ are no longer absorbing. This brings the model back into the class of equilibrium models for which there is no phase transition in one dimension but metastable states can nevertheless persist for long times. The width of the hysteresis cycle, shown in the right panel of Fig. 2, depends on the value of ϵ and the relaxation time T.

We study the asymptotic density as p_1 and p_2 move on a line with slope 1 inside the dashed region of Fig. 1. For p_1 close to zero, the model has only one stable state, close to the state $c = 0$. As p_1 increases adiabatically, the new asymptotic density will still assume this value even when the state $c = 1$ become stable. Eventually the first state become unstable, and the asymptotic density jumps to the stable fixed point close to the state $c = 1$. Going backwards on the same line, the asymptotic density will be close to one until that fixed point disappears and it will jump back to a small value close to zero.

Although not very interesting from the point of view of opinion formation models, the problem of universality classes in the presence of absorbing states have attracted a lot of attention by the theoretical physics community in recent years [11,12]. For completeness we report here the main results [6].

It is possible to show that on the symmetry line one can reformulate the problem in terms of the *dynamics of kinks* between patches of empty and occupied sites. Since the kinks are created and annihilated in pairs, the dynamics conserves the initial number of kinks modulo two. In this way we can present an exact mapping between a model with symmetric absorbing states and one with parity conservation.

Outside the symmetry line the system belongs to the directed percolation universality class [13]. We performed simulations starting either from one and two kinks. In both cases $p_t = 0.460(2)$, but the exponents were found to be different. Due to the conservation of the number of kinks modulo two, starting from a single site one cannot observe the relaxation to the absorbing state, and thus $\delta = 0$. In this case $\eta = 0.292(5)$, $z = 1.153(5)$. On the other hand, starting with two neighboring kinks, we find $\eta = 0.00(2)$, $\delta = 0.285(5)$, and $z = 1.18(2)$. These results are consistent with the parity conservation universality class [3,4].

Let us now turn to the sensitivity of the model to a variation in the initial configuration, i.e. to the study of *damage spreading* or, equivalently, to the location of the chaotic phase. Given two replicas $\boldsymbol{x}$ and $\boldsymbol{y}$, we define the difference $\boldsymbol{w}$ as $\boldsymbol{w} = \boldsymbol{x} \oplus \boldsymbol{y}$, where the symbol $\oplus$ denotes the sum modulus two.

The damage h is defined as the fraction of sites in which $w = 1$, i.e. as the Hamming distance between the configurations $\boldsymbol{x}$ and $\boldsymbol{y}$. We study the case of maximal correlations by using just one random number per site, corresponding to the smallest possible chaotic region [14].

In Fig. 3 the region in which the damage spreads is shown near the lower-right corner (chaotic domain). Outside this region small spots appear near the phase boundaries, due to the divergence of the relaxation time (second-order transitions) or because a small difference in the initial configuration can bring the system to a different absorbing state (first-order transition). The chaotic

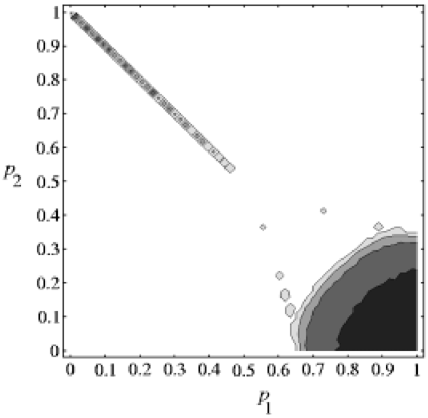


Fig. 3. Phase diagram for the damage found numerically by considering the evolution starting from uncorrelated configurations with initial density equal to 0.5.

domain is stable regardless of the initial density. On the line $p_2 = 0$ the critical point of the density and that of the damage spreading coincide.

3.2 Reconstruction of the Potential

An important point in the study of systems exhibiting absorbing states is the formulation of a coarse-grained description using a *Langevin equation.* It is generally accepted that the directed percolation universal behavior is represented by

$$\frac{\partial c(x,t)}{\partial t} = ac(x,t) - bc^2(x,t) + \nabla^2 c(x,t) + \sqrt{c(x,t)}\alpha(x,t),$$

where $c(x,t)$ is the density field, a and b are control parameters and α is a Gaussian noise with correlations $\langle \alpha(x,t)\alpha(x',t')\rangle = \delta_{x,x'}\delta_{t,t'}$. The diffusion coefficient has been absorbed into the parameters a and b and the time scale.

It is possible to introduce a zero-dimensional approximation to the model by averaging over the time and the space, assuming that the system has entered a metastable state. In this approximation, the size of the original systems enters through the renormalized coefficients $\overline{a}$, $\overline{b}$,

$$\frac{\partial c(x,t)}{\partial t} = \overline{a}c(x,t) - \overline{b}c^2(x,t) + \sqrt{c(x,t)}\alpha(x,t),$$

where also the time scale has been renormalized.

The associated Fokker-Planck equation is

$$\frac{\partial P(c,t)}{\partial t} = -\frac{\partial}{\partial c}(\overline{a}c - \overline{b}c^2)P(c,t) + \frac{1}{2}\frac{\partial^2}{\partial c^2}cP(c,t),$$

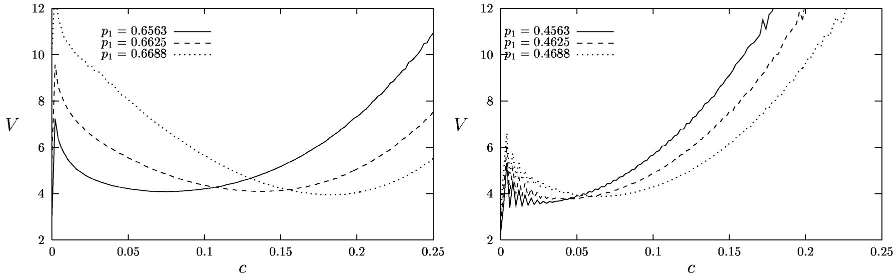


Fig. 4. Reconstruction of potential $V(c)$ for $p_2 = 0$ (left) and for the kink dynamics on the line $p_2 = 1 - p_1$ (right).

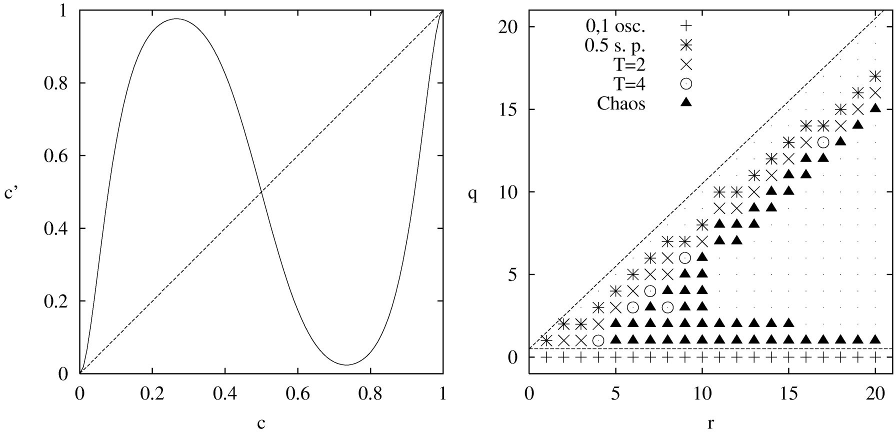


Fig. 5. Case $H = 0$, $J = -\infty$: the mean field map Eq. (2) (left) for $r = 10$ and $q = 2$ and the mean-field $r - q$ phase diagram (right). In the phase diagram the absorbing states are always present. Points mark parameter values for which the absorbing states are the only stable attractors. A plus sign denotes period-2 temporal oscillations between absorbing states, a star denotes the presence of a stable point at $c = 0.5$, a cross (circle) denotes period-two (four) oscillations between two non-zero and non-one densities, triangles denote chaotic oscillations.

where $P(c, t)$ is the probability of observing a density c at time t. One possible solution is a δ-peak centered at the origin, corresponding to the absorbing state.

By considering only those trajectories that do not enter the absorbing state during the observation time, one can impose a detailed balance condition, whose effective agreement with the actual probability distribution has to be checked *a posteriori*. The *effective potential* V is defined as $V(c) = -\log(P(c))$ and can be found from the actual simulations.

In the left panel of Fig. 4 we show the profile of the reconstructed potential V for some values of p around the critical value on the line $p_2 = 0$, over which the model belongs to the DP universality class. One can observe that the curve becomes broader in the vicinity of the critical point, in correspondence of the divergence of critical fluctuations $\chi \sim |p - p_c|^{-\gamma'}$, $\gamma' = 0.54$ [15]. By repeating the same type of simulations for the kink dynamics (random initial condition), we obtain slightly different curves, as shown in the right panel of Fig. 4. We notice that all curves have roughly the same width. Indeed, the exponent γ' for systems in the PC universality class is believed to be exactly 0 [16], as given by the scaling relation $\gamma' = d\nu_\perp - 2\beta$ [15]. Clearly, much more information can be obtained from the knowledge of $P(c)$, either by direct numerical simulations or dynamical mean field trough finite scale analysis, as shown for instance in Ref. [17].

4 Larger Neighborhoods

In order to study the effects of a larger neighborhood and different threshold values q, let us start with the well known two-dimensional "Vote" model. It is defined on a square lattice, with a Moore neighborhood composed by 9 neighbors, synthetically denoted M in the following [8]. If a strict majority rule $q = 4$ is applied (rule M56789, same convention as in Sec. 2) to a random initial configuration, one observes the rapid quenching of small clusters of ones and zero, similar to what happens with rule T23 in the one-dimensional case. A small noise quickly leads the system to an absorbing state. On the other hand, a small frustration $q = 3$ (rule M46789) for an initial density $c^0 = 0.5$ leads to a metastable point formed by patches that evolve very slowly towards one of the two absorbing states. However, this metastable state is given by the perfect balance among the absorbing states. If one starts with a different initial "magnetization", one of the two absorbing phases quickly dominates, except for small imperfections that disappear when a small noise is added.

In the general case, the mean-field equation is

$$c' = \sum_{s=0}^{2r+1} \binom{2r+1}{s} c^s (1-c)^{2r+1-s} p_s, \tag{2}$$

sketched in the left panel of Fig. 5.

We studied the asymptotic behavior of this map for different values of r and q. For a given r there is always a critical q_c value of q for which the active phase disappears, with an approximate correspondence $q_c \simeq 4/5r$. The active phase is favored by the presence of frustrations and the absence of the external pressure, i.e. for $J < 0$ and $H = 0$. We performed extensive computations for the extreme case $J = -\infty$, $H = 0$ corresponding to a society of non-conformists without television. As shown in the right panel of Fig. 5, by increasing the neighborhood size r, one starts observing oscillations in the equilibrium process. This is evident in the limit of infinite neighborhood: the parallel dynamics induced by elections (in our model) makes individual tend to disalign from the marginal majority, originating temporal oscillations that can have a chaotic character. Since the absorbing states are present, and they are stable for $q > 0$, the coherent oscillations of the population can bring the system into one absorbing state. This is reflected by the “hole” in the active phase in the mean field phase diagram.

Preliminary lattice simulations (not shown) reveal that this behavior is still present if there is a sufficiently large fraction of long-range connections due to the small-world effect [7], while the active phase is compact in the short-range case.

5 Conclusions

Although this model is quite rough, there are aspects that present some analogies with the behavior of a real society. In particular, the role of education, represented by the J parameter. In a society dominated by conformists, the transitions are difficult and catastrophic, while in the opposite case of non-conformist people the transitions are smooth. However, in the latter case a great social pressure is needed to gain the majority.

On the other hand, if the neighborhood is large and non-conformism is present, one can observe the phenomenon of political instabilities given by temporal oscillations of population opinion, which could be interpreted as a symptom of healthy democracy.

References

1. Latané, B.: American Psychologist **36** (1981) 343
2. Galam, S., Gefen, Y., Shapir, Y.: Math. J. of Sociology **9**, (1982) 1
3. Holyst, J.A., Kacperski, K, Schweitzer, F.: Physica A 285 (2000) 199
4. Galam, S., Chopard, B., Masselot, A., Droz, M.: Eur. Phys. J. B **4**, (1998) 529
5. Chopard, B., Droz, M., Galam, S.: Eur. Phys. J. B **16**, (2000) 575
6. Bagnoli, F., Boccara, N., Rechtman, R.: Phys. Rev. E **63** (2001) 46116 ; cond-mat/0002361
7. Watts, D. J., Strogatz S. H.: Nature **393** (1998) 440
8. Vichniac, G. Y.: *Cellular Automata Models of Disorder and Organization* In Bienestok, E., Fogelman, F., Weisbuch, G. (eds.), *Disordered Systems and Biological Organization*, NATO ASI Series, b F20/b, Berlin: Springer Verlag (1986) pp. 283-293; `http://www.fourmilab.ch/cellab/manual/cellab.html`
9. Wolfram, S.: Rev. Mod. Phys. **55** (1983) 601
10. Bagnoli, F., Rechtman, R., Ruffo, S.: Phys. Lett. A **172** (1992) 34 (1992).
11. Grassberger, P., von der Twer, T.: J. Phys. A: Math. Gen. **17** (1984) L105; Grassberger, P.: J. Phys. A: Math. Gen. **22** (1989) L1103
12. Hinrichsen, H. Phys. Rev. E **55** (1997) 219
13. Kinzel, W. In Deutsch, G., Zallen, R., Adler, J. (eds.): *Percolation Structures and Processes*, Adam Hilger, Bristol (1983); Kinzel, E., Domany, W.: Phys. Rev. Lett. **53** (1984) 311 ; Grassberger, P.: J. Stat. Phys. **79** (1985) 13
14. Hinrichsen, H., Weitz, J.S., Domany, E.: J. Stat. Phys. **88** (1997) 617
15. Muñoz, M.A., Dickman, R., Vespignani, A., Zapperi, S.: Phys. Rev. E **59** (1999) 6175
16. Jensen, I.: Phys. Rev. E **5**0 (1994) 3263
17. Jensen, I., Dickman, R., Phys. Rev. E **48** (1993) 1710

Cellular Automata Based Authentication (CAA)

Monalisa Mukherjee[1], Niloy Ganguly[2], and P. Pal Chaudhuri[1]

[1] Department of Computer Science & Technology, Bengal Engineering College (D.U),
Botanic Garden, Howrah, West Bengal, India 711103,
{mona, ppc}@cs.becs.ac.in

[2] Computer Centre, IISWBM, Calcutta, West Bengal, India 700073,
n_ganguly@hotmail.com
monalisa_mukherjee@hotmail.com

Abstract. Current demands for secured communication have focussed intensive interest on 'Authentication'. There is a great demand for a high-speed low cost scheme for generation of Message Authentication Code (MAC). This paper introduces a new computational model built around a special class of Cellular Automata (CA) that can be employed for both message and image authentication. Cryptanalysis of the proposed scheme indicates that compared to other known schemes like MD5, SHA1 etc., the current scheme is more secure against all known attacks. High speed execution of the proposed scheme makes it ideally suitable for real time on-line applications. Further, the regular, modular, and cascadable structure of CA with local interconnections makes the scheme ideally suitable for VLSI implementation with throughput in the range of Gigabits per second.

1 Introduction

The human society is currently living in 'Cyber Age'. Phenomenal technological advances of this age have brought unprecedented benefits to the society. However, at the same time this has generated some unique social problems the human society has never encountered in the history of civilization. The issue of 'Cyber Crime' has become a major challenge for law-makers, government officials, social workers and technologists around the globe. *Secured communication in the networked society of cyber age is a pre-requisite for growth of human civilization of twenty-first century.*

Electronic transfer of all types of digital files demands authentication and verification of data source, protection of copyright and detection of intrusion. A strong trend in the development of the mechanisms for authentication of both message and image is based on cryptographic hash functions designed for MD5 by Rivest. However, hash functions are not originally designed for application in the field of authentication. The conventional MD5 based message authentication, as reported in [1], cannot withstand the cryptanalytic attacks.

This paper reports a simple, high speed, low cost authentication scheme for digital messages and images. It employs the computing model of a special class of Cellular Automata (CA) referred to as GF(2^p) CA. The theory of extension field of GF(2^p) has provided the foundation of this model.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 259–269, 2002.

Table 1. The *CA* Rule Table

With XOR (linear *CA*)	With XNOR (complemented rule)
rule 60 : $q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$	rule 195 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_i(t)}$
rule 90 : $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$	rule 165 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_{i+1}(t)}$
rule 102 : $q_i(t+1) = q_i(t) \oplus q_{i+1}(t)$	rule 153 : $q_i(t+1) = \overline{q_i(t) \oplus q_{i+1}(t)}$
rule 150 : $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$	rule 105 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)}$
rule 170 : $q_i(t+1) = q_{i+1}(t)$	rule 85 : $q_i(t+1) = \overline{q_{i+1}(t)}$
rule 204 : $q_i(t+1) = q_i(t)$	rule 51 : $q_i(t+1) = \overline{q_i(t)}$
rule 240 : $q_i(t+1) = q_{i-1}(t)$	rule 15 : $q_i(t+1) = \overline{q_{i-1}(t)}$

2 CA 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 states of its two neighbors and itself (Fig. 1). Each cell stores 0 or 1 in GF(2). The next state function (local transition function) of a cell is defined by one of the 256 (2^{2^3}) rules [4]. Some of the XOR and XNOR rules of GF(2) *CA* are noted in Table 1. A *CA* employing only XOR rules is referred to as Linear, while the ones using both XOR and XNOR are referred to as Additive *CA*. A *CA* with XNOR rules can be viewed as a *CA* with XOR rules and an inversion vector F to account for the XNOR logic function.

Such a CA we have marked as GF(2) CA. In order to enhance the computing power of such a three neighborhood structure, GF(2^p) CA [8] has been proposed.

2.1 GF(2^p) CA

The Fig. 1 depicts the general structure of an n-cell GF(2^p) CA. Each cell of such a CA having p number of memory elements can store an element $\{0, 1, 2, ..., 2^p - 1\}$ in GF(2^p). In $GF(2^p)$ [5], there exists an element α that generates all the non-zero elements, $\alpha, \alpha^2, ...\alpha^{2^p-1}$, of the field. α is termed as the *generator.* α can be represented by a $p \times p$ matrix having its elements as $\{0, 1\} \in GF(2)$. The *matrix representation* of element α^j $(j = 2, 3, \cdots, (2^p - 1))$ for $p = 2$ is shown in Fig. 2.

The connections among the cells of the CA are weighed in the sense that to arrive at the next state $q_i(t+1)$ of i^{th} cell, the present states of $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ are multiplied respectively with w_{i-1}, w_i and w_{i+1} and then added. The *addition* and *multiplication* follows the rule of addition and multiplication defined in GF(2^p). So, under three neighborhood restriction, the next state of the i^{th} cell is given by -
$q_i(t+1) = \phi\left((w_{i-1}, q_{i-1}), (w_i, q_i), (w_{i+1}, q_{i+1})\right)$. ϕ denotes the local transition function of the i^{th} cell and w_{i-1}, w_i & $w_{i+1} \in$ GF(2^p) specify the weights of interconnection. A three neighborhood n cell GF(2^p) CA is equivalent to np cell $3p$ neighborhood GF(2) CA. The structure of GF(2^p) CA provides hierarchy and abstraction that can be exploited in many applications [8].

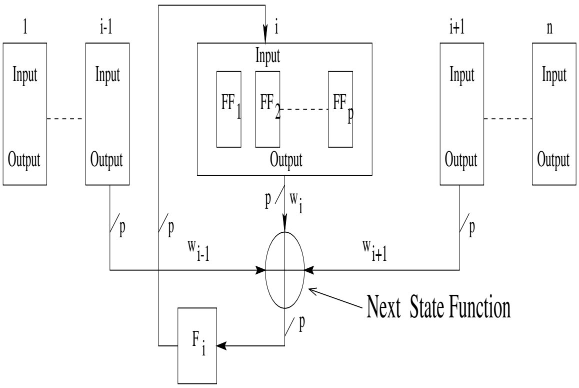


Fig. 1. General structure of a GF(2^p) CA (For p=1, it's a conventional GF(2) CA)

An n cell GF(2^p) CA can be characterized by the $n \times n$ characteristic matrix T, where

$$T_{ij} = \begin{cases} w_{ij}, & \text{if the next state of the } i^{th} \text{ cell} \\ & \text{depends on the present state of the} \\ & j^{th} \text{ cell by a weighed } w_{ij} \in GF(2^p) \\ 0, & \text{otherwise} \end{cases}$$

F = an n symbol inversion vector with each of its element in GF(2^p).

The state of a GF(2^p) CA at time t is an $n-$symbol string, where a symbol $\in$GF(2^p) is the content of a CA cell. If s_t represents the state of the automata at the t^{th} instant of time, then the next state, at the $(t+1)^{th}$ time, is given by $s_{(t+1)} = T * s_t + F,$ and
$s_{(t+n)} = T^n * s_t + (I + T + T^2 + \cdots + T^{n-1}) * F.$
The '*' and '+' operators are the operators of the Galois Field GF(2^p). If the F vector of GF(2^p) CA is an all zero vector, the CA is termed as linear CA, else it is an Additive CA.

In the CA state transition graph, if all the states lie in some cycles, it is called a group CA. For a group CA, det[T]$\neq$ 0. If the characteristic matrix T is singular, that is det[T] = 0, then the CA is a non-group CA. The T matrix of the example non-group GF(2^2) CA of Fig. 2 has the elements in GF(2^2). Its state transition graph has a single component of an inverted tree with a root (a node with self-loop) referred to as 'Attractor'. Consequently, such a CA is marked as Single Attractor CA ($SACA$).

Definition 1 *Dependency matrix D - if all the non-zero weights in T are replaced by 1 then it is referred to as the dependency matrix of the CA in GF(2^p).*

For $p = 1$, dependency matrix is the characteristic matrix T of GF(2) CA (Fig. 2).

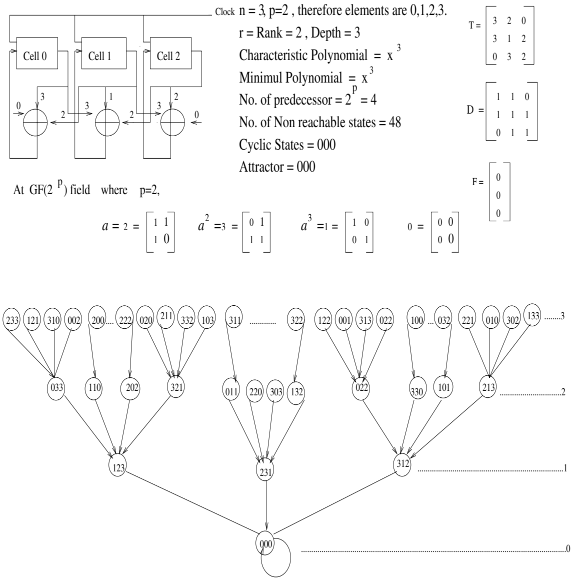


Fig. 2. State Transition Diagram of 3-cell GF(2^2) *SACA*

2.2 Single Attractor Cellular Automata (*SACA*)

The CA belonging to this class and its complemented counterpart referred to as Dual *SACA* display some unique features that have been exploited in the proposed authentication scheme. The T matrix of an n cell GF(2^p) *SACA* is an $n \times n$ matrix with its elements in GF(2^p). The rank, characteristic polynomial and minimal polynomial of the T matrix are :
rank(T) $= n - 1$, rank($T \oplus$ I) $= n$, I being the $n \times n$ identity matrix.
Characteristic polynomial $= \alpha x^n$, Mimimal polynomial $= \alpha x^n$, where $\alpha \in$ GF(2^p).
A few theorems are next introduced without proof. The proof is analogous to GF(2) TPSA (Two Predecessor Single Attractor) CA noted in [2].

Theorem 1 : If the rank of the characteristic matrix T of an n cell GF(2^p) non-group CA is $n - 1$, then each reachable state has 2^p predecessors.

Theorem 2 : Depth of an n cell *SACA* is equal to n

The inversion vector F in the example *SACA* of Fig. 2 is an all 0's vector. A non-zero F leads to its dual counterpart.

Dual SACA

A dual *SACA* also referred to as $\overline{SACA}$ results from an introduction of non-zero inversion vector F with the characteristic matrix T of the *SACA*. $\overline{SACA}$ has identical state transition behavior as that of *SACA* with change of relative

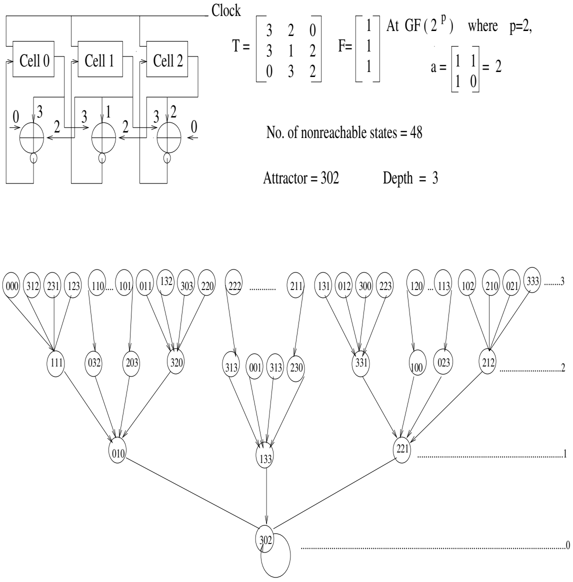


Fig. 3. Structure and state transition graph of a 3 cell GF(2^2) Dual $SACA$

position of states. All the reachable states in a $SACA$ becomes non-reachable in $\overline{SACA}$ [2]. The example CA of Fig. 3 is a dual counterpart of the $SACA$ of Fig. 2. The following Theorem characterizes a $SACA$ and $\overline{SACA}$.

Theorem 3 : If the complement vector F of a GF(2^p) $SACA$ with characteristic matrix T is such that $T^n.F = 0$, and $T^{n-1}.F \neq 0$, then this complemented CA is a dual $SACA$

Detailed characterization of a $SACA$ and its dual are reported in [7]

2.3 Synthesis of $SACA$ and Its Dual

The algorithmic steps for synthesis of an n cell GF(2^p) $SACA$ and its dual are noted below with illustrating example. The **Steps 1** and **2** ensures that the resulting CA is a $SACA$ - the proof is omitted for shortage of space.

Step 1. Generate the dependency matrix D of size $n \times n$ whose 1^{st} cell has no dependency on its neighbors (left, self and right) and the rest of the cells having dependency on its left neighbor only. For a 3 cell GF(2^2) $SACA$, D is: $\begin{pmatrix} 0\,0\,0 \\ 1\,0\,0 \\ 0\,1\,0 \end{pmatrix}$

Step 2. Construct characteristic matrix T of the $SACA$ from D by performing elementary row/column operations such that each of the cells has dependency on left, self and right neighbors.

For a 3 cell GF(2^2) $SACA$, T is: $\begin{pmatrix} 3 & 2 & 0 \\ 3 & 1 & 2 \\ 0 & 3 & 2 \end{pmatrix}$

Step 3. Construct $\overline{SACA}$ by implementing the result of Theorem 3.

3 Cellular Automata Based Authentication (CAA) Scheme for Message/Image

The schemes for message and image authentication are noted along with proof of robustness against the attacks. The GF(2) CA based authentication scheme proposed in [9] is insecure against attacks based on Differential Cryptanalysis. The proposed scheme overcomes the problem.

3.1 $SACA$ as One-Way Hash Function Generator

The proposed scheme employs *keyed one-way hash* function based authentication using GF(2^p) $SACA$ and its dual $\overline{SACA}$. The one-way hash function maps a secret key and an arbitrary length input message data to a fixed length hash output referred to as message digest .

3.2 CAA for Digital Message

Let, A has a message M to send to B and they share a common secret key K. A calculates message digest $C_K(M)$ from M, and K employing one way $SACA$ based hash function. Message M and digest $C_K(M)$ are transmitted to B where B performs the same function on the received message to generate a new digest $C_K(M^{'})$. The message gets authenticated if $C_K(M)$ and $C_K(M^{'})$ matches.

Algorithm 1 Generate_Message_Digest

```
Input: Message M of length $|M\,|$ bits; Private key $\mathcal{P}$: $n \times p$ bits :
- n cell GF($2^p$) SACA and its dual $\overline{SACA}$
Output: Message Digest: $n \times p$ bits
Step 1: Group Message M into k blocks { $M_1$, $M_2$, ... $M_k$ } each of length n symbol ($S_1$, $S_2$, ..., $S_n$) in GF($2^p$)
Let $\mathcal{P}_1 = \mathcal{P}$ (Private Key)
For(i=1 to k)
{
Step 2: Form a tridiagonal matrix $CA_{M_i}$ whose n diagonal elements are n-symbols of $M_i$; off diagonal values are 1 and the remaining all values are zero
Step 3: Run each of the CAs for one step:
(a) Run $CA_{M_i}$ with $\mathcal{P}_i$ as seed to obtain ${\mathcal{P}_i}^1$
(b) Run $SACA$ with ${\mathcal{P}_i}^1$ as seed to obtain ${\mathcal{P}_i}^2$
(c) Run $\overline{SACA}$ with ${\mathcal{P}_i}^2$ as seed to obtain ${\mathcal{P}_i}^3$
Step 4: Let $\mathcal{P}_{i+1} = {\mathcal{P}_i}^3$
}
Step 5: Output $\mathcal{P}_{k+1}$ as the Message Digest
```

3.3 Robustness of CAA for Digital Message

Robustness of the proposed scheme is analyzed against probable attacks.
Attack 1: Brute Force Attack
Birthday attack, Collision attack belong to this category of attacks. An authentication scheme can be made robust against such attacks by increasing message digest/key length. The CAA scheme can easily employ Variable Length key of any size since it employs simple, regular, modular, cascadable structure of CA. So, CAA can be efficiently designed against such attacks.
Attack 2: The Extension Attack or the Padding Attack
This type of attack is not possible for the proposed scheme as it employs a keyed hash function where the key is not a part of the original message.

A detailed description of robustness of CAA against Attack 1 and 2 is reported in [7].

Attack 3: Next the robustness of CAA is tested in respect of the strength of the $SACA$ based hash function employed for the scheme. The attacks are employing much more subtlety than mere brute force attack.

Cryptanalytic attacks attempt to guess whether the function is such that two messages or keys, close to each other in terms of bit distance, produce the outputs which are also close to each other. If it is so, then the code can be broken in much lesser time than exhaustive search. The following two results show that our scheme is protected against such attack.

Result 3(a): Let M be an arbitrary message while $M^{'}$ is another message derived out of M by flipping a randomly chosen bit of M. The corresponding message digests are $C_K(M)$ and $C_K(M^{'})$. From Table 2 this is clear that the difference (performing XOR between $C_K(M)$ and $C_K(M^{'})$) has on the average the same number of zeros and ones.

Table 2 shows that there are equal number of zeros and ones in the output difference which indicates that flipping one bit of a randomly chosen message results in a completely different message digest.

Result 3(b): Let M be an arbitrary message and K be a secret key while $K^{'}$ is another secret key derived out of K by flipping a randomly chosen bit of K. The corresponding message digests are $C_K(M)$ and $C_{K'}(M)$. If the difference between $C_K(M)$ and $C_{K'}(M)$ has almost same number of zeros and ones, then it can be concluded that flipping a bit of Key results in a completely different message digest(Table 2).

In both the attacks, the result becomes better as the value of p increases.

Attack 4: Next CAA is analyzed from the viewpoint of another very important attack called differential attack [10]. The attack analyzes the plaintext pairs along with their corresponding message digest pairs to identify the correlations that would enable identification of the secret key.

Table 2. Results of Result 3(a) and 3(b) on CAA and MD5

Input	Result 3(a)					Result 3(b)				
size	No. of ones for CAA				No. of	No. of ones for CAA				No. of
of	$(C_K(M) \oplus C_K(M^{'})$				ones for	$(C_K(M) \oplus C_{K^{'}}(M)$				ones for
file in	key-length 128		key-length 256		MD5	key-length 1 28		key-length 256		MD5
bytes	p=4	p=8	p=8	p=16	128 bit	p=4	p=8	p=8	p=16	128 bit
3239	34	70	128	122	69	54	63	134	130	64
3239	56	67	124	132	69	70	66	140	132	69
3239	45	66	122	138	70	52	64	136	126	66
65780	55	76	114	138	64	64	66	130	142	70
65780	57	65	140	128	65	45	64	104	134	68
65780	59	65	118	140	67	66	63	118	120	62
259120	38	62	134	136	70	46	69	122	126	67
259120	51	64	130	130	65	55	64	132	128	66
259120	55	66	132	132	67	48	70	140	128	76

For example, let the length of the plain text and message digest are of 8 bit and the fixed bit difference D taken as 3. For a pair of plaintexts X=11001011, $X^{'}$=10011001, corresponding message digests are (say)
MD=00110101, $MD^{'}$=10000110, i.e, with difference $D^{'}$=5. The value of $D^{'}$ is calculated for all plaintext pairs with D=3. Then from the distribution of $D^{'}$, we can calculate the standard deviation (σ). In general, a one-way hash function is said to be protected from differential cryptanalytic attack if σ is lower than 10 % [12].

We have performed differential cryptanalysis on our scheme with 50 different files having 5 different size. For each file, we take 5 different fixed input differences. Table 3 depicts results of differential cryptanalysis on CAA. From Table 3 (Column 2 to 7) this is clear that as p increases (p is the dimension of Galois field GF(2^p)) σ decreases. The experimental results at Table 3 establish that *CAA* can defend differential attack in a better way than MD5 (Column 6).

Execution time

Comparative results for GF(2) CA based authentication algorithm [9], MD5 and CAA at GF(2^p) in respect of *CPU time* are displayed in the Table 3 (Column 9 to 13). These experimental results establish the higher speed of execution of CAA scheme based on GF(2^p) *SACA*. Higher value of p leads to reduction of computation time because rather than handling $np \times np$ matrix with GF(2) elements we deal with $n \times n$ matrix with GF(2^p) elements. In software the speed is almost one and half times more than MD5 at p=16. The throughput of the Hardwired implementation of scheme is of the order of tens of Gigabits/sec.

3.4 CAA for Watermarking

Digital watermarking research has generally focused upon two classes of watermarks, fragile and robust. Fragile watermarks is ideal for image authentication applications [13,14]. In this watermarking it allows a user with an appropriate

Table 3. Differential cryptanalysis for CAA and Comparative speed of CAA and MD5 (in WindowsNT 4.00-1381,IBM)

Input size of file in bytes	Avg.Std.Devn of XOR Distribution for $SACA$ (%)							CPU Time in Seconds				
	key-length 128				key-length 256			MD5	p=1	p=2	p=4	p=8
	p=1	p=2	p=4	p=8	p=8	p=16		method	n=128	n=64	n=32	n=16
1608	9.110	8.950	7.881	5.899	5.660	4.883		0.0549	0.055	0.050	0.040	0.040
35860	14.821	12.111	8.458	6.134	6.123	5.123		0.165	0.147	0.105	0.105	0.087
65780	8.989	7.813	6.657	5.034	5.002	4.986		0.193	0.166	0.129	0.110	0.091
142164	6.824	6.771	5.998	4.823	4.989	5.024		0.2198	0.2053	0.1650	0.118	0.081
259120	14.100	11.783	10.213	7.982	6.102	4.033		0.299	0.271	0.267	0.210	0.200
852984	13.015	12.443	7.893	4.342	3.032	4.003		0.330	0.294	0.252	0.205	0.205

secret key to verify the authenticity, integrity and ownership of an image. If the user performs the watermark extraction with an incorrect key or an image which is not watermarked, the user obtains an Image that resembles noise.

Recent systems apply sophisticated embedding mechanisms, including the use of cryptographic hash functions to detect changes to a watermarked image. This section reports a watermarking scheme that employs CAA based hash functions.

Let the original grey-scale image be X. A bi-level watermark ‘A’ will be inserted in it and again will be extracted from it for authentication. X and A are divided into some equal blocks of size $n \times n$ and say, each block of X is termed as X_r and A as A_r.

Insertion

Let, Image block $X_r = \begin{pmatrix} 255 & 128 \\ 108 & 11 \end{pmatrix}$ or, $\begin{pmatrix} 11111111 & 10000000 \\ 01101100 & 00001011 \end{pmatrix}$ and

Watermark block $A_r = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$.

- Set all LSBs of X_r to 0, $X_r{}' = \begin{pmatrix} 1111111\mathbf{0} & 1000000\mathbf{0} \\ 0110110\mathbf{0} & 0000101\mathbf{0} \end{pmatrix}$ is obtained.
- **Hash $X_r{}'$ using CAA** and the hash output $H_r = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.
- Perform pixel by pixel ex-or operation between H_r and A_r, $(H_r \oplus A_r = C_r)$ and obtain $C_r = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$.
- Insert values of C_r into all LSBs of $X_r{}'$. The resulting watermarked block $X_r{}^w = \begin{pmatrix} 1111111\mathbf{0} & 1000000\mathbf{1} \\ 0110110\mathbf{0} & 0000101\mathbf{1} \end{pmatrix}$ or, $\begin{pmatrix} 254 & 129 \\ 108 & 11 \end{pmatrix}$.

Extraction

- Let, $Y_r = \begin{pmatrix} 254 & 129 \\ 108 & 11 \end{pmatrix}$ or, $\begin{pmatrix} 11111110 & 10000001 \\ 01101100 & 00001011 \end{pmatrix}$ be the watermarked image block.

- Extract all LSBs from Y_r, $C_r = \begin{pmatrix} 0\,1 \\ 0\,1 \end{pmatrix}$ is obtained.
- Set all LSBs of Y_r to 0, and obtain $Y_r{}' = \begin{pmatrix} 1111111\mathbf{0}\ 1000000\mathbf{0} \\ 0110110\mathbf{0}\ 0000101\mathbf{0} \end{pmatrix}$.
- Hash $Y_r{}'$ by CAA and obtain $H_r = \begin{pmatrix} 1\,1 \\ 1\,0 \end{pmatrix}$.
- Perform pixel by pixel ex-or operation between H_r and C_r to obtain water mark image block $A_r = \begin{pmatrix} 1\,0 \\ 1\,1 \end{pmatrix}$.

Analysis and Comparative Study

The inherent advantages of the proposed scheme can be summarised as follows:
(a) The greatest advantage of our scheme is the flexibility of adjusting key size without any overhead. This is possible due to modular structure of Cellular Automata.
(b) The Table 4 shows the result where in each of the image, watermark has been inserted according to proposed scheme (column 4 to 7) gives better PSNR values than MD5 (column 8). Moreover as extension field parameter p increases in CAA, PSNR value improves (Table 4). The robustness of CAA based one-way hash function, as noted in Section 3.3, has resulted in the superior quality watermarked image.

Table 4. Comparison of PSNR values using CAA for different p and MD5

Image name	Data in Bytes	Block size	PSNR Values in dB unit				
			Wong-Memon method				
			p=1	p=2	p=4	p=8	MD5
Sachin	522835	14 x 30	51.126629	51.201994	51.29048	51.541979	51.013072
SkylineArch	964451	72 x 60	52.013388	52.216852	52.427391	52.811862	51.034981
Lena	1064071	60 x 90	53.23367	53.295081	53.463457	53.788033	51.243123
Concord	1485604	80 x 84	53.830272	53.884655	54.020056	54.526984	51.317890
Rabbit	964451	80 x 72	52.177280	52.307440	52.443773	52.725227	51.103782

(c) The most effective attack on image authentication is Holliman-Menon attack or Vector Quantization attack [6]. CAA based watermarking is tuned to counterfeit this attack as a built-in function whereas all other hash functions (including MD5) defend the attack externally which effectively decreases the insertion/extraction speed of watermarking.

4 Conclusion

This paper reports GF(2^p) Cellular Automata (CA) based Authentication (CAA) scheme. The scheme has been employed to insert fragile watermark in images. Security of CAA against known attacks and its execution speed are emphatically better than those of MD5. Future prospective of the CAA lies in building robust watermarking scheme using the proposed CA based one-way hash function and extend the scheme to develop a digital signature scheme for e-commerce application.

References

1. B. Schineier *Applied Cryptography* tJohn Wiley and Sons, 2001.
2. P.Pal Chaudhuri, D.Roy Choudhury, S. Nandi and S. Chattopadhyay *Additive Cellular Automata Theory and Applications.* IEEE Computer Society Press, California, USA, 1997.
3. S. Wolfram *Cryptography with Cellular Automata* Proceedings of Crypto'85, pp.429-432
4. S. Wolfram *Cellular Automata and Complexity* Addition-Wesly Publishing Company, 1994
5. T. R. N. Rao and E. Fujiwara *Error-control Coding for Computer Systems* Prentice-Hall, Englewood Cliffs, N.J, 1989
6. M. Holliman and N. Memon *Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes* IEEE trans. on Image Processing, volume-9, No-3, March,2000
7. M. Mukherjee, B.K.S, N. Ganguly and P. Pal Chaudhuri *GF(2^p) Cellular Automata As A Message Digest Generator* 9 th International Conference on Advance Computing and Communications, December-2001
8. K Paul, D. Roy Chowdhury, P. Pal Chaudhuri *Theory of Extended Linear Machine*, to be published in IEEE, Transaction on Computers.
9. P. Dasgupta, S. Chattopadhyay and I. Sengupta *An ASIC for Cellular Automata based message Authentication* 12 th Int. Conf. on VLSI Design, 1999
10. D. Wagner *Differential cryptanalysis of KHF* 5th Int. Workshop on Fast Software Encryption, 1998
11. B. Preneel and P. van Oorschot *MD-x MAC and building fast MACs from hash functions* CRYPTO 95 Proceedings, Vol.963, D. Coppersmith ed., Springer-Verlag, 1995.
12. E. Biham and A. Shamir *Differential Cryptanalysis of DES-like Cryptosystems* Journal of Cryptology, 4(1991), 3-72
13. J. Fridrich *Security of Fragile Authentication Watermarks with Localization* Proc. of SPIE Photonic West, Electro Imaging 2002, Security and Watermarking of Multimedia Contents, January-2002
14. P. W. Wong and N. Memon *Secret and Public Key Image Watermarking Schemes for Image Authentication and ownership verification* IEEE trans. on Image Processing, volume-10, No-10, October,2001

Cellular Automata Machine for Pattern Recognition

Pradipta Maji[1], Niloy Ganguly[2], Sourav Saha[1], Anup K. Roy[1], and
P. Pal Chaudhuri[1]

[1] Department of Computer Science & Technology, Bengal Engineering College (D U),
Howrah, India 711103,
{pradipta@,sous@,anup@,ppc@}cs.becs.ac.in
[2] Computer centre, IISWBM, Calcutta, India 700073, n_ganguly@hotmail.com

Abstract. This paper reports a *Cellular Automata Machine* (*CAM*) as a general purpose pattern recognizer. The *CAM* is designed around a general class of *CA* known as *Generalized Multiple Attractor Cellular Automata* (*GMACA*). Experimental results confirm that the sparse network of *CAM* is more powerful than conventional dense network of Hopfield Net for memorizing unbiased patterns.

1 Introduction

This paper reports the design of a Cellular Automata Machine (CAM) to address the problem of Pattern Recognition. The design is based on an elegant computing model of a particular class of Cellular Automata (CA) referred to as Generalized Multiple Attractor CA (GMACA). The extensive experimental results confirm that CAM provides an efficient and cost-effective alternative to the dense network of Neural Net for solving pattern recognition problem.

The *Associative Memory Model* provides an excellent solution to the problem of pattern recognition. This model divides the entire state space into some pivotal points (a, b, c, d of *Fig.1*) that represent the patterns learnt. The states close to a pivotal point get associated with a specific learnt pattern. Identification of an input pattern (without or with distortion due to noise) amounts to traversing the transient path (*Fig.1*) from the given input pattern to the closest pivotal point. *As a result, the process of recognition becomes independent of the number of patterns learnt.*

In early 80's, the seminal work of Hopfield [1] made a breakthrough by modeling a *recurrent, asynchronous, neural net* as an *associative memory* system. But, the dense network of neural net and its complex structure has partially restricted its application. Search for alternative model around simple sparse network of *Cellular Automata* (*CA*) continued [2,3,4,5,6]. The *simple, regular, modular, cascadable local neighborhood* structure of *Cellular Automata* (*CA*) serves as an excellent sparse network model of associative memory. Such a structure can be efficiently realized with *VLSI* technology.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 270–281, 2002.

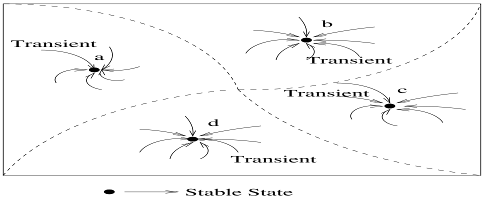


Fig. 1. Model of Associative Memory

In this paper we propose the design of a *CAM* for pattern recognition. The *CAM* basically acts as an *Associative Memory* which is also referred to as *Content Addressable Memory*. Thus the acronym *CAM* stands for both *Cellular Automata Machine* acting as *Content Addressable Memory*. **The memorizing capacity of *CAM*, as established in this paper, can be found to be better than that of conventional Hopfield Net by 33%.**

2 Cellular Automata Preliminaries

A *Cellular Automaton* (*CA*) consists of a number of cells organized in the form of a lattice. It evolves in discrete space and time. The next state of a cell depends on its own state and the states of its neighboring cells. In a two state 3-neighborhood *CA*, there can be a total of 2^{2^3} - that is, 256 distinct next state functions of a cell referred to as the *rule* of *CA* cell [7].

A special class of *CA* referred to as *Multiple Attractor CA* (*MACA*) designed with rule vector $< 150, 102, 60, 150 >$ is shown in *Fig.2*. Its state space gets divided into four attractor basins; each basin contains an attractor (with self loop) and transient states. *MACA* employing linear/additive *CA* rules have been widely studied in the book [8].

Generalized Multiple Attractor Cellular Automata *(GMACA) employs non-linear CA rules with attractor cycle length greater than or equal to 1. It can efficiently model an associative memory [9,10,11].* The *Fig.3* illustrates the state space of a 4-cell *GMACA* with rule vector $< 202, 168, 218, 42 >$. The state space of this *CA* is divided into two attractor basins. The non-cyclic states are referred to as *Transient States* in the sense that a *CA* finally settles down in one of its attractor cycles after passing through such transient states.

In order to model an *associative memory* for a given set of patterns $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_i, \cdots \mathcal{P}_k\}$, following two relations has to be satisfied by the *GMACA*.

R1: Each attractor basin of the *GMACA* should contain one and only one pattern ($\mathcal{P}_i$) to be learnt in its attractor cycle; the corresponding basin is referred to as $\mathcal{P}_i$-basin.

R2: The hamming distance (*HD*) of each state $\mathcal{S}_i \in \mathcal{P}_i$-basin with $\mathcal{P}_i$ is lesser than that of $\mathcal{S}_i$ with any other $\mathcal{P}$'s. That is, $HD(\mathcal{S}_i, \mathcal{P}_i) < HD(\mathcal{S}_i, \mathcal{P}_j)$, $\mathcal{P}_j \in \mathcal{P}$ and $\mathcal{P}_j \neq \mathcal{P}_i$.

While the relation **R1** ensures uniqueness of the stored patterns, the relation **R2** ensures recognition of patterns with distortion due to noise.

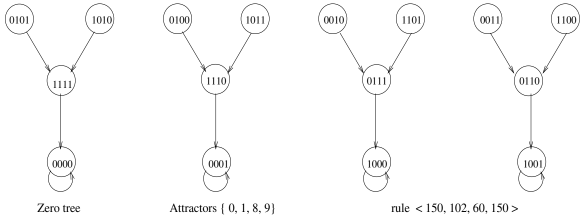


Fig. 2. State space of a 4-cell *MACA* divided into four attractor basins

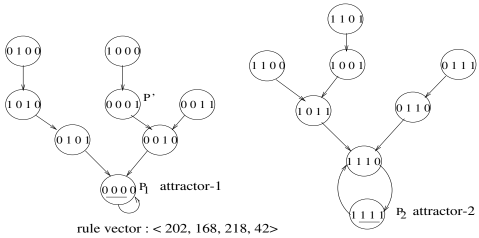


Fig. 3. State space of a GMACA divided into two attractor basins

Example 1 *Suppose, we want to recognize two patterns* $\mathcal{P}_1 = 0000$ *and* $\mathcal{P}_2 = 1111$ *of length 4 with single bit noise. We first synthesize a CA (rule vector) for which the state transition behavior of* $GMACA$ *is similar to that of* Fig.3*, that maintains both* **R1** *and* **R2***.*

It learns two patterns, $\mathcal{P}_1 = 0000$ *and* $\mathcal{P}_2 = 1111$*. The state* $\acute{\mathcal{P}} = 0001$ *has the hamming distances 1 and 3 with* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *respectively. Let* $\acute{\mathcal{P}}$ *be given as the input and its closest match is to be identified with one of the learnt patterns. The recognizer designed with the* $GMACA$ *of* Fig.3 *is loaded with* $\acute{\mathcal{P}} = 0001$*. The* $GMACA$ *returns the desired pattern* $\mathcal{P}_1$ *after two time steps.*

3 Synthesis of CA Machine (CAM)

The CAM is synthesized around a $GMACA$. The synthesis of $GMACA$ can be viewed as the training phase of CAM for Pattern Recognition. The $GMACA$ synthesis scheme outputs the rule vector of the desired $GMACA$ that can recognize a set of given patterns with or without noise.

3.1 GMACA Evolution

The $GMACA$ synthesis procedure consists of three phases. It accepts the patterns to be learnt as the input. It cycles through these phases till desired $GMACA$ is reached as its output or the specified time limit gets elapsed with null output.

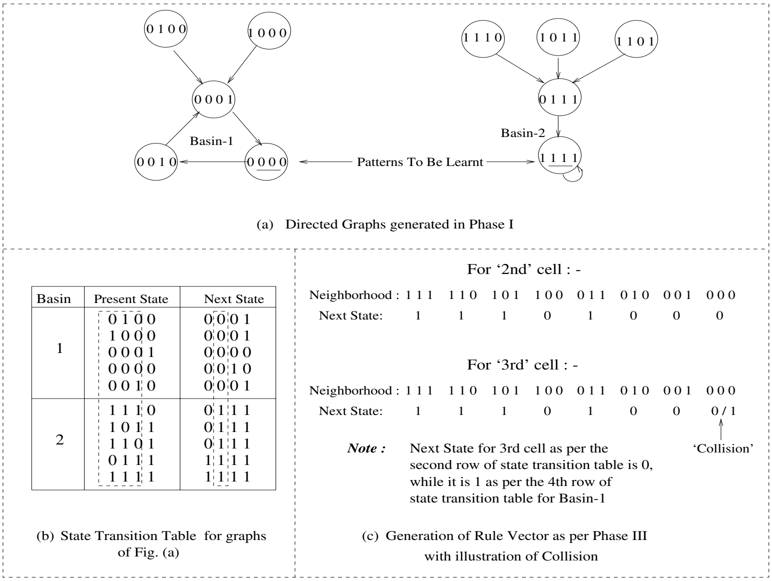
Let, the input be denoted as k number of n bit patterns to be learnt. The output should be an n-cell $GMACA$ with k number of attractor basins, with each attractor basin having a unique pattern from the input patterns to be learnt.

Phase I - Since, the state transition diagram of a $GMACA$ can be conceived as a graph, we first randomly generate k number of *directed graphs* with each node coded as an n-bit string. The cycle length l permissible for the generated graph is assumed to be less than or equal to L_{max}; where L_{max} is the *maximum permissible length* of the attractor cycle. Each graph represents a basin of the candidate $GMACA$ to be synthesized; while its cycle represents an attractor cycle. Each directed graph has a unique pattern $\mathcal{P}_i$ in its attractor cycle, while those with limited noise added to $\mathcal{P}_i$ are the patterns in the $\mathcal{P}_i$-basin.

The number of nodes p of each graph is equal to the number of states in the corresponding basin - that is, the patterns without or with specified noise. So,

$$p = \sum_{r=0}^{r_{max}} \binom{n}{r} \tag{1}$$

where, r is the number of noisy bits and r_{max} is the *maximum permissible noise* that can be tolerated by the $GMACA$ based pattern recognizer. After mapping a particular pattern (say P_i to be learnt) on to the cyclic node of a graph, we randomly map other patterns to be covered by the P_i basin at different nodes of same graph. Note that the $\mathcal{P}_i$-basin of the $GMACA$ to be synthesized covers the states with permissible noise of r bits ($r = 0$ to r_{max}) added to $\mathcal{P}_i$. A design example follows.



(a) Directed Graphs generated in Phase I

Basin	Present State	Next State
1	0 1 0 0	0 0 0 1
	1 0 0 0	0 0 0 1
	0 0 0 1	0 0 0 0
	0 0 0 0	0 0 1 0
	0 0 1 0	0 0 0 1
2	1 1 1 0	0 1 1 1
	1 0 1 1	0 1 1 1
	1 1 0 1	0 1 1 1
	0 1 1 1	1 1 1 1
	1 1 1 1	1 1 1 1

(b) State Transition Table for graphs of Fig. (a)

For '2nd' cell : -

Neighborhood :	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
Next State:	1	1	1	0	1	0	0	0

For '3rd' cell : -

Neighborhood :	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
Next State:	1	1	1	0	1	0	0	0 / 1

'Collision'

Note : Next State for 3rd cell as per the second row of state transition table is 0, while it is 1 as per the 4th row of state transition table for Basin-1

(c) Generation of Rule Vector as per Phase III with illustration of Collision

Fig. 4. Randomly Generated Directed Graphs with state transition tables and CA Rules

Example 2 *The* Fig.4 (a) *represents two arbitrary graphs generated in Phase I for* $n = 4$ *and* $r_{max} = 1$. *Patterns to be learnt* $P_1 = 0000$ *and* $P_2 = 1111$ *are mapped onto the nodes of the attractor cycle of length 3 and 1 respectively. The sets* 0001, 0010, 0100, 1000 *and* 1110, 1101, 1011, 0111 *are the noisy patterns with noise of 1 bit added to* $\mathcal{P}_i$ $(i = 1, 2, \cdots)$ *respectively. These are mapped in two attractor basins as shown in* Fig.4 (a).

Phase II - From all the graphs selected in *Phase I*, we generate a state transition table. The *Fig.4 (b)* represents state transition tables derived from two directed graphs shown in *Fig.4 (a)*. Total number entries in the state transition table is $k \cdot p$, where k is the number of patterns to be learnt and p is the number of states in the graph generated as per the *Equation 1*. For the example, graphs of *Fig.4 (a)*, the number of patterns to be learnt (k) is 2 ($\mathcal{P}_1$ and $\mathcal{P}_2$) and the number of states in each basin (p) is 5. So the total number entries in the state transition table of *Fig.4 (b)* is 10.

Phase III - Design the rule vector of the $GMACA$ cells from the state transition table.

Consider i^{th} cell whose rule is to be identified. We concentrate on 3 columns - $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ columns of all the $k \cdot p$ number of patterns of the state transition table of *Fig.4 (b)*. Suppose, for a present state configuration of the i^{th} cell, the next state is '0' for n_0 times and '1' for n_1 times, then state '0' and '1' collides with each other to become the next state of the cell for that configuration.

The *Fig.4 (c)* represents the neighborhood configurations along with the next state of 2^{nd} and 3^{rd} cells of the patterns of two basins noted in *Fig.4 (b)*. For 2^{nd} cell, there is no collision between next state '0' and '1' of the cell for 8 possible configurations. Whereas for '000' neighborhood configuration, the next state of 3^{rd} row of *Fig.4 (b)*) for 1 time and '1' (4^{th} row of *Fig.4 (b)*) for 1 time. That is, $n_0 = 1$ and $n_1 = 1$. So, for '000' configuration, the next state of 3^{rd} cell may be '0' or '1' generating an instance of collision.

In order to resolve this conflict we introduce the following heuristics.

3.2 Resolution of Collision

- If $n_0 \simeq n_1$, the collision between state '0' and '1' is high. In that case, we randomly decide the next state of a cell.
- If $n_0 >> n_1$ or $n_0 << n_1$, the collision is minimum. In that case, the next state of a cell is '0' if $n_0 > n_1$, otherwise '1'.

Design of a synthesis algorithm to arrive at an effective $GMACA$, we observed, is a **hard problem**. So, we fall back on the *Simulated Annealing* (SA) program to solve this problem.

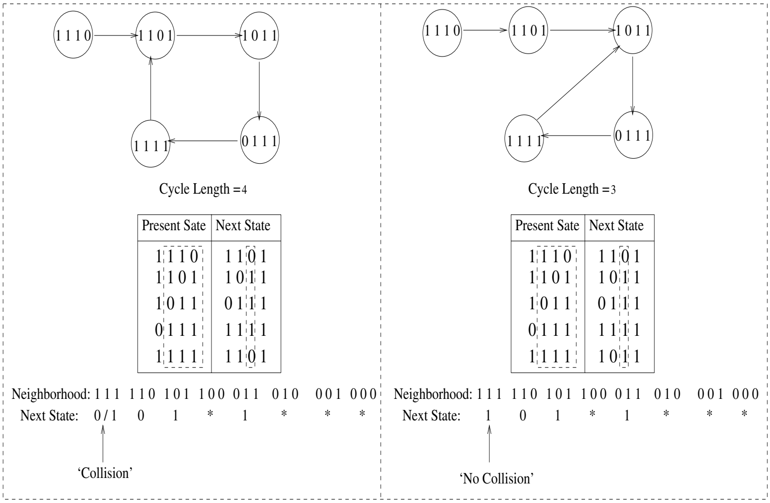
3.3 Simulated Annealing Program to Evolve Effective GMACA

Simulated annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems. The $GMACA$ synthesis scheme can be elegantly mapped to this Monte Carlo approach. The current state of a thermodynamic system is analogous to the current solution of synthesis scheme, while the energy level for the thermodynamic system is analogous to at the *Cost Function*. Finally, the ground state is analogous to the desired *GMACA* rule space. So, we employ and appropriately tune the *Simulated Annealing* to find out appropriate graphs to arrive at the desired *GMACA*.

Cost Function: To generate *GMACA* rules through *Phase I-III*, we define a **Cost Function** ($\mathcal{C}$) which is given by

$$\mathcal{C} = 1- \mid \frac{(n_0 - n_1)}{(n_0 + n_1)} \mid \tag{2}$$

where n_0 and n_1 are the occurrence of state '0' and '1' of a cell for a particular configuration respectively. If $n_0 \simeq n_1$, then the value of $\mathcal{C}$ is high; whereas if $n_0 >> n_1$ or $n_0 << n_1$, then $\mathcal{C}$ becomes zero and we get effective *GMACA* configurations.



Present Sate	Next State
1 1 1 0	1 1 0 1
1 1 0 1	1 0 1 1
1 0 1 1	0 1 1 1
0 1 1 1	1 1 1 1
1 1 1 1	1 1 0 1

Neighborhood:	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
Next State:	0 / 1	0	1	*	1	*	*	*

Present Sate	Next State
1 1 1 0	1 1 0 1
1 1 0 1	1 0 1 1
1 0 1 1	0 1 1 1
0 1 1 1	1 1 1 1
1 1 1 1	1 0 1 1

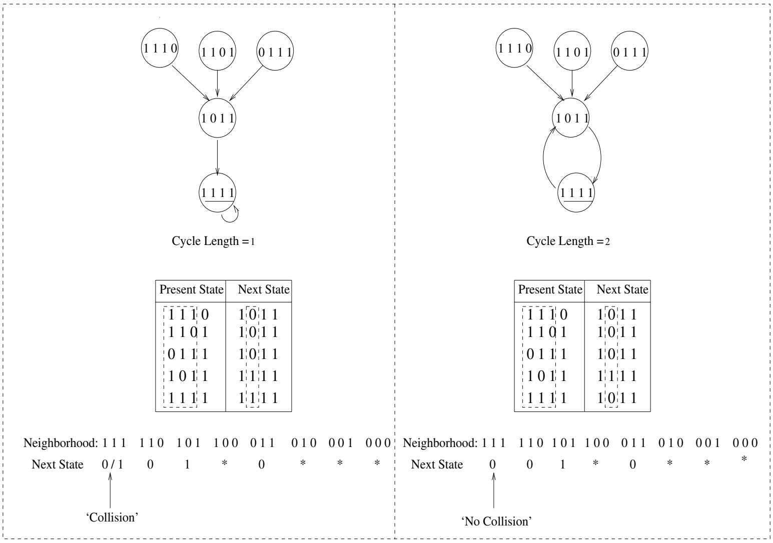
Neighborhood:	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
Next State:	1	0	1	*	1	*	*	*

Fig. 5. Example of Cycle Length Reduction of a Directed Graph

In Simulated Annealing an initial temperature ($TEMP_{Initial}$) is set. The temperature decreases exponentially during the process. At each temperature point ($Temp_{point}$) action is taken on the graphs based on the value of *Cost Function*. The entire process continues till temperature becomes zero.

Based on the value of *Cost Function*, the new set of graphs are generated. The entire process - generation of graphs, and its evaluation through the *Cost Function* - that is, *Phase I* to *III*, continues till temperature becomes zero. So, the emphasis is to arrive at graphs with low value of *Cost Function*. This demands low collision. Following two schemes are employed to reduce the collision on the randomly generated graph with attractor cycle length $l \leq L_{max}$ in *Phase*

I of synthesis scheme, where L_{max} denotes *maximum permissible length* of the attractor cycle.



Present State	Next State
1 1 1 0	1 0 1 1
1 1 0 1	1 0 1 1
0 1 1 1	1 0 1 1
1 0 1 1	1 1 1 1
1 1 1 1	1 1 1 1

Neighborhood:	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
Next State	0 / 1	0	1	*	0	*	*	*

Present State	Next State
1 1 1 0	1 0 1 1
1 1 0 1	1 0 1 1
0 1 1 1	1 0 1 1
1 0 1 1	1 1 1 1
1 1 1 1	1 0 1 1

Neighborhood:	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
Next State	0	0	1	*	0	*	*	*

Fig. 6. Example of Cycle Length Increment of a Directed Graph

Scheme 1 : Reduction of the cycle length of a given graph In this case, we reduce the attractor cycle length of a given graph. The *Fig.5* illustrates an example of this technique along with the state transition table and next state function. In *Fig.5*, when the cycle length of the graph is 4, there is a 'collision' between state '0' and '1' for '111' configuration of 3^{rd} cell; whereas when the cycle length is reduced from 4 to 3, there is no 'collision' for same configuration.

Scheme 2 : Increment of the cycle length of a given graph In this case, we increase the cycle length of a given graph. The *Fig.6* illustrates an example of this technique. When cycle length of the given graph is 1, there is a 'collision' between state '0' and '1' for '111' configuration of 2^{nd} cell; whereas when the cycle length is incremented from 1 to 2, the 'collision' disappears.

In *Schemes 1* and *2*, we change the state transition table by changing the cycle length of the given graphs. As a result, the collision between state '0' and '1' of a particular configurations of a cell is changed. Consequently, the *cost function* is also changed.

The cost value is evaluated from *Equation -2*. There are two types of solution based on cost value - Best Solution(BS) and Current Solution(CS). A New Solution(NS) at immediate next $Temp_{point}$ compares its cost value with CS. If NS has better cost value than CS, then NS becomes CS. The new solution(NS) is also compared with BS and if NS is better, then NS becomes BS. Even if NS is not as good as CS, NS is accepted with a probability. This step is done typically to avoid any local minima. The complete algorithm is presented below:

Algorithm 1 Evolving GMACA

```
Input : Pattern Size (n), Pattern Set to be learnt,
        Initial temperature ($Temp_{initial}$)
Output :   GMACA Rule.
$Temp_{point} = Temp_{initial}$
Initialize CS, BS as zero rules.
while $Temp_{point} > 0$
{
  if $Temp_{point} > 0.50 \times Temp_{initial}$
     Randomly generate graph as guess soln.
  else
     Generate graph by Scheme 1 or 2.
  Generate state transition table and rule table.
  NS = GMACA - Rule
  $\delta_{cost}$ = cost value(NS) - cost value(CS)
  if $\delta_{cost} < 0$
  {
     CS = NS
     if cost value(NS) < cost value(BS)
     {
        BS = NS
     }
  }
  else
     accept CS = NS with prob. $e^{-|\delta_{cost}|/Temp_{point}}$.
  Reduce $Temp_{point}$ exponentially.
}
```

The time required to find out an n-cell *GMACA* increases with the value of r_{max} which is the *maximum permissible noise*. The number of nodes in a graph p increases with r_{max}. So, we investigate the minimum value of r_{max} for which cost function attains minimum value close to zero.

3.4 Minimum Value of Maximum Permissible Noise (r_{max})

r_{max} specifies the noise level at the training/synthesis phase. To find out the minimum value of r_{max}, we carry out extensive experiments to evolve pattern recognizable n-cell *GMACA* for different values of n. The results are reported in *Table 1*.

Column I of *Table 1* represents noise present in training phase; whereas *Column II* represents the percentage of convergence for different noise value per bit in identification/recognition phase. The results of the *Table 1* clearly establish that in the training (synthesis) phase if we consider that the patterns are corrupted with single bit noise, then the percentage of convergence at the recognition is better irrespective of noise level. **So, the minimum value of r_{max} is set to 1 for which *GMACA* based associative memory performs better**. Then, *Equation 1* reduces to $p = 1 + n$.

Table 1. Computation of Minimum value of r_{max}

Training	Percentage of Convergence								
Noise	For $n = 20$, $k = 5$			For $n = 30$, $k = 7$			For $n = 45$, $k = 10$		
(r_{max})	$r = 1$	$r = 2$	$r = 3$	$r = 1$	$r = 2$	$r = 3$	$r = 1$	$r = 2$	$r = 3$
1	84.57	64.65	44.87	84.88	67.57	50.90	79.30	63.92	51.06
2	83.62	63.98	44.65	81.57	61.99	44.59	75.65	59.41	46.31
3	82.67	61.70	41.55	81.57	61.79	44.56	74.89	59.34	46.01

The reason for such a phenomenon can be explained as follows. *If we consider noise in more than one bit, then the total number of entry in state transition table is higher. As a result the Simulated Annealing program fails to arrive at the desired GMACA.*

3.5 Selection of Attractor Cycle Length

In the context of the observation noted in *Section 3.4*, we have analyzed the attractor cycle length of the evolved *GMACA*. We have observed that the length of the attractor cycle of the evolved *GMACA* is equal to one in majority of cases.

The reason for which attractor cycle length of *GMACA* is equal to one is as follows. *If the cycle length of attractor is equal to one, same neighborhood configurations of i^{th} cell of attractor map to same next state more times; in effect collision of state '0' and '1' gets reduced. As a result, the convergence rate of the Simulated Annealing program gets accelerated.*

4 Experimental Results and Performance Analysis

In this section, we report experimental results based on randomly generated data set for different values of n (number of bits in a pattern) and k (number of patterns to be learnt) to analyze the performance of *CAM*. The memorizing capacity, evolution time and identification/recognition complexity of the proposed model confirm that *GMACA* based *CAM* can be employed as an excellent pattern recognition machine.

4.1 Memorizing Capacity

For an associative memory to be useful, the stored patterns should be stable in the configuration space in the sense that if the network state is a stored pattern, the state should not change. Moreover, if the stored patterns are correlated their stability becomes less likely. The maximum capacity of a conventional Hopfield Network is roughly $0{\cdot}14n$ random patterns [12].

The experiments to evolve pattern recognizable n-cell *GMACA* for different values of n are carried out. For each n, 15 different sets of unbiased patterns to be trained are selected randomly. The number of patterns to be learned by the *CAM* is progressively increased.

Table 2. Comparison of Memorizing Capacity of *GMACA* and Hopfield Net

Size of	Memorizing Capacity of Network	
Pattern (n)	*GMACA*	Hopfield Net
10	4	2
20	5	3
30	7	5
40	10	6
50	12	8
60	13	9
70	15	11
80	18	12
90	20	14
100	23	15

Table 3. Evolution time for Synthesize GMACA

Size of Pattern (n)	No of Patterns (k)	Initial Temp (T)	Evolution Time (min)
10	4	15	0.43
20	5	15	1.06
30	7	15	1.55
40	10	20	3.01
50	12	25	3.35
60	13	25	4.52
70	15	30	7.03
80	18	35	7.45
90	20	30	9.21
100	23	40	15.08

The *Table 2* demonstrates the potential of *CAM* as an associative memory model. *Column II* of *Table 2* depicts the maximum number of patterns that an n-cell *GMACA* can memorize. The results of Hopfield Net on the same data set are provided in *Column III* for the sake of comparison. The experimental result clearly indicates that :

(i) **the memorizing capacity of *GMACA* is found to be more than 20% of its lattice size**; and

(ii) **it is superior to conventional Hopfield Net by 33%**.

4.2 Evolution Time

The *Table 3* represents the evolution time to evolve *GMACA* by *Simulated Annealing*. *Column I* and *II* of *Table 3* represent different *CA* size (n) and number of attractors (k) respectively; while the *Column III* depicts the *Initial Temperature* required to find out the best possible *GMACA* configuration by *SA*. In *Column IV*, we provide the evolution time required to synthesize *GMACA*. Growth of time, as the results indicate, is super linear in nature.

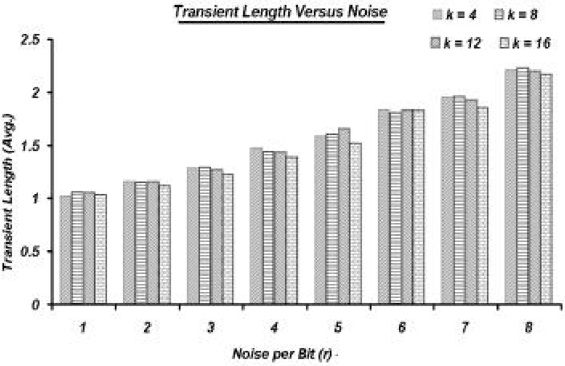


Fig. 7. Graph of Transient Length versus Noise for $n = 50$

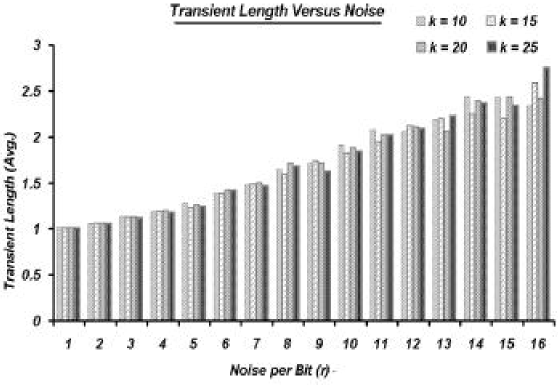


Fig. 8. Graph of Transient Length versus Noise for $n = 100$

4.3 Identification/Recognition Complexity

The *Fig.7* and *8* represent time taken to recognize a noisy pattern for $n = 50$ and $n = 100$ respectively. All the results reported in *Fig.7* and *8* ensure that the time taken to recognize a noisy pattern is independent of k. Also, it does not depend on the size of the patterns learnt (n). It depends on the transient length of the *CA*, which is constant. **Hence, the cost of computation for the entire recognition/identification process is constant.**

5 Conclusion

This paper reports the pattern recognizable capability of *Cellular Automata Machine* (*CAM*). Extensive experimental results reported in this paper confirm that the sparse network of *CAM* out-performs the conventional Hopfield Net.

References

1. J. J. Hopfield, *"Pattern Recognition computation using action potential timings for stimulus representations",* Nature, 376: 33-36; 1995.
2. M. Chady and R. Poli, *"Evolution of Cellular-automaton-based Associative Memories",* Technical Report no. CSRP-97-15, May 1997.
3. J. H. Moore, L. W. Hahn, *"Multilocus pattern recognition using one-dimensional cellular automata and parallel genetic algorithms",* Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001).
4. R. Raghavan, *"Cellular Automata In Pattern Recognition",* Information Science, 70: 145-177; 1993.
5. K. Morita, S. Ueno, *"Parallel generation and parsing of array languages using reversible cellular automata",* Int. J. Pattern Recognition and Artificial Intelligence, 8: 543–561; 1994.
6. E. Jen, *"Invariant strings and Pattern Recognizing properties of 1D CA",* Journal of statistical physics, 43, 1986.
7. S. Wolfram, *"Theory and application of Cellular Automata",* World Scientific, 1986.
8. P. Pal Chaudhuri, D Roy Chowdhury, S. Nandi and S. Chatterjee, *"Additive Cellular Automata, Theory and Applications, VOL. 1",* IEEE Computer Society Press, Los Alamitos, California.
9. N. Ganguly, A. Das, P. Maji, B. K. Sikdar, and P. Pal Chaudhuri, *"Evolving Cellular Automata Based Associative Memory For Pattern Recognition",* International Conference on High Performance Computing, Hyderabad, India, 2001.
10. N Ganguly, P Maji, A Das, B K Sikdar, P Pal Chaudhuri, *"Characterization of Non-Linear Cellular Automata Model for Pattern Recognition",* 2002 AFSS International Conference on Fuzzy Systems, Calcutta, India, 2002.
11. P Maji, N Ganguly, A Das, B K Sikdar, P Pal Chaudhuri, *"Study of Non-Linear Cellular Automata for Pattern Recognition",* Cellular Automata Conference, Yokohama National University, Japan, 2001.
12. J. Hertz, A. Krogh and R. G. Palmer, *"Introduction to the theory of Neural computation",* Santa Fe institute studies in the sciences of complexity, Addison Wesley, 1991.

Cellular Automata Model of Drug Therapy for HIV Infection

Peter Sloot*[1], Fan Chen[1], and Charles Boucher[2]

[1] Faculty of Sciences, Section Computational Science, University of Amsterdam
The Netherlands. {sloot, fanchen}@science.uva.nl
[2] Department of Virology, University Hospital Utrecht, Utrecht University
The Netherlands. C.Boucher@azu.nl

Abstract. In this study, we employ non-uniform Cellular Automata (CA) to simulate drug treatment of HIV infection, where each computational domain may contain different CA rules, in contrast to normal uniform CA models. Ordinary (or partial) differential equation models are insufficient to describe the two extreme time scales involved in HIV infection (days and decades), as well as the implicit spatial heterogeneity [4,3, 10]. R.M.Zorzenon dos Santose [13] (2001) reported a cellular automata approach to simulate three-phase patterns of human immunodeficiency virus (HIV) infection consisting of primary response, clinical latency and onset of acquired immunodeficiency syndrome (AIDS), Here we report a related model. We developed a non-uniform CA model to study the dynamics of drug therapy of HIV infection, which simulates four- phases (acute, chronic, drug treatment responds and onset of AIDS). Our results indicate that both simulations (with and without treatments) evolve to the relatively same steady state (characteristic of Wolfram's class II behaviour). Three different drug therapies (mono-therapy, combined drug therapy and highly active antiretroviral therapy HAART) can also be simulated in our model. Our model for prediction of the temporal behaviour of the immune system to drug therapy qualitatively corresponds to clinical data.

1 Introduction

1.1 Biological Background of HIV Infection

The infection of human immunodeficiency virus (HIV), causing AIDS (acquired immunodeficiency syndrome), is almost invariably a progressive, lethal disease with insidious time course. Currently, clinicians identified two common laboratory markers for detection of disease progression (the amount of virus (HIV-RNA) and the number of T helper cells (CD4 T cells) in blood. Immune response for typical virus infection varies from days to weeks, but HIV infection typically follows a three-phase pattern (See also Fig. 1).

* contact author

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 282–293, 2002.

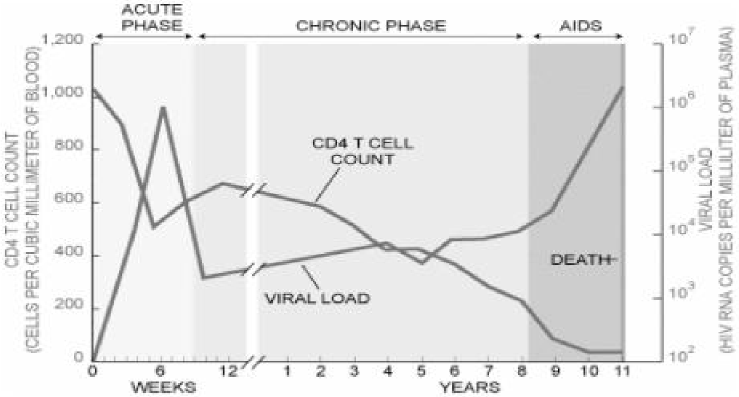


Fig. 1. Common pattern of HIV infection in a typical untreated patient indicates a three-phase evolution. The two lines represent CD4 T cell count and viral load respectively. (Image: Bryan Christie, July 1998 Scientific American)

- During a few weeks (varying from two to six weeks), a transient and dramatic jump of plasma virion level is present with a marked decrease of immune cell count (CD4 T helper cells), following by a sharp decline.
- In the subsequent chronic phase (varying from one to ten or more years, on average eight to ten years), the immune system partially eliminates the HIV virus and the rate of viral production reaches a lower, but relatively steady, state that varies greatly from patient to patient. Their apparent good health continues because CD4 T cell levels remain high enough to preserve defensive responses to other pathogens. But over time, CD4 T cell concentrations gradually fall.
- An outbreak of the virus (varying from one to two years), together with constitutional symptoms and onslaught by opportunistic diseases, cause death [2].

1.2 Biological Background of Drug Therapy of HIV Infection

A vaccine would certainly be ideal for preventing infection by HIV and thus for avoiding AIDS when immunity is severely impaired. The near-term prospects for a vaccine are poor due to error occurrence during each transcription of HIV. Therefore, for the immediate future, many scientists are concentrating on improving the therapy.

- Currently, there are fifteen drugs licensed for treatment of individuals infected with HIV. These drugs belong to two classes, one inhibiting the viral enzyme reverse transcriptase and the other inhibiting the viral protease. These drugs are used in combination therapy to maximally inhibit viral replication and decrease HIV-RNA to below levels of detection levels (currently defined as below 50 copies per ml) in blood. In one class, the nucleoside analogues resemble the natural substances that become building blocks of HIV-DNA; and when reverse transcriptase tries to add the drugs to a developing strand of HIV-DNA, the drugs prevent completion of the strand. The other agent in this class, non-nucleoside reverse transcriptase inhibitors, composed of other kinds of substances, constitute the second class of anti-retrovirals. The other class, the protease inhibitors, blocks the active, catalytic site of the HIV protease, thereby preventing it from cleaving newly made HIV proteins.

- HIV therapy is classified into three classes: mono-therapy, combined therapy and triple therapy. Mono-therapy (such as based on reverse transcriptase inhibitor) or combined drug therapy (reverse transcriptase and protease inhibitors) are considered to suppress the viral multiplication. Because of incompletely blocking the replication pathway and occasionally creation of a resistant virus strain, the CD4 T counts will come back to the pre-treatment baseline within many weeks (Fig. 2). The problem of drug resistance in the treatment has become an increasing significant barrier in the effectiveness of AIDS immune-therapy.
- Currently, there is no single class of drug that can completely prevent HIV from replicating. Treatment with drug combinations is in only 50% of the cases successful in inhibiting viral replication to undetectable levels. In the remaining 50% of cases viruses can be detected with a reduced sensitivity to one or more drugs from the patients regimen. Theory and clinical trials indicate that the best way to achieve maximum viral suppression is through highly active anti-retroviral therapy (HAART), which consists of triple therapy including two nucleoside analogues and a protease inhibitor.

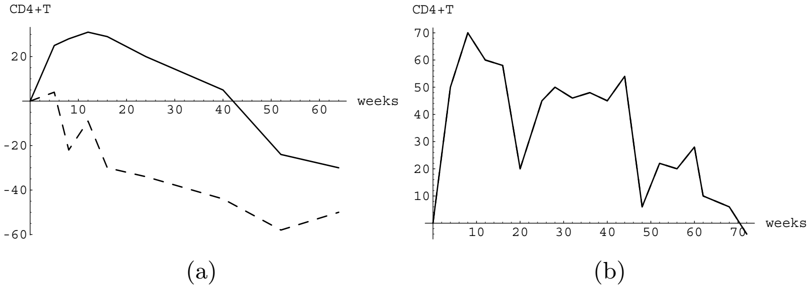


Fig. 2. Clinical data for mono-therapy (CD4 + T count is compared with baseline): (a) This study administered patients either with a placebo control (hash line) or AZT (solid line) for 62 weeks. The treatment started when CD4 T counts were between 200 and 500/ml [1]. (b) The results indicated that the effects of mono-therapy AZT on non-progressors can not be sustained above base line for more than 70 weeks [9].

1.3 Uniform and Non-uniform CA

Cellular automata (CA) provide us with a means to model complex dynamical phenomena by reformulating the macroscopic behaviour into microscopic and mesoscopic rules that are discrete in space and time. The states of the discrete elements (cells) organised in a regular grid, are updated synchronously according to a uniform local interaction rule [8]. Uniform CAs have three notable features: massive parallelism, locality of cellular interactions and simplicity of cells (finite state machines). Non-uniform CAs have first been investigated by Vichniac [11] who discussed a one-dimension CA in which a cell probabilistically selects one of two rules, at each time step. In this study, we use non-uniform CAs to explore the huge space of this complex system. The three essential CA features are preserved in this non-uniform model.

1.4 Computational Task and Related Work

Modelling the population dynamics of cells in immune response relevant to HIV has recently attracted a considerable interest [2,3,5,6,7,12]. Currently, the only two ways to model this dynamics of the immune response with respect to the pathology and therapy of HIV infection are analytic PDE and ODE models and cellular automata models. Analytical approaches are successful to describe different aspects of HIV infection dynamics [4,3,6]. But they have strong limitations to describe the two time scales observed in the time course of infection in term of weeks and years and have serious difficulties in exploiting spatial information. Cellular automata are recently regarded as a good strategy to model this spatial-temporal dynamics with emphasis on local interactions. Mielke et al., developed a fuzzy interaction model for mutating HIV with a fuzzy set of 10 interactions for macrophages, helper cells, cytotoxic cells and virion [5]. Hershberg et al., (2001) indicated, using a microscopic simulation, that the time course of AIDS is determined by the interactions of the virus and the immune cells in the shape space of antigens, and that it is the virus's ability to move more rapidly in this space (its high mutability) which cause the time course and eventual "victory" of the disease [2]. This model clearly showed the three stages of the disease. A simple set of CA rules was used to model the evolution of HIV infection by Zorenon dos Santos et al. (2001). The three phase patterns were also presented and the results indicated that the infected cells organise themselves into spatial structures, which are responsible for the decrease in the concentration of uninfected cells, leading to AIDS [13]. This CA model inspires the drug therapy simulation presented here. In this model we can investigate the HIV infection dynamics with therapy using microscopic simulations. The main ingredients in our model are destruction of previously emerged spatial patterns (wave-like and solid-like structures) and reconstruction of new spatial patterns (wave-like structures) due to incorporation of the drug therapy concept. In the sequel we will refer to the Zorenon dos Santos's model as the HIV Infection Model (HI model) and our model as the Drug Therapy of HIV Infection Model (DTHI model).

2 The Computational Model

Here we illustrate the basic concepts of DTHI. The first subsection introduces shortly the context of the problem, including the models of ODE/ PDE. The second section presents the CA rules and corresponding description for mimicked biological concepts in the HI model as well as the correctness for one of rules. The third section defines our DTHI model.

2.1 The Problem of Modelling Drug Therapy

With continuous progress of medical and biological research, three kinds of therapy have been gradually developed. The long-term survival with combined drug therapy is considered to be longer than with mono-therapy. The appearance of resistance virus against HAART is apparently much longer than with combined drug therapy. Up to now, the extension of long-term survival with HAART is not yet known. Mathematical ODE/PDE models have difficulties to simulate the four phases (acute, chronic, drug treatment responds and AIDS) in one model but have also difficulties to unify three therapies into one model. Because these models could not describe two kinds of time scales (weeks in primary response and years in the clinical latency and AIDS) which might be related to two kinds of interactions: one local and fast, and the other long-ranged and slow [13].

2.2 The HI Model

Here we first review the rules and biological descriptions of HIV infection model (HI) with Moore neighbourhood and periodic boundary from reference [13].

[Rule 1] Update of a healthy cell.

(a) If it has at least one infected-A1 neighbour, it becomes an infected-A1 cell.

- The spread of the HIV infection by contact before the immune system had developed its specific response against the virus.

(b) If it has no infected-A1 neighbour but does have at least R $(2 < R < 8)$ infected-A2 neighbours, it becomes infected-A1.

- Before dying, infected-A2 cells may contaminate a healthy cell if their concentration is above some threshold.

(c) Otherwise it stays healthy.

[Rule 2] An infected-A1 cell becomes an infected-A2 cell after τ time steps.

- An infected cell is the one against which the immune response has developed a response hence its ability to spread the infection is reduced. The τ represents the time required for the immune systems to develop a specific response to kill an infected cell. A time delay is requested for each cell because each new infected cell carry a different lineage (strain) of the virus. This is the way to incorporate the mutation rate of the virus in this model. On the average, one mutation is produced in one generation due to the error occurrence during HIV transcription. Assume that mutation in each trial is varied in this model due to the stochastic characteristics.

[Rule 3] Infected-A2 cells become dead cells.

- The depletion of the infected cells by the immune response.

[Rule 4]

(a) Dead cells can be replaced by healthy cells with probability p_{repl} in the next step ($p_{repl} = 99\%$) or remain dead with probability 1 - p_{repl}.

- The replenishment of the depleted cells mimics the high ability of the immune system to recover from the immuno-suppression generated by infection. As a consequence, it will also mimic some diffusion of the cells in the tissue.

(b) Each new healthy cell introduced, may be replaced by an infected-A1 cell with probability p_{infec} ($p_{infec} = 10^{-5}$).

- The introduction of new infected cells in the system, either coming from other compartments of the immune system or from the activation of the latent infected cells.

2.3 The DTHI Model

Based on the model summarized above, we incorporate the drug therapy process into the CA model for drug therapy. All approved anti-HIV, or anti-retroviral, drugs attempt to block viral replication within cells by inhibiting either reverse transcriptase or the HIV protease. In addition to the ‘delayed’ infection modelled in Rule 1b and the latent infection in Rule 4b, the main source of HIV infection in the HI model is Rule 1a. We limit the range of HIV infection (infected A1 cells) by giving a rank level N ($0 \leq N \leq 7$). This mimics the principle that the drug prevents the virus from replication, resulting in less efficient infection. N represents the effectiveness of each drug. The bigger N, the less efficient the drug. Different drug therapies are modelled by different response functions P_{resp} over the time. P_{resp} represents the response function for each drug therapy, which have effects on the infected A1 cells after the starting of a drug therapy. This models the fact that the drug therapy will not immediately influence all of infected A1, but rather it will affect part of them at each time step. Over time these effects of drug therapy can (and will) decay. At the same time, this also mimics the concept of drug resistant virus strains.

[Modified Rule 1 (a)] Update of a healthy cell:
If there is one A1 neighbour during the time of drug therapy, N ($0 \leq N \leq 7$) neighbour healthy cells become infected-A1 in the next time steps with probability p_{resp}. Otherwise, all of eight neighbours become infected-A1 cells. N is related to effectiveness of each drug. Non-uniform CA rules is used when the therapy starts. At the time step t_c during the therapy, $p_{resp}(t_c)$ infected-A1 cells have rank N and $1 - p_{resp}(t_c)$ infected-A1 cells have the max rank eight.

[Modified Rule 3] We propose to adapt Rule 3 by adding ‘In The Next Time Step’. This mimics the fact that infected-A2 cell will also be present in the lymph-node for a short time but with less infection ability, compared to infected-A1 cells.

[The rest of rules] don’t change.

3 Results

3.1 DTHI Model Simulations

To repeat the HI model, we use the same parameters as reference [13], using a Moore neighbourhood, periodic boundary conditions on a lattice of 700 sites, initial infected A1 cells (with $P_{HIV} = 0.05$), $\tau = 4$ and $R = 4$. Because the delay parameter τ may vary from 2 to 6 weeks, and the number of infected-A2 neighbours vary from 3 to 7 due to some of threshold. In Fig. 3, we show the densities of healthy, infected (A1 and A2) and dead cells using the modified Rule 3 with the other rules the same as in the HI paper. The results are averaged over 500 simulations. The variance which do not show in this paper is consistent with Fig. 2 in reference [13]. The reason to do this modification of the Rule 3 comes from two facts. Analytically, infected A2 cells will be only present transiently in the lattice during execution of a set of rules if we don't modify Rule 3. Infected A1 cells, however do have an opportunity to be present because of its related τ time steps. Moreover, there is hardly opportunity for Rule 1b to contribute. As a consequence the evolution speed with which the cells enter the AIDS phase is too fast, without correction and Rule 1b has no chance to be activated. Our simulations show that the density of the dead cells in Fig. 3 are shifted one time step with respected to the results shown in reference [13]. This results in a required conservation of densities of healthy, infected and dead cells, normalized to unity in each time step.

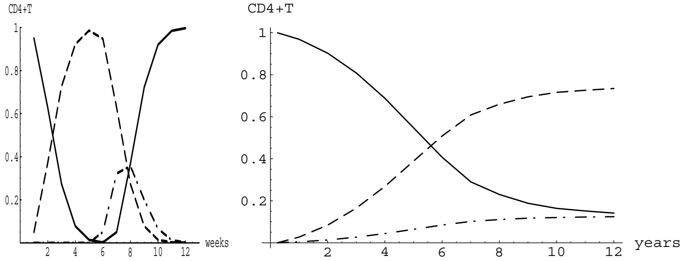


Fig. 3. Three-phase dynamics with two time scales (weeks and years) were obtained which were primary response (within 12 weeks), clinical latency (within 10 years) and AIDS (steady state after 10 years). The solid, hash and hash-dot lines represent healthy, infected (A1 and A2) and dead cells respectively. The density of dead cells shifted one time step, compared with the result of HI model. The evolution speed of cells with corrected rule was consistent with the result of HI model, but was slower than the result without correction.

3.2 Introduction of Drug Therapy into the DTHI Model

In the previous sections we introduced new rules to model the response of the system to drug therapy. In figure 4 four different time scales (weeks and years) can be observed. The data shown were averaged over 500 simulations. The first acute phase indicates the fast proliferation of the original HIV strains before the actual immune system response. This phase ends when specific immune response occurs for these strains. The next phase, the chronic phase that takes years, is the phase where the viral load increases slowly and CD4 counts decrease slowly. When CD4 T counts drop to a certain level (normal 200 to 500 counts per ml), the drug therapy is started. In this phase, virus replication is blocked and CD4 T counts increase. Once resistant strains against the drugs evolve, the last phase of the disease occurs disrupting the whole immune system.

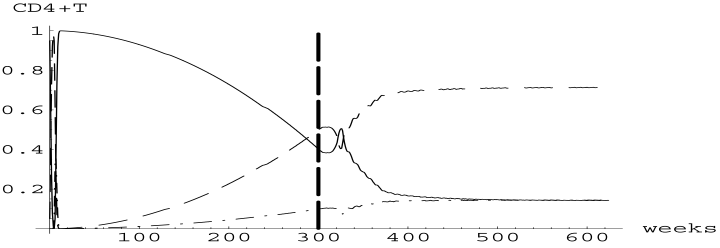


Fig. 4. Four-phase dynamics with two time scales (weeks and years) were obtained, which were qualitatively comparable with clinical data. The solid, hash and hash-dot lines represent healthy, infected (A1 and A2) and dead cells respectively. The vertical hash line indicates the starting point of the therapy. The profile indicated that after primary response, the CD4 T cells decreased gradually in the latency period. Once the therapy started, CD4 T cell count increased due to the drug therapy. Finally they evolved into AIDS state due to the resistance against drug.

Our simulation results indicate that the extension of long-term survival is dependent on the drug effectiveness (N) and the response function (P_{resp}). The high quality of the drug (modelled by small N) efficiently prevents the virus from replication, and thus few resistant new viruses are generated. As a consequence, a relatively prolonged long-term survival is obtained such as shown for $N = 0$ in Fig. 5. We can also simulate HAART treatment by selecting a suitable P_{resp} response function in our model such as $P_{resp}2$ and $P_{resp}3$ in Fig. 6.

In our simulation model, we get insights into local behaviour and spatial structures. Two typical spatial structures wave-like (left structure) and solid-like structures (right structure) are shown in Fig. 7a. The solid-like structures spread in all directions and wave-like structures generate a propagating front wave with width τ+1." After the therapy started, original spatial structures disappeared

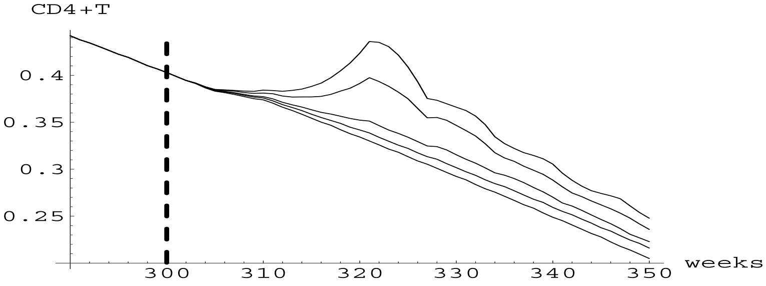


Fig. 5. Different drug effectiveness with the same response function (P_{resp}) was observed, which were $N = 0$, $N = 1$, $N = 4$, $N = 7$ and no treatment locating from the top to the bottom in the figure respectively. The results here qualitatively simulated mono-therapy and combined therapy.

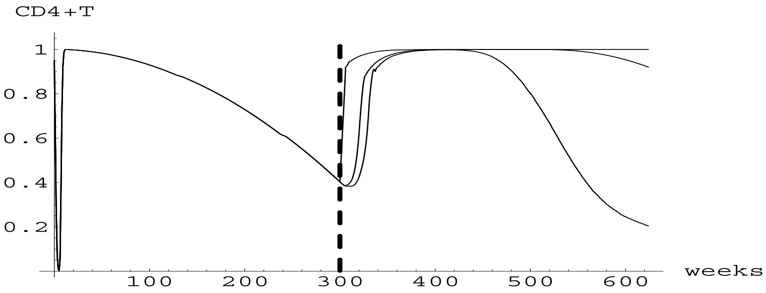
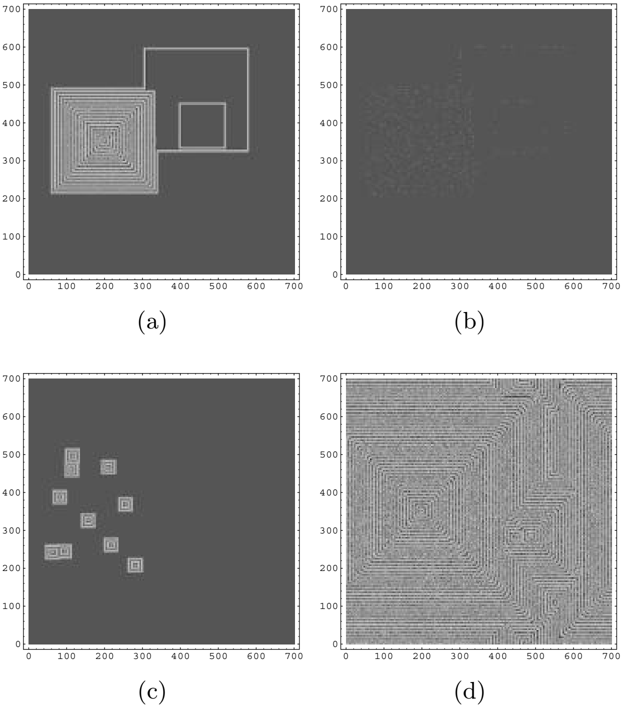


Fig. 6. Different response functions (P_{resp}) with the same drug effectiveness ($N = 0$) were obtained, which were P_{resp1},P_{resp2} and P_{resp3} locating from the top to the bottom of the figure respectively. $P_{resp}1$ simulated the completely recovering therapy. $P_{resp}2$ and $P_{resp}3$ simulated the current HAART therapy. Vertical hash line represents the starting of the therapy.

due to the limitation of infection, and only new resistant virus against drug were left in the lattice (Fig. 7b). They developed and formed new virus sources wave-like structures (Fig. 7c). Eventually, they evolved into AIDS phase. Only wave-structures were left and covered the whole lattice. The steady states was reached, in which the concentrations of each cell were kept relatively fixed and patterns of structures were unchanged (Wolframs class II behaviour) (Fig. 7d).



(a) (b)

(c) (d)

Fig. 7. Four snapshots of the whole lattice (700 X 700) at different time steps in the trial of Fig. 5 ($P_{resp}3$), simulated by the DTHI model. The dark grey, light grey, white and black represent healthy, infected A1, infected A2 and dead cells respectively. Figures (a) to (d) indicated time steps 300, 400, 500 and 600 respectively. The treatment starts at 300 time steps.

4 Discussions and Conclusions

The main success of the present model is the adequate modelling of the four-phases of HIV infection with different time scales into one model. Moreover, we could also integrate all of the three different therapy procedures into one model. The simulations show a qualitative correspondence to clinical data. During the phase of drug therapy response, temporal fluctuations for $N > 3$ were observed, this is due to the relative simple form of the response distribution function (P_{resp}) applied to the drug effectiveness parameter N at each time-step. Our simulation results indicate that, in contrast to ODE/PDE, our model supports a more flexible approach to mimic different therapies through the use of mapping the parameter space of P_{resp} to clinical data. P_{resp} is different functions of time step, corresponding to different therapies. In this paper, we employ different constant P_{resp} over time step for mono-/combined therapy and linear P_{resp} over time step for HAART therapy. Therefore there is ample room to incorporate biologically more relevant response functions into the model. This future work requires in depth investigation of the parameter space of P_{resp}.

The results from Fig. 3 with respect to the amount of CD4 T counts are not completely supported by clinical data. The number of CD4 T cells should completely go down at least to a level that is undetectable. We will investigate the influence of P_{resp} on the steady state in our model. The chosen value of P_{HIV} in this paper ($P_{HIV} = 0.05$) is too large with respect to data known for clinic. A more realistic value would be 1 infected cell per 10^2 to 10^3 cells, resulting in $P_{HIV} = 0.005$. This effect will also be investigated. Clinical data indicate a increased sensitivity of T-cells over time, probably due to activation of the immune system. This will be modelled by making P_{infec} a function of the number of infected cells. Finally, it is known that in the early stages of infection, virus replication is confined to monocytic white blood cells. Only in later stages, CD4 T cells will become the new target cells. This trophysm effect will be studied in the future.

Acknowledgement. The authors would like to thank the anonymous referees for many valuable comments and suggestions that have improved the quality of this paper. We also like to thanks suggestions of Roeland Merks.

References

1. M.A. Fischl, D.D. Richman, Hansen. N., A.C. Collier, J.T. Carey, M.F. Para, W.D. Hardy, R. Dolin, W.G. Powderly, J.D. Allan, and et al. The safety and efficacy of azt in the treatment of subjects with mildly symptomatic HIV type 1. *Annals Int. Med.*, pages 727–737, 1990.
2. U. Hershberg, Y. Louzoun, H. Atlan, and S. Solomon. HIV time hierarchy: Winning the war while, loosing all the battles. *Physica.*, pages 178–190, 2001.
3. D. E. Kirschner and G. F. Webb. A mathematical model of combined drug therapy of HIV infection. *J. Theoret. Med.*, pages 25–34, 1997.
4. D. E. Kirschner and G. F. Webb. Understanding drug resistance for mono-therapy treatment of HIV infection. *Bull. Math. Biol.*, pages 763–185, 1997.
5. A. Mielke and R. B. Pandey. A computer simulation study of cell population in a fuzzy interaction model for mutating HIV. *Physica A*, 251:430–438, 1998.
6. A. S. Perelson. Modelling the interaction of the immune system with HIV. *In: C. Caastillo-Chavez (ed): Mathematical and Statistical Approaches to AIDS Epidermiology. Lecture Notes in Biomathematics, Springer-Verlag*, 83:350–370, 1989.
7. N. Stilianakis, C.A.B. Boucher, M.D. De Jong, R. Van Leeuwen, R. Schuurman, and R.J. De Boer. Clinical data sets of HIV-1 reverse transcriptase-resistant mutants explained by a mathematical model. *J. of Virol.*, pages 161–168, 1997.
8. T. Toffoli and N. Margolus. *Cellular Automata Machines.* Cambridge, Massachusetts: The MIT Press, 1987.
9. S. Vella, M. Giuliano, L.G. Dally, M.G. Agresti, C. Tomino, M. Floridia, A. Chiesi, V. Fragola, M. Moroni, M. Piazza, and et al. Long-term follow-up of zidovudine therapy in asymptomatic HIV infection: results of a multicenter cohort study. *J. AIDS.*, pages 31–38, 1994.
10. D. Verotta and F. Schaedeli. Non-linear dynamics models characterising long- term virological data from aids clinical trials. *Math. Biosci.*, pages 1–21, 2002.

11. G. Y. Vichniac, P. Tamayo, and H. Hartman. Annealed and quenched inhomogeneous cellular automata. *J. Statistical Phys.*, 45:875–883, 1986.
12. D. Wodarz, K. M. Page, R. A. Arnout, A. R. Thomsen, J. D. Lifson, and M. A. Nowak. A new theory of cytotoxic t-lymphocyte memory: implications for hiv treatment. *Philos. Trans. R. Soc. Lond. (B Biol. Sci.)*, 355(1395):329–343, 2000.
13. R. M. Zorzenon dos Santos and S. Coutinho. Dynamics of HIV infection: A cellular automata approach. *Phys. Rev. Lett.*, 87(16):168102–1–4, 2001.

Cellular Automata Approaches to Enzymatic Reaction Networks

Jörg R. Weimar

Institute of Scientific Computing, Technical University Braunschweig,
D-38092 Braunschweig, Germany
J.Weimar@tu-bs.de, http://www.jweimar.de

Abstract. Cellular automata simulations for enzymatic reaction networks differ from other models for reaction-diffusion systems, since enzymes and metabolites have very different properties. This paper presents a model where each lattice site can can contain at most one enzyme molecule, but many metabolite molecules. The rules are constructed to conform to the Michaelis-Menten kinetics by modeling the underlying mechanism of enzymatic conversion. Different possible approaches to rule construction are presented and analyzed, and simulations are shown for single reactions and simple enzyme networks.

1 Enzymatic Reaction Networks

Most reactions in biological systems are catalyzed by enzymes. These enzymes are complex molecules (they are proteins) which are not consumed in the reaction, but simply facilitate the reaction of smaller molecules called metabolites (such as sugar). In a biological cell, thousands of different enzymes are active. Each enzymatic reaction takes molecules of one or more metabolite species, the substrates of this reaction, and converts them into molecules of one or more other species, the products. The products are again substrates to other reactions, and thus the reactions form complex networks. Enzymatic reactions can be described and modeled on different levels of detail:

Static Interaction Networks: A first level is the static interaction network, such as the pathways collected in the KEGG database [9,10,11], or in the Boehringer-Mannheim map [13]. Such interaction networks can be constructed from purely qualitative information without relying on any quantitative information, such as reaction rates (except possibly for stoichiometric coefficients). Some analyses can extract additional information from these networks, such as inferring elementary metabolic flux modes [14,15] (although these rely on the information whether a given reaction is reversible or not, which in turn is a semi-quantitative information on the order of magnitude of the equilibrium constant). Simply on the basis of such interaction networks, no quantitative time-dependent simulation is possible.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 294–303, 2002.

Reactive Networks: Once quantitative data is available, reaction rates for reactions such as the conversion of a substrate S to a product P by an enzyme E can be described by some rate law, usually in the form of a Michaelis-Menten (MM) law:

$$\frac{d[S]}{dt} = \frac{V_{\max}[E][S]}{K_m + [S]} \tag{1}$$

with the maximum conversion rate $V_{\max}$ and the Michaelis-Menten coefficient K_m.

A more detailed description would contain rates for the elementary reactions leading to such an overall MM rate [1]:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_2}{\rightarrow} E + P. \tag{2}$$

Here, several rate constants need to be measured, which is usually not done or not possible. Most reactions in a realistic reaction network are more complicated, since they involve two or more substrates and two or more products, such as energy-providing ATP. In these cases, more coefficients need to be specified, and in addition the mechanism must be specified, such as BiBiRandom (in which case the two substrates may bind in any order) or Ordered BiBi (where the substrates must bind in a specific order). Other effects, such as competitive inhibition complicate the situation further.

Given a network of enzymatic reactions and the corresponding reaction rates, one can simulate the network by solving the ODEs numerically, or using stochastic simulation methods [7,8]. A static approach is to analyze steady states, parameter dependences or sensitivities, etc..

Space and Transport Phenomena. The space and diffusion or other transport phenomena are usually only taken into account by compartmentalizing the system where necessary, but they can also be included explicitly (as reaction-diffusion equations, or probabilistically in the stochastic simulation approach [6,17]). The most detailed simulation would be a full molecular dynamics simulation of the cell, but this is by far not feasible yet.

In this paper I present an approach based on cellular automata for simulating enzymatic reaction networks including diffusive transport.

2 Scales

To establish the conditions for a simulation model, let us first look at the scales of space, time, diffusion coefficients, and molecule concentrations and numbers. First, concentrations and counts: The volume of a typical procaryotic cell is $10^{-15}l$, the typical diameter is $1\mu m = 10000\mathring{A}$. The diameter of a typical enzyme is $100\mathring{A}$, while the diameter of a typical metabolite is $6\mathring{A}$ and of one atom $2\mathring{A}$. The concentrations of the involved molecular species differ widely. For metabolites they can be between $1\mu M$ and $5mM$, while for enzymes they can be in the range of $10nM \ldots 1\mu M$. This means that a complete cell with volume $10^{-18}m^3$, we find between 1 and 60000 molecules of a given enzyme species, and between $10^5 and 10^9$ molecules of a given metabolite species. Simply from the size of the enzyme, we can calculate that in a space volume of size $(100\text{Å})^3$, there can be O(1) enzyme molecules, but many metabolite molecules. In this paper I consider a simulation with lattice sites which have approximately this size and can contain at most one enzyme molecule of any species (exclusion principle) and an arbitrary number of metabolite molecules. Since we use integers to count the molecules instead of calculating with concentrations, our results will be more similar to the stochastic models than to the PDE models.

The diffusion coefficient for a typical metabolite is around $10^{-9}\frac{m^2}{s}$, for an enzyme that is not bound to a membrane it varies, but is at least two orders of magnitude smaller. The time scales of events inside a cell vary greatly, from $10^{-12}s$ (dissociation events) to $100s$ (fastest cell division). In the cellular automaton model, our diffusion method dictates a connection between time and space scales and diffusion coefficients. In our case, a lattice spacing of 100 Å and typical metabolite diffusion coefficients imply a time step of around $2.5 * 10^{-8}s$, which means that millions of time steps are necessary to simulate dynamical changes of metabolite concentrations.

3 Modeling Enzymes

In our model, each lattice site (we do not use the term cell as usually used in cellular automata to avoid confusion with the biological cell) can contain at most one enzyme molecule but many metabolites. We first consider the options for modeling the enzymatic reaction in one such site.

We show in section 3.1 that directly using the Michealis-Menten rate law is not possible in the discretized setting of cellular automata, then demonstrate in section 3.2 how to obtain correct results by directly simulating the mechanism that was approximated by the Michealis-Menten rate law.

3.1 Michaelis-Menten Rate Law

The first possibility is to directly use a Michaelis-Menten rate law. The rate law in Eq. (1) contains the concentration of the enzyme $[E]$, which in one site is always zero or one. If we assume that the concentrations of metabolites can be simply calculated from the metabolite numbers present in the cell, we have the following cellular automata rule: In time Δt, the probability of converting one molecule S to a molecule P is zero if there is no enzyme E present, otherwise

$$Q_{S\to P}(S) = \frac{\Delta t}{\alpha}\, V_{\max}\frac{\alpha S}{K_m + \alpha S}, \tag{3}$$

where α is a scaling factor to convert molecule counts per lattice site into concentrations. Here we will show that in our setting this method does not give correct results. The formula Eq. 3 is only correct in the limit $\Delta t \to 0$, so that at most one reaction takes place in one time step. On the other hand, in this limit the Michaelis-Menten simplification is not valid, but should be replaced by the more detailed description in Eq. (2). Another problem appears if the counts S are small (e.g., because concentrations $[S]$ are low or the volume of one lattice size is small). In the case where S is rarely greater than one, which happens when $K_m \ll \alpha$, Eq. (3) reduces to three cases for the probability $Q_{S \to P}$:

$$\begin{aligned} Q_{S\to P}(0) &= 0; \\ Q_{S\to P}(1) &= \frac{\Delta t}{\alpha}\ V_{\max}\frac{\alpha}{K_m+\alpha} = \beta; \\ Q_{S\to P}(S>1) &\approx \frac{\Delta t}{\alpha}\ V_{\max}. \end{aligned}$$

This one-parameter equation described only by β is clearly not a good representation of the original MM law. In fact, we can calculate the average rate of conversion $S \to P$ as a function of the concentration $[S]$ if we assume that the numbers of molecules S at a lattice site follow a Poisson distribution:

$$P(S=k) = \frac{\lambda^k}{k!}e^{-\lambda} \tag{4}$$

with $\lambda = [S]/\alpha$. This distribution has the mean λ and is attained in the absence of reactions and if no limits are imposed on the number of metabolite molecules per cell (if there were limits, it would be a binomial distribution).

On average, the number of molecules transformed is the product of the probability of finding k molecules with the probability of converting one of them, summed over all values k:

$$\begin{aligned} \langle Q\rangle &= \sum_{k=0}^{\infty} P(S=k)\ \ Q_{S\to P}(k) && \text{(5a)} \\ &= V_{\max}\ e^{-\lambda}\ (-\lambda)^{-K_m/\alpha}\left(\Gamma\left(\frac{\alpha+K_m}{\alpha},-\lambda\right)-\Gamma\left(\frac{\alpha+K_m}{\alpha}\right)\right), && \text{(5b)} \end{aligned}$$

which for $K_m \gg 1$ is almost equal to the correct MM rate, whereas for small K_m, it is too small (here Γ is the Gamma-function). In Figure 1 we can see a comparison of the exact rate with the average of 100 trials and the predicted average from Eq. (5).

An attempt to correct the discrepancy by changing the function $Q_{S\to P}$ does not lead to an exact solution, since we have some restrictions: Q must be nonnegative for all k. To make an exact fit, we would be required to use an alternating diverging series for $Q(k)$. Note that for polynomial reaction rate laws, such a correction is possible and can be given in analytic form [19], while for the MM rate law (which is not polynomial) this is not possible.

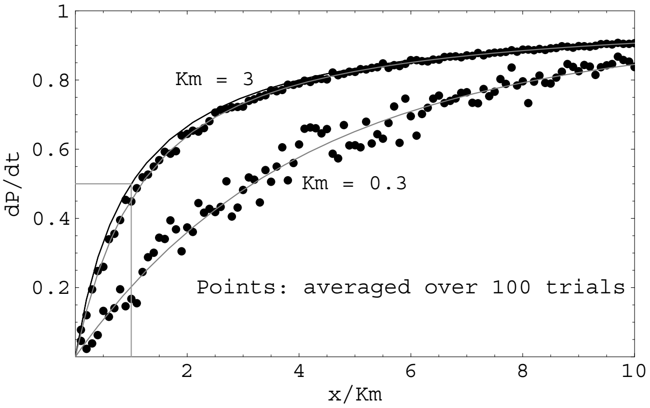


Fig. 1. Correct MM reaction rate (re-scaled by K_m, with $V_{\max} = 1, dt = 1$) (top curve) compared to the average of 100 random realizations (dots) and predicted average from Eq. (5) (two gray curves below the top curve).

3.2 Detailed Simulation of the Mechanism

A second possibility is to model the mechanism that leads to the MM rate law in detail. For the simple unimolecular reaction, this mechanism is described by Eq. (2). We can directly model this in the automaton, since at each lattice site we have at most one enzyme molecule, and therefore we can keep track of the state of this molecule: unbound or bound to S. The transitions between the states are governed by simple linear rate laws, which we can model exactly in the CA. We therefore have the following processes:

$$\begin{aligned} E \quad + nS &\to ES + (n-1)S && \text{with probability } p_1(n); \\ ES + nS &\to E \quad + (n+1)S && \text{with probability } p_{-1}; \\ ES + nP &\to E \quad + (n+1)P && \text{with probability } p_2, \end{aligned}$$

where the probabilities can be calculated from the rate laws as $p_1(n) = \Delta t\ k_1\ n$, $p_{-1} = \Delta t\ k_{-1}$, $p_2 = \Delta t\ k_2$ and p_{-1} and p_2 are independent of the metabolite numbers present in the cell. One problem with this approach is that the rates k are usually not known. We can assume values by taking into account that $V_m = k_2$ and $K_m = (k_{-1} + k_2)/k_1$ and that k_2 is smaller than k_1 and k_{-1}. If we simply assume $k_{-1} = c\ k_2$, we obtain $k_1 = (c+1)\ V_m/K_m$. One would assume that c should be > 1, but actually $c = 1$ works just as well. Note that now the fastest processes are $(c+1)$ times faster than before, which means that we need to use a smaller time step. With such a model we can perform a number of time steps and show that the average rate of transformation from S to P converges to the rate given by the MM law for all values of K_m.

4 Diffusion

Diffusion of the metabolite molecules can be simulated using a number of techniques. Since we model individual molecule counts, we have to make sure that the number of molecules is conserved by the diffusive operation. This is not easily verified in standard cellular automata. Two techniques have been developed which make conservation of particles easy in cellular automata: Partitioned cellular automata and block cellular automata.

The first kind is used in the lattice gas automata [2,4,5,18,19], where each lattice site has a number of channels to the neighboring sites, which can be occupied by at most one particle at a time. Since the transport of particles through these channels in the synchronous cellular automaton simply represents a permutation of the channel contents, one can easily verify that no particles get lost (assuming correct treatment of the boundaries).

The second technique is to subdivide the lattice into blocks, and at each step do some exchange of the contents of the sites within one block [3,12,16,20]. Since only a few cells are involved, it is easy to verify whether conservation laws are observed. In subsequent time steps, the block boundaries are changed to make information exchange across the whole lattice possible. Note that such block cellular automata are equivalent to classical cellular automata, since each can be simulated by the other (with some duplication of cell content and extension of the neighborhood).

Here we use this second technique to simulate diffusion. We use blocks of size two, which are placed on the lattice in all possible orientations in subsequent time steps (four orientations in two dimensions, six in three dimensions). For the exchange between the two cells within a block one can use different prescriptions: One possibility was already mentioned by Gillespie [7,8] as an extension of his stochastic simulation method to spatially distributed systems: If the two cells contain n_1 and n_2 particles respectively, move $(n_1 - n_2)/2$ particles from cell 1 to cell 2 (or reverse, if the difference is negative). This approach models the macroscopic diffusive flux proportional to the gradient in concentration. Better suited to the stochastic simulation is a microscopic approach: Consider all the particles in both cells as independent, and let each particle move to the neighboring cell with a fixed probability p. Then the number of particles to be moved from cell 1 to cell 2 is obtained by sampling a binomial distribution with parameters p and n_1, while the number of particles moving from cell 2 to cell 1 is given by a binomial distribution with parameters p and n_2. Thus the number of particles exchanged between the two cells is governed by the difference between two binomial distributions. This approach leads to fluctuating particle numbers in all cells even in the absence of reactions, which is appropriate when particles represent molecules. The particle numbers in the cells will be distributed according to a Poisson distribution Eq. (4).

For the calculations of effective reaction rates, as in Eq. (5), the averaged (macroscopic) diffusion operator leads to a distribution that is more compact than the Poisson distribution, e.g. a two-valued distribution around the aver-

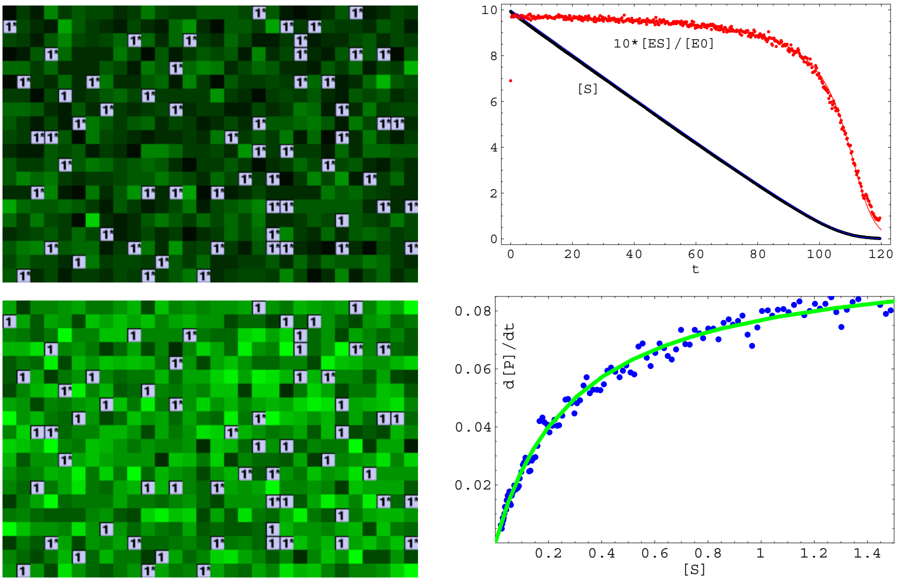


Fig. 2. Simulation of one reaction using the CA. Cells marked by a 1 contain an enzyme molecule, when marked with “1*” the enzyme is bound to a substrate molecule. The other cells show the relative number of P molecules. On the left we show snapshots at time $t = 50\Delta t$ and $t = 110\Delta t$, on the right the time evolution of $[S]$ and $[ES]/[E_0]$ (the fraction of substrate-bound enzyme molecules), and right bottom the dependence of $d[P]/dt$ on $[S]$ (for a larger 3-D system of 30^3 sites). The lines show the theoretical predictions from the MM rate law.

age concentration. Here we use the microscopic procedure to ensure consistent results.

5 CA Simulations

Figure 2 shows a simulation with such a cellular automaton for a system with only one enzymatic reaction: $E + S \to E + P$. We use the parameters $V_m = 1, K_m = 0.3, c = 1, \Delta t = 0.01$. Here the time step is limited by the restriction that at most one molecule can react at one time step (since the enzyme can only bind one substrate molecule at a time), and we start the simulation with a high substrate concentration of 10 molecules per site. We observe that the CA model corresponds to the predictions from the macroscopic Michaelis-Menten rate law. In this example diffusion of the metabolites is comparably fast, and the spatial dimension does not have a measurable influence.

In a second test, we use this model to simulate a toy network of unidirectional enzymatic reactions with four enzymes and four metabolites.

$$A \xrightarrow{E_1} B \xrightarrow{E_2} C \xrightarrow{E_3} D \xrightarrow{E_4} A \qquad (6)$$

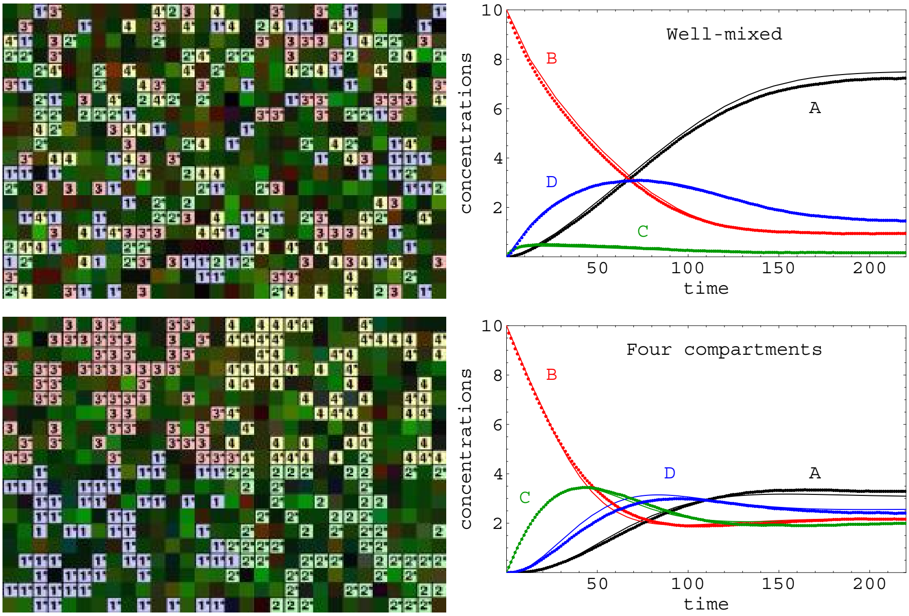


Fig. 3. Simulation of four-enzyme system with different spatial distributions of the enzymes. Top row: random distribution of enzymes (with equal numbers for all enzymes, and probability of 0.2 for each site). Bottom row: distribution in four quadrants. The concentration time series shown at right are compared with the predictions from the differential equations using Michaelis-Menten kinetics. For the four quadrant model, we compare with a compartmentalized ODE model, since the well-mixed ODE model, which is the same in both cases, clearly gives wrong results. The initial conditions for the metabolites are identical in both cases, and the number of enzyme molecules of each type is also identical in both cases.

The parameters for the different enzymes are: E_1: $V_{m1} = 1$, $K_{m1} = 0.3$, E_2: $V_{m2} = 2$, $K_{m2} = 1$, E_3: $V_{m3} = 3$, $K_{m3} = 0.3$, E_4: $V_{m4} = 3$, $K_{m4} = 3$, the initial condition is $[B] = 10$ and $[A] = [C] = [D] = 0$.

We simulate two different situations in Figure 3, one where the enzymes are randomly distributed in space, and one where the enzymes are separated in different compartments, and the metabolites must diffuse from one enzyme to the other. In the first case, we compare the time evolution of the metabolite concentrations with the solution of the ordinary differential equations, using the Michaelis-Menten approximation for the construction of the ODEs. In the second case (with identical parameters except for the spatial distribution of enzymes), we observe a significantly different time evolution and also a very different steady state. We can model this situation by constructing a system of ODEs where each quadrant is treated as a different compartment, the reaction rates are adjusted in the compartments (e.g., in the compartment corresponding to the left lower quadrant, only enzyme E_1 is present, but in concentration four times higher than in the first model. Therefore only the reaction $A \to B$ takes place, with rate four times higher), and diffusion of all species between neighboring compartments is introduced. The lines in the time evolution plot correspond to the solution of this ODE.

The large differences in the two solutions show that spatial dependences should not be ignored when whole-cell models are constructed.

6 Conclusion

We have described a cellular automaton model for enzymatic reaction networks. This model is based on block-cellular automata to ensure conservation of particles and assumes that at most one enzyme can be present at any lattice site. The enzymes can change state by binding to metabolite molecules, and the state changes are described by probabilistic rules derived from the enzyme kinetics to be simulated. The quantitative correctness of the rules has been demonstrated by analytic arguments and by comparison of simulations with the ODE solution. This model can incorporate complex processes, where an enzyme binds several ligands, possibly in specific order. The explicit inclusion of space makes possible the detailed investigation of phenomena that depend on diffusion of different species. As an example we have demonstrated that simply placing the enzymes at different regions leads to a significantly different average behavior. In this case the behavior can be well approximated by a compartmentalized ODE model, but in more complex geometries of biological relevance, this is not necessarily the case.

The model presented here is a microscopic model, since individual enzyme molecules are explicitly represented, and the corresponding fluctuations take place, which sets the model apart from numerical methods for solving the averaged PDE. We showed that enzymatic reactions cannot be simulated by the same techniques used e.g., in reactive lattice gas automata [2], since the rate law is not polynomial.

As further steps, we will apply this cellular automaton approach to more complex enzymatic reaction networks and try to obtain biologically meaningful spatial distributions of the individual enzymes.

References

[1] Hans Bisswanger. *Enzymkinetik*. Wiley-VCH, Weinheim, 2000.

[2] Jean Pierre Boon, David Dab, Raymond Kapral, and Anna Lawniczak. Lattice gas automata for reactive systems. *Physics Reports*, 273(2):55–147, 1996.

[3] B. Chopard and M. Droz. Cellular automata approach to diffusion problems. In P. Manneville, N. Boccara, G. Y. Vichniac, and R. Bidaux, editors, *Cellular automata and modelling of complex physical systems*, pages 130–143, Berlin, Heidelberg, 1989. Springer-Verlag.

[4] David Dab and Jean-Pierre Boon. Cellular automata approach to reaction-diffusion systems. In P. Manneville, N. Boccara, G. Y. Vichniac, and R. Bidaux, editors, *Cellular automata and modelling of complex physical systems*, pages 257–273, Berlin, Heidelberg, 1989. Springer-Verlag.

[5] David Dab, Jean-Pierre Boon, and Yue-Xian Li. Lattice-gas automata for coupled reaction-diffusion equations. *Phys. Rev. Lett.*, 66(19):2535–2538, 1991.

[6] Micheal A. Gibson. *Computational Methods for Stochastic Biological Systems.* Phd thesis., Calif. Inst. Technology, 2000.

[7] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Computational Physics*, 22:403–434, 1976.

[8] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem*, 81:2340–2361, 1977.

[9] M. Kanehisa. Kegg. http://www.genome.ad.jp/kegg/.

[10] M. Kanehisa. Toward pathway engineering: a new database of genetic and molecular pathways. *Science & Technology Japan*, 59:34–38, 1996.

[11] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28:27–30, 2000.

[12] Norman Margolus and Tomaso Toffoli. *Cellular automata machines. A new environment for modeling.* MIT Press, Cambridge, MA, USA, 1987.

[13] Gerhard Michal. *Boehringer Mannheim Biochemical Pathways Map.* Roche and Spektrum Akademischer Verlag.

[14] Stefan Schuster. Studies on the stoichiometric structure of enzymatic reaction systems. *Theory Biosci.*, 118:125–139, 1999.

[15] Stefan Schuster and Claus Hilgetag. On elementary flux modes in biochemical reaction systems at steady state. *J. Biol. Syst.*, 2:165–182, 1994.

[16] T. Toffoli and N. Margolus. Invertible cellular automata: a review. *Physica D*, 45:229–253, 1990.

[17] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier, 1992.

[18] Jörg R. Weimar. *Cellular Automata for Reactive Systems.* PhD thesis, Université Libre de Bruxelles, Belgium, 1995.

[19] Jörg R. Weimar. *Simulation with Cellular Automata.* Logos-Verlag, Berlin, 1998.

[20] Jörg R. Weimar. Simulating reaction-diffusion cellular automata with JCASim. In T. Sonar, editor, *Discrete Modelling and Discrete Algorithms in Continuum Mechanics*, pages 217–226. Logos-Verlag, Berlin, 2001.

Modelling Surface Flows for Macroscopic Phenomena by Cellular Automata: An Application to Debris Flows

Donato D'Ambrosio[1], Salvatore Di Gregorio[1], Giulio Iovine[2], Valeria Lupiano[2], Rocco Rongo[3], and William Spataro[1]

[1]Univ. of Calabria, Dept. of Mathematics & Center of High-Performance Computing, Arcavacata, 87036 Rende (CS), Italy
{d.dambrosio, toti.dig, spataro}@unical.it
[2]National Research Council - Institute of Research for Hydrogeological Protection (CNR-IRPI), Via Cavour, 87030 Rende (CS), Italy
{iovine, lupiano}@area.cs.cnr.it
[3]Univ. of Calabria, Dept. of Earth Sciences, Arcavacata, 87036 Rende (CS), Italy
rongo@unical.it

Abstract. Cellular automata are good candidates for modelling and simulating complex dynamical systems, whose evolution depends on the local interactions of their constituent parts. Many macroscopic phenomena have such features, but their complexity involves sometime the interaction of heterogeneous processes, whose composition is not immediate in a cellular automaton frame. Furthermore the managing of flows could be not trivial, because cellular automata and derived models as lattice-gas methods cannot always be applied successfully to macroscopic cases. We propose some empirical and practical rules for modelling some macroscopic phenomena with surface flows. An application to complex debris flows is exhibited together with the results of simulations of a real case.

1 Introduction

Cellular Automata (*CA*) are one of the first Parallel Computing models [1]; they capture the peculiar characteristics of systems, whose global evolution may be described on the basis of local interactions of their constituent parts (locality property).

A homogeneous *CA* [2] can be considered as a d-dimensional space, the cellular space, partitioned into regular cells of uniform size, each one embedding an identical finite automaton, the elementary automaton (*ea*). Input for each *ea* is given by the states of the *ea* in the neighbouring cells, where neighbourhood conditions are determined by a pattern, which is invariant in time and constant over the cells. At time t=0 (step 0), *ea* are in arbitrary states and the *CA* evolves changing the state of all *ea* simultaneously at discrete times, according to the transition function of the *ea*.

Complex phenomena modelled by classical *CA* involve an *ea* with few states (usually no more than a dozen) and a simple transition function, easily specified by a lookup table [3].

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 304-314, 2002.

Many complex macroscopic phenomena, which own the same locality property of *CA*, seem difficult to be modelled in the classical *CA* frame, because they consist often of processes involving different frameworks of space and time and need transition rules, far from a typical transition function of a finite automaton.

This paper suggests some mechanisms (when they are practicable) which permit to define the macroscopic phenomenon in terms of a *CA* formalism [4]. Different applications [5], [6] were developed according to some of these specifications. A particular complex example is here exhibited: landslides characterized by very rapid debris flows with strong soil erosion, generating an avalanche effect.

The second section considers some *CA* criteria for modelling complex macroscopic phenomena together with a practical approach for modelling flows, the third section presents the model SCIDDICA (release S3hex) for very rapid debris flows with strong soil erosion, the fourth section shows a result of the simulations of the landslides occurred in Sarno (1998, Italy); some conclusions are reported at the end.

2 Some *CA* Criteria for Modelling Macroscopic Phenomena

The criteria here reported are empirical and may be applied only to particular classes of macroscopic phenomena, involving surface flows. The proposed recipes don’t guarantee by themselves success in the simulation. In fact, when the model is validated on real cases of a certain typology, the model applications in similar conditions are reliable.

The physical laws expressed for complex phenomena involve systems of differential equations; they are difficult to be managed, even considering numerical methods, which are often revealed inapplicable.

The approximations, that are here proposed, are rough, however they want to translate in a context of locality with discrete time and space the conservation laws of physics and the features of minimisation, typical of physical equations, concerning the energy variations.

2.1 Model Specifications for Time and Space

Simulations of real macroscopic events need to define a correspondence of the system and its evolution with the model and the corresponding simulations. At least, the dimension of the cell and the time correspondence to a *CA* step must be fixed. They are defined as global parameters, because their values are global to all the *CA*; of course other global parameters could be necessary.

In order to fix these two essential global parameters, further points must be considered, when the phenomenon is complex and can involve time and/or space heterogeneity in the sense specified later on.

The state of the cell must account for all the characteristics, relevant to the evolution of the system and relative to the space portion corresponding to the cell. Each characteristic could be individuated as a substate; the permitted values of the substate must form a finite set. The set of possible states of the cell is given by the Cartesian product of the sets of substates. In case that one of the characteristics (e.g. a physical quantity) is usually expressed in terms of a continuous variable referred to a space point, the cell must correspond to a portion of space small enough so that a single value may be attributed to full space of the cell. The continuity of the variable is not a problem; in practical case the utilised variables have a finite number of significant digits and a finite range of permitted values, then the set of utilised values could be extremely large, but always finite.

As the state of the cell can be decomposed in substates, the transition function may be split in many local interactions: the "elementary" processes. Such local interactions could be inhomogeneous in space and/or time: the opportune dimension of a cell can vary for different local interactions; very fast local interactions need a step corresponding to short times on the same cell size; the appropriate neighbourhoods for different local interactions could be different. An obvious solution to these problems is the following: the smallest dimension of a cell must be chosen among the permitted dimensions of all the local interactions. Then it is possible to define for each local interaction an appropriate range of time values in correspondence of a *CA* step; the shortest time necessary to the local interactions must correspond to a step.

It is possible, when the cell dimension and the *CA* step are fixed, to assign an appropriate neighbourhood to each local interaction; the union of the neighbourhoods of all the local interactions must be adopted as the *CA* neighbourhood. A lookup table could be unpractical to describe the local interactions. Each local interaction may be espoused by means of procedures involving the proper substates and neighbourhood of the local interaction.

Considering these premises, it is expedient to consider the transition function step divided in as many phases as the elementary processes; the substates involved in each process will be updated each time at the phase end.

2.2 A Practical Approach for Modelling Surface Flows

A delicate point concerns the modelling of flows in a *CA* context. Solutions to this problem were proposed especially for microscopic simulation (e.g., Boltzmann lattice [7]). A practical approach for modelling surface flows is here proposed.

Macroscopic phenomena involving surface flows may be often modelled by two-dimensional *CA*, when the third dimension, the height, may be included as a property of the state of the cell (a substate). This condition permits to adopt a simple, but effective strategy, based on the hydrostatic equilibrium principle in order to compute the cell outflows [4].

Let us focus for simplicity on a single cell (individuated as the "central" cell). It can be considered limited to the universe of its neighbourhood, consisting of m cells: the central cell and of the remaining cells (individuated as the "adjacent" cells). Index 0 individuates the central cell, indexes 1, 2 … m-1 individuate the adjacent cells.

On the basis of this assumption, the outflows from the central cell to the adjacent cells depend on the hydrostatic pressure gradients across the cells, due to differences in heights (for trivial instance, altitude plus debris thickness).

Two quantities must be identified in the central cell: the fixed part ($q[0]$) and the mobile part (p) of the height. The mobile part represents a quantity that could be distributed to the adjacent cells (the debris thickness, for instance), whilst the fixed part cannot change value (the altitude, for instance). So the height of the central cell is the sum of two terms $p+q[0]$; $q[i]$, $1 \leq i \leq m-1$ is the height of the *i*-th adjacent cell of the neighbourhood, where the distinction between mobile and fixed part is not necessary, taking in account that only the mobile part of the central cell may be distributed. The flow from the central cell to the *i*-th neighbouring cell will be denoted by $f[i]$, $0 \leq i < m$, where $f[0]$ is the part of p which is not distributed. Let $q'[i]=q[i]+f[i]$, $0 \leq i \leq m-1$ be the sum of the content of a neighbouring cell, plus the flow from the central cell, and let *q'_min* be the minimum value for $q'[i]$.

Thus, the determination of outflows, from the central cell to the adjacent cells, is based on the local minimisation of the differences in "height", given by the following expression:

$$\sum_{i=0}^{m-1} (q'[i] - q'\_min) \qquad \textbf{(1)}$$

The "minimisation" algorithm, i.e. the algorithm for the minimisation of differences, and correlated theorems are not treated here, but they can be found in [4].

Furthermore, the minimum "imbalance" conditions cannot be always achieved in a *CA* step, so a relaxation rate, depending on both the cell size and the duration of the *CA* step, must be considered. The relaxation rate p_r, specified by a multiplicative factor, can assume values between *0* and *1*: $0<p_r \bullet 1$.

This mechanism involves particular care in the space and time settlement: the size of the cell limits at the top the *CA* step, because the outflow rate may not be so rapid that the outflow overcomes the neighbourhood boundaries in a step.

3 The CA Model SCIDDICA for Debris Flows

SCIDDICA (Simulation through Computational Innovative methods for the Detection of Debris flow path using Interactive Cellular Automata - to be read "'she:ddre:*CA*", as the acronym was devised to mean "it slides" in Sicilian), is a *CA* model developed in order to simulate the behaviour of landslides that can be typologically defined as "flows" [8]. These are a good application field for *CA*, as they can be considered to evolve in terms of local processes.

The new release S3hex with hexagonal tessellation is here presented.

3.1 The Problem of Modelling Debris Flows

Analytical solutions to the differential equations (e.g. the Navier-Stokes equations) governing debris flows are a hopeless challenge, except for few simple, not realistic, cases. The possibility to successfully apply numerical methods for the solution of differential equations have been elevated considerably because of the continuous rise of computing power, even if there are still difficulties to obtain high performances regarding their implementation on parallel computing machines to exploit the maximum computational power. It must be said that *CA* are a particular model that can be easily implemented in parallel computing [2], [9].

Moreover, the complexity of the problem resides both in the difficulty of managing irregular ground topography and in complications of the equations, that must also be able to account for flows that can range, rheologically, from nearly Newtonian fluids to brittle solids by means of water loss.

Some authors proposed *CA* or *CA*-like models for flow type landslides.

Barca et al. [10] designed 3-dimensional *CA* models with a cellular space divided in cubic cells, but computational high complexity and costs did not permit to apply the model to the simulation, except for few cases of small and simple landslides.

Sassa [11] adopted *CA*-like numerical method to finite differences for a simplified solution of debris flow equations and applied it to the M. Ontake landslide. His approach accounts for the physics of the phenomenon, but the simulation suffered because of cell dimension, too large to obtain an accurate description of the event.

Di Gregorio et al. [9] developed a simple 2-dimensional *CA* model (first release of SCIDDICA) and validated it by simulating the Mt. Ontake landslide.

Segre & Deangeli [12] presented a 3-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.

SCIDDICA was further improved, introducing correlating empirical parameters of the model to physical ones, and applied again to the M. Ontake landslide [13].

Avolio et al. [14] applied a modified release of SCIDDICA to the Tessina landslide, also performing a risk analysis for the threatened area.

Malamud & Turcotte [15] presented a very simple *CA* "sand pile" model to be applied to landslides from a statistical viewpoint in order to forecast the frequency-area distribution of landslides triggered by earthquakes.

Clerici & Perego [16] simulated the Corniglio landslide using a simple *CA* model in order to capture the blockage mechanisms for that type of landslide.

Finally, a preliminary extension of SCIDDICA was developed in order to capture the characteristics of the extremely complex landslides of Sarno [17].

3.2 The Hexagonal Model SCIDDICA

The latest hexagonal release of SCIDDICA is a significant extension of the models applied successfully to the landslides of Tessina and Mount Ontake. Such extension involves new substates, new procedures, new parameters because of the landslides of Sarno appear to be a more complex phenomenon, especially for their avalanche effect in soil erosion during the phenomenon evolution.

The hexagonal *CA* model SCIDDICA is the quintuple

$$\text{SCIDDICA}= <R, X, Q, P, \sigma>$$

- *R* is the set of regular hexagons covering the finite region, where the phenomenon evolves.
- *X* identifies the geometrical pattern of cells, which influence any state change of the central cell (the central cell itself and the six adjacent cells):
- *Q* is the finite set of states of the *ea*; it is equal to the Cartesian product of the sets of the considered substates:

$$Q = Q_a \times Q_{th} \times Q_r \times Q_d \times Q_m \times Q_o^6 \times Q_i^6$$

where:

  - Q_a is the cell altitude
  - Q_{th} is the thickness of landslide debris

- Q_r is the “run up” height (depends on the potential energy and expresses the height that can be overcome by the debris flow)
- Q_d is the maximum depth of detrital cover, that can be transformed by erosion in landslide debris (it depends on the type of detrital cover)
- Q_o is the debris outflow from central cell to adjacent cell (six components)
- Q_i is the debris inflow from adjacent cell to central cell (six components)

The elements of Q_a express the value of the altitude; also the elements of Q_r are expressed as a length; the elements of Q_{th} and Q_d represent the material quantity inside the cell, expressed as thickness/depth; then Q_o and Q_i are expressed in terms of length by homogeneity.

- *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:

$$P=\{p_c, p_t, p_{adh}, p_f, p_r, p_{rl}, p_{mt}\}$$

where:

- p_c is the edge of the cell;
- p_t is the temporal correspondence of a step of SCIDDICA;
- p_{adh} is the adhesion value, i.e. the debris thickness, that may not removed;
- p_f is the friction threshold for debris outflows;
- p_r is the relaxation rate of debris landslide outflows;
- p_{rl} is the run up loss at each *CA* step;
- p_{mt} is the activation threshold of the mobilisation;
- p_{er} is the erosion depth parameter;

- $\sigma: Q^7 \rightarrow Q$ 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 so initialised:

Q_a assumes the morphology values except for the detachment area, where the thickness of the landslide mass is subtracted from the morphology value;

Q_{th} is zero everywhere except for the detachment area, where the thickness of landslide mass is specified;

Q_r assumes initial values equal to the substate Q_{th};

Q_d assumes initial values corresponding to the maximum depth of the mantle of detrital cover, which can be eroded;

Q_o and Q_i are zero everywhere.

At each next step, the function σ is applied to all the cells in *R*, so that the configuration changes in time and the evolution of the *CA* is obtained.

3.3 The SCIDDICA Transition Function (S3hex Release)

Four local processes are considered for the release S3hex of SCIDDICA:

- altitude, run up and debris thickness variation by detrital cover mobilisation;
- run up variation by friction;

- debris outflows determination by application of the minimisation algorithm;
- debris inflows determination, debris and run up variation due to outflows.

In the following, a sketch of the local elementary processes will be given, which is sufficient to capture the mechanisms of the transition function. Δ_Q_x means variation of the substate Q_x; the energy associated to the debris inside a cell is measured by the value $Q_r[0]*Q_{th}[0]$; the execution of an elementary process updates the substates.

Mobilisation Effects. When the energy value overcomes an opportune threshold ($Q_r[0]*Q_{th}[0]>p_{mt}$), depending on the soil features and its saturation state then a mobilisation of the detrital cover occurs proportionally to the quantity overcoming the threshold. The depth of the erosion (altitude variation Δ_Q_a) is given by the following expression $(\Delta\_Q_a=-(Q_r[0]*Q_{th}[0]-p_{mt})*p_{er}$; then trivially $\Delta\_Q_{th}=-\Delta\_Q_a$; $\Delta\_Q_r=-\Delta\_Q_a$.

Friction Effect. The effect of the friction is modelled, considering a constant run up loss p_{rl} at each SCIDDICA step.

Outflows Determination. Very rapid debris flows imply often a run up effect, depending on the energy associated to debris flow. So the height minimisation algorithm [4] is applied, considering the height fixed part of the central cell as $q[0]=Q_a[0]+p_{adh}[0]$, the height mobile part as $p[0]=Q_r[0]-p_{adh}[0]$, the height of the adjacent cell i, $1\leq i\leq 6$, as $q[i]=Q_a[i]+Q_{th}[i]$.

A preliminary test is executed in order to account the friction effects, that prevent debris outflows, when the height difference between the two cells is insufficient; the condition is expressed by the formula $(q[0]+p[0]-q[i])<p_f$.

Note that in order to account for the ability of the flowing debris of climbing a slope of given partial height, the volume occupied by the debris in the central cell is, ideally, assumed to be equal to cell area multiplied by run-up. In this way, given an amount of debris in the cell, its volume (and then its thickness - which we use in the computation) enlarges to a higher value (fictitious swelling).

The minimisation algorithm then returns outflows values that are fictitious swelling and must be normalised by the multiplicative factor $(Q_{th}[0]-p_{adh}[0])/(Q_r[0]-p_{adh}[0])$; furthermore another multiplicative factor must be considered: the relaxation rate p_r.

Outflows Effects. Debris inflows are trivially derived by the outflow values: cell A outflow toward adjacent cell B is cell B inflow from adjacent cell A.

The new value of $Q_{th}[0]$ is given, considering the balance of inflows and outflows:

$$Q_{th}[0]+\sum_{j=1}^{6}(Q_i[j]-Q_o[j]) \tag{2}$$

The run-up determination is calculated as the average weight of Q_r, by considering both the remaining debris in the central cell and the inflows:

$$\left((Q_{th}[0]-\sum_{j=1}^{6}Q_o[j])\times Q_r[0]+\sum_{j=1}^{6}(Q_i[j]\times Q_r[j]\right)\Bigg/\left(Q_{th}[0]+\sum_{j=1}^{6}(Q_i[j]-Q_o[j])\right) \tag{3}$$

4 Simulations of the Curti Debris Flow of May 1998

The Curti landslide was selected among the whole population of slope movements triggered in Campania by prolonged rainfalls on May 1998 [18]. On that occasion, hundreds of small soil slips originated at Pizzo d'Alvano, in the volcaniclastic mantle, on the uppermost portions of the slopes, and transformed into fast-moving debris flows. These generally involved the entire depth of available detrital cover, eroding it down to the bedrock, and increased their initial volume by scraping off material along the path. Landslides hit the urbanised areas at the base of the massif (villages of Sarno, Siano, Bracigliano, Quindici), killing 161 people and leaving more than 1000 others homeless.

In particular, the selected landslide started as a soil slip at 783 m a.s.l. (right above a minor outcrop of bedrock). The sliding mass of volcaniclastic terrain, of about 100 m^3, rapidly transformed into a fast flowing mixture of mud, debris and water, running down the slope along a smooth pre-existing channel. After encountering a notable bedrock outcrop (about 75 m high), the flow enlarged and entered the main channel, triggering some "secondary" debris slides on both flanks (cf. "a", "b", "c" and "d" in Fig.1). At about 200 m a.s.l., the phenomenon subdivided into two distinct flows which, *CA*. 460 m down slope, merged again and the hit the urban area of Curti (at 115 m a.s.l.), killing two people. The total length of the Curti debris flow is greater than 1750 m. According to field evidence, along both the upper and middle reach of the path, the detrital cover was completely eroded by the flowing mass. As a consequence, the debris flow reached the base of the massif with an estimated volume of about 81000 m^3.

The thickness of the mantle of detrital cover (available for the erosion along the path), as evaluated through detailed field surveying, was given as input matrix to the model. Note that "complete" erosion of the detrital cover was allowed along the path of the debris flow – as suggested by field observations. The extent of this matrix is about 2.7 times greater than the real landslide.

At present, values of global parameters have been assigned on the basis of a trial-and-error procedure, starting with physically reasonable values and gradually modifying them until satisfactory results obtained. Initial values have been chosen by considering both the cell size, and their expected influence on the rheology of the phenomenon. By simply changing these values, simulation after simulation, and by comparing the results with the map of the real event, the set of provisional "optimal" values for the particular case of study has then been obtained.

A quantitative evaluation of the performed simulations can be made, in a GIS environment, by comparing the extent of the "real" landslide (as mapped through field surveying and air photo-interpretation) and the simulated ones. The area A_b affected by both the simulated and the real debris flow was computed, as well as sectors which pertain to only one case (A_r = real and not simulated, A_s = simulated and not real; note that the error A_r is more serious than A_s).

In Fig.1, the best simulation obtained with S3hex release of SCIDDICA is shown. The ratios of A_b, A_r and A_s to the extent of the real debris flow, are 73%, 27% and 21%. Simulation may be considered successful in terms of extent of debris path, if quality of available input data is considered; furthermore erosion and deposit obtained by simulation are satisfying, where a comparison with real event is possible.

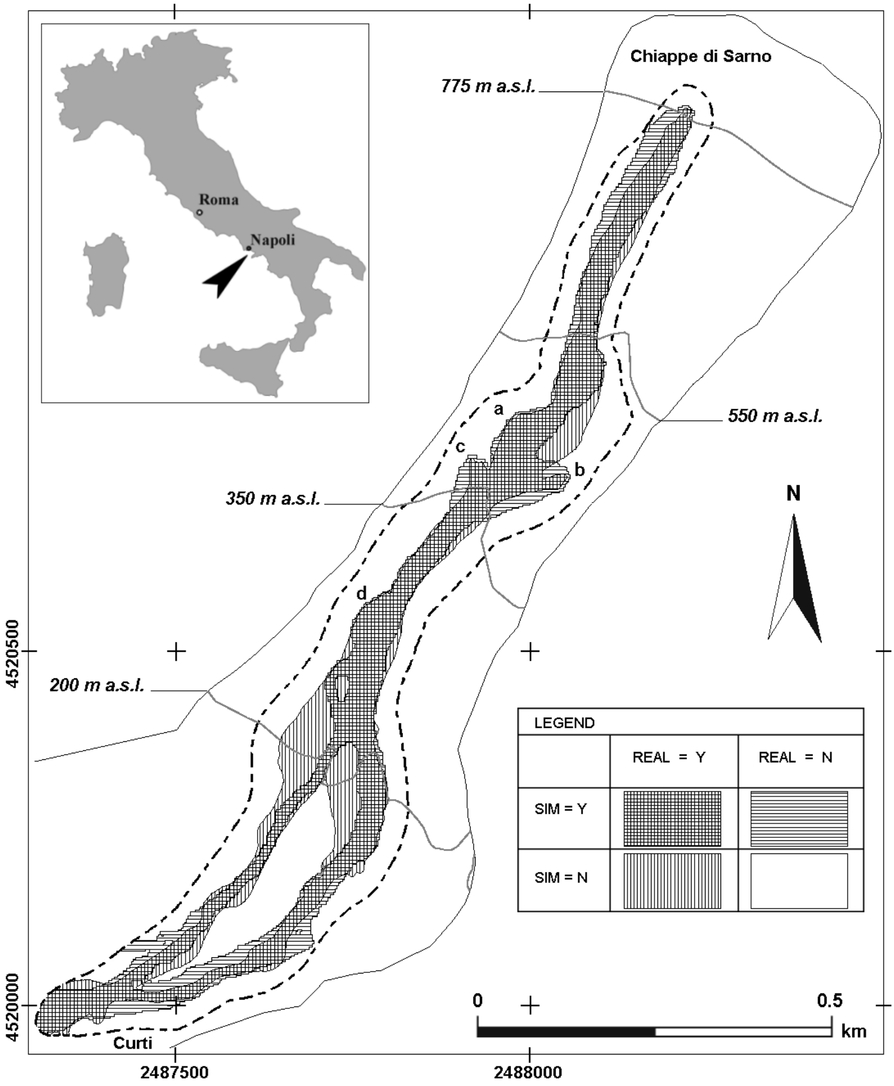


Fig. 1. The results of the application of the SCIDDICA S3hex release are evidenced by the composition of the real and simulated landslide; *a*, *b*, *c* and *d* individuate secondary soil slips

The global parameters of the simulation in fig. 1 are: p_c=1.25m (apothem of the hexagonal cell); p_t≈0.3s (?); p_{adh}=0.001m; p_f=0.1m; p_r=1; p_{rl}=0.6m; p_{mt}=3.5m^2; p_{er}=0.015m. The simulation takes 11498 steps; afterwards the mass moving in the simulation is null.

5 Conclusions

Simulations of the Chiappe di Sarno-Curti debris flow, occurred at Sarno in May 1998, proved to be consistent with the observed path of the actual landslide, suggesting that SCIDDICA could be usefully applied in debris-flows hazard analyses.

The complexity of this debris landslide with avalanche features urged our research group "Empedocles" to examine critically the methods developed [4] for modelling macroscopic phenomena with surface flows [5], [6], [13], [14]. Corrections and extensions, here adopted, can be extended to other similar phenomena. Significant improvements, in our opinion, could be added. A limit of this approach is the implicit managing of the time and an understimation of inertial effects; future innovations will concern this problem.

Acknowledgements. The development of the the SCIDDICA model (Sarno release) and performed simulations were partially funded by Regione Campania, Italy (*Project "Sviluppo di una Modellistica Sperimentale–Temporale dei Processi Evolutivi dell' Ambiente e del Territorio per la Mitigazione dei Rischi"*). The authors thank Ms M. Adinolfi as a supporter for the carried out project; Ms L. Monti for the project scientific organisation.

References

1. von Neumann, J.: Theory of self reproducing automata. Uni. of Illinois Press, Urbana (1966)
2. Worsch, T.: Simulation of Cellular Automata. FGCS, 16, (1999) 157-170
3. Toffoli, T., Margolus, N.: Cellular Automata Machines. MIT Press, Cambridge (1987)
4. Di Gregorio, S., Serra R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. FGCS, 16, (1999) 259-271
5. Barca, D., Crisci, G.M., Di Gregorio, S., Nicoletta, F.P.: Cellular Automata for simulating lava flows: a method and examples of the Etnean eruptions. Transport Theory and Statistical Physics, 23 1-3 (1994), 195-232
6. 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, Vol. 26(1), (2001), pp. 33-39.
7. Succi, S., Benzi, R., Higuera, F.: The lattice Boltzmann equation: a new tool for computational fluid dynamics. Physica 47 D (1991) 219-230
8. Cruden, D.M., Varnes, D.J.: Landslide Types and Processes. In: Turner, A.K., Schuster, R.L., (eds.): Landslides: Investigation and Mitigation. Special Report 247, Transportation Research Board, NRC, National Academy Press, Washington D.C., (1996) 36-75.
9. Di Gregorio, S., Nicoletta, F., Rongo, R., Sorriso-Valvo, M., Spezzano, G., Talia D.: Landslide Simulation by Cellular Automata in a Parallel Environment. Mango Furnari, M. (ed.): Proceedings of 2nd International Workshop "Massive Parallelism: Hardware, Software and Applications", World Scientific, Singapore (1995) 392-407
10. Barca, D., Di Gregorio, S., Nicoletta, F.P., Sorriso-Valvo, M.: A Cellular Space Model for Flow type Landslides. In: Messina, G., Hamzda, M.H., (eds.): Computers and their Application for Development. Proc. Int. Symp. IASTED (Taormina), (1986) 30-32
11. Sassa, K.: Motion of Landslides and Debris Flows. Report for Grant-in-Aid for Scientific Research by the Jap. Ministry of Edu., Science and Culture (Project No. 61480062), (1988).

12. Segre, E., Deangeli, C.: Cellular Automaton for Realistic Modelling of Landslides. Nonlinear Processes in Geophysics, 2(1), (1995) 1-15.
13. 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 (2), (1999) 97-100.
14. Avolio, M.V., Di Gregorio, S., Mantovani, F., Pasuto, A., Rongo, R., Silvano, S., Spataro, W.: Simulation of 1992 Tessina landslide by CA model and future hazard scenarios. JAG, 2 (1), (2000) 41-50.
15. Malamud, B.D., Turcotte, D.L.: Cellular Automata models applied to natural hazards. IEEE Computing in Science & Engineering. 2 (3), (2000) 42-51
16. Clerici, A., Perego, S.: Simulation of the Parma River blockage by the Corniglio landslide (Northern Italy). Geomorphology, 33 (2000) 1-23.
17. D'Ambrosio, D., Di Gregorio, S., Iovine, G., Lupiano, V., Rongo, R., Spataro, W.: First simulations of the Sarno debris flows through Cellular Automata modelling. Geomorphology, (2002) in press.
18. Del Prete, M., Guadagno, F.M., Hawkins, A.B.: Preliminary report on the landslides of 5 May 1998, Campania, southern Italy. Bull. Eng. Geol. Env. **57**, (1998) 113-129.

Simulation Framework for the Autobahn Traffic in North Rhine-Westphalia

Sigurður F. Marinósson[1], Roland Chrobok[1],
Andreas Pottmeier[1], Joachim Wahle[2], and Michael Schreckenberg[1]

[1] Gerhard-Mercator-University Duisburg, Physics of Transport and Traffic,
Lotharstr. 1, 47048 Duisburg, Germany,
[marinosson,chrobok,pottmeier]@traffic.uni-duisburg.de,
schreckenberg@uni-duisburg.de,
http://www.traffic.uni-duisburg.de

[2] TraffGo GmbH, Grabenstr. 132, 47057 Duisburg, Germany, wahle@traffgo.com,
http://www.traffgo.com

Abstract. In the last decade there has been a continuous progress in the development of cellular automata models of vehicular traffic. The most recent models are able to reproduce free flow, spontaneous jam formation, synchronized traffic, as well as meta-stability. However, these models have been developed and tested on topologically simple road networks and the translation to large and topologically complex real road networks is non-trivial. In this paper we describe the cellular automaton model we use to simulate the traffic on the autobahn network in North Rhine-Westphalia and discuss some of the challenges that arise when using this model on such a huge and topologically complex network.

1 Introduction

Efficient vehicular transport of persons and goods is of vital importance to any modern society. In densely populated areas the capacity of the road network is often to its limits and frequent traffic jams cause a significant economic damage. Moreover, in these areas, it is usually hardly possible or socially untenable to build more roads. An intelligent use of the resource 'traffic infrastructure' is therefore economically crucial. The German state North Rhine-Westphalia (NRW) is an example of such a densely populated area where the capacity of the road network is not able to satisfy the traffic demand during the rush-hours. Every day there are traffic jams on the autobahns in the Rhine-Ruhr region (Dortmund, Duisburg, Düsseldorf, Essen, etc.) and in the area around Cologne and Leverkusen. To make things even worse, the traffic demand is still growing. For this reason, new information systems and traffic management concepts are clearly needed.

Data regarding the traffic state on the autobahns in NRW are mainly provided through more than 3,500 loop detectors and infrared or video detection devices. These devices are locally installed and deliver measured data to central servers minute by minute. The measured quantities include *the number of*

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 315–324, 2002.

passenger cars passed, *the number of lorries passed*, *the average speed of the passenger cars*, and *the average speed of the lorries*. Our approach to generate the traffic state in the whole autobahn network from these locally measured quantities is to feed the data into an advanced cellular automaton traffic simulator. The simulator does not only deliver information about the traffic states in regions not covered by measurement, but also delivers reasonable estimates for other valuable quantities like travel times for routes, a quantity that is not directly accessible through the measurements of the detectors.

2 Simulation Model

Because data is fed real-time into the simulator it has to be efficient, that is, at least real time. Due to their design cellular automata models are very efficient in large-scale network simulations [1,2,3,4,5]. Models which reproduce the dynamic phases of traffic are still under debate. For this reason an object-oriented design of the simulator is advantageous because it allows a flexible use of different cellular automata models through inheritance of classes. The first cellular automaton model for traffic flow that was able to reproduce some characteristics of real traffic, like jam formation, was suggested by Nagel and Schreckenberg [6] in 1992. We will give a brief review of their basic model before we describe the more advanced cellular automaton model used by the simulator, which includes anticipation, brake-lights, and asymmetric rules for lane changes.

2.1 The Nagel-Schreckenberg Model

In the Nagel-Schreckenberg model the road is represented by a one dimensional lattice which is subdivided in cells with a length of 7.5 m. Each cell is either occupied by one vehicle or is empty. In every time-step $t \to t+1$ the following update rules are applied to the cars in the lattice in parallel:

- Step 1: Acceleration:
$$v_n(t+\frac{1}{3}) := \min(v_n(t)+1, v_{\max}).$$
- Step 2: Braking:
$$v_n(t+\frac{2}{3}) := \min(v_n(t+\frac{1}{3}), d_n(t)-1).$$
- Step 3: Randomization with probability constant $p \in]0,1[$:
$$v_n(t+1) := \begin{cases} \max(v_n(t+\frac{2}{3})-1, 0), & \text{with probability } p, \\ v_n(t+\frac{2}{3}), & \text{default.} \end{cases}$$
- Step 4: Move (drive):
$$x_n(t+1) := x_n(t) + v_n(t+1).$$

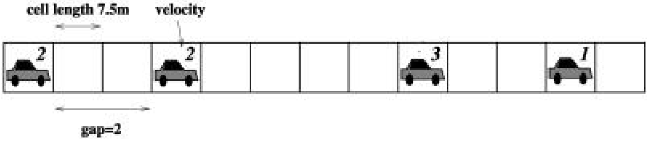


Fig. 1. Part of a road in the Nagel-Schreckenberg model

x_n is the position of the nth car and $v_n \in \{1, 2, \ldots, v_{\max}\}$ its velocity. $v_{\max}$ denotes the maximum velocity and d_n is the number of empty cells (gap) in front of car n. One time-step corresponds to 1 s in real time.

The first two rules (step 1 and 2) describe a somehow optimal driving strategy: the driver accelerates if the vehicle has not reached the maximum velocity $v_{\max}$ and brakes to avoid accidents, which are explicitly excluded. However, drivers do not react in this optimal way: they vary their speed without any obvious reason, reflected by the *braking noise* p (step 3). It mimics the complex interactions between the vehicles and is also responsible for spontaneous formation of jams.

2.2 More Realistic Cellular Automaton Model

The first model implemented in our simulator was the basic Nagel-Schreckenberg cellular automaton model, but a more realistic traffic flow is obtained by using smaller cells and by extending it with *velocity dependent randomization*, *anticipation*, and *brake lights*.

Smaller cells allow a more realistic acceleration and more speed bins. We are currently using a cell size of 1.5 m, which corresponds to speed bins of 5.4 km/h and an acceleration of 1.5 m/s^2 (0−100 km/h in 19 s). A vehicle occupies 2−5 consequent cells. By using velocity dependent randomization [7] meta-stable traffic flows are modeled by the simulator, a phenomenon observed in empirical studies of real traffic flows [12,13,14]. By including anticipation and brake lights [8,10] in the modeling, the cars not solely determine their velocity in dependency of the distance to the next car in front, but also take regard to its speed and whether it is reducing the speed or not. These modifications of the Nagel-Schreckenberg model imply that we have to add some new parameters to the model.

When the simulation algorithm decides if a car n should brake or not it does not only look how far away the next car m in front is, but makes an estimate of how far the car m will move during this time-step (anticipation). Note, that the moves are done in parallel, so the model remains free of collision. This leads to the effective gap

$$d_{n,m}^{\text{eff}}(t) := d_{n,m}(t) + \max(v_m^{\min}(t) - d_S, 0)$$

seen by car n at time t. In this formula $d_{n,m}(t)$ is the number of free cells between the front of the car n and the back of the car m, d_S is a safety distance, set equal to 6 cells (9 m) in our model, and

$$v_m^{\min}(t) := \min(d_{m,l}(t), v_m(t)) - 1,$$

where $d_{m,l}(t)$ is the number of free cells between the car m and its next car in front l, is a lower bound of how far the car m will move during this time-step.

Brake lights are a further component of the anticipated driving. They allow cars to react to disturbances in front earlier by adjusting their speed. Empirical observations suggest [15,16] that drivers react in a temporal- rather than a spatial-horizon. For this reason the velocity-dependent temporal interaction horizon

$$t_n^S(t) := \min(v_n(t), h)$$

is introduced to the model. The constant h determines the temporal range of interaction with the brake light $b_m(t)$ of the next car m in front. The car n does only react to $b_m(t)$ if the time to reach the back of the car m, assuming constant velocity ($v_n = const.$) and that the car m stands still, is less than $t_n^S(t)$, that is,

$$t_n^h(t) := \frac{d_{n,m}(t)}{v_n(t)} < t_n^S(t).$$

In our model we take h equal to 7 s.

The third modification of the Nagel-Schreckenberg model implemented in the simulator is a velocity dependent randomization, which means that the probability constant p is replaced with a probability function dependent on the velocity of the car. Further, the probability is also a function of the brake light of the next car in front. In every time-step for every car n with car m next in front, the probability that the car n brakes is

$$p = p(v_n(t), b_m(t)) := \begin{cases} p_b, & \text{if } b_m(t) = \text{on and } t_n^h(t) < t_n^S(t), \\ p_0, & \text{if } v_n(t) = 0 \text{ and } (b_m(t) = \text{off or } t_n^h(t) \geq t_n^S(t)), \\ p_d, & \text{default.} \end{cases}$$

In our model we take p_b equal to 0.96, p_0 equal to 0.1, and p_d equal to 0.01.

To sum up, to move the cars forward in the network the algorithm executes the following steps in parallel for all cars n:

- Step 0: Initialization:
  For car n find next car in front m. Set $p := p(v_n(t), b_m(t))$ and $b_n(t+1) := \text{off}$.
- Step 1: Acceleration:

$$v_n(t+\frac{1}{3}) := \begin{cases} v_n(t), & \text{if } b_n(t) = \text{on or } (b_m(t) = \text{on and } t_n^h(t) < t_n^S(t)), \\ \min(v_n(t)+1, v_{\max}), & \text{default.} \end{cases}$$

- Step 2: Braking:

$$v_n(t+\frac{2}{3}) := \min(v_n(t+\frac{1}{3}), d_{n,m}^{\text{eff}}(t)).$$

  Turn brake light on if appropriate:

$$\text{if } v_n(t+\frac{2}{3}) < v_n(t), \text{ then } b_n(t) := \text{on}.$$

- Step 3: Randomization with probability p:

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

  Turn brake light on if appropriate:

$$\text{if } p = p_b \text{ and } v_n(t+1) < v_n(t+\frac{2}{3}), \text{ then } b_n(t+1) := \text{on.}$$

- Step 4: Move (drive):

$$x_n(t+1) := x_n(t) + v_n(t+1).$$

2.3 Free Lane Changes

Free lane changes are needed so that cars can overtake slower driving cars and lorries. When designing rules for the free lane changes, one should take care of that cars taking over do not disturb the traffic on the lane they use to overtake to much, and one has to take account of German laws, which prohibit overtaking a car to the left. Further, it is advantageous to prohibit lorries to drive on the leftmost lane in the simulation, because a lorry overtaking another lorry forces all cars on the left lane to reduce their velocity.

One more variable is needed for the free lane changes, $l_n \in \{\text{left}, \text{right}, \text{straight}\}$ notes if the car n should change the lane during the actual time-step or not. This variable is not needed if the lane changes are executed sequentially, but we prefer a parallel update of the lane changes for all cars and that renders this variable necessary. For the left free lane changes the simulator executes the following steps parallel for all cars n:

Overtake on the Lane to the Left:

- Step 0: Initialization:
  For car n find next car in front m on the same lane, next car in front s on the lane left to car n, and the next car r behind car s. Set $l_n :=$ straight.
- Step 1: Check lane change:

$$\text{if } d^{\text{eff}}_{n,m}(t) < v_n(t) \text{ and } d^{\text{eff}}_{n,m}(t) < d^{\text{eff}}_{n,s}(t) \text{ and } d^{\text{eff}}_{r,n}(t) > v_r(t), \text{ then set } l_n := \text{left.}$$

- Step 2: Do lane change:

$$\text{if } l_n = \text{left, then change lane for car } n \text{ to the left.}$$

The definition of the gaps $d^{\text{eff}}_{n,s}(t)$ and $d^{\text{eff}}_{r,n}(t)$ is an obvious extensions of the definition above, one simply considers a copy of the car n on its left side. These overtake rules used by the simulator can verbally be summed up as follows: First, a vehicle checks if it is hindered by the predecessor on its own lane. Then it has to take into account the gap to the successor and to the predecessor on the lane to the left. If the gaps allow a safe change the vehicle moves to the left lane. For the right free lane changes the simulator executes the following steps parallel for all cars n:

Return to a Lane on the Right:

- Step 0: Initialization:
  For car n find next car in front s on the lane right to car n and next car r behind car s. Set $l_n :=$ straight.
- Step 1: Check lane change:

$$\text{if } d^{\text{eff}}_{n,s}(t) > v_n(t) \text{ and } d^{\text{eff}}_{r,n}(t) > v_r(t), \text{ then set } l_n := \text{right}.$$

- Step 2: Do lane change:

$$\text{if } l_n = \text{right, then change lane for car } n \text{ to the right.}$$

Thus, a vehicle always returns to the right lane if there is no disadvantage in regard to its velocity and it does not hinder any other car by doing so.

It should be noted that it is not possible to first check for all lane changes to the left and to the right and then perform them all parallel without doing collision detection and resolution. This comes because there are autobahns with three lanes and more. To overcome this difficulty the simulator checks and performs the left lane changes in every odd time-step and the right lane changes in every even time-step. For a systematic approach to multi-lane traffic, i.e., lane-changing rules, see, for example, [17]. For a detailed discussion of the different models see [18,19,20] and the references therein.

3 Network Structure

A crucial point in the design of every simulator is the representation of the road network. Like in other simulators (e.g., [1,21]) the network consists of basic elements, links and nodes. A link is a directed bundle of parallel lanes or, more casually, simply a piece of autobahn. A vehicle on a link has local coordinates (cell and lane) with respect to the link. A node is a connection between two links. It stores information about where the exit is on the link to be left, about how to leave the link (lane change, drive out of it), and how to calculate the new local coordinates on the target link (cell offset, lane offset).

By combining links and nodes one is able to build the complex structures of the autobahn network. Examples for these structures are:

- junctions, where vehicles enter or leave the autobahn,
- intersections, at which two autobahns are connected, and
- triangular intersections, where two autobahns meet, but one ends.

The complexity of an intersection can be derived from Fig. 2. Other geometries are rarely found in the autobahn network in NRW. However, they can be constructed easily with the elements used here.

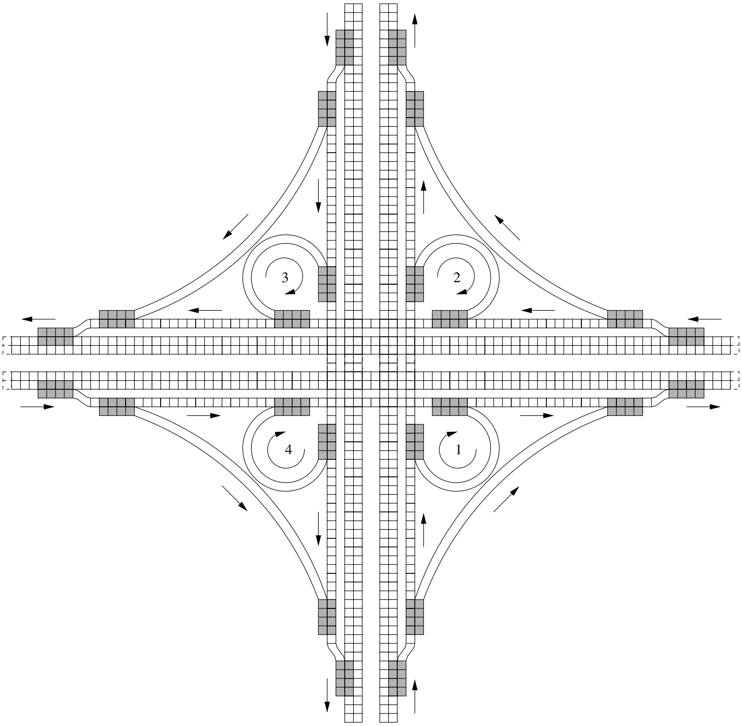


Fig. 2. The complex structure of an intersection.

Using these links and nodes the autobahn network of NRW was reconstructed. It comprises 3,688 links, 830 on- and off-ramps, and 67 intersections. The overall length of the lanes is approximately 12,200 km, corresponding to more than 8 million cells. The data used for the network were extracted from the *NW*-SIB, a Geographic Information System (GIS) database provided by the state of NRW.

4 Additional Rules for Complex Real Networks

The cellular automaton model for traffic flow used by the simulator was designed to be able to reproduce the main aspects of the fundamental diagram for real traffic flows (car flow as a function of cars per km). This ability was verified by testing it on topologically simple networks. When simulating the traffic on a large and topologically complex network, like the autobahn network in NRW, some extensions to the cellular automaton model have to be considered. One is the guidance of vehicles and another is a strategy to integrate the measured flow from the loop detectors into the simulation.

4.1 Guidance of the Vehicles

A real driver usually has the intention to reach some goal with his driving. This makes it necessary to incorporate routes in the modeling. In principle, there are two different strategies to solve this problem. One can assign an origin and a destination to the road user and then guide him through the network according to this route [2,4]. For our network origin-destination information with a sufficient temporal and spatial resolution is not available. Therefore, the vehicles are guided in the network according to the probabilities calculated on the basis of the measured data. This means that a car is not guided through the whole network, but every time it reaches a new link it will decide in accordance with the measured probabilities how it leaves the link.

To implement this we use forced lane changes. Forced lane changes are necessary so that the cars can drive from on-ramps on the autobahn, from the autobahn on off-ramps, when the autobahn narrows, and when cars drive from one particular section of the autobahn on another over an intersection. Forced lane changes differ from free lane changes in a fundamental way. While free lane changes give vehicles the opportunity to overtake vehicles driving slower and thus reduce disturbances, forced lane changes stem from the need to reach a node and are obviously an additional source for disturbances.

The simulator uses gradually increasing harsh measures to force lane changes. At the beginning of an area where a car could change to the target lane, it does so, if the gap is sufficiently large and no car is severely hindered. At the end of the area it will bully into any gap regardless of velocity differences. Further, a vehicle driving on its target lane should not leave the lane to overtake. An efficient implementation of this strategy is to store the lane change information in the cells. This gives a fast access through the coordinates of a vehicle. Of course this information depends on the node chosen and whether the vehicle is a lorry or a passenger car. Because of this every link has several versions of the lane change information.

4.2 Sensor Input

To incorporate the real world measurements from the loop detectors into the simulation vehicle-moving, inserting, and removing algorithms have to be applied. This is done at the so-called checkpoints, which are located at those places in the network where a complete cross-section is available, i.e., all lanes are covered by a loop detector. Every time, when checkpoint-data is provided the simulator uses the measured values to adjust the traffic state in the simulation. The first step is to try to move vehicles behind the checkpoint in front of it and vice versa. If this is not enough to adjust the traffic state, cars are inserted or removed. This should be preferred to pure insert/removal strategies, because these can completely fail due to positive feedback if a non-existing traffic jam is produced by the simulation. In this case the simulation measures a low flow in comparison with the real data, so cars are added periodically to the ever growing traffic jam leading to a total breakdown.

For realistic results it is further important to minimize the perturbation of the dynamics present in the network due to the data integration. Therefore, we propose a method which follows the idea to add the cars to the network "adiabatically", i.e., without disturbing the system. If the number of cars crossing the checkpoint is lower than measured by the detector, cars are inserted with regard to the measured mean velocity and mean gap, so that the system is not disturbed, i.e., no car has to brake due to the insertion. This method is therefore called "Tuning of the Mean Gap" [3]. If it is not possible to add the required number of cars some vehicles are left out. Although this is not correct, it is more important to keep the dynamics of the system untouched. It turns out that the strategy is capable of reproducing the traffic state quite well [3].

5 Summary and Outlook

In this paper we present a simulator of the autobahn network in North Rhine-Westphalia. The simulator uses an advanced cellular automaton model of traffic flow and adjusts the traffic state in accordance with measurements of the real traffic flow provided by more than 3,500 loop detectors installed locally on the autobahn. The cellular automaton model, the abstraction of the network, the guidance of the vehicles, and data integration strategies to periodically adjust the traffic flow in the simulation in accordance with the measured flow on the autobahn were discussed.

The simulation performs in a multiple real-time on a modern personal computer (Athlon$^{\text{TM}}$ XP 1600+, 512 MB DDR-RAM) and in regard to the ever growing computational power it looks promising to combine the simulation with historical data for traffic forecast. A further application for the simulator is to research the influence of new roads or road-works on the traffic flow. Finally, we are extending the simulator so it can be used for the research of traffic flow control. This is not as simple as it might seem, because any information about the current traffic state available to the public is likely to influence the strategy of the drivers [22].

Acknowledgement. The authors would like to thank the "Landesbetrieb Straßenbau NRW" for data support and the Ministry of Economics and Small Businesses, Technology and Transport for North Rhine-Westphalia for financial support.

References

1. Esser J., Schreckenberg M. (1997) Microscopic simulation of urban traffic based on cellular automata. Int. J. of Mod. Phys. C **8**, 1025–1036
2. Nagel K., Esser J., Rickert M. (2000) Large-scale traffic simulations for transport planning. In: Stauffer D. (Ed.), Ann. Rev. of Comp. Phys. **VII**, 151–202, World Scientific, Singapore
3. Kaumann O., Froese K., Chrobok R., Wahle J., Neubert L., Schreckenberg M. (2000) On-line simulation of the freeway network of North Rhine-Westphalia. In: Helbing D., Herrmann H., Schreckenberg M., Wolf D. (Eds.) (2000) Traffic and Granular Flow '99. Springer, Heidelberg, 351–356

4. Rickert M., Wagner P. (1996) Parallel real-time implementation of large-scale, route-plan-driven traffic simulation. Int. J. of Mod. Phys. C **7**, 133–153
5. Schreckenberg M., Neubert L., Wahle J. (2001) Simulation of traffic in large road networks. Future Generation Computer Systems, **17**, 649–657
6. Nagel K., Schreckenberg M. (1992) A cellular automaton model for freeway traffic. J. Physique I **2**, 2221–2229
7. Barlovic R., Santen L., Schadschneider A.,Schreckenberg M. (1998) Metastable states in cellular automata for traffic flow. Eur. Phys. J. B **5**, 793–800
8. Barrett C., Wolinsky M., Olesen M. (2000) Emergent local control properties in particle hopping traffic simulations. In: Helbing D., Herrmann H., Schreckenberg M., Wolf D. (Eds.) (2000) Traffic and Granular Flow '99. Springer, Heidelberg
9. Knospe W., Santen L.,Schadschneider A., Schreckenberg M. (1999) Disorder effects in cellular automata for two-lane traffic. Physica A **265**, 614-633
10. Knospe W., Santen L.,Schadschneider A., Schreckenberg M. (2000) Towards a realistic microscopic description of highway traffic. J. Phys. A **33**, L1–L6
11. Pfefer R. (1976) New safety and service guide for sight distances. Transportation Engineering J. of Am. Soc. of Civ. Engineers **102**, 683-697
12. Helbing D. (1996) Empirical traffic data and their implications for traffic modelling. Phys. Rev. E **55**, R25
13. Kerner B., Rehborn H. (1997) Experimental properties of phase transitions in traffic flow. Phys. Rev. Lett. **79**, 4030-4033
14. Treiterer J. (1975) Investigation of traffic dynamics by areal photogrammatic techniques. Tech. report, Ohio State University Tech. Rep. PB 246, Columbus, USA
15. George H. (1961) Measurement and Evaluation of Traffic Congestion. Bureau of Highway Traffic, Yale University
16. Miller A. (1961) A queuing model for road traffic flow. J. of the Royal Stat. Soc. **B1**, 23, University Tech. Rep. PB 246, Columbus, USA
17. Nagel K., Wolf D. E., Wagner P., Simon P. (1998) Two-lane traffic rules for cellular automata: A systematic approach. Phys. Rev. E **58**, 1425–1437
18. Helbing D., Herrmann H., Schreckenberg M., Wolf D. (Eds.) (2000) Traffic and Granular Flow '99. Springer, Heidelberg
19. Schreckenberg M., Wolf D. (Eds.) (1998) Traffic and Granular Flow '97. Springer, Singapore
20. Chowdhury D., Santen L., Schadschneider A. (2000) Statistical Physics of Vehicular Traffic and Some Related Systems. Physics Reports **329**, 199–329
21. Yang Q., Koutsopoulos H. N. (1996) A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transp. Res. C **4**, 113–129
22. Wahle J., Bazzan A., Klügl F., Schreckenberg M. (2000) Anticipatory Traffic Forecast Using Multi-Agent Techniques. In: Helbing D., Herrmann H., Schreckenberg M., Wolf D. (Eds.) Traffic and Granular Flow '99. Springer, 87–92

Cellular Automata Based Temporal Process Understanding of Urban Growth

Jianquan Cheng and Ian Masser

Urban Planning and Management Division,
International Institute for Geo-Information Science and Earth Observation (ITC),
Hengelosestraat 99, P.O.Box 6, 7500 AA, Enschede,
The Netherlands
{Jianquan, Masser}@itc.nl

Abstract. Understanding of urban growth process is highly crucial in making development plan and sustainable growth management policy. As the process involves multi-actors, multi-behavior and various policies, it is endowed with unpredictable spatial and temporal complexities, it requires the occurrence of new simulation approach, which is process-oriented and has stronger capacities of interpretation. In this paper, A cellular automata-based model is designed for understanding the temporal process of urban growth by incorporating dynamic weighting concept and project-based approach. We argue that this methodology is able to interpret and visualize the dynamic process more temporally and transparently.

1 Introduction

The city is a typical complex system, which is characterized with a self-organization property [1, 2]. Understanding of urban development process is highly crucial in urban development planning and sustainable growth management. Urban development process involves multi-actors, multi-behaviors and various policies, which results in their spatial and temporal complexities. Due to the hidden complexity of reality, our science has become less orientated to prediction but more an aid to understanding, to structure debate [3]. Couclelis [4] first put forward the idea of spatial understanding support system (SUSS). Proper understanding of complex system is the prerequisite to its prediction.

Cellular automata (CA), a technique developed recently, has been receiving more and more attention in GIS modeling due to its simplicity, transparency, strong capacities for dynamic spatial simulation, and innovative bottom-up approach. Numerous literatures can be seen even in the field of urban growth CA modeling on various scales (regional, municipal and town) e.g. [5-9].

In contrast to classic CA, more and more modifications have been made to improve its modeling capacity such as multi-states of cell, relaxing size of neighborhood with

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 325-336, 2002.

distance-decay effects, and linkage with complexity theory [10]. As the core of CA model, transition rules have also been modified and expanded to include notions such as hierarchy, self-modification, probabilistic expressions, utility maximization, accessibility measures, exogenous links, inertia, and stochasticity; in fact, many-if not all-urban CA bear little resemblance to the formal CA model [11]. Nevertheless, interpretation of transition rules, which is highly important for urban planners, still receives little attention in process modeling. Most studies focus on how to make complicated models.

The previous studies of urban CA models ignore the fact that urban growth is a dynamic process rather than a static pattern. Similar patterns, the final outputs of CA simulation do not indicate similar processes. Thus, the transition rules tested are not evidential to explain the complex spatial behavior. Therefore, process rather than pattern oriented simulation should be the major concern of urban growth CA modeling. This point is started to be aware in some journals [11]. In GIS field, [12] applied fuzzy spatio-temporal interpolation to simulate changes that occurred between snapshots registered in a GIS database. The main advantage of the research lies in its flexibility to create various temporal scenarios of urbanization processes and to choose the desired temporal resolution. The author also declared that the approach does not explicitly provide causal factors, thus it is not an explanatory model.

In summary, we need to take spatial and temporal process into CA modelling to achieve stronger interpretation capacities of causal factors. With this in mind, this paper is organized into four sections. Following the introduction, the next section discusses in detail a proposed methodology, which mainly comprises dynamic weighting and mathematical models of local growth. One of major features in our CA model is to utilize dynamic weighting for linking pattern and process. Sections 3 moves to the implementation of the methodology by a case study area from Wuhan City, P.R.China. Section 4 ends with some discussion and conclusions.

2 Methodology

As a typical self-organizing social-economic system (SOS), urban system modelling must call for an innovative bottom-up simulation approach. Complexity of urban growth comprises the multiplicity of spatial patterns and social economic processes, nonlinear interactions among numerous components and heterogeneity over a variety of spatial and temporal scales. Intuitively, the complexity of urban growth process can be transferred into spatial and temporal complexity when projected onto land system. The understanding rather than prediction of urban growth process based on SOS mechanisms is a feasible way. This understanding must be based on the integration of top-down and bottom-up approach.

As an effective bottom-up simulation tool, CA firstly offers a new thinking way for dynamic spatial modelling, and secondly provides a laboratory for testing human being's decision making. However, the complexity of urban growth determines that the classic CA must be modified in order to deal with practical issues (the details are described in [4]). In this paper, we develop a modified CA model for understanding the spatial and temporal processes of urban growth based on dynamic weighting concept and project-based approach to be described below.

2.1 Temporal Heterogeneity (Dynamic Weighting)

[13] applied logistic regression method for modelling land development patterns in two periods (1979-1987 and 1987-1992) based on parcel data extracted from aerial photos. They found that the major determinants of land development have changed significantly, e.g. from proximity to inter-city highways to proximity to city streets. Likewise, if we shrink the long period (1979-1992) to shorter period such as 1993-2000 and also from the whole city to smaller part. The same principle should be working as well. As a consequence, the factors influencing local growth should be assigned with dynamic weight values.

Obviously temporal pattern from time t_1 to t_n, is influenced by highly complicated spatial and temporal processes. However, similar patterns can result from numerous different processes. As a consequence, the understanding of process is more important than that of pattern. Pattern is only a phenomena but process is the essence. The interaction between pattern and process is a non-linear iteration function like other phenomena: fractal, chaos etc. which are typically represented by non-linear iteration function (eq.1).

$$X_{t+1}=f(x_t) \tag{1}$$

In the case of urban growth, temporal complexity might be indicated by:

- Compared with major roads, minor roads especially in new zones, which are also new development units, may have certain time delay in affecting local growth, i.e. between T_0 and T_n, not immediately from T_0;
- The spatial impacts of various factors such as road, center, rail are not simultaneous temporally in effecting local growth;
- Neighbourhood effects may suffer from temporal variation, for example, it may be stronger in T_0 than in T_n, or vice versa.

Figure 1 is only an example of temporal complexity involved in urban growth. T_1,T_2,T_3 indicate time series. The same spatial pattern results from three (in reality, more) distinguishing temporal process, which reflect the spatial and temporal interactions between road-influenced and center-based local growth. The arrows indicate the trend of temporal development, from which we can define them as three different temporal processes (convergence, sequence and divergence). The basic principle behind this phenomena is that various physical factors like road and center point take temporally varied roles in the course of local growth. In the first one, road is more important than center at time T_1, but less important at T_2. It means that local growth occurs along road first and then moves to the center. The third one takes an opposite effect. If we use D to denote the total amount of local growth, D_l for the lower part along road, D_u for the upper part along road, D_c for the center part and D_t for the continuous development till time t. Hereby, $D{=}D_n = D_l + D_u + D_c$. W_r and W_c represent the weight value of factor ROAD and CENTER respectively. We are able to detect the following rules:

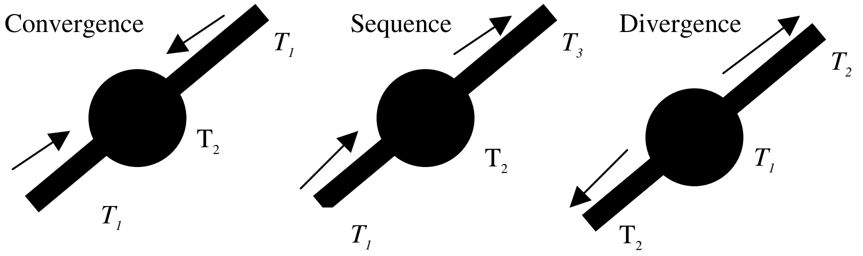


Fig. 1. Temporal Heterogeneity

In Convergence: if $D_t < D_l + D_u$, $W_r \to 1$, $W_c \to 0$ (at T_1); if $D_t > D_l + D_u$, $W_r \to 0$, $W_c \to 1$ (at T_2);

In Sequence: if $D_t < D_l$, $W_r \to 1$, $W_c \to 0$ (at T_1); if $D_t > D_l$ and $D_t < D_l + D_c$, $W_r \to 0$, $W_c \to 1$ (at T_2); if $D_t > D_l + D_c$ and $D_t < D$, $W_r \to 1$, $W_c \to 0$ (at T_3);

In Divergence: if $D_t < D_c$, $W_r \to 0$, $W_c \to 1$ (at T_1); if $D_t > D_c$ and $D_t < D$, $W_r \to 1$, $W_c \to 0$ (at T_2).

Here, symbol "→" means "approaching to or close to". The three cases imply that the temporal complexity could be represented and understood through dynamic weighting. It means that factor weight is a function of temporal development amount, i.e.

$$W_i\,(t) = f_i\,(D_t) \tag{2}$$

In principle, the function $f_i\,(D_t)$ should be continuous, which may be a step linear or more complicated non-linear function. In practice, the function $f_i\,(D_t)$ has to be simplified through discretization. It implies that the whole period needs to be divided into few cases t_1~t_n, each with varied weight values.

2.2 Generalized Mathematical Models

Urban growth process is effected by many factors, which may change their influential roles spatially and temporally. The spatial heterogeneity phenomena (heterogeneity in a spatial context means that the parameters describing the data vary from place to place) suggests that the search for general laws frequently fail in practice and it is being replaced by local area analysis like Geographically Weighted Regression (GWR) and others in the field of spatial statistics. As a consequence, a project-based local growth modelling is more reasonable for understanding of complex urban growth process. The spatial extent described below is limited to individual large-scale project.

$$L(t)|_{t=n} = L_d(x,y) \tag{3}$$

Here, L_d is the actual area of land development of one project d in the whole period [$t=1\sim n$]. L_d in principle should result from traditional top-down socio-economic models. Here it is assumed to be a known value. $L(t)$ is the simulated area of land development of same d till time t. $L(t)$ will be calculated from the later section. As an example, we only refer to project d; the others follow the same procedures.

$$C_i(t) = f_i(L(t)) \tag{4}$$

Here, $C_i(t)$ is the temporal contribution of factor i to $L(t)$ at time t, which is a function rather than a subjective parameter. We strongly argue that C_i should vary with $L(t)$. $W_i(t)$ in eq. 5 is a relative weight of factor i $(\ 0\square\ W_i(t)\ \square 1)$.

$$W_i(t) = \frac{C_i(t)}{\sum_i C_i(t)} \tag{5}$$

To some extent, eq. 4 and 5 indicate a dynamic feedback between $w_i(t)$ and $L(t)$, which can be utilized to represent the complex interactions between pattern and process. Actually, $L(t)$ is to quantify temporal pattern, and the process can be described by multiple $w_i(t)$. The non-linear iteration function $f(L(t))$ (together with $L(t)=g(W_i(t)$) exhibit the complexity property of the interactions between pattern and process like *'chicken and egg'*. When $f_i(L(t))$ is constant, C_i is becoming universe temporally, which is treated in most CA applications, however, it can not model the temporal process. The design of function $f_i(L(t))$ is a critical point. It needs numerous experimental tests, which is based on the theoretical understanding of the interaction *"chicken and egg"*. However, model from the experimental tests with higher temporal resolution are able to theoretically abstract the hidden temporal rules. The development potential of each cell j at time t is defined as:

$$P_j(t) = (1 + ln(\xi)^{\alpha})(\sum_{i=1}^{k} W_i(t) * V_{ij}(t)) \prod_{i=k+1}^{m} \omega_i \qquad (6)$$

Assuming that totally *m* constraints (*1≤ i ≤ m*) are considered, when *k+1≤ i ≤ m,* ω_i (binary variable: 0 or 1) are restrictive constraints such as water body, slope etc, which may include local, regional and global levels with equal weight.

When *1≤ i ≤ k*, they are non-restrictive constraints. $W_i(t)$ is the weight value of constraint *i* computed from eq.5. For proximity variables like the distance to major road, here a negative exponential function is employed to calculate $V_{ij}(t)$. Urban models based on economic theory [14], and discrete choice theory [15] had made widespread uses of the negative exponential function.

$$V_{ij}(t)=e^{-\phi\, dij} \qquad 0<V_{ij}(t)<1 \qquad (7)$$

d_{ij} is the value of proximity variable *i* at cell *j.* ϕ is the density gradient for quantifying its spatial influence. Usually, $0<\phi<1$, and ϕ varies with factor *i*. Eq.7 is actually to standardise non-restrictive variables.

In order to generate the patterns that are closer to reality, a stochastic disturbance is introduced as $(1+\ln(\xi)^{\alpha})$ [16]. ξ is a random variable within [0~1]. α is a parameter controlling the size or strength of the stochastic perturbation. *P(t)* represents the potential or probability of development of cell *j* at time *t,* which is the major driving force of local growth.

In our model, neighbourhood size is not universal globally and is locally parameterised, which varies with different projects. Neighbourhood effect i.e. 'action-at-distance' is also represented as one or two non-restrictive variables in eq.6, which indicate the spatial influences of developed cell (including both the new and the old) on its surrounding sites.

$$P_j(t) \rightarrow P_{j'}(t) \qquad (8)$$

$$\Delta L(t) = L(t) - L(t-1), \quad L(0)=0 \qquad (9)$$

Eq.8 builds a transition from $P_j(t)$ to $P_{j'}(t)$, which is to reorder from maximum to minimum. *ΔL(t)* in eq.9 is the land development demand at the phase from *t-1* to *t* as *L(t)* is the accumulative amount of land development till time *t.*

Based on equation 8 and 9, the state of cell *j* at time *t+1* can be determined as follows:

$$S_j(t+1)= \begin{cases} 1 & \text{if } P_j(t)=P_{j'}(t) \text{ and } j' \in [1 \sim \Delta L(t)] \\ 0 & \text{if not} \end{cases} \qquad (10)$$

Simply, totally *ΔL(t)* cells will be selected at time *t* for the transition from developable land (0) to urban (1) according to their development potential values $P_j(t)$. *L(t)* is to be determined as below.

Another advantage of project-based CA modelling is able to control the temporal development pattern of each project. Previous studies suggest that urban development process (*L(t)* in eq.4) follows a logistic curve over time [17]. The logistic curve is illustrated as eq. 11.

$$L(t)=1/(a+b*e^{(-c*t)}) \qquad (11)$$

Assuming that $L(0)=L_0=1/(a+b)=1$, $L(n)=L_n=1/(a+be^{(-cn)})=L_d$, the parameters *a* and *b* can be calculated as the functions of parameter *c* (eq.12):

$$a=1-b \qquad b=\frac{1-ln}{ln(e^{-cn}-1)} \qquad (12)$$

The shape of logistic curve usually represents the speed of urban development over time, which is controlled by the parameter *c* and *n*. Here, in simplicity, temporal control is classified as three types: slow growth, normal or basic growth and quick growth, which indicates three distinguishing scenarios (eq.13). Of course, you can define more classes or even use fuzzy logic.

$$\begin{array}{c} \text{Quick growth: } c*n > 25 \\ \text{Basic growth: } c*n < 25 \text{ and } > 15 \\ \text{Slow growth: } c*\mathrm{n} < 15 \end{array} \qquad (13)$$

The selection of temporal control pattern is a top-down process of decision-making as shown in equation 14. Where *y* denotes the real time-year (*1~m*) such as 1993 (*y*=0) and 2000 (*y*=7), which is different from iteration number *t* (*1~n*) in simulation.

$$G(y)= L_d(y) \qquad y \le m \qquad (14)$$

G(y) denotes the total growth of the whole study area till year *y*, $L_d(y)$ represents the total growth of only project *d* till year *y*. The assignment of $L_d(y)$ should be determined from a top-down social-economic model. Eq.14 also offers a link between local growth and global development. It is a feedback between top-down and bottom-up decision-making.

$$L_i(y)=h(L_i(t)\,)\quad y=1,\,2,...m;\quad t=1,\,2,\,...,n;\;\; n>m \tag{15}$$

Eq.15 establishes a transition from $L_i(t)$ *to* $L_i(y)$. In the previous researches of CA application, a linear function is applied, i.e. $t=\lambda * y$. Here λ is assumed to be a constant, which means equal growth rate. In reality, function *h* could be a non-linear function of iteration number *t*, which can be tested experimentally through visual exploration.

3 Implementation

3.1 Case Study

Wuhan is the largest mega city in central China. In1999, it had around 4 million non-agricultural population, 4 times more than that of 1949. During the last 5 decades, Wuhan underwent rapid urban growth from 3000 ha of built-up area in 1949 to 3,0151 ha in 2000. As a result, Wuhan is a fresh and typical case for understanding the dynamic process of Chinese cities.

With the assistance of topographic maps of 1993 and SPOT Pan/Xs images of 2000, we found that land cover change in the period 1993-2000 was dominated by strong spatial agglomeration of a few large-scale projects, which take over 60% of total change. As a consequence, the understanding of local growth process of each project is highly crucial to that of whole study area. Here, *Zukou* car manufacturing center, the largest project, is taken as a case study for testing the methodology proposed. The influential factors include major roads, minor roads, master planning, physical constraints from water body. The cell size in this research is 100x100 m^2.

3.2 CA Simulation

The validation of parameters has been proven difficult for urban CA modelling [6, 16] in particular when factors and parameters considered are voluminous. Here, we think that manual test is much quicker and also more interpretable, which is based on the modeler's reasonable understanding of urban growth process and visual exploration of model outputs. The impact of each factor or parameter is assessed by changing its value and holding the others constant. In this case study, the major parameters include "distance to minor road" (OR), "distance to major road"(MR), "distance to center"(CN), "density of neighbouring new development" (DN), " OR density gradient", "MR density gradient", "CN density gradient", and "Master planning". So their relative importance (weight values) could be assigned quantitatively by manual test, further improvement can be done by limited number of automatic search like 1000 iterations.

Model accuracy depends on measure approach to comparing simulated and actual patterns. [6] chose four ways to statistically test the degree of historical fit (three r-squared fits and one modified Lee-Sallee shape index). The last one is a measurement of spatial fit between the simulated and the actual growth. Supposed that the actual is denoted by set A, the simulated B; the index is equal to $(A \cap B)/(A \cup B)$ mathematically. This simple measure of shape was computed through counting the union and the intersection of their total areas on a pixel x pixel basis, and then dividing the intersection by the union. For a perfect match, the Lee Sallee measure gives a value of 1.0, and for all others ranging from 0 to 1. Clark reported the practical accuracy of his model is only 0.3 [6]. Other measures like fractal and Moran I index are also frequently used for global pattern comparison e.g.[18]. In this paper, we use consistency co-efficient (*CC*) (spatial match between the simulated and the actual) and Lee-Sallee index (*LI*) for goodness of fit evaluation. Mathematically, CC is equal to $(A \cap B) / A$. As the total number of pixels is set the same for the simulated as the actual, apparently here $LI=CC/(2-CC)$. Following this formula, the Lee-Sallee index of *Zuankou* is computed and listed in table 1. The model accuracy is 55% in *CC* and 39% in *LI*, which is greater than Clark's [6].

Assisted with SPOT images of 1995, 2000 and IRS images of 1997, we are able to judge the temporal development pattern of *Zuankou,* compared with other parts of Wuhan city. In 1993, *Zuankou* was still completely rural and nearly half constructed in 1995. There was not much change from 1997 and 2000. So its temporal growth pattern is defined as "Quick". The number of iteration is defined as 50 (n=50) as principally the greater the number is, the finer discriminative capacity the model has, which results in higher accuracy. Therefore, when c=0.5, c*n=25. As described in equation 15, the result of simulation is $L_i(t)$, which is different from yearly actual amount $L_i(y)$. We need a transition from $L_i(t)$ to $L_i(y)$. In simplicity, we just use equal time interval, i.e. a linear function: $y = t/7$. As t ranges from 1 to 50 and y is from 1 to 7, $L_i(y)= \Sigma L_i(t)$ $(t$ from $7*(y-1)+1$ to $7*y)$. A new layer with 7-year urban growth (from 1993 to 2000) is input into animation software for dynamic exploration. This animation is helpful for comparing the distinguishing temporal development processes of various projects.

Two models of *Zuankou* in Table 1 have similar model accuracy and also similar pattern (the CA model is over till the 28th step). However, their temporal processes shown in Figure 2 are quite different. The mode of temporal control is set the same (c=0.5). Model 1 exhibits a more random process. Model 2 shows a more organized process. Model 2 is based on the assumption that new development in *Zuankou* first occurred in the center, then along the major road and finally spread from the center. The assumption corresponds to a temporal process that is spatially controlled to by three sets of weight values (Table 1). In other words, the temporal process can enable us better understand the organized local growth. If we explore the changes of weight values, it can be found that the major changes are indicated in major road and center. As explained in section 2 (eq. 3 and 4), weight values should be the functions of temporal development demand. Table 1 also shows the functions are highly complicated in reality. A universal or standard function is not available. Rather it should be simplified and based on local knowledge. Model 2 actually is based on the interviews with local planners.

Table 1. Test of temporal heterogeneity (Zuankou)

Results	*Model 1*	*Model 2*		
Total cells	1390	1390		
Accuracy (CC)	*55%*	*55%*		
Lee-Sallee Index	*38%*	*38%*		
Stochastic (α=10%)	*1%*	*1%*		
Iteration number	*50*	*50*		
Neighb. size	*6*	*6*		
c	*0.5*	*0.5*		
Temporal division	*100%*	*<15%,*	*15%-50%,*	*>50%*
Major road (MR)	0.2	-	0.5	0.05
Minor road (OR)	0.3	-	0.1	0.15
Centers (CE)	-	0.7	-	0.5
Neighb. (new)	0.3	0.3	0.1	0.15
Master Planning	0.2	-	0.3	0.15
Total	100%	100%	100%	100%
Gradient for MR	0.0005	0.0005		
Gradient for OR	0.002	0.002		
Gradient for CE	-	0.0004		

Note("-" :0)

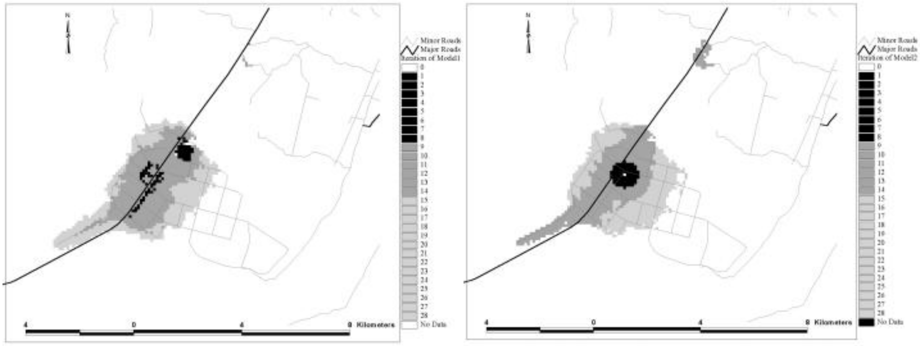


Fig. 2. Test of temporal heterogeneity (Model 1 & Model 2)

4 Discussion and Conclusions

Although the accuracy of two CA models is only 55%, simulation model accuracy, to some extent, depends on the complexity and stochasticity of real city and also the availability of more detailed information. From the previous part of this research, the accuracy of global pattern model that is based on logistic regression analysis with 10 explanatory variables is only around 70%. If more detailed data like control plan scheme is available, more rigorous model calibration will become possible. From the angle of spatial modelling, as criticized by other researchers, CA is not an appropriate tool on micro scale, we need to integrate agent-based techniques e.g.[19]. Models of complex systems with geographic properties, such as city and ecology systems, usually involve spatial and temporal processes, which are difficult to embed within proprietary GIS. Most CA software available such as AUGH, DUEM either lack GIS functions or do not fit specific complex city. A loose coupling strategy is still preferred, which is also adopted in this research.

We can not ignore the fact that any advanced modelling techniques including CA must be based on the proper understanding and abstract of the system studied. The more proper, the more accurate it is. The ability of science to understand the real world is to a large extent dependent on knowledge constrained by the limits of our understanding of complexity.

CA is only a simulation tool for testing user's understanding. Limited by existing GIS theory and technique, the identification of spatial and temporal heterogeneity can not be completed without the assistance of local knowledge as rich historical data layers do not guarantee the improvement of model calibration. It implies that local knowledge is an important ancillary data sources for CA modelling. During modeling, temporal control, dynamic weighting, and manual test need more local knowledge. For the division of temporal process, due to limited temporal resolution, local knowledge is a key source of qualitative information.

The major purpose of CA simulation is to generate alternative scenarios for decision support in a smart growth management. Apparently, the methodology developed here can be extended in this direction. As it is based on the soft systems thinking, which stresses the role of users' subjectivity; Local planners' intention can be transformed into spatially and temporally explicit weight values and certain parameters.

References

1. Xie, Y. and Batty, M., Automata-based exploration of emergent urban form. Geographical System, (1997) 83-102.
2. Allen, P.M., Cities and regions as evolutionary, complex systems. Geographical systems, (1997) 103-130.

3. Batty, M. and Torrens, P.M., Modeling complexity: the limits to prediction. CyberGeo, (2001) .
4. Couclelis, H., From cellular automata to urban models: new principles for model development and implementation. Environment and Planning B, (1997) 165-174.
5. Batty, M., Xie, Y., and Sun, Z., Modelling urban dynamics through GIS-based cellular automata. Computers, Environment and Urban Systems, (1999) 205-233.
6. Clarke, K.C. and Gaydos, L.J., Louse-coupling a CA model and GIS: long-term urban growth prediction for San Franciso and Wanshington/Baltimore. Interlational Journal of Geographical Information Science, (1998) 699-714.
7. White, R. and Engelen, G., High resolution integrated modelling of the spatial dynamics of urban and regional systems. Computers, Environment and Urban systems, (2000) 383-440.
8. Yeh, A.G. and Xia, L., A constrained CA model for the simulation and planning of sustainable urban forms by using GIS. Environment and Planning B, (2001) 733 - 753.
9. Ward, D.P., Murray, A.T., and Phinn, S.R., A Stochastically constrained cellular model of urban growth. Computers, Environment and Urban Systems, (2000) 539-558.
10. O'Sullivan, D., Graph-cellular automata: a generalised discrete urban and regional model. Environment and Planning B, (2001) 687-705.
11. Torrens, P.M. and O'Sullivan, D., Editorial: Cellular automata and urban simulation: where do we go from here? Environment and Planning B, (2001) 163-168.
12. Dragicevic, S., Marceau, D.J., and Marois, C., Space, time, and dynamics modeling in historical GIS databases: a fuzzy logic approach. Environment and Planning B, (2001) 545- 562.
13. Wu, F. and Yeh, A.G.-O., Changing spatial distribution and determinants of land development in Chinese cities in the transition from a centrally planned economy to a socialist market economy: a case study of Guangzhou. Urban Studies, (1997) 1851-1879.
14. Muth, R., Cities and Housing: The Spatial Pattern of Urban Residential Land Use, Chicago University Press, Chicago, IL. (1969)
15. Anas, A., Residential Location Markets and Urban Transportation: Economic Theory, Econometrics and Policy Analysis with Discrete Choice Models, Academic Press, New York (1982)
16. Li, X. and Yeh, A.G.-O., Calibration of cellular automata by using neural networks for the simulation of complex urban systems. Environment and Planning A, (2001) 1445-1462.
17. Herbert, D.T. and Thomas, C.J., Cities in Space: City as place, 3nd edn, David Fulton Publishers, London (1997)
18. Wu, F., An experiment on the generic polycentricity of urban growth in a cellular automata city. Environment and Planning B, (1998) 731-752.
19. Bernard, R.N., An application of agent-based modelling to planning policy: the case of rent control, in Department of urban planning and policy development, Rutgers University, (1999)

Playing with Automata. An Innovative Perspective for Gaming Simulation

Ivan Blecic[1], Arnaldo Cecchini[2], Paola Rizzi[1], and Giuseppe A. Trunfio[3]

[1] Dept. of Planning - Faculty of Urban Planning - Istituto Universitario di Architettura di Venezia; {ivan, rizzi}@iuav.it

[2] Dept. of Architecture and Urban Planning - Faculty of Architecture - University of Sassari; cecchini@uniss.it

[3] Department of Mathematics, University of Calabria; andrea.trunfio@tiscali.it

1. Cellular Automata and Territorial Planning

In this article we will advance – from different points of view – a methodological and operative proposal of an "environment" based on cellular automata useful for the design of models of some efficacy for territorial analysis and planning. Furthermore, we well suggest a modality of its use that permits the coupling of a cellular automata models with other instruments in general, and with a role-play games in particular.

1.1 The Crisis of "Old Planning"

The traditional paradigm of urban and territorial planning based on "demiurgical" conception of planner's role has, for different reasons, gone through a profound crisis: numerous and different "styles" of masterplan inspired by such conception, described efficaciously by Hall [14] and Alexander [1], have come to an arrest – after some initial positive results – in the face of certain insurmountable obstacles.

Among these obstacles, the most relevant on our opinion appear to be:

- the crisis of reference epistemological paradigms;
- the difficulty of prediction;
- the difficulty to take into account all social and individual "forces" that "give shape" to the city;
- the co-presence of many levels of decision-making instances (both in vertical and in horizontal sense);
- the expansion of phenomena related to globalisation and the affirmation of the "liberist" ideology and that of the "unique thought";
- the growth and diffusion of movements of protest and "protagonism" on various levels.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 337-348, 2002.

2. The Role of Models and Cellular Automata

Obviously, we all have our own “vision” of the future of the city, and that is obviously a *non est disputandum*: anyone, each individual, each social class, has right on its own vision, more or less freely chosen, that then compete and collide with other visions.

But if (and as long) a vision is accepted and shared, the decision about all the rest should be defined on the basis of the case in question, it is in that case a *true problem.*

But in order to solve the problem it is necessary to understand the real situation, which presupposes the use of instruments and tools for reading, interpretation and prediction, the use of a “tool-box” with all necessary models.

The role of models in the government of the city is thus central. But which models?

2.1 Characteristics of a Model

We would only like to recall the qualifying elements “good” urban models should have (maybe not all of them and not immediately):

- it should not be a black-box; it is essential that those who use it for planning as well as those whom the plan is directed to, understand how it works and why;
- a model should predict and should take into account actions and reactions of social actors as well as their interests, conscious or not, disclosed or not, rational or not;
- a model should be such as to enable the assessment of as many alternatives as possible, as well as their comprehension and their differences;
- a model should be compatible with other models, even if differing in formulation and techniques used;
- a model should be parsimonious, should not require an excessive number of variables, an excessive amount of data and an excessive computational power;
- a model should be flexible for different situations and contexts, and should permit to be fed, processed and handled with what is at hand;
- a model should be fast to build, at least compared with the time-line of the project the model is built for;
- a model should be re-usable and anyway should never be a *hapax legomenon.*

Many of the above mentioned characteristics are innate in models based on the “artificial life” approach, and more specifically based on cellular automata; many, but not all, and not all to an equally adequate extent.

We are not proposing this technique as just another occasional fashion, or as the final solution, or as the “right” model, but nonetheless we think it to be a possibly useful model for coping, in many senses, with the complexity and for obtaining a good collection of answers.

No more, no less.

Subsequently, we will describe analytically some main elements of a generalised cellular automata model.

3. A Generalised Cellular Automata Model

3.1. Aims

In order to allow a realistic modelling of real phenomena, the proposed model aims at reaching the following objectives:

- to cope with and to supersede intrinsic rigidities of classical formulation of CA;
- – the *spatial and temporal "stationarity" of neighbourhoods*. In this work, the neighbourhoods are redefined – in general – as sets of cells whose state can influence the variation of the current cell's state. The neighbourhood, that can also modify in time, is hence defined on the basis of not necessarily topological relations between objects of the simulated reality.
- – the *regularity in "discretisation" of physical system*. The graphical representations associated to cells are considered attributes whose particular appearance is not *a priori* subject to constraints of spatial or temporal regularity.
- – the *spatial and temporal "stationarity" of transition functions*. The transition function (transition rule) of a cell is defined as a rule depending also on local cell's parameters as well as on global constants and variables.
- – the *limitations with respect to external events*. By introducing the concept of variable parameters, it is possible that global phenomena influence the evolution of the simulation, even on local level.
- to design and develop a programmable simulation environment based on CAs with features permitting the following:
- – to separately model phenomena of different nature interacting within the same physical scenario: this is obtained with the structuring in *layers* of the scenario. Within different layers, various relevant phenomena are simulated according to assigned rules.
- – to include within a CA-based simulation other calculation models that dynamically determine values of one or more variable parameters.
- – to obtain an interactive simulation environment capable to import and export scenarios from and into a GIS in an automatic or a guided manner.

3.2. The Model

The founding element of any CA model is the *cell*. In our CA, on the basis of the homogeneity of forms assumed by certain attributes, *n* different *types* of cells are defined.

Let $\mathbf{c}^{(i,t)}$ indicate the configuration of a generic cell, of *i*-th type, at the moment *t* of the simulation.

A set $\boldsymbol{C}$ of $\mathbf{c}^{(i,t)}$ cells homogenous by type belongs to a *layer of cells*, which will be indicated as $\mathbf{L}^{(i,t)}$ and is defined as follows:

$$\mathbf{L}^{(i,t)}(\boldsymbol{C}, \mathbf{P}_{\mathrm{L}}) \qquad (1)$$

where $\mathbf{P}_{\mathrm{L}}=\{p_{\mathrm{L}1}, p_{\mathrm{L}2},\ldots, p_{\mathrm{L}c}\}$ is a set of *c layer's parameters* that, according to a transition function and together with other values, concur to the evolution of the scenario. With the notation $\mathbf{L}^{(i,t)}.\boldsymbol{C}$ and $\mathbf{L}^{(i,t)}.\mathbf{P}_{\mathrm{L}}$ will be indicated the elements of $\mathbf{L}^{(i,t)}$ (each layer can contain only cells of one type; it is not excluded more than one layer containing cells of the same type).

Thus, the configuration $\mathrm{CA}^{(t)}$ of a specific cellular automaton at the moment t is entirely defined by the pair $\mathbf{P}_{\mathrm{G}}$ and $\mathbf{L}$:

$$\mathrm{CA}^{(t)}=(\mathbf{P}_{\mathrm{G}},\ \mathbf{L}) \tag{2}$$

where:

- $\mathbf{P}_{\mathrm{G}}=\{p_{\mathrm{G}1}, p_{\mathrm{G}2},\ldots, p_{\mathrm{G}g}\}$ (also $\mathrm{CA}^{(t)}.\mathbf{P}_{\mathrm{G}}$) is a set of g global parameters that concur in regulating the evolution of the CA according to cells' transition functions
- $\mathbf{L}=\{\mathbf{L}^{(1)}, \mathbf{L}^{(2)},\ldots, \mathbf{L}^{(l)}\}$ (also $\mathrm{CA}^{(t)}.\mathbf{L}$) is a set of all l layers of cells.

Both the parameters $p_{\mathrm{G}k}$ and the parameters $p_{\mathrm{L}k}$ can assume values in predefined domains, continuous or discrete, and can vary in time according to assigned laws.
In particular, it is assumed that:

– for a generic p_{Lk}, the possibility to vary with respect to the moment t, with respect to the values of other parameters of layer $p_{Lj} \in \mathbf{P}_{Lk} \subset \mathbf{P}_{L}$ where $j \neq k$, and with respect to global parameters according to an assigned function $p_{Lk} = \mathrm{f}_k(t, \mathbf{P}_{\mathrm{L}k}, \mathbf{P}_{\mathrm{G}})$;

– for a generic p_{Gk}, the possibility to vary with respect to the moment t, and with respect to the values of other global parameters $p_{Gj} \in \mathbf{P}_{Gk} \subset \mathbf{P}_{G}$ where $j \neq k$ and according to an assigned function $p_{Gk} = \mathrm{f}_k(t, \mathbf{P}_{Gk})$;

– for both types of parameters, the possibility to depend of a generic calculation model that evolves in parallel with the cellular automaton.

3.3. The Cell

The generic cell of i-th layer at the instant t is defined as:

$$\mathbf{c}^{(i,t)}\ (s, \mathbf{H}, \mathbf{V}, \mathbf{P}, \phi, \Sigma, o) \tag{3}$$

where the symbols between the parenthesis indicate *attributes* characterising the cell. As we shall explain later, the form attributes assume determines the type, or better the *class*, of the cell. Being $\mathbf{c}^{(i,t)}.\mathrm{X}$ the attribute X of the cell $\mathbf{c}^{(i,t)}$ in the instant t of the simulation, in (4) we will have that:

- s is the cell's *state* that assumes values from a set of discrete of continuous states $\boldsymbol{S}^{(i)}$.
- $\mathbf{H} \subseteq \mathbf{L}^{(i)}.\boldsymbol{C}$ is the set of cells constituting the *horizontal neighbourhoods* (cfr. fig. 1). These are cells that, according to the transition functions ϕ, can determine a change of the state s of the generic cell belonging to the set $\mathbf{L}^{(i)}.\boldsymbol{C}$. The cells of set $\mathbf{H}$, that potentially can change during the evolution of the system, can be defined:

– as an explicit assignment of specific cells;

– as a result of a generic *query* on the set of cells $\mathbf{L}^{(i)}.\boldsymbol{C}$;

A particular topologies of neighbourhood is comprised implicitly in the latter possibility.

- $\mathbf{P}=\{p_1, p_2,\ldots, p_m\}$ is a set of *m local cell's parameters* that, together with layer's parameters $\mathbf{L}^{(i,t)}.\mathbf{P}_L$ and global parameters $\mathbf{P}_G$, determines the transition from one state of the cell to another. The generic $p_k \in \mathbf{P}_k$, where $\mathbf{P}_k$ is the domain of variation of parameter, can be calculated:

– with respect to the time *t*, with respect to values of other parameters $p_j \in \mathbf{P}_k \subset \mathbf{P}$ (where $j \neq k$) and to global and layer's parameters, according to a function $p_k = f_k(t, \mathbf{P}_k, \mathbf{P}_L, \mathbf{P}_G)$, assigned analytically of in a tabular form;

– according to a generic calculation model evolving parallel with the CA;

– according to a *transferring function* $\sigma_k \in \Sigma$ (cfr. fig. 1), starting from states *s* of a defined subset of cells $\mathbf{V}_k = \{\mathbf{c}_1^{(j)}, \mathbf{c}_2^{(j)},\ldots, \mathbf{c}_v^{(j)}\}$ belonging to a layer $\mathbf{L}^{(j)}$ where $j \neq i$:

$$p_k = \sigma_k(\mathbf{c}_1^{(j)}.\mathrm{s}, \mathbf{c}_2^{(j)}.\mathrm{s},\ldots, \mathbf{c}_v^{(j)}.\mathrm{s}) \qquad \textbf{(4)}$$

The subset of cells $\mathbf{V}_k$ can be defined analogously as with the definition of horizontal neighbourhoods. If $q \leq m$ is the number of parameters depending on the state of groups of cells belonging to layers different than $\mathbf{L}^{(i)}$, the *vertical neighbourhoods of cells* is defined as the set:

$$\mathbf{V}=\{\mathbf{V}_1, \mathbf{V}_2,\ldots, \mathbf{V}_q\} \qquad \textbf{(5)}$$

where:

- ϕ is the *transition rule* of the cell: the state $\mathbf{c}^{(i,\,t+1)}.s$ is obtained from the current state $\mathbf{c}^{(t)}.s$ following the relation:

$$\mathbf{c}^{(i,t+1)}.s = \phi(\mathbf{c}^{(i,t)}.s,\ \mathbf{c}_1^{(i,t)}.s, \mathbf{c}_2^{(i,t)}.s,\ldots, \mathbf{c}_h^{(i,t)}.s;\ \mathrm{AC}^{(t)}.\mathbf{P}_G,\ \mathbf{L}^{(i,t)}.\mathbf{P}_L,\ \mathbf{c}^{(i,t)}.\mathbf{P}) \qquad \textbf{(6)}$$

where $\{\mathbf{c}_1^{(i,\,t)}, \mathbf{c}_2^{(i,\,t)},\ldots, \mathbf{c}_h^{(i,\,t)}\}=\mathbf{c}^{(i,\,t)}.\mathbf{H} \subseteq \mathbf{L}^{(i)}.C$ is the horizontal neighbourhood at the moment *t*. The global and layer variable parameters are updated (in this order) before the application of the rule ϕ.

- *o* is a graphical object, eventually geo-referenced, that can be chosen from a finite set *O* of objects each characterised by an adequate vectorial graphical description. Any relevant entity of the simulated phenomenon can be represented by one or, if necessary, can be "discretised" and represented by more than one graphical objects

In the exposed definition of cells, we have implicitly abandoned the classical "discretisation" of space in regular grid: in fact, potentially, different cells belonging to the same layer can be represented by different graphical objects. Furthermore, it is not necessary that the entirety of graphical objects representing cells of a layer constitutes a complete and exhaustive partitioning of the space.

In particular, a cell belonging to a layer can be seen as an *object*-instance of a *class* (the class of *that* layer's cells) characterised by common *properties*: the domain of state variation $\mathbf{S}^{(i)}$, the number *m* of parameters with respective domains of definition and variation $\mathbf{P}_1$, $\mathbf{P}_2$,...,$\mathbf{P}_m$, the form of transition rules, the form of horizontal and vertical neighbourhoods determination rules, the form of parameters evolution rules, the transferring functions Σ and connection with other simulation models.

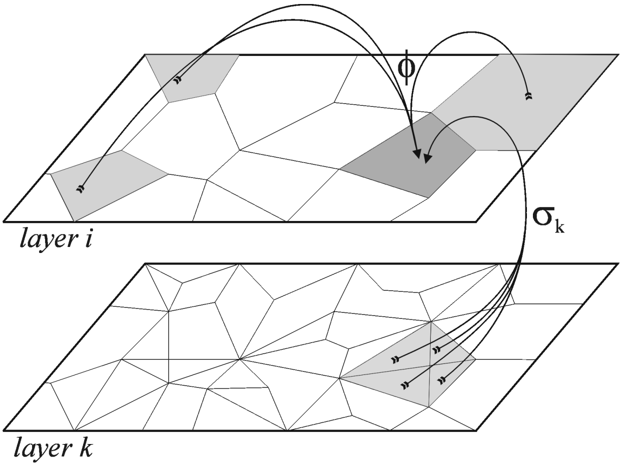


Fig. 1. Relations s between cells and layers in the proposed model

3.4. "Ingredients" of a Cellular Automaton

The generalisation proposed by our model requires, for each application to a concrete case-study, a preliminary structuring-programming and initialisation. The design of an cellular automaton requires in fact two subsequent phases:

1. **Design of the structure and rules**. This is done through the definition of the number and types of the following elements::

- the number of global parameters g: $\mathbf{P}_G=\{p_{G1}, p_{G2},\ldots, p_{Gg}\}$ and their respective domains of definition $\mathbf{P}_{G1}, \mathbf{P}_{G2},\ldots, \mathbf{P}_{Gg}$;
- the eventual functions of variation of parameters $p_{Gk} = f_k(t, \mathbf{P}_{Gk})$;
- the number of cell types n and the number of layers l;
- for each layer:
  - the number of layer's parameters c: $\mathbf{P}_L=\{p_{L1}, p_{L2},\ldots, p_{Lc}\}$ and their respective domains of definition $\mathbf{P}_{L1}, \mathbf{P}_{L2},\ldots, \mathbf{P}_{Lc}$;
  - the eventual functions of variation of parameters: $p_{Lk} = f_k(t, \mathbf{P}_{Lk}, \mathbf{P}_G)$;
  - the domain of definition of cell states $\boldsymbol{S}^{(i)}$;
  - the rules for the definition of sets $\mathbf{H} \subseteq \mathbf{L}^{(i)}.\boldsymbol{C}$ of cells constituting the *horizontal neighbourhood*;
  - the number of local parameters m: $\mathbf{P}=\{p_1, p_2,\ldots, p_m\}$ and their respective domains of definition $\mathbf{P}_1, \mathbf{P}_2,\ldots,\mathbf{P}_m$.
  - the eventual functions of variation of parameters: $p_k = f_k(t, \mathbf{P}_k, \mathbf{P}_L, \mathbf{P}_G)$;
  - the eventual rules for the definition of cell's *vertical neighbourhood* $\mathbf{V}=\{\mathbf{V}_1, \mathbf{V}_2,\ldots, \mathbf{V}_q\}$ and their respective sets Σ of *transferring functions* σ_k;
  - the eventual definition of graphical object type characterising cells.

2. **Scenario Initialisation**. This comprises the definition of initial values of global and layer parameters, the definition of initial values of cells' states and the initialisation of graphical objects associated to cells.

3.5. Definition of Neighbourhoods

The neighbourhoods $\mathbf{c}^{(i,t)}.\mathbf{H}$ and $\mathbf{c}^{(i,t)}.\mathbf{V}$ are in general the result of queries executed on sets of cells, using operators and predefined functions (logical, mathematical, geographical, etc.). Hence, formally there is no difference between sets $\mathbf{c}^{(i,t)}.\mathbf{H}$ and $\mathbf{c}^{(i,t)}.\mathbf{V}$ and their generating queries. "Cross-references" in queries are contemplated: references to the state $\mathbf{c}^{(i,t)}.s$ and to parameters $\mathbf{c}^{(i,t)}.\mathbf{P}$ of current cell, to layer's parameters $\mathbf{L}^{(i,t)}.\mathbf{P}_L$ and to global parameters $\mathbf{P}_G$, as well as to the target cells' states.

For the way of cell's neighbourhood formulation, the concept of proximity does not necessarily have a topological meaning: the criteria of Euclidean distance between cells and predefined patterns (Von Neumann, Moore, Margolus, etc.) are comprised as special cases of generic queries. Furthermore, the neighbourhood does not remain constant during the simulation but can evolve since a query can be reiterated with a defined frequency or can be activated under certain circumstances and configuration of the scenario.

3.6. Rules of Evolution

In order the cellular automata environment be truly general, a generic transferring function σ_k ,the transition function ϕ, as well as eventual functions $f(t)$ of variation of parameters regulated by dynamics esternal to the CA, should possibly be described with an adequate programming language. Such approach will potentially pose no limit to the generalisation of rules governing the CA.

The σ_k is implemented in form of a generic analytical function, as one of the classical aggregation and dis-aggregation available if GIS-based environments (sum, mean, probabilistic distribution, etc.).

It is important to mention that, in general, the ϕ is a rule with a modifiable form during the simulation: the flow-control instructions (for instance IF <*condition*> THEN <*instruction*>) permit in fact to subordinate the execution of parts of the rule to specific conditions of the state of the system. Rules can also be codified in probabilistic terms, allowing the possibility to design a non-deterministic automaton.

Furthermore, if a parameter is given the role of a counter, the system have the possibility to establish different frequency of application of transition rules for each layer and of transferring function between linked layers.

The mechanism of information transferring function between layers (aggregation – dis-aggregation), a potentially manifold spatial extension of cells belonging to different layers, as well as the possibility of inclusion of global variable parameters in transition rules, permit the modelling of "global events" having effects and influences on local level. This principle constitute a necessary derogation of the principle of "locality" embodied in classical cellular automata.

3.7. Some Remarks about the Architecture of the System

In order to satisfy all the functional requests described above, it is necessary to design the general architecture of the software system as modular as possible.

The core of the system will consist of a generalised kernel for cellular automata management ("*ACKer*"). Such a kernel will be automata execution and elaboration engine, according to the rules and following instructions given through a scripting language designed for the purpose.

Alternatively, it will be possible to access the CA database and to programme automata through a common programming language[1] using a purposely developed class library (CA-SDK – Cellular Automata Software Development Kit).

At the front-end of such an architecture, generalised or purpose-oriented user interfaces can be developed.

The following figure illustrates schematically the outlined architecture.

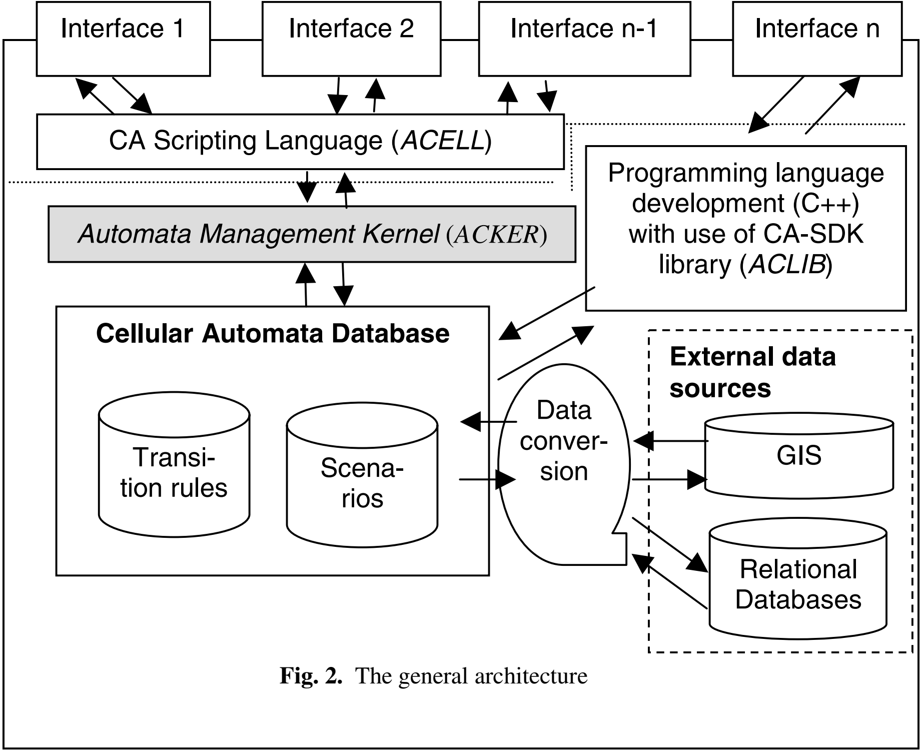

Interface 1
Interface 2
Interface n-1
Interface n
CA Scripting Language (*ACELL*)
Programming language development (C++) with use of CA-SDK library (*ACLIB*)
Automata Management Kernel (*ACKER*)
Cellular Automata Database
Transition rules
Scenarios
Data conversion
External data sources
GIS
Relational Databases

Fig. 2. The general architecture

Let us spend some words on few elements of the above scheme.

Interfaces. These are user-friendly interfaces developed *ad hoc* for the treatment of particular problems. They could be of a generalised kind – allowing the definition, management and execution of traditional "generalised" automata –, but could also be stand-alone applications that make use of particularly designed automata e.g. for the purpose of a gaming simulation. The interfaces can be developed such as to interact with the database through the kernel (ACKER), using the scripting language (ACELL), or can be directly developed using the cellular automata SDK libraries.

Scripting Language *ACELL.* Working in tight relation with the kernel, the scripting language should put at users disposal a set of commands, functions, data-

[1] Initially the library will be developed for use in C++ programming language.

types, and instructions for flow management in order to be a generalised instrument for interaction with the cellular automata database. With such a language it will thus be possible to define scenarios, set up transition rules, establish criteria of rules' application as well as fix a series of general automata execution parameters. Furthermore, the scripting language will allow complex querying and exportation, *in itinere* or *ex-post*, of results and of an automaton states.

Kernel ACKER. This is the true execution and elaboration "engine". It receives instructions in form of scripting language, access directly the CA database and provides outputs. In order to make it as portable as possible, it will be developed in C++ language and subsequently compiled for different hardware platforms.

Cellular Automata Database. As already anticipated, the aim of this research programme is to formulate an as abstract as possible general definition of cellular automata. Consequently, and in the first place, a scenario will not be treated – as normally happens – as a collection of elements (cells) each characterised by three values[2], but as a vector, or better being said, as an object, being this term borrowed from a well-known computer science paradigm. Such an object will hence be defined by a set of *properties* (variables that on their turn can be objects as well), *methods* and *events* with attributes of *inheritance* offered by the mentioned *object-oriented* paradigm[3]. Such data structuring and architecture offers greater degrees of freedom and a sufficient level of abstraction in order to be able to threat a series of problems (some mentioned above) and to render the system interoperable with external data sources or with other simulation models.

4. A Model to Play with

As widely known, gaming-simulation is a technique used in learning contexts [11] or as an instrument of participation, since it improves the understanding of different viewpoints, helps a better comprehension of a problem and permit an exploration of possible solutions [7]. A gaming-simulation can be *defined as the simulation of effects of decisions taken through assumption of roles complying to defined rules*.

The simulation model can be of any kind, and consequently also a CA-based one.

Attention, we do not want here to propose the use of CAs for playing with the behaviour of abstract systems as those based on the game of *Life* or on one of its numerous evolutions [3, 6, 15, 18].

Attention, we do not want here to propose the use of CAs for playing with the behaviour of "concrete" ecological systems (in other words, of efficacious and simplified models of real systems), a well know set of games that makes a large

[2] As known, an automaton cell is defined by a state value, but in order to operate first abstraction, it is useful to consider each cell distinguished by three values: its state and *x* and *y* co-ordinate.

[3] Terms such are *propery, method, events, inheritance* are not to be intended in their common-sense meaning, but should be referred to the specific terminology of the "object-oriented" paradigm. For the reasons of space, we cannot here even briefly sketch their specific meaning, so be it allowed to us to suggest to our reader any good book that treat that question.

portion of Eigen and Winkler's fundamental work on games [13], as we have developed ever since the far 1988 examples and applications (see e.g. [18] and [4]).

Attention, we do not want here to propose the use of CAs for playing with the behaviour of "concrete" territorial and spatial systems, as we have experimented those since the far 1989 with the simulation of "all possible cities" *FICTIES* [9, 10].

In all these cases, we were fundamentally dealing with useful, amusing and didactically efficacious competition against the "machine", trying to understand with the use of CAs how a system could work, how to exalt auto-organisation, realise equilibri, evolutions, both in general and in specific contexts.

We do not want to propose an application – still somewhat closer to our current proposal – realised using CAT for the simulation of an interactive situation of conflict [17]; however, it is a case worth to be described more in detail.

The starting point was a situation of ecological conflict of a predator/prey kind.

Initial scenario was distinguished by an environment where building elements were water, land, grass, flowers and rabbits.

The game consisted in a search for possible behaviour and reciprocal influences of each state in proximity to others, and in a definition of transition rules from one state to another, given the initial conditions. Even if initially few participants had difficulties in putting the evolution of the scenario in relation with the transition rules, gradually emerged the importance of this instrument as a pre-diction tool, from the moment the type of input given to the system could have been controlled.

The groups of players designed transition ruled imagining rabbits particularly cruel and angry if too numerous, and thus eating each other, or those reproducing dense flower area being desertified, etc. All rules were collected and publicly exposed. The scenario had been executed and results were subsequently discussed.

Let us see the mechanisms of the game:

1. the starting point is a scenario made of land, grass, rabbits and marguerites;
2. the transition rules are defined by groups of players and then implemented;
3. groups decide, from time to time, which rules to activate contemporarily;
4. discussion: the participants' problem is: why on Earth are we doing this?

At that point the participants were encouraged to modify the interpretation, taking into account the existing scenario but applying it to a plausible situation of a Lebanon city, and immediately the above mentioned states became Christians, Muslims (Lebanese or refugees), Druses and UNIFIL (*United Nations Interim Forces In Lebanon*). And the given initial scenario became a precarious equilibrium between different religious groups living in homogenous areas of the city.

The first imagined "event" was a bomb explosion, and the question was how could each group react on it.

The aim was to describe possible behaviour of "actors" under a strong pressure: Muslims claiming UNFIL's responsibility, Christians accusing Druses, UNFIL supporting Christians. The possible transition rules describing these behaviours were seek for, for example:

- protection against Christians' attacks: a Druse becomes UNFIL if surrounded by 3 Christians;
- preservation of a precarious equilibrium: an UNFIL stays UNFIL if surrounded by 3 Christians and by 3 Muslims;

- conversion or marriage: a Christian becomes Muslim if surrounded by 3 Muslims and 1 Druse.

This was only and example. However, what we have in mind here is a true gaming-simulation on territorial and spatial dynamics with situation more or less similar as follows:

- the players are given roles (in the line of principle associated with one or more states), objectives, and possibilities of action related to the definition of transition rules and neighbourhoods;
- an initial scenario is defined (e.g. a homogeneous distribution of "populations" in different areas);
- a "game-master" has the possibility to define a fixed set of transition rules and a set of possible "events";
- players can define simultaneously a set of *n* transition rules related to the behaviour of the state associated to their role;
- the model is ran for the first time for k_1 number of turns;
- one or more "events" get activated;
- players are given the possibility to simultaneously re-write their *n* transition rules;
- the model is executed for the second time for k_2 number of turns;
- the fulfilment of given objectives is verified and a results evaluation is collectively discussed.

Obviously, this is only an example of a possible game as far as "contents" are concerned (other could be related to spatial functions, land-values, demographic dynamics, etc.)

5. Conclusions

As it can easily be seen, our research sits in a zone of frontier, and that requires an explorative use of cellular automata in order to permit moments of comprehension and analysis of a plausible territorial system development and to verify the effects of decisions taken through interaction of actors as social subjects.

The use of cellular automata as reference model for a gaming simulation serves to make actors more conscious about retroactive and interactive effects normally present in territorial dynamics, and it serves this purpose better that any other model: the question here is not to interact with other players and to input results of such an interaction into a "black-box" that simulates effects; rather, here we have a direct control over the model, over its internal *rationale*, its mechanisms, rules, responses.

We insist underlining the usefulness of such an approach from conceptual and practical standpoint: users – persons involved in the game – are not only in the position to understand their respective points of view, but are also enabled to understand internal functioning of the system and thus to understand better the consequences of their actions – and that is an essential component of any process of true participation [8].

It seems to us that we are looking at a true perspective of a new and promising work.

References

1. Alexander E. R. (1992) Approaches to planning : introducing current planning theories, concepts and issues Amsterdam, Gordon and Breach 1992
2. Berlekamp E. R., Conway J. H. e Guy R. K. (1992) Winning ways : for your mathematical plays New York, Academic Press
3. Buso M. and Rizzi P. (1999) Cellular Automata Applications: from environmental education to urban and regional analysis, paper presented to CUPUM'99 6th Conference , Venice
4. Cecchini A. (1999) "Gli Automi Cellulari un utile, efficace e semplice strumento per comprendere, descrivere e prevedere le dinamiche del sistema città" in Cecchini A. (a cura di) Meglio meno, ma meglio Automi Cellulari ed Analisi Territoriale Milano, Franco Angeli 1999
5. Cecchini A. (2001) "Lucumie: a family of games for training and skills improvement. Some considerations on the astuteness of the history" in Anticipating the Unexpected – ISAGA'99 Conference proceedings of the 30th Annual Conference Sydney, International Simulations and Gaming Association (ISAGA) 2001
6. Cecchini A. (2001) "Participation to the government of cities: some considerations, some ideas, a methodology, some proposals" in INPUT 2001 Proceedings Bari, INPUT 2001 and CUPUM 2001 Proceedings Honululu, CUPUM 2001 (with P. Ardu, I. Blecic, G. Gigante, P. Rizzi, A. Vania)
7. Cecchini A. and Viola F "Ficties (Fiktive Stadte): Eine Stadtbausimulation" in Wissenschaflichte Zeitschrift der Hochschule fu Architektur und Bauwesen, Weimar Heft 2.1990
8. Cecchini A. and Viola F. (1996) "Artificial Worlds and Learning" in Besussi E. e Cecchini A. (eds) Artificial Worlds DAEST, Venezia 1996 Cecchini A. and Rizzi P. (2001) "The reasons why cellular automata are a useful tool in the working-kit for the new millennium urban planner in governing the territory" in INPUT 2001 Proceedings Bari, INPUT 2001 and CUPUM 2001 Proceedings Honululu, CUPUM 2001
9. Cecchini A. and Rizzi P. (2001) "Are Urban Gaming Simulations Useful" in Simulation and Games Special Issue, 2001
10. Eigen M. e Winckler R. (1982) Laws of the game: how the principles of nature govern chance London, Allan Lane 1982
11. Hall P. (1988) Cities of tomorrow: an intellectual history of urban planning and design in the twentieth century London, Basil Blackwell
12. W. Kriz, P. Rizzi (1997) "Planspiele für die Umwelterziehung", in *Psychologie in Österreich* n2 Juni
13. Oc T., Carmona M. e Tiesdell S. (1997) "Needs of the Profession into the Next Millennium: Views of Educators and Practitioners" in AESOP News Summer 1997
14. Rinaldi E. (1998) AUGH! Users Guide Venezia, DAEST-IUAV 1998
15. Rizzi P. (2002) "Komunikacja miedzykulturalna a gry sysimulacyjne" in Forum Europejskie, n3 zima 2002, Katedra Europeistyki Uniwersytet Jagiellonski, Krakow, pp.163-170
16. Rizzi P. (1997) "Co-evolutive Games" Proceedings of ISAGA, Tilburg NL

Urban Cellular Automata: The Inverse Problem

Giovanni A. Rabino and Alessandra Laghi

Politecnico di Milano
Dipartimento di Architettura e Pianificazione
Piazza Leonardo da Vinci 32, Milan, I-20133 Italy
giovanni.rabino@polimi.it
alessandra.laghi@libero.it

Abstract. The issues regarding complex systems and the validation of their models has recently come on the fore; as Cellular Automata do belong to this category, they are directly involved in this revision. A major issue arising from the debate regards the procedure adopted to test models of these systems: application of a priori ipotheses to one case study. This kind of procedure is seen as unreliable, and as a generator of misleading models, whose predictions do not have solid foundations. Analyzing the problem in a general perspective, it (that is, the choice of the family of models to use) could be formulated as an inverse problem, based on an inductive method, which tries to formulate rules gaining information from data and doing the least number of a priori ipotheses.

1. Introduction

The problem of validating models of complex systems (namely, Cellular Automata) has recently arisen in geography, pointed out in a paper by M. Batty and P. Torrens [1]. In this paper it was highlighted the difficulty of validating such kind of model simply running it to reproduce the dynamics of one case study. The main issue against this procedure is the complexity of the system itself, which makes very difficult test the hypotheses made a priori in structuring the model; as a consequence, simulations and predictions are not sufficiently reliable. Since the complexity of cities is a matter of fact, the problem right now is try to find a new approach to model them, in order to prevent the application of this kind of models only for “story telling” purposes.

This paper presents the problem of modeling land use dynamics using Cellular Automata (CA from this point forward), particularly focusing on the creation of transition rules. A new approach is used: the creation of rules starts from data analysis instead of a priori hypotheses, and the perspective is inductive instead of deductive.

The second section of this paper develops the general issue of inverse problem, while the third relates it to urban CA. The fourth section presents the available data, the operational problems arisen and their solutions and the first results. Finally, the fifth section is for conclusions and suggestions for further research.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 349-356, 2002.

2. The Inverse Problem in Modeling

We define *inverse* the problem faced by an observer who is dealing with a system whose dynamics is unknown; he tries to understand how it works, recording outputs without interacting with it and/or analyzing how outputs change if he perturbs the system.

The adjective “inverse” means that the order of cause and effect is reverse: the observer knows effects instead of causes and tries to deduce causes going backward; the unknown quantity of this problem is a function, that is the set of rules that determine the evolution of the system.

Solving this kind of problems is strictly dependent on the quality of the available data, because it affects the uniqueness of solutions. A good general rule is that, in order to determine a function of n variables, one should collect data that also depend on n variables.

Often inverse problems solving is difficult because of the problem itself is *ill posed*. In an ill posed problem small changes in the observations may correspond to big changes in the phenomenon being observed. Ill posed problems are difficult to deal with because the algorithms to solve them tend to be *unstable*, that means they are likely to produce very different solutions if inputs are changed just a little (Fig. 1).

When the problem is ill posed accurate observations are not enough; it is important to have prior information about the unknown system, derived from oneself experience. As a beginning, you may consider a very simple set of rules, even if it introduces strong simplifications, and then make corrections to make the model fit reality.

When you deal with well posed problems (which are characterized by existence, uniqueness and stable solutions) more data available can improve the reconstruction; on the contrary, more data in solving an unstable problem can be a double-edge sword. In fact, new data might generate a solution very different from the previous, posing the observer in front of a choice; this is the problem of *data consistency* [2].

A general method of solution for this kind of problem, adopted in our case studies as well, is called *successive approximation method*, and consists in changing progressively the starting structure of the elements of the model, testing the new model after each change through simulations [3]. This procedure addresses the full non-linearity of the problem, even if it could be computationally intensive.

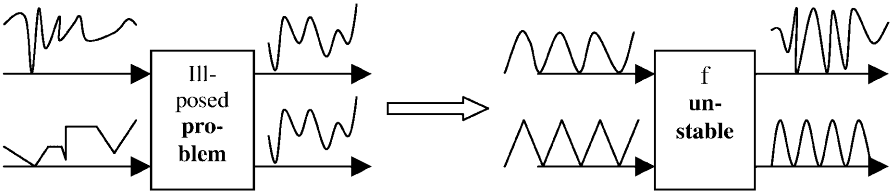


Fig. 1. Example of ill posed problem

3. Urban Cellular Automata and Inverse Problem

As far as an Urban Cellular Automaton is concerned, the challenge is trying to reproduce the behavior of the system "city"; so, the core of this research is to determine the transition rules of an Urban CA. The data we need are maps displaying cells' characteristics, i.e. land use in our case studies; maps are like pictures of the city taken in different years, and their sequence displays the changes that affected the city. What we are looking for are the rules that governed these changes, and they should be derived by the comparison of couples of successive maps, trying to extract the rules that directed the transitions happened between the maps. It is the same situation that faces a chess player who compares two configurations on a chessboard and tries to extract the moves that were made.

The most important issues are data availability and their quality. The first is relevant because we are dealing with a system that cannot be reproduced in laboratory, so we are not able to gain more data easily. Two important factors regarding data quality are the span of time covered by data and the frequency they are collected with; both these issues do affect greatly the precision we may obtain in structuring the model with a given data-set. In fact, it is worth reminding the reader that data are the starting point of the procedure and, of course, different data do lead the observer on different paths.

Even if the purpose of this approach is to extract as more as possible information from data, the observer has often to choose, because there are some critical points where data cannot help him anymore; of course every choice must be tested against the evolution of the city. The first choice that is necessary is the category of models in which looking for the best fitting for our case studies; evidently, we have chosen CA. This implies that we will use a kind of "if...and…then" rules; but rules are only one of the elements of a CA, so a choice must be made for grid, neighborhood, states and time. Because of these elements are strictly connected each other, the setting of each element influences the result we obtain in structuring the transition rules.

We are free to decide which kind of grid is the best, its dimensions, the function that gives the neighborhood, the number of land uses to consider and the span of a time step Δt. We choose to adopt a regular grid, made of 100m X 100m square cells; the neighborhood is a Moore one, considering only the first eight cells around, and its class is the prevalent among its cells; there are 12 land uses: residential continuous urban fabric, residential discontinuous urban fabric, industrial areas, commercial areas, public and private services, roads and railways, abandoned land, construction sites, green urban areas, crops, forests, canals and rivers. The calibration of time step, actually, is a different kind of freedom, because the frequency data are collected with inevitably forces the choice of the time step and, in the end, the precision we get in predictions made using the model. In fact, rules derived from the comparison of maps N years far in time can be applied to make predictions on N years time intervals; you cannot hope to get more precise results with that set of rules.

Moreover, the issue about asynchronous transition is not faced in this paper, and all transitions will be assumed as synchronous.

4. An Application: Cellular Automata of Land-Use Dynamics

4.1 Data and Steps in Inverse Problem Solving

Data available as a starting point of our study are maps of four European cities, displaying the land use according to the Corine legend. For each city, four maps are available, referring to different years and covering a period of about 40 years; in details: Milan (1955, 1965, 1980, 1997), Palermo (1955, 1963, 1989, 1997), Grenoble (1948, 1960, 1981, 1997) and Prague (1953, 1968, 1989, 1998), as shown in Fig. 2.

A number of steps are needed to reach our goal, headed on one hand to understand which changes have affected every city during the period of analysis, on the other to decompose these changes in their one-step evolution:

1. comparing successive maps of each city, in order to draw three transition matrices for each city; every matrix is square, 12X12 (as a consequence of our legend that takes into account 12 land uses), and everyone of its elements means the number of cells migrated from the row class to the column class (or the probability of migration, after dividing for the total of the corresponding row);
2. drawing transition rules that *effectively* took place;
3. conveying all the information above in operational rules for the automaton. This is a fundamental issue: information gained from data is not immediately useful for the definition of transition rules because of the different span of real (that is, coming from data) transitions; so it is necessary to elaborate a procedure that allows us to obtain one-step transitions, that we may call "fictitious", from the real ones. The composition of fictitious transitions should have the real corresponding transition as a result; this must be proved applying rules to a map and comparing the result with the real subsequently map.

The approach used to manage all the issues arisen is based on *successive approximations*: taking a simple set of rules as a starting point and then refining it on the basis of the distance between simulations and reality.

4.2 Operational Problems

The first important operational problem is about the determination of the simulation step, necessary to run the CA. As is shown in Table 1, the number of years between maps is not the same, and we should keep in mind two issues in our calculations:

1. the number of steps dividing each transition must be an integer;
2. the approximation to reduce this number to an integer must be as least as possible.

As a consequence, considering three different spans, all feasible for the life of a construction site (respectively 36, 40 and 42 months), the best choice resulting is 36 months, because every approximation is of one year, more or less, and this does not implies significant changes in the distribution of land uses.

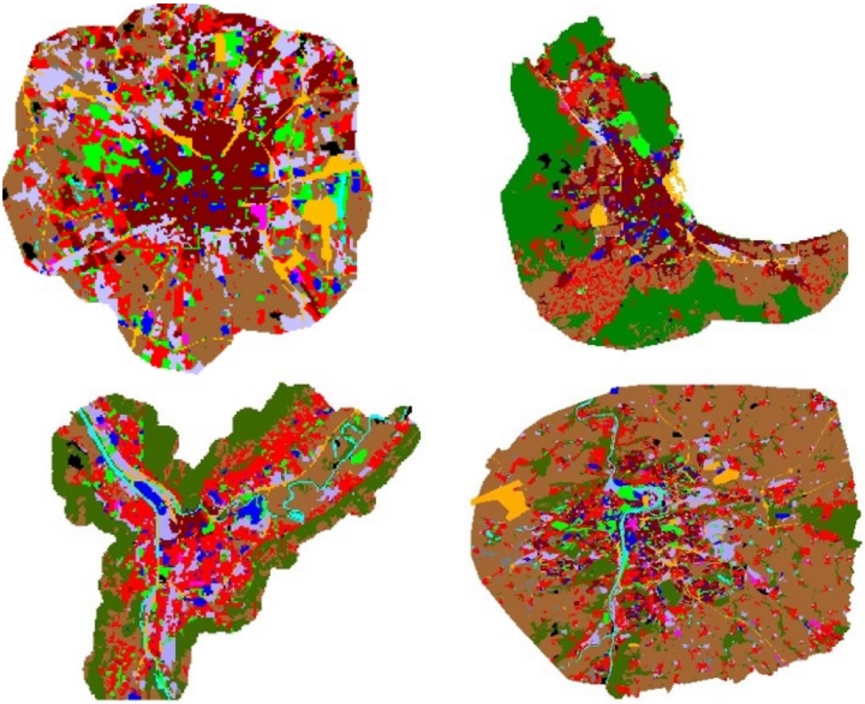


Fig. 2. Examples of maps available: starting from top right and proceeding clockwise you see Milan 1997, Palermo 1997, Prague 1998 and Grenoble 1997

Table 1. Summary of transitions' span for each city (years).

CITIES	TRANSITION 1	TRANSITION 2	TRANSITION 3
Milan	10	15	17
Palermo	8	26	8
Grenoble	12	21	16
Prague	15	21	9

Table 2. Summary of the number of steps for each transition

CITIES	TRANSITION 1	TRANSITION 2	TRANSITION 3
Milan	3	5	6
Palermo	3	9	3
Grenoble	4	7	5
Prague	5	7	3

The second operational problem that was faced regards transition probabilities [4]. In fact, the set of rules obtained from the real transitions' analysis is not deterministic; this means that the same condition in the *if...and* part of the rule brings to different final class, and this characteristic must be inherited by one-step transition rules. Unfortunately, probabilities gained from the comparison of successive maps refer to the real transition itself, and cannot be used to weight one-step transition rules. Modeling a transition as a *Markov Chain*, and assuming that transition probabilities are the same for every step belonging to the same real transition, a good estimation is

$$\hat{p}_{ii} = \sqrt[n]{p_{ii\,OSS}} \qquad \hat{p}_{ij} = \frac{p_{ij\,OSS}}{\sum_{k=0}^{n} \hat{p}_{ii}^{\,n-k}\,\hat{p}_{jj}^{\,k}} \qquad \textbf{(1)}$$

where p_{ii} is the probability to remain in the same class i, p_{ij} is the probability to move from class i to class j, and n is the number of steps in the observed transition.

The third and last issue to solve refers to the role of neighborhoods in transitions. In fact, till now we have said nothing about this role, because we have analyzed transitions in terms of *how many* cells change/do not change their class without considering *which* cells change; modeling through a CA requires the examination of the influence of the neighborhood as well. So, due to the lack of data (see above), as a first approximation we assume that the neighborhood does not change during an "observed" transition; for instance, considering the first transition of Grenoble from 1948 to 1960, the only neighborhood we are able to calculate is the one in 1948, but we assume that it keeps the same in 1951, 1954, 1957. This hypothesis involves that the number of changes of land use does not affect the prevalent class of the neighborhood itself, and allows us to use the rules obtained from real transitions' analysis, that explain *how and why* the change took place, *weighted* by the probabilities referred to one-step transition.

4.3 Results

Here the major results, that emerged from the analysis, follow:

1. transition rules are stochastic, as mentioned at the end of the previous paragraph;
2. rules can be grouped in five different types, depending on the relations among the classes involved:

- a,a,a: the cell has a neighborhood homogeneous to her class and remains in that class;
- a,b,a: the cell does not change class even if its neighborhood belongs to a different class;
- a,b,b: the cell is absorbed by its neighborhood;
- a,a,b: the cell moves to a different class, even if the neighborhood is homogeneous to its starting class;

- a,b,c: all the three elements are different, and there is probably a deeper functional relation between the class involved.

3. as far as the number of changes is concerned, all the cities have been facing a descending trend, that leads them to keep the existing distribution of land uses.
4. there are some differences among cities concerning the different growth of some kind of land uses, especially public and private services and discontinuous residential urban fabric;
5. some classes are faster than others to spread, and the accuracy to examinate transition rules must be valuated according to this issue.

The last observation is a consequence of the results obtained through the first simulations. In fact, as a starting point, we adopted a constant threshold on probability for skimming the list of rules drawn from the comparison, in order to choose only the most significant ones; the threshold was 0.1. Evidently, the word "significant" has a wide range of meanings depending on the class it refers to, because after applying these rules, results showed that this threshold was good for some classes, but too high for others, especially the first six classes, that represent human constructions. The sequence of maps generated through simulations, that should be as close as possible to the real sequence, is too conservative for Milan, and fails especially in reproducing the fast growth of the first transition; better results are available for Prague and for Palermo (except in the 26 years long transition), that are affected by a slower growth.

The error in the maps showed in Table 3, calculated as the percentage of different land uses assigned by the CA to each cell in respect to the real map, depends directly on the speed of growth of each city, because transitions lasting the same number of steps do have different errors.

Table 3. Summary of errors in each transition (%); in parenthesis, the span of the transition in number of steps

CITIES	TRANSITION 1	TRANSITION 2	TRANSITION 3
Milan	34,5 (3)	23,8 (5)	15,7 (6)
Palermo	15 (3)	33,5 (9)	2,1 (3)
Grenoble	4,8 (4)	26,6 (7)	9,1 (5)
Prague	8,4 (5)	12,4 (7)	3,2 (3)

5. Conclusions and Further Research

Fundamental aspects of this method have been showing their capability to reach the goal, but some modifications have to be made in order to take into account the problems arisen. First, it is necessary to define precisely the threshold, which must depends on the city (and, as a consequence, on available data), and on the particular class considered.

Moreover, it is necessary to investigate quantitatively the role of stochasticity in determining the difference between the real map and the simulated one. In fact, even if we have a list of all the rules with their exact probabilities, we could not obtain two identical maps after simulations, because of the different combinations of probabilities.

Further research should also expand the analysis considering wider neighborhoods, which comprise more than one frame around the cell, if necessary assigning an influence decreasing with distance.

In the end, you might consider to review the family of models considered in this analysis, changing completely or simply evolving forward a hybrid model, integrating our CA with spatial interaction models and/or multiagent systems, which are capable to reproduce such dynamics that may escape the CA.

References

1. Batty, M., Torrens, P.: Modeling complexity - The limits to prediction, paper presented at the 12th European Colloquium on Quantitative & Theoretical Geography, September 7-11, 2001, St. Valery en Caux, France
2. Cheney, M.: Inverse boundary-value problems. American Scientist vol. 85 (1997) 448-455
3. Groetsch, C.W.: Inverse problems in the mathematical sciences, Vieweg, Braunschweig, Wiesbaden (1993)
4. Najim, K., Poznyak, A.S., Gomez-Ramirez, E.: Self-learning control of finite Markov chains, Marcel Dekker Inc., New York, NY (2000)

Regional Controllability with Cellular Automata Models

Samira El Yacoubi[1], Abdelhaq El Jai[1], and Nezha Ammor[2]

[1] Laboratory of Systems Theory , University of Perpignan, 52, Avenue de Villeneuve,
66860 Perpignan, France
{yacoubi, eljai}@univ-perp.fr
http://www.univ-perp.fr/see/rch/lts/index.htm

[2] Department of mathematics, University Mohammed V, Agdal
Rabat, Morocco
nammor@yahoo.fr

Abstract. In relation with spatio-temporal systems theory some regional analysis aspects were recently developed [6, 24] and well studied in continuous systems described by partial differential equations. The purpose of this paper is to give a comparative study of the regional controllability by means of cellular automata models. We show through various examples how the main features of regional controllability may be simply described and implemented by cellular automata approach. To solve this problem, we propose one of the most efficient evolutionary techniques based on genetic algorithms.

1 Introduction

The basic concepts of systems theory including controllability, observability, identification, stability analysis, have been introduced to analyse systems behaviour taking into account input-outputs [2, 5, 13, 17]. They have been widely studied using essentially partial differential equations models. However, these equations remain very hard to implement for predictive descriptions and evolutions, particularly in the case of controlled systems. The main results have been found only for linear systems. For non-linear systems, many problems are still open. Cellular automata (CA's) models can offer in this case some additional results in so far as they can describe many nonlinear phenomena by means of simple local rules.

The concept of control is recently introduced in CA's models as a forcing function called $u(t)$ which takes values at each time t in a discrete set of all possible inputs U [7, 12]. The regional controllability is an extension of controllability concept which studies whether a given system may be steered from any initial state at time t_0 to any final state within a finite time $T - t_0$. The notion of regional controllability is motivated by various practical application in industry, environment, etc. It consists of finding an appropriate control which allow the system to achieve some objectif not on the whole domain but only on a subregion. We consider in this paper a numerical approach of the regional

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 357–367, 2002.

controllability based on genetic algorithms which have been successfully used for both uniform and non-uniform CA's [18]. We consider first the problem of finding a CA rule which performs the task consisting on steering the system at a given time T to a desired configuration on a subregion of the domain. The exhibited evolution of such CA's rules is given for both one and two dimensional examples. The control problem is considered then and formulated such an additional term which excites a given local CA dynamics. In both cases, explicite forms of rules and controls are given.

The paper is organized as follows: first, we recall some basic definitions regarding CA's and their relation to systems theory. In the third section, we give some ideas on genetic algorithms. The fourth section states the regional controllability problem by means of CA's models. In the fifth one, we give various illustrative examples in one and two dimensions. We finish with some concluding remarks and comments.

2 Cellular Automata Definitions

2.1 Introduction

CA's are simple mathematical models which provide a powerful and interesting tool for describing complex space-time phenomena. They are discrete dynamical systems which are often delineated as a counterpart to partial differential equations, as they also demonstrate the capability to describe continuous distributed dynamic systems. Since their introduction in the late of 1940s by Stanislas Ulam and the work of John von Neumann (1966), many other scientists have applied CA's approach to a wide range of problems. We can cite John Horton Conway (1970) with his famous game of life which constitutes a very good example in computer science, Stephen Wolfram (1980) who gave a classification of CA's and developed a very good study establishing that CA's evolution may reproduce behaviours of many continuous systems. In recent years, CA's have already become a very popular tool for simulating the behaviour of complex physical processes see e.g. [3, 19, 22].

2.2 Cellular Automata Architecture and Dynamics

CA's are discrete models whose behaviour is completely specified in terms of simple local relations. They are constructed as follows: time is discrete and progresses in steps. A D-dimensional infinite space is partitioned into discrete elements (the CA's lattice $\mathcal{L}$) according to a given geometry. Boundary conditions can be set to define a finite lattice. Each cell or small region of space takes a value in a discrete finite state set $\mathcal{S}$ and updates itself independently basing its new state on the states and location of a set of cells (the neighbourhood), usually formed by its immediate surrounding. The neighbourhood of size n may be defined as a mapping $N : \mathcal{L} \longrightarrow \mathcal{L}^n$ where $N(c) = \{c_1, c_2, \ldots, c_n\}$. The local dynamics can be given in several ways. It is usually expressed as a function which specifies the transition rule and defined by

$$
f : \begin{array}{ccc} \mathcal{S}^n & \longrightarrow & \mathcal{S} \\ s_t(N(c)) & \longrightarrow & s_{t+1}(c) \end{array} \tag{1}
$$

where $s_{t+1}(c)$ denotes the state of the cell c at time $t+1$. The CA evolves through a succession of global states or configurations which define its trajectory, by the iteration of its global rule

$$
F : \begin{array}{ll} \mathcal{S}^{\mathcal{L}} & \rightarrow \mathcal{S}^{\mathcal{L}} \\ s & \rightarrow F(s) \end{array} \tag{2}
$$

In this case, $F(s) = f(s(N))$ and s denotes a CA configuration defined as a mapping which associates a value in $\mathcal{S}$ to each element of $\mathcal{L}$. The global dynamics of a CA can be visualized in the so-called phase space (also state space) of F.

2.3 Additive Cellular Automata

In the deterministic case, there exist different classes of transition functions which are encountered in real problems. Nevertheless, the basic results on CA's deal with additive (linear and homogeneous) global dynamics but only in one-dimensional case. Considering addition in $\mathcal{S}$ modulo its cardinal k, addition in the configurations space $\mathcal{S}^{\mathcal{L}}$ is defined by :

$$
\forall s_1, s_2 \in \mathcal{S}^{\mathcal{L}}, \forall c \in \mathcal{L}, (s_1 + s_2)(c) = s_1(c) + s_2(c) \tag{3}
$$

Definition 1. *A global dynamics F is additive if for every pair of configurations $s_1, s_2 \in \mathcal{S}^{\mathcal{L}}$,*

$$
F(s_1 + s_2) = F(s_1) + F(s_2) \tag{4}
$$

This definition is equivalent to the local condition of additive CA's

$$
f(s_t(N(c))) = \sum_{1 \leq i \leq n} a_i s_t(c_i) \tag{5}
$$

for some scalar $a_0, a_1, \cdots, a_n$ which are called the weights or coefficients of cells in the neighbourhood N.

Transition functions which give equal weight to all the cells in the neighbourhood is a particular case of the so called **totalistic** rule which have the form

$$
f(s_t(N(c))) = \varphi\left(\sum_{c' \in N(c)} s_t(c')\right) \tag{6}
$$

It implies that the updated state value of the central cell depends only on the sum of all previous state values in the neighbourhood cells.

2.4 Cellular Automata in Systems Theory

CA's models have been extensively used as a modelling tool to approximate nonlinear discrete and continuous dynamical systems in a wide range of applications. However the inverse problem of determining the CA that satisfies some specified constraints has received a little attention. Identification is one of these inverse problem which was thoroughly studied for CA's models by Adamatzky [1]. It is understood as extracting an adequate CA model from a given set of consecutive configurations (snapshots) of a completely unknown automaton in order to produce the same observed evolution. Few works dealing with structural identification or parameter estimation problem of CA's models have been developed in [9]. Another interesting inverse problem is to find an appropriate CA rule capable of steering a given system from an initial state to a desired configuration during a time horizon T. In a previous work [8], we considered an exemple of controllability problem from a numerical point of view using genetic programming techniques in the case of additive CA's. The obtained results are quite promising and stimulate further research in this direction.

3 Genetic Algorithms

Evolutionary computation techniques have received in the last two decades increasing attention regarding their potential as optimization tools for engineering problems [10, 11, 14]. The different developed methodologies are focussed on the possibility of solving problems by evolving an initially random population of candidate solutions, through the application of operators inspired by natural selection. Genetic algorithms constitue a very important evolutionary method which has been successfully implemented for various problems including optimization, machine learning, operation research, immune systems, ecology, population genetics and so on [15]. A standard genetic algorithm proceeds as follows:

1. Generating a random initial population of individuals in a space of potential solutions called the search space. Each individual is represented by a finite string of symbols (known as the genome) to encode a possible solution in the search space.
2. At generation g (evolutionary step), individuals of population $P(g)$ are decoded and evaluated according to a given fitness function (quality criterion).
3. Individuals of the next generation $P(g+1)$ are selected according to their computed fitness values and new individuals are generated by genetically inspired operators. The most important and well known genetic operators are :
   (a) *Crossover* which operates on two selected individuals (parents) by exchanging parts of their genomes (encodings) and produces two new individuals called offspring. It is performed with probability p_{cross} which defines the crossover rate.
   (b) *Mutation* which is carried out by randomly sampling new points in the search space, with some probability p_{mut}. It is introduced to prevent premature convergence to local optima.
4. The next generation is then in turn evaluated.

Remark 1. It should be noted that convergence of genetic algorithms is not guaranteed. We have then to specify a termination condition as for instance the attainment of an appropriate fitness level.

4 Problem Statement

4.1 Preliminaries

Let us first recall some definitions related to controllability and regional controllability of distributed parameter systems. Consider a continuous system described by a partial differential equation defined on an open bounded set $\Omega \in I\!\!R^n$

$$\begin{cases} \dot{z}(t) = Az(t) + Bu(t) & t \in]0, T[\\ z(0) = z_0 \end{cases} \tag{7}$$

Under some specific assumptions satisfied by the operators A and B and given smooth boundary conditions, we can ensure the existence and uniqueness of the system solution denoted by $z(x,t)$. Let us now consider a given desired state z_d defined on Ω which is assumed to be enough smooth. The controllability problem consists in finding a control u in a given regular space which steers the system to z_d at a given time T. A large variety of works dealing with the controllability problem have been developed in a big literature see e.g [5, 13, 17].

Inspired by real applications, the controllability concept was recently relaxed to regional controllability which consist in steering a system from a given initial state to a prescribed state defined only on a subregion $\omega \subset \Omega$. This notion was firstly introduced in [6, 24] and various works related to regional analysis which concern essentially linear systems have been developed.

In order to define the above mentioned notions in terms of CA's models, we need to introduce control as an external input affecting the evolution even if CA's in the classical sense are autonomous systems. Let $\mathcal{A} = (\mathcal{L}, \mathcal{S}, N, f)$ be a CA and $\mathcal{L}_1$ be a subset of the lattice $\mathcal{L}$.

Definition 2. *A control u is any real value which is assigned to $\mathcal{L}_1$ ($\mathcal{L}_1$ may be reduced to one cell c and varying in time and space). $\mathcal{A}$ is then a non-autonomous CA denoted by $(\mathcal{L}, \mathcal{S}, N, f, u)$ with a modified form of transition function which can be denoted bu f_u and expressed as follows*

$$s_{t+1}(c) = f_u(s_t(N(c))) = f(s_t(N(c))) + \chi_{\mathcal{L}_1} u_t(c) \tag{8}$$

where $s_{t+1}(c)$ is the state of the cell c at time $t+1$ and $s_t(N(c))$ denotes the state of its neighbourhood at time t. The global controlled CA function is then denoted by F_u.

Remark 2. One of the main problems for introducing a control term is how to combine u with F in order to have a well-defined global CA function. Another problem concerns the definition of a suitable topology for these models. In this context, additive CA's defined on a finite state set with a group structure form an interesting class in which topological properties have been widely explored by many authors.
It should be finally stressed that formula (8) is not the only way to introduce control, it may be expressed by various forms of disturbance of the CA's dynamics.

4.2 Regional Controllability Problem

For an initial configuration s_0, the configuration at time T is obtained by $s_T = F_u^T(s_0)$. Let s_d be a given desired configuration defined on a subregion ω of $\mathcal{L}$. Let d be a distance defined on the configuration space $\mathcal{S}^{\mathcal{L}}$ by :

$$\forall (s_1, s_2) \in \mathcal{S}^{\mathcal{L}} \times \mathcal{S}^{\mathcal{L}} \;\; d(s_1, s_2) = \mathrm{card}\{c \in \mathcal{L} \mid s_1(c) \neq s_2(c)\} \tag{9}$$

It is easy to verify that (9) is a well defined distance on $\mathcal{S}^{\mathcal{L}}$.

Definition 3. *The CA $\mathcal{A}$ is said to be regionally controllable for $s_d \in \mathcal{S}^{\omega}$ if there exists a control u in an appropriate way ((8) for instance) such that*

$$s_T = s_d \quad on \;\; \omega \tag{10}$$

To solve this problem by a theoretical approach is still very hard owing to the fully discrete nature of the CA's models. One of their main deficiencies is the lack of structural spaces. This work is intended to be an attempt to formulate and solve numerically the regional controllability problem using genetic algorithms. The first problem concerns an unknown local CA dynamics which steers the system from an appropriate initial configuration to a desired one within a finite time T. The second problem is a properly control one in which we try to find an additional term to a given local dynamics in order to achieve the same goal. The one and two-dimensional cases are both considered and the expressions of CA's rules and controls are obtained.

5 Simulation Examples

5.1 Extracting Local Rules

Example of 1-D Cellular Automaton. Consider a one-dimensional CA which consists of a lattice of cells c_i, $i = 1, \cdots, N$ arranged in a line. Each cell takes its value in a discrete set $\mathcal{S} = \{0, 1, 2\}$. The cell states evolve synchronously in discrete time steps according to the states of their neighbours $\{c_{i-1}, c_i, c_{i+1}\}$ subject to a local transition function to be determined. With $N = 40$, we consider a subregion $\omega = \{c_{11}, \cdots, c_{20}\}$ and a prescribed state s_d defined by $s_d(c_i) =$

1, $\forall\ 11 \leq i \leq 20$. Denote by $N_{s_d}(t)$ the value of $d(s_t, s_d)$ where s_t is the obtained CA configuration at time t. The goal is to investigate the suitable transition rule capable of steering the system from a given initial state to a prescribed final configuration within some number of time steps T. This can be formulated by $N_{s_d}(T) = 0$ on ω.

Within an arbitrary initial population of individuals (rules in this case), we simulate for each rule the CA evolution and we keep the best fitness value $v_f = -N_{s_d}(t)$. The implementation of the genetic algorithm provides at each generation a rule which gives the best fitness value. The presented results are obtained with the two following parameters : 0.6 for crossover probability and which belongs to the interval $[0.6, 1]$ and 0.003 for mutation probability which must be in $[0.001, 0.005]$. The extracting rule has the form :

$$s_{t+1}(c_i) = [s_t(c_{i-1}) \oplus 2] \otimes [s_t(c_i) \oplus 2] \otimes [s_t(c_{i+1}) \oplus 2] \qquad (11)$$

where $\oplus$ and $\otimes$ indicate addition and multiplication modulo 3, respectively. Starting with an arbitrary initial configuration, the simulation results give the following evolution with rule (11)

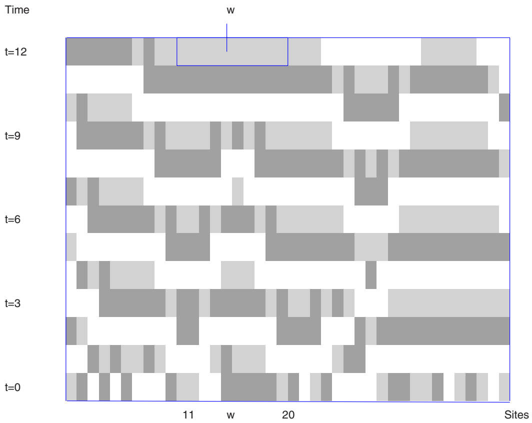


Fig. 1. Evolution of the 1-D CA rule achieving regional controllability in ω at $T = 12$. Cells with state 0, 1 and 2 are represented by the white, gray and black regions respectively. ω is represented by gray squares.

2-D Cellular Automaton Example. Consider a square lattice composed of cells $c_{i,j}$, $i, j = 1, \cdots, 10$ that evolve in a discrete state set $\mathcal{S} = \{0, 1, 2\}$. The cell state at time t, $s_t(c_{i,j})$ is updated according to the states of its surrounding von Neumann neighbourhood. The subregion is given by $\omega = \{c_{i,j} \mid 4 \leq i, j \leq 6\}$ and the problem is to find a local rule which allows the system to reach at time T the configuration defined by $s_d(c_{i,j}) = 1\ \ \forall c_{i,j} \in \omega$. A genetic algorithm as for 1-D CA's is implemented and the obtained result gives

$$s_{t+1}(c_{i,j}) = (s_t(c_{i,j-1}) \oplus s_t(c_{i,j+1}) \oplus 1) \otimes (s_t(c_{i-1,j}) \oplus 2) \otimes (s_t(c_{i+1,j}) \oplus 2) \quad (12)$$

The subregion ω where the cell states coincide with s_d appears clearly in the final CA configuration at $T = 16$ (Fig. 2).

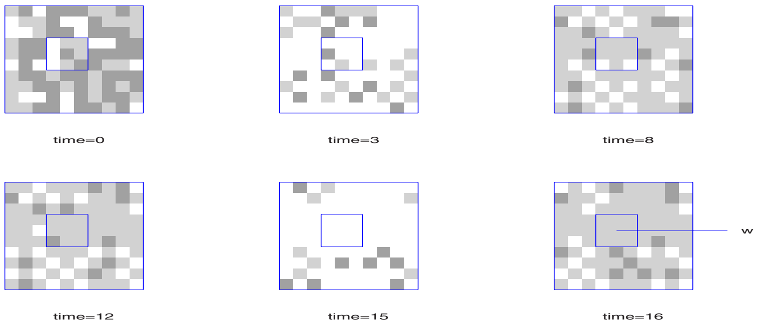


Fig. 2. Configurations of the 2-D CA governed by rule (12) at several times. Starting with a given initial configuration, regional controllability is guaranteed on ω at $T = 16$. The white, gray and black squares represent cells states 0, 1 and 2, respectively. ω is the surrounded area consisting of 3×3 cells and represented by gray squares

5.2 Control Problem

We consider in this section the problem of finding a control value u_t at each time t that disturbs a given local cellular automaton rule in order to achieve regional controllability on a subregion ω of the lattice $\mathcal{L}$. We examine also one and two dimensional cases and give explicit forms of controls.

One Dimensional Example. Let consider a lattice formed by $N = 40$ cells indexed as c_i, $i = 1, \cdots, N$. Each state cell takes its value in $\mathcal{S} = \{0, 1, 2\}$ and is updated according to the states of $N(c_i) = \{c_{i-1}, c_i, c_{i+1}\}$ subject to the following transition function

$$f(s_t(N(c_i))) = f_1(s_t(N(c_i))) + u_t(c_{i_0}) \quad (13)$$

where f_1 is a totalistic rule defined by

$$f_1(s_t(N(c_i))) = s_t(c_{i-1}) \oplus s_t(c_i) \oplus s_t(c_{i+1}) \quad (14)$$

and u_t is a control assumed to be active only on cell c_{i_0}. The regional controllability problem is considered with $\omega = \{c_{11}, \cdots, c_{20}\}$ and s_d defined by

$s_d(c_i) = 1, \;\; \forall\; 11 \leq i \leq 20$. With $i_0 = 15$ the problem is solved using classical genetic algorithms and the obtained solution has an explicit form given by :

$$u_t = a_t \otimes b_t \otimes c_t \tag{15}$$

where $a_t = s_t(c_{14})$, $b_t = s_t(c_{15}) + 2$ and $c_t = s_t(c_{16})$.

By successive applications of the control (15), we obtain in Fig. 3, the evolution from a given initial configuration to the desired configuration which is reached after $T = 23$ steps

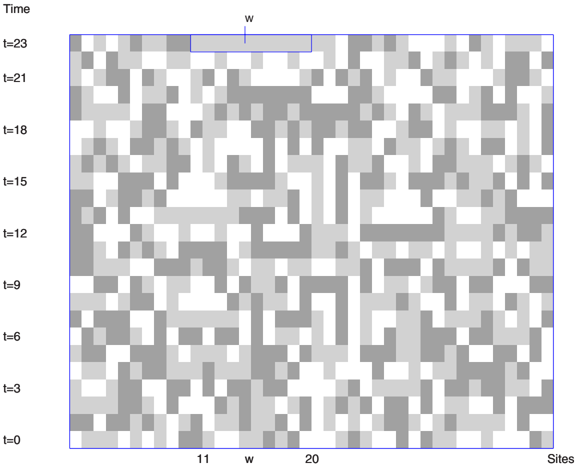


Fig. 3. The successive configurations from time 0, obtained by application of control.

Two Dimensional Case. Consider a square lattice composed with 10×10 cells indexed as $c_{i,j}$, $i,j = 1, \cdots, 10$. The state set is given by $\mathcal{S} = \{0,1,2\}$. If $s_t(c_i, j)$ denotes the state of $c_{i,j}$ at time t, the CA evolution obeys the following rule

$$f(s_t(N(c_{i,j}))) = f_1(s_t(N(c_{i,j}))) + u_t \chi_{\mathcal{L}_1} \tag{16}$$

where $\mathcal{L}_1$ is a sublattice of $\mathcal{L}$ which denotes the support of the control u_t and N designates the von Neumann neighbourhood. We consider the particular case when f_1 is of totalistic type which calculates $s_{t+1}(c_{i,j})$ as a sum modulo 3 of its neighbouring cell states $s_t(c_{i,j-1}) \oplus s_t(c_{i,j+1}) \oplus s_t(c_{i-1,j}) \oplus s_t(c_{i+1,j})$.
Given the same desired configuration $s_d(c_{i,j}) = 1 \; \forall c_{i,j} \in \omega$, the aim is to find a control u_t which is active on $\mathcal{L}_1$ and allows the system to reach the state s_d at time T on the subregion $\omega = \{c_{i,j}, \; 4 \leq i,j \leq 6\}$. For $\mathcal{L}_1 = \{c_{3,3}, c_{3,4}, c_{4,3}, c_{4,4}\}$, the used genetic algorithm gives a result in the form :

$$u_t = s_t(c_{i+1,j}) \oplus 1$$

which makes the system regionally controllable on ω (see Fig. 4).

Remark 3. For all the considered examples, periodic boundary conditions were used. It should be noted that for both one and two dimension, the necessary time T to reach the desired state for extracting CA's rules is less than for finding the control. The search space in the first case is bigger than in the second one. We suppose that time horizon T is not fixed. The constraint of fixing T a priori seems to be very restrictive and makes the problem harder.

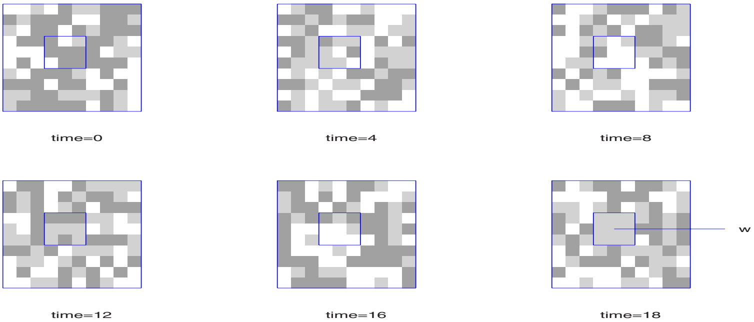


Fig. 4. Evolution of the controlled CA. The desired state is achieved on the subregion $\omega = \{c_{i,j}, \mid 4 \leq i, j \leq 6\}$ at time $T = 18$.

6 Concluding Remarks

Many analysis and control problems via CA's models are still open because of the complexity of CA's behaviour even described by simple local rules. Some numerical approaches has been successfully tested to solve simple and particular problems related to controllability, spreadability and identification of CA's models [7, 8, 12, 9]. In this paper, the problem of regional controllability of CA's has been considered and a computational approach based on genetic algorithms has been proposed and implemented for various examples. Our main goal is to illustrate the ability of CA's models to perform computational tasks that are difficult to do with numerical analysis of partial differential equation (PDE). In PDE modelling, the problem of regional controllability has been simulated only for one dimension. The 1D considered example allows to compare CA's and PDE approaches. The 2D case is very illustrative because no simulation has been done in two-dimensional systems with PDE models.

With the aim of studying all the aspects of systems theory by means of CA's models, the present paper constitues an interesting outline. It goes without saying that much can still be done in this connection.

References

1. Adamatzky, A.: Identification of Cellular Automata. Taylor & Francis Ed. (1994)

2. Butkovskii, A.G., Egorov, A.I., Luries, K.A.: Optimal Control of Distributed Systems. SIAM J. Cont., Vol. 6, N3 (1968)
3. Chopard, B., Droz, M.: Cellular Automata Modelling of Physical Systems. Collection Alea-Sacley, Cambridge University Press, Cambridge (1998)
4. Conway, J.H.: Game of Life. Academic Press (1976)
5. El Jai, A., Pritchard, A.J.: Sensors and Controls in the Analysis of Distributed Systems. J. Wiley, Texts in Applied Mathematics Series (1988)
6. El Jai, A., Simon, M.C., Zerrik, E., Pritchard, A.J.: Regional Controllability of Distributed Parameter Systems. Int. J. Control, Vol. 62, N6 (1995) 1351–1365
7. El Yacoubi, S., El Jai, A., Jacewicz, P.: Lucas: an Original Tool for Landscape Modelling. Environmental Modelling and Software (to appear, 2002)
8. El Yacoubi, S., Jacewicz, P.: Cellular Automata and Controllability Problem. CD-Rom Proceeding of the 14th Int. Symp. on Mathematical Theory of Networks and Systems, june 19-23, Perpignan, France (2000)
9. El Yacoubi, S., Ucinski, D.: Estimation de Paramètres de Modèles d'Automates Cellulaires. IEEE Int. Conf. on Automatics CIFA-2002, Nantes, 8-10 july (2002)
10. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway, NJ (1995)
11. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press (1975)
12. Jacewicz, P., El Yacoubi, S.: A Genetic Programming Approach to Structural Identification of Cellular Automata. In: Wyrzykowski, R., Mochnacki, B., Piech, H., Szopa, J. (eds): 3rd Int. Conf. on Parallel Processing and Applied Mathematics, Kazimierz Dolny, Poland, 14-17 Septembre (1999) 148–157
13. Lions, J.L.: Contrôle Optimal des Systèmes Gouvernés par des Equations aux Dérivées Partielles. Dunod, Paris (1968)
14. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
15. Mitchell, M.: An Introduction to Genetic Algotithms. MIT Press, Cambridge, MA (1996)
16. von Neumann, J.: Theory of Self-Reproducing Automata. (edited and completed by Arthur Burks), University of Illinois Press (1966)
17. Russell, D.L.: Controllability and Stabilizability Theory for Linear Partial Differential Equations. Recent Progress and Open Questions. SIAM Rev. 20 (1978) 639–739
18. Sipper, M.: Evolution of Parallel Cellular Machines. The Cellular Programming Approach. Lecture Notes in Computer Science, Vol. 1194, Springer-Verlag, Berlin (1997)
19. Toffoli, T.: Cellular Automata as an Alternative to (rather than approximation of) Differential Equations in Modeling Physics. Physica D, vol. 10 (1984) 117–127
20. Ucinski, D., El Yacoubi, S.: Simulating Cellular Automata with Maple. MapleTech, Vol. 1 (1998) 1–7
21. Ulam, S.: Random Processes and Transformations. Proceedings of the Int. Congress on Mathematics, Vol. 2 (1952) 264–275
22. Weimar, J.: Simulation with Cellular Automata. Logos-Verlag, Berlin (1998)
23. Wolfram, S.: Cellular Automata and Complexity : Collected Papers. Addison-Wesley Publishing Company (1994)
24. Zerrik, E.H.: Analyse Régionale des Systèmes à Paramètres Distribués. Thèse de Doctorat d'Etat, Université de Rabat, Maroc (1993)

Author Index

Albuquerque, Paul 220
Ammor, Nezha 357
Azpeitia, Iker 132

Bagnoli, Franco 249
Bandini, Stefania 156, 202
Bandman, Olga 10
Blecic, Ivan 337
Boucher, Charles 282

Cattaneo, Gianpiero 82
Cecchini, Arnaldo 337
Chaudhuri, P. Pal 56, 259, 270
Chen, Fan 282
Cheng, Jianquan 325
Chopard, Bastien 231
Chrobok, Roland 315

D'Ambrosio, Donato 304
Dennunzio, Alberto 82
Dhar, Sandip 56
Droz, Michel 190
Dupuis, Alexandre 220

Franci, Fabio 249
Fujiwara, Norio 69

Galam, Serge 1
Ganguly, Niloy 56, 259, 270
Gnaciński, Piotr 104
Gregorio, Salvatore Di 304

Höns, Robin 92

Ibáñez, Jesús 132
Iovine, Giulio 304

Jai, Abdelhaq El 357
Jiménez-Morales, Francisco 32

Kinder, Michael 210
Kirchner, Ansgar 239

Laghi, Alessandra 349
Lupiano, Valeria 304

Maeda, Masashi 69
Maji, Pradipta 56, 270
Makowiec, Danuta 104
Manzoni, Sara 156
Marchese, Fabio M. 178
Marconi, Stefan 231
Marinósson, Sigurður F. 315
Masser, Ian 325
Mühlenbein, Heinz 92
Mukherjee, Monalisa 259

Nagel, Kai 20
Nishinari, Katsuhiro 239

Pavesi, Giulio 202
Pękalski, Andrzej 190
Pottmeier, Andreas 315

Rabino, Giovanni A. 349
Rechtman, Raúl 249
Rizzi, Paola 337
Roli, Andrea 144
Rongo, Rocco 304
Roy, Anup K. 270

Saha, Sourav 270
Schadschneider, Andreas 239
Schönfisch, Birgitt 210
Schreckenberg, Michael 315
Serra, Roberto 168
Sikdar, Biplab K. 56
Simone, Carla 156
Sipper, Moshe 123
Sloot, Peter 282
Spataro, William 304
Stauffer, André 123
Succi, Sauro 114

Tomassini, Marco 44
Trunfio, Giuseppe A. 337

Umeo, Hiroshi 69

Venzi, Mattias 44
Villani, Marco 168

Wahle, Joachim 315
Weimar, Jörg R. 294

Yacoubi, Samira El 357

Zambonelli, Franco 144