Faculty of Bioscience Engineering

Academic year 2013-2014

Stability of stochastic cellular automata

Wouter Van der Meeren
Promotors: Prof. dr. Bernard De Baets
Dr. ir. Jan Baetens

Master thesis submitted in fulfilment of the requirements for the degree of
Master of Science in Bioscience Engineering: Food Science and Nutrition

Ghent, June 2014

Bernard De Baets:

Jan Baetens:

Wouter Van der Meeren

Preface

Allereerst wil ik mijn promotoren, dr. ir. Jan M. Baetens en prof. dr. Bernard De Baets, bedanken om mij de kans te geven op dit boeiende onderwerp te werken. Ook wil ik hen bedanken voor de begeleiding en nuttige input gedurende de uitwerking en bij het uitschrijven van deze thesis. Ook bedank ik de universiteit voor het ter beschikking stellen van de HPC infrastructuur. Voorts wil ik mijn collega-thesisstudenten en iedereen in lokaal A1.079 bedanken voor de toffe sfeer, de steun en een komische noot op momenten dat het nodig was, alsook voor hulp bij moeilijkheden met Mathematica, de HPC, inkscape, tikz, en andere LaTeX-gerelateerde zaken, en nog zoveel meer. Ten slotte wil ik ook alle andere leden van de vakgroep Wiskundige Modellering, Statistiek en Bio-informatica bedanken voor de feestjes, toffe pauze en lekkere zoetigheden. I would also like to thank Witold Bołt for the input and remarks regarding my first results.

Wouter Van der Meeren
Gent, 2 juni 2014

Table of Contents

List of Symbols

b	Birth rate
d	Death rate
$\lvert\mathcal{E}\rvert$	Ensemble size
ϵ_t	Number of defects at time step t
H	Hamming distance
i	Index to refer to a specific cell
J	Jacobian matrix
λ	Lyapunov exponent
λ_m	Mean-field estimate of the upper bound of the maximal Lyapunov exponent
λ_n	Normalised Lyapunov exponent
ℓ	Lempel-Ziv complexity
ℓ_n	Normalised Lempel-Ziv complexity
$\bar{\mu}$	Geometric mean of the average proportion of cells in a cell's neighbourhood that affect that cell
$\bar{\mu}_d$	$\bar{\mu}$ as determined in the case of a deterministic cellular automaton
$\bar{\mu}_s$	$\bar{\mu}$ as determined in the case of a stochastic cellular automaton
$\lvert N\rvert$	Size of the neighbourhood
N	Neighbourhood
$\mathcal{N}^{\mathcal{M}}$	Moore neighbourhood
ν	Minimal required proportion of the circumference of a cell that has to be occupied by another cell for that cell to be considered as the first cell's neighbour

$\mathcal{N}^{\mathcal{V}}$	Von Neumann neighbourhood
$\mathcal{N}^{\mathcal{V},\nu}$	Restricted von Neumann neighbourhood
p	Probability of applying a rule to a certain cell
Φ, Ψ	Families of transition functions of a (stochastic) cellular automaton
$\mathcal{P}^n$	Set of nth order polytopes
ρ	Density of a cellular automaton
S	Set of all states
$s(c_i, t)$	State of a cell c_i at time step t
σ_i	Sum of the states of the cells in the neighbourhood of $c_i : \sum_{j=1}^{\lvert N(c_i)\rvert} s(c_{i_j}, t)$
T	Number of time steps
t	Time step
$\mathcal{T}$	Tessellation
$\lvert\mathcal{T}\rvert$	Tessellation size
$\overline{V}$	Mean connectivity
z	Number of cells with state 1 in a cell's neighbourhood, without the centre cell

List of Acronyms

CA	Cellular automaton
ECA	Elementary cellular automaton
HPC	High performance computing
IC	Initial condition
IP	Initial perturbation, initial defect
LZ	Lempel-Ziv complexity
MLE	Maximal Lyapunov exponent
nLZ	Normalised Lempel-Ziv complexity
nMLE	normalised maximal Lyapunov exponent
PDE	Partial differential equation
SCA	Stochastic cellular automaton
SECA	Stochastic elementary cellular automaton
std	Standard deviation

Summary

The aim of this thesis is to assess the influence of stochasticity on the stability of cellular automata, the discrete counterparts of partial differential equations. The stability is assessed following a Lyapunovian approach, which measures the rate of divergence or convergence of infinitely close phase space trajectories. The stability assessment was performed on both theoretical and practical models. The results for the former indicate that distinct behavioural classes can be found. In both the one-dimensional CAs and those involving two spatial dimensions a class is found that does not arise in the other. In the last chapter, the stability of a practical model is tested for two kinds of neighbourhoods. From this stability analysis it was concluded that the model could be classified in one of the classes established based on theoretical models and that stochasticity is not the only source of changes in stability behaviour. It is also likely that there is more often a large effect of stochasticity in theoretical models than in practical models.

Dutch summary

Het doel van deze thesis is het onderzoeken van de invloed die stochasticiteit heeft op cellulaire automaten, de discrete alter egos van partiële differentiaalvergelijkingen. De stabiliteit wordt bepaald aan de hand van een Liapunoviaanse aanpak die de snelheid van divergentie van twee oneindig dichte fasebanen meet. De stabiliteitsbepaling werd uitgevoerd op zowel theoretische als praktische modellen. De resultaten van de theoretische modellen geven aan dat verschillende gedragsklassen gedefinieerd kunnen worden, maar dat zowel in modellen met één ruimtelijke dimensie als deze met twee ruimtelijke dimensies een klasse gevonden werd die niet in de andere terug te vinden was. In het laatste hoofdstuk wordt de stabiliteit van een praktisch model nagegaan voor twee soorten buuromgevingen. Uit deze stabiliteitsanalyse werd besloten dat het model in kwestie kan ingedeeld worden in een van de klassen die opgesteld werden op basis van de theoretische modellen en dat stochasticiteit niet de enige bron van stabiliteitswijzigingen is. Het is ook mogelijk dat het bijzondere gedrag dat in sommige theoretische cellulaire automaten naar voor komt, veel minder zal teruggevonden worden in praktische modellen.

1. Introduction

Stochastic cellular automata (SCAs) are often used for modelling natural and physical processes, such as the dynamics of forests (Kubo, 1996) and the interaction between chemical substances (Van der Weeën et al., 2011, 2013). Besides, Reichenbach et al. (2007) have shown that a SCA can be used to simulate the interaction between microorganisms in such a way that qualitatively similar spatial aggregates of microorganisms emerge in silico as they do in vitro. Although Lyapunov exponents as a means to determine the stability of cellular automata (CAs) were already established two decades ago to assess the stability of deterministic CAs, in the works of Wolfram (1984), Shereshevsky (1992) and Bagnoli et al. (1992), the effect of stochasticity on the stability of elementary CAs (ECAs) has not yet been examined. Since SCAs are so often used in practical models, it might be important to assess the influence of stochasticity on the stability of SCAs. The goal of this thesis is to examine the stability of certain theoretical SCAs and to classify them based on their behaviour. The insights obtained by studying these theoretical models will then be used to arrive at a better understanding of a SCA mimicking real-life processes.

2. Literature study

2.1 Introduction

Spatio-temporal models always involve three domains, namely the space, time and state domain. Usually, the state domain is the dependent domain. A simple example is the one dimensional wave equation for a perfectly flexible string with fixed ends at the same height (Armstead and Karls, 2006):

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad 0 \leq x \leq a, \quad t \geq 0,$$

where c is a constant, x represents the space variable, which is the position along the string, a is the maximum value of x, t is the time and u is the state, which is the height of the string at time t and position x. This model is based on a partial differential equation (PDE), in which space, time and state are continuous. In general, each of these three domains can be either discrete or continuous. The commonly used names of the eight possible combinations of discrete and continuous domains are given in Table 2.1. There are, however, also hybrid models that combine two or more paradigms of those listed in Table 2.1.

Table 2.1: Classification of continuous (C) and/or discrete (D) models and their most common name (adapted by Baetens (2012) from Berec (2002)).

Space	Time	State	common name
C	C	C	PDE-based models
C	C	D	Spatial point models
C	D	C	Reaction-diffusion models
C	D	D	Agent-based models
D	C	C	Spatially implicit models
D	C	D	Interacting particle system
D	D	C	Coupled-map lattice
D	D	D	Cellular automata

When building models based on (P)DEs, continuity of the universe is assumed. At macro scale this seems to be appropriate. Yet, in real life many things are discrete, such as the smallest measurable distance, the Planck distance, which is 1.6162×10^{-35} m, the smallest electric charge, i.e. the charge of an electron, which is -1.602×10^{-19} C, and so on. A more important problem of continuous models is that they are often based upon mathematical equations for which there

exist no closed-form solution, and thus have to be solved numerically. Such numerical methods (e.g. finite-difference, finite-elements and finite-volume methods) involve the discretisation of the continuous equations, which leads to truncation and approximation errors (Katz, 2009). To avoid these disadvantages discrete models can be used. These are intrinsically discrete and the numerical integration is an exact process. However, discrete models also have disadvantages. For instance, in order to know the state of the system after a number of time steps, the system has to be evaluated at every intermediate time step (Toffoli, 1984). Furthermore, the scale of the discrete units of nature is too small to use as the dimension of a spatial entity in discrete models. Therefore, an error is introduced by transferring reality to the model.

2.2 Cellular automata

2.2.1 Formalism

The discrete systems discussed in this thesis are cellular automata (CAs). A cellular automaton (CA) $\mathcal{C}$ can be defined as follows (Baetens and De Baets, 2010).

Definition 2.1 *A cellular automaton $\mathcal{C}$ is a quintuple*

$$\mathcal{C} = \langle \mathcal{T}, S, s, N, \Phi \rangle,$$

where

(i) $\mathcal{T}$ is a countably infinite tessellation of an n-dimensional Euclidean space $\mathbb{R}^n$, consisting of cells c_i, $i \in \mathbb{N}$, which are identified with n-polytopes p_i^n.

(ii) S is a finite set of k states, often $S \subset \mathbb{N}$.

(iii) The output function $s : \mathcal{T} \times \mathbb{N} \to S$ yields the state value of cell c_i at the t-th discrete time step, i.e. $s(c_i, t)$.

(iv) The neighbourhood function $N : \mathcal{T} \to \cup_{p=1}^{\infty} \mathcal{T}^p$ maps every cell c_i to a finite sequence $N(c_i) = (c_i)_{j=1}^{|N(c_i)|}$, consisting of $|N(c_i)|$ distinct cells c_{i_j}.

(v) $\Phi = (\phi_i)_{i \in \mathbb{N}}$ is a family of functions,

$$\phi_i : S^{|N(c_i)|} \to S,$$

each ϕ_i governing the dynamics of cell c_i, i.e.

$$s(c_i, t+1) = \phi_i(\tilde{s}(N(c_i), t)),$$

where $\tilde{s}(N(c_i), t) = (s(c_{i_j}, t))_{j=1}^{|N(c_i)|}$.

Aside from the CAs introduced in Definition 2.1 another family of CAs can be defined, being the stochastic CAs (SCAs).

Definition 2.2 *A stochastic CA $\mathcal{S}$ is a sextuple*

$$\mathcal{S} = \langle \mathcal{T}, S, s, N, \Phi, \Psi \rangle,$$

where premises (i)-(iv) of Definition 2.1 still hold, and for which $\Phi = (\phi_i)_{i\in\mathbb{N}}$ *and* $\Psi = (\psi_i)_{i\in\mathbb{N}}$ *are families of functions,*

$$\phi_i : S^{|N(c_i)|} \to S,$$

$$\psi_i : S^{|N(c_i)|} \to S,$$

each ϕ_i *and* ψ_i *governing the dynamics of cell* c_i*, i.e.,*

$$s(c_i, t+1) = \begin{cases} \phi_i(\tilde{s}(N(c_i), t)), & \textit{with probability p,} \\ \psi_i(\tilde{s}(N(c_i), t)), & \textit{with probability 1-p,} \end{cases}$$

where $p \in [0, 1]$ *represents the probability that the transition of a cell* c_i *occurs according to* ϕ_i*.*

Definitions 2.1 and 2.2 are the most general definitions of (S)CAs. In practice, there are some simplifications. The first one relates to the presumed infinite tessellation. This is impracticable, but can be approximated by a finite tessellation with periodic boundary conditions. A second simplification is possible by assuming that the transition function is spatially homogeneous. In the remainder of this thesis, CAs with a homogeneous transition function will be used, referred to as homogeneous CAs.

Definition 2.3 *A homogeneous CA is a CA fulfilling premises (i)-(iv) of Definition 2.1, and for which* $\Phi = (\phi_i)_{i\in\mathbb{N}}$ *is a family of functions,*

$$\phi_i = \phi_j = \phi : S^{|N(c_i)|} \to S,$$

ϕ *governing the dynamics of every cell* c_i*, i.e.*

$$s(c_i, t+1) = \phi(\tilde{s}(N(c_i), t)).$$

Similarly, a homogeneous SCA can be defined. The most studied CAs are those with only two states, usually denoted 0 and 1 (i.e. $S = \{0, 1\}$).

2.2.2 One-dimensional cellular automata

The simplest CAs are those with only one spatial dimension. The neighbourhood of a cell c_i in these CAs is given by $\{c_{i-r}, c_{i-(r-1)}, \ldots, c_{i-1}, c_i, c_{i+1}, \ldots, c_{i+(r-1)}, c_{i+r}\}$, where r is the range of the neighbourhood (see Figure 2.1). Within the family of one dimensional CAs the simplest ones are the members of the family of elementary CAs (ECAs). These are CAs with range $r = 1$, i.e., every cell has three neighbours (the cell to the right, the cell itself and the cell to the left), such that $s(c_i, t+1) = \phi(s(c_{i-1}, t), s(c_i, t), s(c_{i+1}, t))$. The remainder of this section will only consider ECAs. Because the neighbourhood of every cell consists of three cells and every cell has two possible states, there are eight possible neighbourhood configurations. Each configuration leads to an evolution of the middle cell to 0 or to 1. Hence, there are $2^8(256)$ possible rules within the family of ECAs. The rule numbers are based on their binary notation. For instance, Table 2.2 illustrates that the sum of the products of the values of $s(c_i, t+1)$ and the decimal numbers gives rise to rule number 150, or, mathematically:

...	c_{i-r}	$c_{i-(r-1)}$	...	c_{i-1}	c_i	c_{i+1}	...	$c_{i+(r-1)}$	c_{i+r}	...

Figure 2.1: Numbering convention of cells for a 1D CA.

Table 2.2: Rule table for the one-dimensional ECA 150.

$N(c_i, t)$	$s(c_i, t+1)$	binary to decimal
111	1	$2^7(128)$
110	0	$2^6(64)$
101	0	$2^5(32)$
100	1	$2^4(16)$
011	0	$2^3(8)$
010	1	$2^2(4)$
001	1	$2^1(2)$
000	0	$2^0(1)$

$$\begin{aligned}
&\phi(1,1,1)*2^7+\phi(1,1,0)*2^6+\phi(1,0,1)*2^5+\phi(1,0,0)*2^4 \\
&\qquad\qquad +\phi(0,1,1)*2^3+\phi(0,1,0)*2^2+\phi(0,0,1)*2^1+\phi(0,0,0)*2^0 \\
&= 1*128+0*64+0*32+1*16+0*8+1*4+1*2+0*1 \\
&= 150
\end{aligned}$$

Although there exist 256 ECAs, their analysis can be restricted to the 88 minimal representative ECAs, because of symmetries (Vichniac, 1990). This set encompasses the behaviour of all 256 ECAs, thus the behaviour of all 256 ECAs can be described with less computational resources. The minimal rules are: 0-15, 18, 19, 22-30, 32-38, 40-46, 50, 51, 54, 56-58, 60, 62, 72-74, 76-78, 90, 94, 104-106, 108, 110, 122, 126, 128, 130, 132, 134, 136, 138, 140, 142, 146, 150, 152, 154, 156, 160, 162, 164, 168, 170, 172, 178, 184, 200, 204, 232.

2.2.3 Two-dimensional cellular automata

Because the cells of 2D CAs are 2nd order polytopes (i.e. polygons), the number of neighbours depends on the neighbourhood definition. Typically, two neighbourhood types are considered being the Moore neighbourhood and the von Neumann neighbourhood (Baetens and De Baets, 2010).

Definition 2.4 *The Moore neighbourhood $\mathcal{N}_i^{\mathcal{M}}$ of a polygon $p_i^2 \in \mathcal{T}$ contains those polygons $p_j^2 \in \mathcal{T}$ that share a vertex with p_i^2, i.e., $\mathcal{N}_i^{\mathcal{M}} = \{p_j^2 \in \mathcal{T} \mid \mathcal{P}_i^0 \cap \mathcal{P}_j^0 \neq \emptyset\}$*

Definition 2.5 *The von Neumann neighbourhood $\mathcal{N}_i^{\mathcal{V}}$ of a polygon $p_i^2 \in \mathcal{T}$ contains those polygons $p_j^2 \in \mathcal{T}$ that share a line segment with p_i^2, i.e., $\mathcal{N}_i^{\mathcal{V}} = \{p_j^2 \in \mathcal{T} \mid \mathcal{P}_i^1 \cap \mathcal{P}_j^1 \neq \emptyset\}$*

Two-dimensional CAs can be based upon a regular or irregular tessellation, called regular CAs and irregular CAs, respectively. In the latter case, the length of the line segments that a neighbour shares with a cell varies between neighbours. Therefore, a restricted von Neumann neighbourhood can be defined, containing those neighbours whose shared line segment makes up a minimal proportion of the cell's circumference (Baetens and De Baets, 2010).

Definition 2.6 *The restricted von Neumann neighbourhood* $\mathcal{N}_i^{\mathcal{V},\nu}$ *of a polygon* $p_i^2 \in \mathcal{T}$ *contains those polygons* $p_j^2 \in \mathcal{T}$ *that share a line segment* p_r^1 *with* p_i^2 *and, make up at least a prescribed proportion* $\nu \in [0,1]$ *of* c_i*'s circumference, i.e.,* $\mathcal{N}_i^{\mathcal{V},\nu} = \{p_j^2 \in \mathcal{T} \mid \mathcal{P}_i^1 \cap \mathcal{P}_j^1 \neq \emptyset \wedge O_{ij} \geq \nu\}$*, where* O_{ij} *is the proportion represented by* p_r^1 *in* p_i^2*'s circumference.*

Baetens and De Baets (2013a) revealed that, when using a restricted neighbourhood, the behaviour of a CA is influenced by the choice of the prescribed proportion of the circumference of a cell that should be occupied by the shared line segment of a neighbour (ν) (see Section 2.3.3).

For a regular, square tessellation with a von Neumann neighbourhood there are $2^{2^5} = 2^{32} \approx 10^9$ possible rules and for a Moore neighbourhood there are $2^{2^9} = 2^{512} \approx 10^{154}$. However, in the framework of theoretical investigations, the evolution of 2D CAs is mostly based on the sum of the states within the neighbourhood. Essentially, the two most important types of 2D CAs are totalistic and outer-totalistic CAs. Both are order-invariant, meaning that their transition function does not depend on the ordering imposed on $\tilde{s}(N(c_i), t)$ (as opposed to Definition 2.1) (Baetens and De Baets, 2010).

Definition 2.7 *A totalistic CA is an order-invariant CA for which* $S \subset \mathbb{N}$*, and for which there exists a* $\Omega : \mathbb{N} \to S$ *such that* $s(c_i, t+1) = \Omega(\sigma_i)$*, where* $\sigma_i = \sum_{j=1}^{|N(c_i)|} s(c_{i_j}, t)$*.*

Definition 2.8 *An outer-totalistic CA is an order-invariant CA for which* $S \subset \mathbb{N}$*, and for which there exists a* $\Omega^* : \mathbb{N} \times S \to S$ *such that* $s(c_i, t+1) = \Omega^*(\sigma_i^*)$*, where* $\sigma_i^* = \sum_{j=1}^{|N(c_i)|} s(c_{i_j}, t) - s(c_i, t)$*.*

In the remainder of this thesis, the rule numbering convention introduced by Baetens and De Baets (2010) for totalistic and outer-totalistic CAs will be used, since it is applicable to both regular and irregular CAs. For that purpose, a new parameter θ is introduced to overcome the possible unboundedness of σ_i in irregular tessellations. This parameter is an upper bound on the sum of neighbouring states, σ_i,s such that $\Omega(\sigma_i) = \Omega(\theta)$, whenever $\sigma_i \geq \theta$. The rule number of a k-state, θ-sum totalistic CA, further referred to as a (k, θ) totalistic CA, is:

$$\begin{aligned} R_\theta^T &= z_\theta k^\theta + z_{\theta-1} k^{\theta-1} + \ldots + z_2 k^2 + z_1 k + z_0 \\ &= \sum_{f=0}^{\theta} z_f k^f \,, \end{aligned}$$

where z_f is the state value of c_i at time step $t+1$ if σ_i at time step t is f. The rule number for a k-state, θ-sum outer-totalistic CA, further referred to as a (k, θ) outer-totalistic CA, is:

$$\begin{aligned} R_\theta^{OT} = & z_{\theta,k-1}k^{k(\theta+1)-1} + z_{\theta,k-2}k^{k(\theta+1)-2} + \ldots + z_{\theta,0}k^{k\theta} \\ & + z_{\theta-1,k-1}k^{k\theta-1} + z_{\theta-1,k-2}k^{k\theta-2} + \ldots + z_{\theta-1,0}k^{k\theta-k} \\ & + \ldots \\ & + z_{1,k-1}k^{2k-1} + z_{1,k-2}k^{2k-2} + \ldots + z_{1,0}k^{k} \\ & + z_{0,k-1}k^{k-1} + z_{0,k-2}k^{k-2} + \ldots + z_{0,0} \\ = & \sum_{f=0}^{\theta}\sum_{g=0}^{k-1} z_{f,g}k^{kf+g}\,, \end{aligned}$$

where $z_{f,g}$ is the state value of c_i at time step $t+1$ if σ_i and $s(c_i, t)$ at time step t are equal to f and g, respectively.

2.2.4 Asynchronous cellular automata

Most CAs are synchronous, meaning that each cell is updated at every time step. On the other hand, when the cells are updated asynchronously, the CA is referred to as an asynchronous CA. There are various asynchronous update methods. Bandini et al. (2010) describe various update methods, for example the random independent, the cyclic order, the random order and the exponentially clocked method. The random independent method updates one cell in every time step, which is picked at random. In the cyclic order method, every cell is given a random number and one cell is updated per time step in the order of the assigned number. The random order method is similar to the cyclic order method, but the order assigned to the cells is changed every time all cells have been updated. The exponentially clocked method assigns to every cell a waiting time according to an exponential distribution with mean one, after which the cell with the shortest waiting time is updated and subsequently is assigned a new waiting time.

2.2.5 Representation and visualisation of cellular automata

There are two possible ways of representing a CA. In case of regular CAs, the most straightforward way is by identifying every cell with an element of a matrix (for 2D CAs) or a list (for 1D CAs). Another representation uses a graph, where the nodes of the graph represent the cells of the CA and the edges represent the neighbourhood relations. This representation is mainly used for irregular CAs, but can be used for all CAs.
Visualisation can also be done in two ways, similar to the representation. Either the graph or a tessellation of all cells can be drawn. For 1D CAs, the cells are not drawn as 1-polytopes (line segments), as would be expected, but they are drawn as squares. In the case of irregular CAs it is not always possible to draw a tessellation, since neighbourhood relations can be too complex to map the underlying graph to a tessellation. When a CA is visualised as a tessellation, the state of the cells is mostly indicated by assigning a colour to every state and colouring the cells in the tessellation accordingly. For 2-state CAs, black (for 1) and white (for 0) are typically used. Since 1D CAs have only one spatial dimension, it is possible to show the evolution throughout the different time steps by means of a space-time diagram. In this diagram the time is on the

vertical axis with the configuration at $t = 0$ at the top of the diagram (see Figure 2.2, 2.3, 2.4 and 2.5).

2.2.6 Applications of cellular automata in food and related sciences

Various applications of (S)CA-based models have been described in food and related sciences. Van der Weeën et al. (2013) describe a model for oil migration in chocolates. This model is built upon a SCA with 50×260 square cells, seven states (cavity, solid cocoa butter, liquid cocoa butter, sugar, cocoa particle, nut oil, hazelnut particle) and a neighbourhood of range two. Periodic boundary conditions are used to mimic the continuous space along the horizontal axis and fixed boundary conditions are used for the boundaries with air and filling (Van der Weeën et al., 2013).

Van der Weeën et al. (2011) compare a SCA- and a PDE-based model for the formation of CO_2 in the reaction: $CaCO_3 + 2HCl \rightarrow CaCl_2 + H_2O + CO_2$. Both models were fitted to data and their accuracy was compared. The difference in accuracy between both models is small, but the CA-based model has a better fit to the data, indicating that it would be the better choice.

A model for competition between three species is developed by Reichenbach et al. (2008). This model has four states, being the three species (A, B and C) and an empty state ($\emptyset$). Between the three species, cyclic domination occurs, meaning that species A outperforms species B, species B outperforms species C and species C outperforms species A. For every cell, one cell in its neighbourhood is selected at random to compete with. This model leads to complex patterns where the density of each of the species fluctuates.

Another SCA-based model is set up by Wimpenny and Colasanti (1997). It simulates the growth of a biofilm for a given concentration of substrate and a given number of attached microorganisms. The SCA is implemented in an object-oriented language and two objects were defined, a microorganism and a compartment. Both objects have their own functionalities. A microorganism is defined by a location in a tessellation, a nutritional state and a function for growth. A compartment is defined by its position, its number of resources and a function for the distribution of resources.

All of the above mentioned models are based on 2D SCAs with more than two states. Despite the simplicity of ECAs, they have some applications in food and related sciences. Diao et al. (2008) use ECAs to predict transmembrane regions of membrane proteins. Since transmembrane regions consist of approximately 20 amino acids, the considered amino acid sequences have this length. This means that for a protein with n amino acids, $n - 20 + 1$ segments are considered as possible transmembrane regions. For these segments, the amino acids are replaced by a series of five binary numbers, leading to ECAs with 100 cells. To check interactions between the different amino acids in these regions, the ECAs are evolved for a given number of time steps using rule 84 with reflecting boundary conditions. For this ECA the Lempel-Ziv complexity is calculated. This measure partially reflects sequence effects. Based on this measure and the abundance of the 20 existing regular amino acids, each protein part is expressed as a vector whose position gives an indication whether that part is a transmembrane region or not.

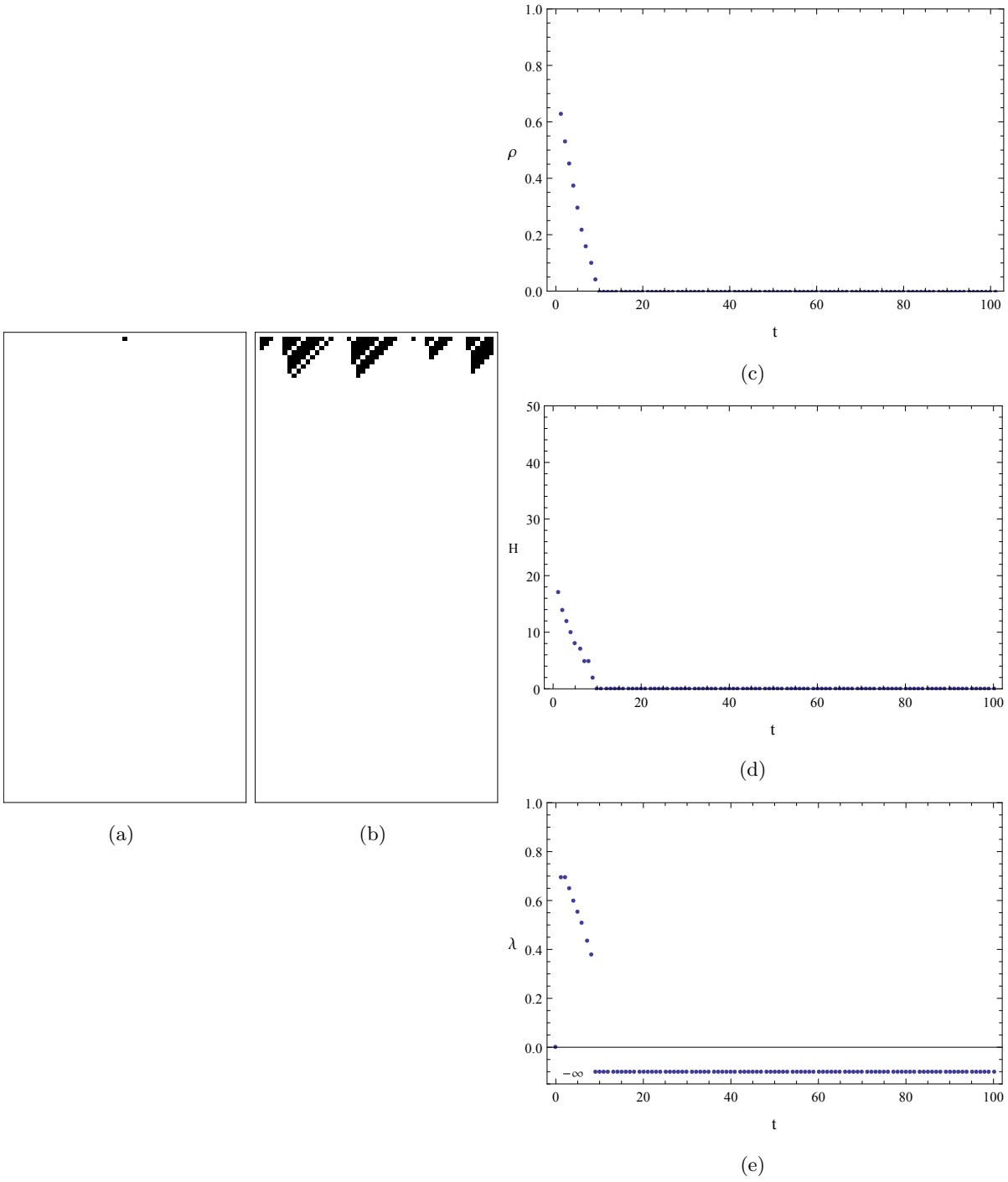


Figure 2.2: A space-time diagram of a one-dimensional CA (rule 168) starting from 2 initial conditions (a and b), its density (c), Hamming distance (d) and Lyapunov exponent (c) as function of the time step, starting from a random configuration.

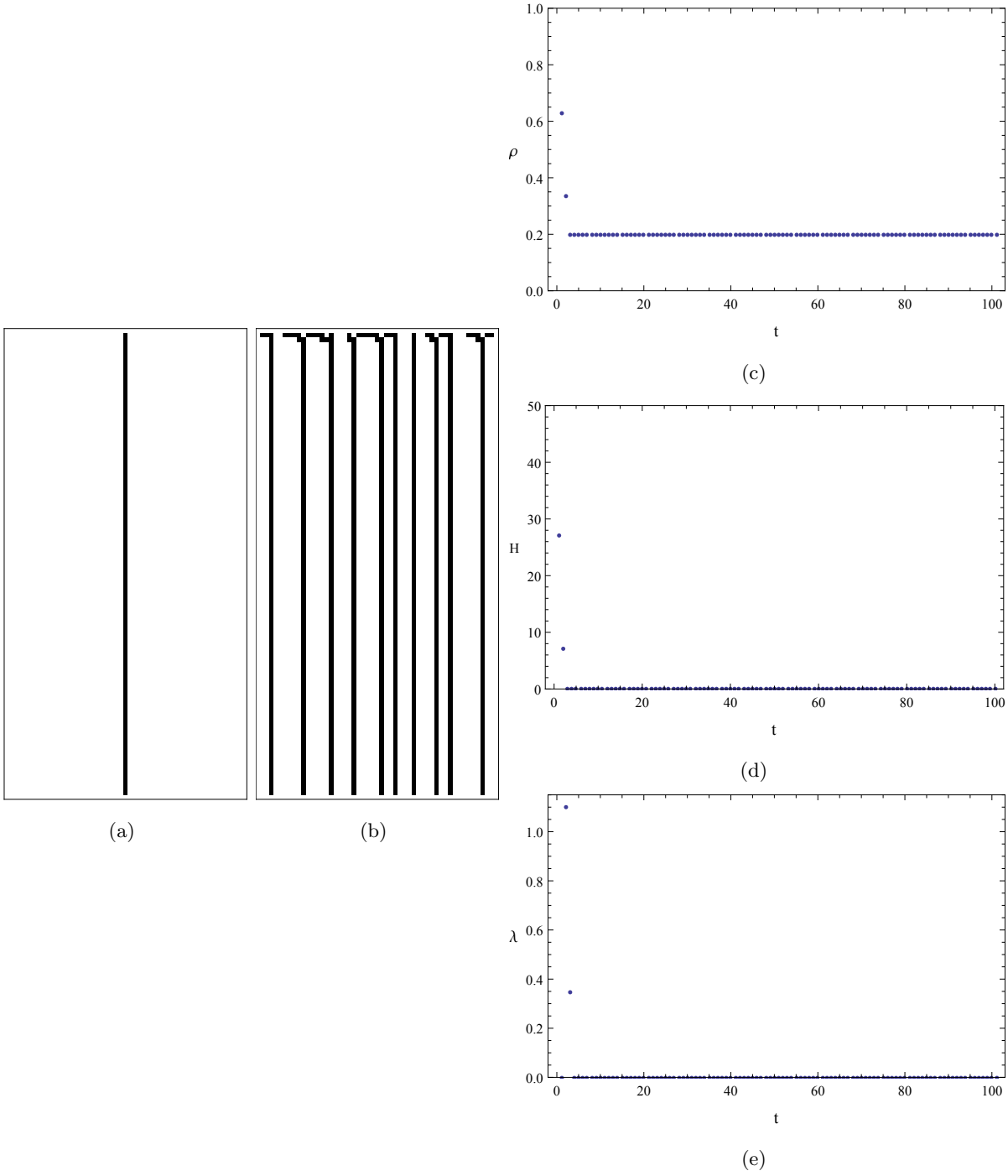


Figure 2.3: A space-time diagram of a one-dimensional CA (rule 100) starting from 2 initial conditions (a and b), its density (c), Hamming distance (d) and Lyapunov exponent (e) as function of the time step, starting from a random configuration.

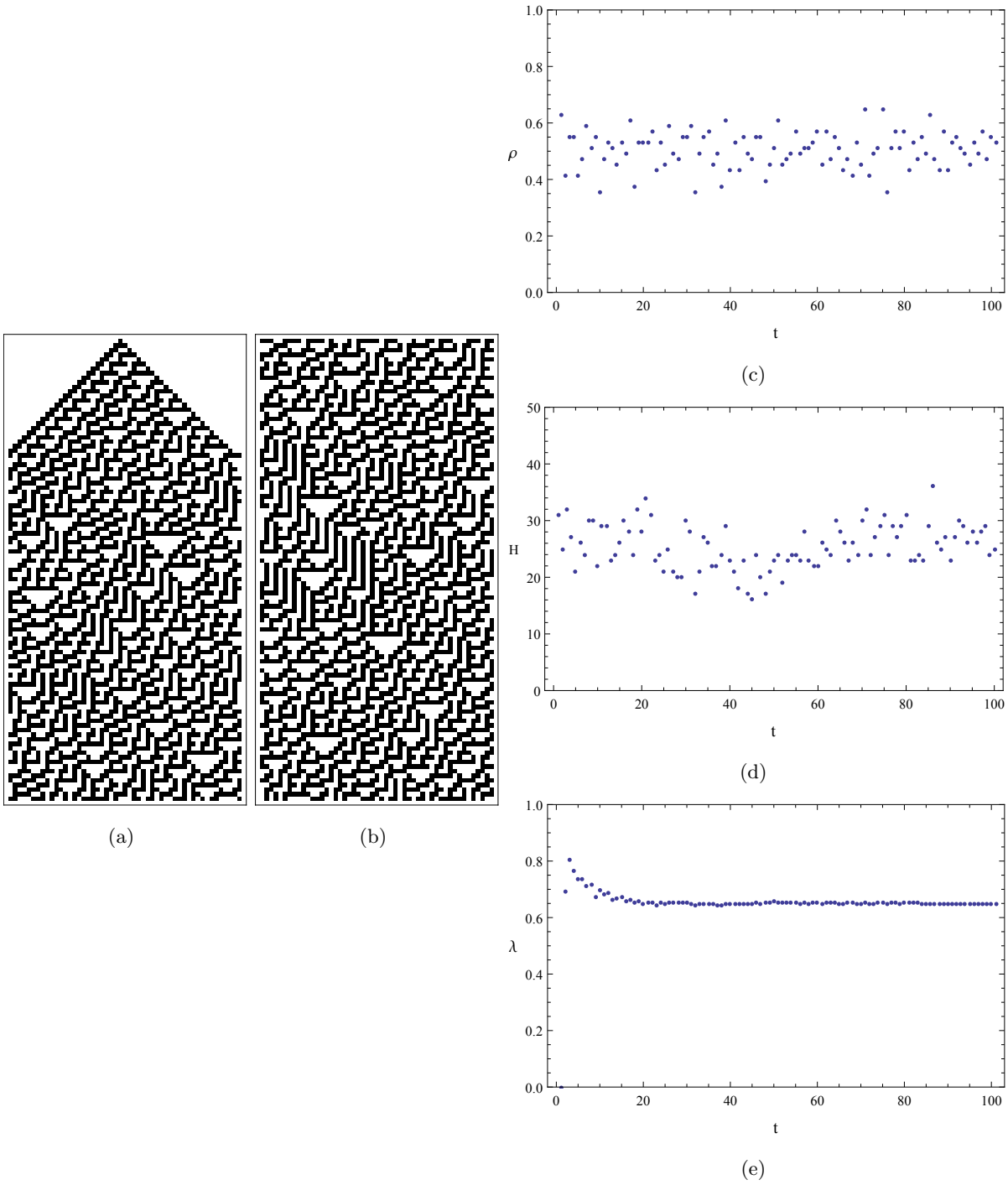


Figure 2.4: A space-time diagram of a one-dimensional CA (rule 30) starting from 2 initial conditions (a and b), its density (c), Hamming distance (d) and Lyapunov exponent (e) as function of the time step, starting from a random configuration.

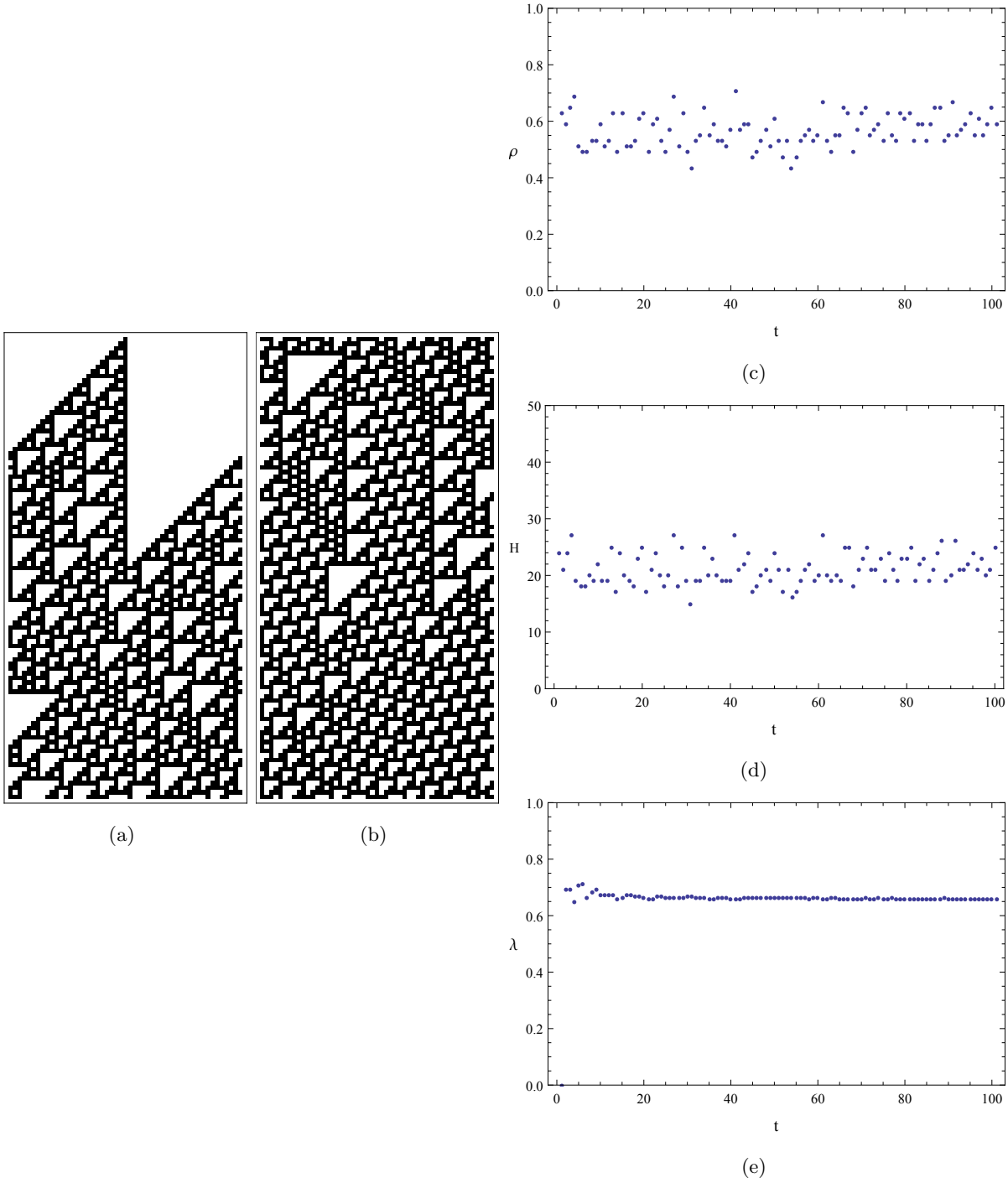


Figure 2.5: A space-time diagram of a one-dimensional CA (rule 110) starting from 2 initial conditions (a and b), its density (c), Hamming distance (d) and Lyapunov exponent (e) as function of the time step, starting from a random configuration.

Another two-state model is the one by Kubo (1996) about gap dynamics in forests. The two states in this model are gap and non-gap, represented as 0 and +, respectively. A gap is considered as an area in which the height of the canopy is lower than a certain threshold. Many studies showed that gap formation is more likely to happen at the edges of already existing gaps (Kubo, 1996). Therefore, the speed of gap formation is dependent on the number of gap sites in a cell's neighbourhood, denoted by $\delta n(0)$. The speed of gap closing can either be constant (See Eq. (2.1)), or it can depend on either the total number of non-gaps or the number of non-gaps in a cell's neighbourhood.

$$(0) \underset{d+\delta n(0)}{\overset{b}{\rightleftarrows}} (+) \tag{2.1}$$

More CA approaches for biological modelling can be found in the paper of Ermentrout and Edelstein-Keshet (1993).

2.3 Stability of cellular automata

Based on their stability, CAs can be classified according to the classification scheme proposed by Wolfram (1984). It discriminates between four classes, of which the first three exhibit behaviour that is similar to attractors found in continuous dynamical systems (Wolfram, 1984):

Class 1: Evolution leads to a homogeneous state

Class 2: Evolution leads to a set of stable or periodic structures

Class 3: Evolution leads to a chaotic pattern

Class 4: Evolution leads to complex localised structures, sometimes long-lived

This classification has so far been proven appropriate for every kind of CA, but it is hard to use, because it is subjective (since it is based on visual inspection) (Baetens, 2012). Table 2.3 shows the classification of the 256 ECAs according to Wolfram (1984). A typical evolution of a rule belonging to Class 1, 2, 3 and 4 can be found in Figures 2.2, 2.3, 2.4 and 2.5, respectively.

Table 2.3: Wolfram's classification of elementary CAs (Wolfram, 1984).

Class	Rule number
1	0, 8, 32, 40, 64, 96, 128, 136, 160, 168, 192, 234, 235, 238, 239, 248-255
2	1-7, 9-17, 19-21, 23-29, 31, 33-39, 42-44, 46-53, 55-59, 61-63, 65-74, 76-85, 87, 88, 91-95, 97-99, 100, 103, 104, 107-109, 111-119, 123, 125, 127, 130-134, 138-145, 148, 152, 154-159, 162-164, 166-167, 170-181, 184-191, 194, 196-224, 226-233, 236, 237, 240-247
3	18, 22, 30, 45, 60, 75, 86, 89, 90, 101, 102, 105, 122, 126, 129, 135, 146, 149-151, 153, 161, 165, 182, 183, 195
4	41, 54, 106, 110, 120, 121, 124, 137, 147, 169, 193, 225

There are various other, quantitative, ways of assessing the stability of a CA. The majority of such measures rely upon Lyapunov exponents (Bagnoli et al., 1992) and Boolean derivatives

(Vichniac, 1990). In the remainder, both measures will be introduced for two state CAs. On the other hand, measures like the density (ρ) and Hamming distance (H) have also been suggested, but these have proven not to be convergent for many rules, which limits their discriminative power (see Figures 2.2, 2.3, 2.4, 2.5). The density of a CA at a given time step is the proportion of cells that have a state 1, i.e.

$$\rho(t) = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} s(c_i, t)\,.$$

The Hamming distance in this context is the distance between the CA's configuration at two subsequent time steps, i.e.

$$H(t) = \sum_{i=1}^{|\mathcal{T}|} |s(c_i, t) - s(c_i, t+1)|\,.$$

2.3.1 Boolean derivatives

Boolean derivatives are the discrete counterparts of derivatives for real functions. A Boolean derivative determines whether or not changing the value of a cell c_j at the $t-$th time step influences the value of a cell c_i at time step $t+1$ (Bagnoli et al., 1992). Mathematically, this Boolean derivative can be expressed as:

$$\Phi'_{ij} = \frac{\partial s(c_i, t+1)}{\partial s(c_j, t)} = \phi(s(c_1,t), s(c_2,t), \ldots, s(c_i,t), \ldots, s(c_j,t), \ldots) \\ \oplus\, \phi(s(c_1,t), s(c_2,t), \ldots, s(c_i,t), \ldots, \overline{s}(c_j,t), \ldots),$$

where $\overline{s}(c_j, t)$ is the Boolean complement of $s(c_j, t)$ and $\oplus$ is the sum mod 2 operator. The Boolean derivative is always false (i.e. 0) if $c_j \notin N(c_i, t)$ because ϕ only takes the cells in $N(c_i, t)$ as its arguments. The Boolean Jacobian matrix J is composed of the Boolean derivatives of every cell with respect to every cell, i.e. its elements are given by:

$$J_{ij} = \begin{cases} \dfrac{\partial s(c_i, t+1)}{\partial s(c_j, t)} & , \text{ if } c_j \in N(c_i), \\ 0 & , \text{ else.} \end{cases}$$

Higher-order Boolean derivatives can also be calculated (Vichniac, 1990). Vichniac (1990) proved that higher-order derivatives can, when starting from a single one on a background of zeroes, be used to predict the value of a cell after T time steps without having to evolve the system for $T-1$ intermediate time steps. The difference in computational cost between using higher-order derivatives and evolving the CA for intermediate time steps is an indication for the complexity of a rule, with the difference decreasing as complexity increases (Vichniac, 1990).

Based on the Jacobian, another measure of complexity can be calculated (Baetens and De Baets, 2010). This measure, denoted $\bar{\mu}$ and referred to as the input sensitivity, represents the geometric mean over T time steps of the average proportion of cells c_j in $N(c_i)$ that affect $s(c_i, t+1)$:

$$\bar{\mu} = \left(\prod_{t=1}^{T} \mu(t) \right)^{1/T}, \tag{2.2}$$

where

$$\mu(t) = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \frac{1}{|N(c_i)|} \sum_{j:c_j \in N(c_i)} J_{ij} \,. \tag{2.3}$$

2.3.2 Lyapunov exponents

Lyapunov exponents were originally conceived for continuous dynamical systems. They are a measure for the average rate of exponential divergence of two nearby trajectories in phase space (Eckmann and Ruelle, 1985). Mathematically, this rate can be expressed as:

$$\lambda = \lim_{t\to\infty} \lim_{\boldsymbol{\delta}\mathbf{x}(0)\to\mathbf{0}} \frac{1}{t} \log\left(\frac{|\boldsymbol{\delta}\mathbf{x}(t)|}{|\boldsymbol{\delta}\mathbf{x}(0)|}\right),$$

where log is the natural logarithm and $\boldsymbol{\delta}\mathbf{x}(0)$ represents the initial separation between two nearby trajectories. The limit $\lim_{\boldsymbol{\delta}\mathbf{x}(0)\to 0}$ is necessary, because otherwise the trajectories would diverge exponentially and would no longer be nearby. The number of Lyapunov exponents of a continuous dynamical system equals the dimensionality of the phase space. The ensemble of all Lyapunov exponents is called the Lyapunov spectrum Λ. These exponents are the natural logarithms of the eigenvalues of:

$$\lim_{t\to\infty} \left(\left(\prod_{l=0}^{t-1} J^l \right)' \left(\prod_{l=0}^{t-1} J^l \right) \right)^{\frac{1}{2t}},$$

where $'$ represents the transpose and J^l is the Jacobian at time l. The theorem of Oseledec (1968) guarantees that this limit exists. If one of the Lyapunov exponents is larger than zero, the system is unstable.

Lyapunov exponents for discrete dynamical systems were first introduced by Wolfram (1984). Wolfram (1984) conceived left and right Lyapunov exponents which represent the speed at which a defect propagates to the left or to the right in a 1D CA, respectively. Yet, the rate of propagation of defects to the left or right can only grow linearly, such that exponential divergence cannot be obtained. Besides, they are only of use for 1D CAs. Therefore, such directional Lyapunov exponents are not frequently used as a measure of chaotic behaviour. Nowadays, the maximal Lyapunov exponent (MLE), denoted as λ, is used far more often. The MLE can be assessed through a damage spreading analysis. By means of such an analysis the propagation of defects throughout the tessellation is studied. A defect (defective cell) is a cell whose state is different in two configurations (Baetens and De Baets, 2010). Usually, a damage spreading analysis starts from two initial configurations that differ in only one cell, i.e. there is one defect. This is the smallest possible separation between two nearby trajectories in discrete systems such as CAs. These two configurations are then evolved and the number of defects is tracked as the system evolves. If we denote the number of defects on time step t as ϵ_t, the finite-time MLE can be defined as (Bagnoli et al., 1992):

$$\lambda(t) = \frac{1}{t} \log\left(\frac{\epsilon_t}{\epsilon_0}\right), \tag{2.4}$$

where ϵ_0 is the initial number of defects. The MLE is then defined as $\lambda = \lim_{t\to\infty} \lambda(t)$. Counting the number of defects, however, is not as straightforward as would be expected, since defects

can cancel each other in configuration space due to the discrete nature of the system. Also, as with directional Lyapunov exponents, the propagation rate would never become exponential. Thirdly, since we work with a finite tessellation, the number of defects is limited whereas time is infinite. These three reasons would lead to $\lambda = -\infty$ for every rule, if we would simply count the number of defects in configuration space. These problems can be overcome by keeping track of the multiplicity of defects in every cell (Baetens and De Baets, 2013b). Bagnoli et al. (1992) used the Boolean Jacobian matrix (see Section 2.3.1) to calculate the propagation of defects. For a given MLE, the number of defects at a given time step can be calculated, given the initial number of defects:

$$\epsilon_t = \epsilon_0 \, e^{\lambda t}.$$

Since both the proportion of cells in a cell's neighbourhood that affect a cell ($\bar{\mu}$) and the Lyapunov exponent (λ) are good measures of stability, they are often used together. This is shown in Figure 2.6 for the 88 minimal ECA rules, or at least those that do not have $\lambda = -\infty$ for all members of an ensemble of different initial configurations (IC) and different initial defects (initial perturbations (IP)). The reason for using this ensemble will be explained in Section 3.2. A mean-field estimate of the upper bound on the Lyapunov exponent can be found by considering the time- and space-averaged proportion of cells in a cell's neighbourhood that affect this cell ($\bar{\mu}$) and the space-averaged number of cells in a cell's neighbourhood, referred to as the mean connectivity of a tessellation, and denoted in the remainder as $\overline{V}$. The mean-field estimate of the number of defects at a given time step is:

$$\epsilon_{t,m} = \epsilon_0 \, (\overline{V}\bar{\mu})^t,$$

which leads to the mean-field estimate of the upper bound on the MLE (λ_m):

$$\lambda_m(\bar{\mu}) = \log(\overline{V}\bar{\mu}). \tag{2.5}$$

Contrary to the directional Lyapunov exponent, the MLE is applicable to all tessellations.

Based on the MLE, Baetens and De Baets (2010) proposed a new classification:

Class 1: $\lambda = -\infty$ for all members of the ensemble

Class 2: $\lambda = -\infty$ for some, but not all members of the ensemble

Class 3: $\lambda \geq 0$ for all members of the ensemble

CAs belonging to Class 1 are referred to as unconditionally superstable. For CAs belonging to Class 2, a distinction is made between those that give rise to a positive MLE for most members of the ensemble and those that give rise to a homogeneous state for most members of the ensemble, referred to as conditionally unstable and conditionally stable, respectively. The CAs belonging to Class 3 are referred to as unconditionally unstable (Baetens and De Baets, 2010). Class 2 exists because the most stable state can, for some CAs, not be reached from every configuration due to their discrete nature. This will be explained in more detail in Section 3.2.

Baetens et al. (2012) assessed the stability of CAs that are updated according to the four asynchronous methods discussed in Section 2.2.4 and compared the Lyapunov exponents with the

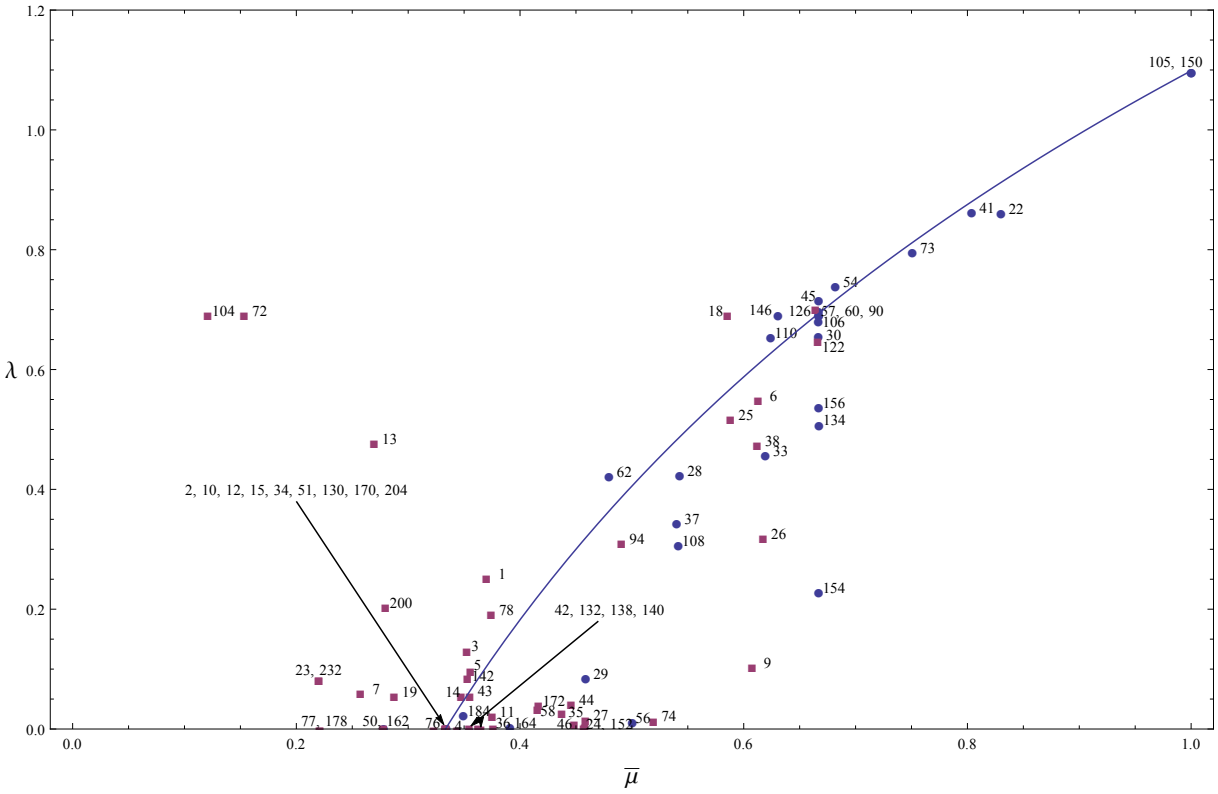


Figure 2.6: Lyapunov exponents as a function of the input sensitivity $\bar{\mu}$ for the 88 minimal ECAs (dots) and the mean-field estimate of the MLE (line). Rules for which some of the Lyapunov exponents in the ensemble are $-\infty$ are represented by squares. Rules that have a positive Lyapunov exponent for all members of the ensemble are represented by circles.

ones obtained for their synchronously updated counterparts. Yet, since in these four asynchronous methods only one cell is updated every time step, defects cannot spread at the same speed as is the case for synchronous update methods. Therefore, the Lyapunov exponents of synchronous and asynchronous methods have to be normalised by dividing them by their respective upper bounds. The upper bound on the MLE of asynchronously updated CAs where one cell is updated at every time step is given by (Baetens et al., 2012):

$$\lambda_m(\bar{\mu}) = \log\left(\frac{|\mathcal{T}| - 1 + \overline{V}\bar{\mu}}{|\mathcal{T}|}\right), \tag{2.6}$$

where $\bar{\mu}$ is the one obtained in the synchronous case.

At this point, it should be noted that chaos apparently present in configuration space is not always linked to unstability in phase space, and vice versa. And, last but not least, the MLE is a measure for the effect of single defects, whereas in configuration space two or three adjacent cells can cancel each other.

2.3.3 Effect of topology

Topology refers to the whole of size and neighbourhood relations of a CA. Baetens et al. (2013) revealed that the tessellation size has an influence on the stability of a measure, leading to the concept of a representative tessellation size beyond which the value of a measure is no longer

influenced by the tessellation size. Beside variability of a measure, the actual value itself can change depending on the topology (Baetens and De Baets, 2013a). Baetens and De Baets (2013a) used the value of the proportion of the circumference of a cell that is occupied by the shared line segment of a neighbour (ν) as a parameter to assess the influence of topology on the behaviour of a CA. They revealed topological bifurcation points, where the Lyapunov exponent jumps from a positive value to 0 or $-\infty$ or vice versa.

3. Materials and methods

3.1 Materials

The algorithms were developed on a desktop computer provided by Ghent University, Department of Mathematical Modelling, Statistics and Bioinformatics. The full-scale in silico experiments were performed on the high performance computing (HPC) infrastructure, and more specifically on the Gengar and Raichu clusters. The specifications of the computing infrastructure can be found in Table 3.1.

Table 3.1: Properties of the computational infrastructure.

Desktop	
	Intel® Core™ 2 CPU 6300 @ 1.86GHz
	1.99 GB of Ram
Gengar	
	156 computing nodes
	dual-socket quad-core Intel Xeon Harpertown (L5420) @ 2.5 GHz
	16 GB RAM/node
Raichu	
	64 computing nodes
	dual-socket octa-core Intel Xeon Sandy Bridge (E5-2670) @ 2.6 GHz
	32 GB RAM/node

Mathematica versions 9.0.0 and 9.0.1 (Wolfram Research Inc.) were used for the development of algorithms and conducting the in silico experiments.

3.2 Methods

3.2.1 Initial condition and initial perturbation

For the simulations of 1D CAs, a tessellation of 500 cells was used. It is assumed that this tessellation size is representative, since it is almost as big as the 512 used by Bagnoli et al. (1992). Because not every configuration of a CA can be reached from a given initial configuration (IC) (Bagnoli et al., 1992), it is possible that the stability of a rule depends on the initial configuration and/or the initial perturbation (IP), i.e. the initial defect. However, it is impossible to calculate the MLE for all possible initial perturbations of all possible initial configurations. Bagnoli

et al. (1992) avoided this problem by swapping positions of two random cells at every time step, i.e. giving the CA a push. Hence, when a CA gets to a configuration from which it cannot evolve to a more stable state, this extra push can force the CA towards the most stable configurations. Another way of handling this problem is by assessing the MLE for an ensemble of perturbations of a random initial configuration (Baetens and De Baets, 2010). Baetens and De Baets (2010) then made a distinction between rules based on the number of perturbations that lead to negative Lyapunov exponent (see Section 2.3.2). In this dissertation, the effect of different initial conditions and perturbations on the numerically obtained value of the MLE was tested, prior to running full-scale in silico experiments. The results can be found in Figure 3.1. From this figure it is clear that no distinction, i.e. no consistent difference, can be observed between the outcome of the approach involving different initial conditions of which the same cell is perturbed, and the one involving one IC where each time a different cell is perturbed. Only small differences can be found between the figures on the left and the respective figures on the right, meaning that there is no difference in either using various initial conditions or various initial perturbations. Therefore, in the remainder of this work, a combination is used, i.e. different random initial configurations with each time a different random initial perturbation. The choice of the ensemble size, denoted as $|\mathcal{E}|$, is based on the computing time needed. By examining the results from the timing tests (see Section 3.2.2), it was clear that an ensemble size of 100 would lead to excessive required computing time, yet 8, like used by Baetens and De Baets (2010) would for some 1D CAs not be sufficient. A compromise was found in an ensemble size of 30.

For 2D CAs, the topology designed by Baetens and De Baets (2010) is used (See Figure 3.2). This is an irregular tessellation of 675 cells, where the square cells on the sides were inserted to facilitate the implementation of periodic boundary conditions. Similar to 1D CAs, an ensemble of 30 combinations of random initial conditions and initial perturbations is used.

3.2.2 Time steps and timing

Since it is impossible to determine the Lyapunov exponent after an infinite number of time steps, an error will be made by determining the Lyapunov exponent after a finite number of time steps. To determine whether the number of time steps used is sufficiently large, a convergence threshold for the Lyapunov exponent of 0.001 is used. This threshold is based on Baetens et al. (2012). Figure 3.3 shows the maximum of the absolute difference in Lyapunov exponents between the last two, of 500, time steps, for the ensemble, for the 88 minimal rules. Rules 1, 9, 22, 33 exceed the threshold, but this may be disregarded since this is the maximum value of the absolute difference over an ensemble of 30 members and only four rules exceed the threshold. The increased accuracy achieved by increasing the number of time steps does not compensate the additional computing resources needed.
This is especially the case for SCAs, since each time step only a fraction p of all cells is updated, on average. Therefore, SCAs need to be evolved for more time steps to have an equal number of updated cells as in the case of deterministic CAs. If the number of time steps for deterministic CAs is denoted as T, and the number of time steps for SCAs is denoted T^*, the following relation

between T and T^* holds:

$$T^* = \frac{T}{p}. \tag{3.1}$$

The resemblance with asynchronous CAs, for which the number of time steps was derived by Baetens et al. (2012), can be seen easily. In this dissertation 100 equidistant update probabilities values will be considered, leading, through Eq. (3.1), to a 520 fold computing time required for SCAs compared to deterministic CAs.

Timing of the script executions was done on the Gengar cluster, but the batch job execution was done on the Raichu cluster, which is about one third faster than the Gengar cluster. Timing is given in Table 3.2. The final setup that was chosen is the one where each script covers one rule, 30 combinations of IC and IP and 500 time steps.

Table 3.2: Timing of different experimental setups on the Gengar cluster.

$\lvert\mathcal{E}\rvert$	# rules	# time steps	# p-values	Stochastic	Timing (h)
1	88	500	/	No	00:17:30
100	1	500	/	No	00:19:30
10	1	500	/	No	00:02:00
1	1	$^{500}/_{p}$	100	Yes	02:20:00

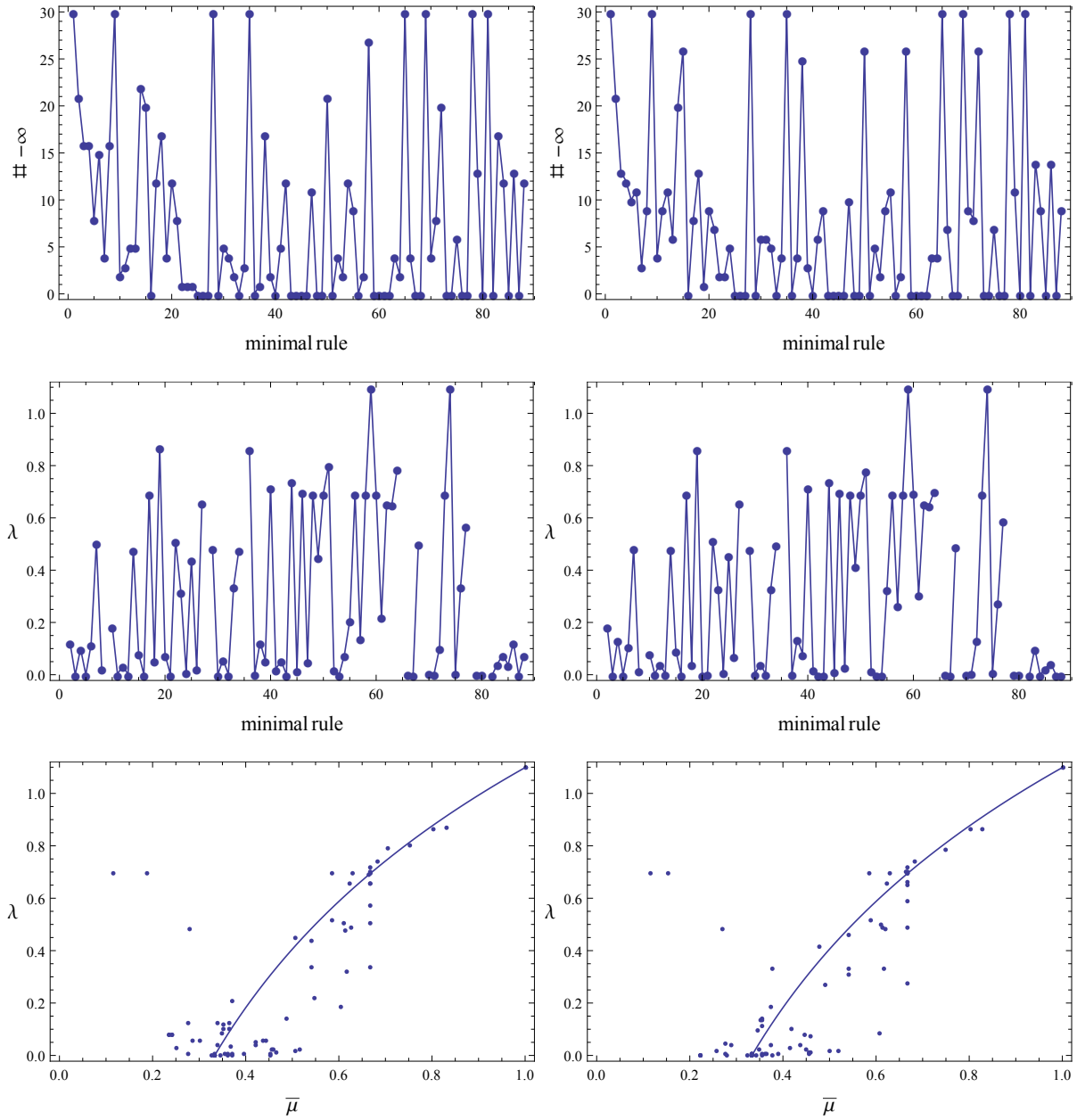


Figure 3.1: Comparison between the situation for the 88 minimal rules with 1 initial condition (IC) and 30 initial perturbations (IP) (left) and the situation for 30 IC where one IP is used (right) over 500 time steps, in the case of deterministic CAs. The top figures concern the number of Lyapunov exponents that are $-\infty$, the middle figures concern the average of the finite Lyapunov exponents and the lower figures represent the average of the finite Lyapunov exponents as a function of the input sensitivity ($\bar{\mu}$). Lines between data points are drawn solely to facilitate comparison.

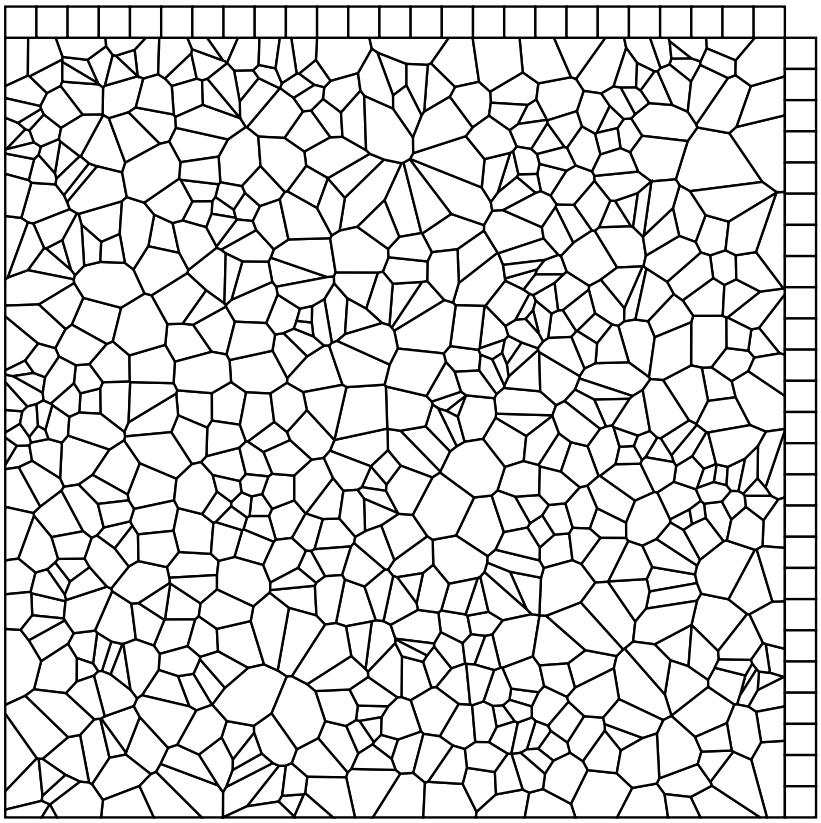


Figure 3.2: The tessellation designed by Baetens and De Baets (2010)

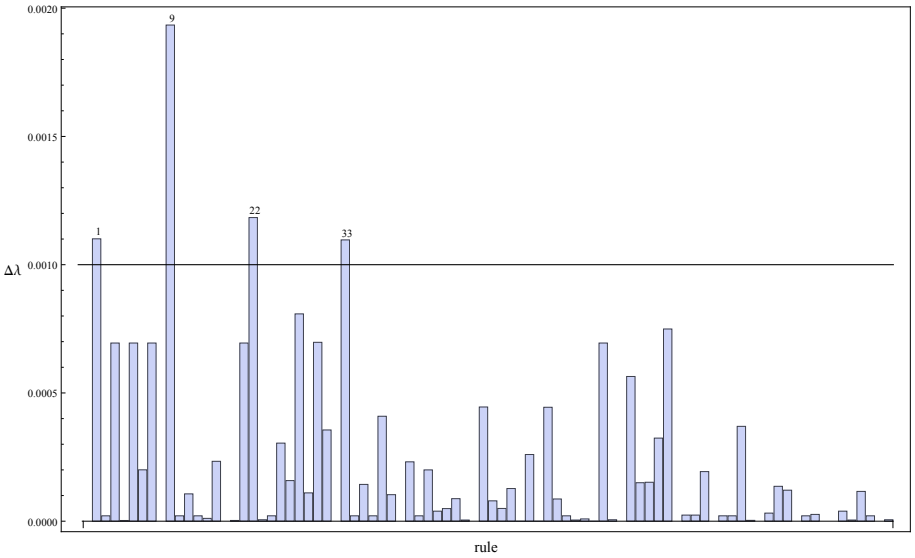


Figure 3.3: Difference in Lyapunov exponents between two consecutive time steps after 500 time steps

4. Stochastic cellular automata

Stochastic cellular automata (SCAs) have been introduced in Definition 2.2. In this chapter, mainly stochastic elementary cellular automata (SECAs) will be examined in detail. The last section will briefly discuss two-dimensional SCAs. More specifically, the case that will be examined is the one where the transition function ψ (see Definition 2.2) is given by:

$$\psi(\tilde{s}(N(c_i), t) = s(c_i, t),$$

meaning that with a probability $1 - p$ the rule is not applied. First, the effect of stochasticity on the space-time configuration will be examined for one exemplary rule per Wolfram class (Wolfram, 1984). Subsequently, the Lyapunov exponents (λ) and input sensitivity ($\bar{\mu}$) will be examined for all 88 minimal SECAs. Also the number of infinite MLEs will be examined.

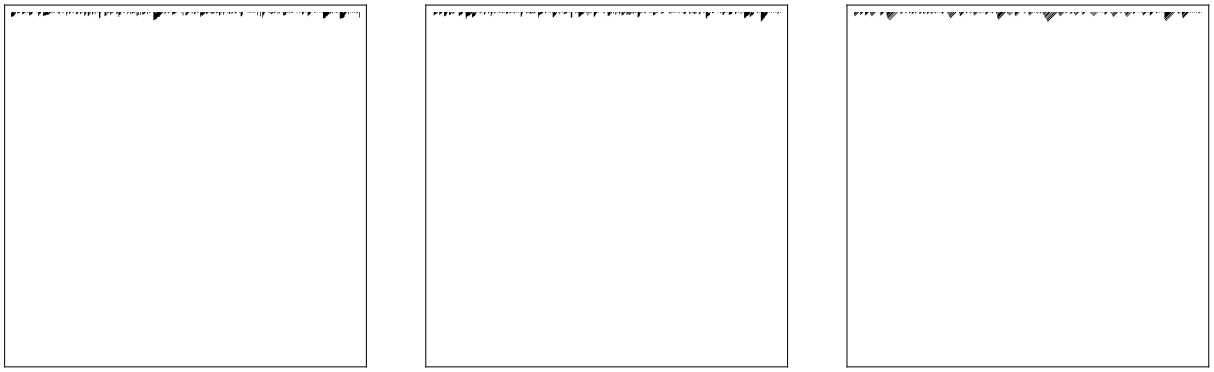
Essentially, the family of SECAs that is examined here may be seen as a family of SCAs where ϕ and ψ are two different rules chosen from the 256 rules of ECA. In this case, however, ψ always equals rule 204. Continuing along this line of reasoning leads to the realisation that deterministic ECAs are not only SECAs with $p = 1$, but also SCAs where $\phi = \psi$.

4.1 Visual examination

For a visual examination, the same four rules as the ones considered throughout Chapter 2 will be used (Figures 4.1, 4.2, 4.3 and 4.4, for rules 168, 100, 30 and 110, respectively). Since for values of p smaller than 1 the number of time steps is increased to maintain the number of cell updates (cfr. Eq. (3.1), the number of time steps in the corresponding space-time diagrams will also increase. To be able to compare diagrams across different values of p, there are two options. The first option involves adjusting the aspect ratio of the diagrams, which would, however, lead to increasingly flatter cells with decreasing p. The other option is to show only as many time steps as in the deterministic case, which is opted for in this dissertation. This means that for the case where $p = 0.5$ the configuration at only every other time step will be shown. However, for some values of p an additional minor adjustment of the aspect ratio is unavoidable, because it is impossible to show part of a time step.

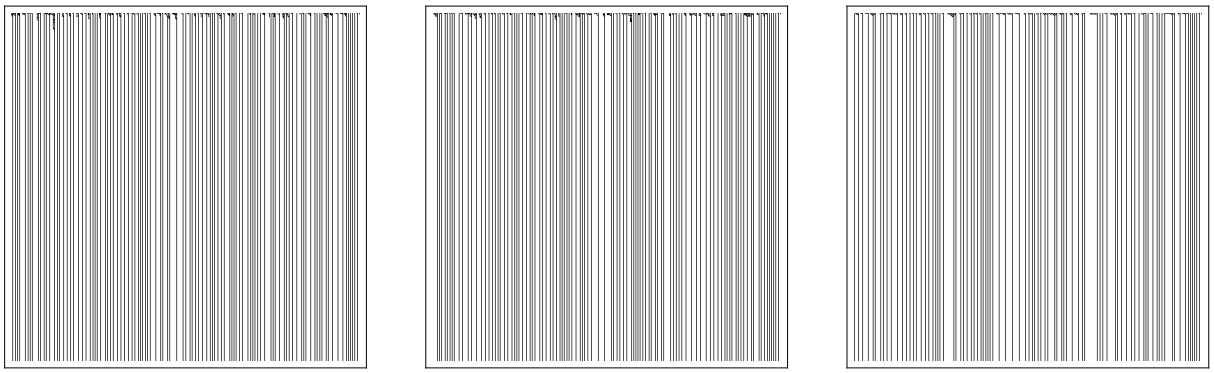
Rules 168 and 100, of which the space-time diagrams for different update probabilities are displayed in Figures 4.1 and 4.2, do not exhibit very interesting behaviour as a function of p, because, after a few time steps, the configuration reaches a fixed point in phase space. The behaviour of rules 30 and 110, displayed in Figures 4.3 and 4.4, on the contrary, seems to be greatly dependent on the update probability. The largest difference can be seen between the

deterministic case ($p = 1$) and the case where p is 0.8. From Figure 4.4 it can be deduced that adding stochasticity can give rise to more chaos in systems that are typically recognised by their more structured, long-lived patterns (Wolfram Class 4), whereas Figure 4.3 indicates that it can reduce chaos in systems that have a chaotic deterministic evolution (Wolfram Class 3). Later on, it will be found that Class 2 ECAs can also exhibit chaotic behaviour when stochasticity is introduced, but Class 1 ECAs always are unconditionally superstable.



(a) (b) (c)

Figure 4.1: Evolution of SECA 168 from one random initial condition, with p respectively 0.05 (a), 0.5 (b) and 1 (c)

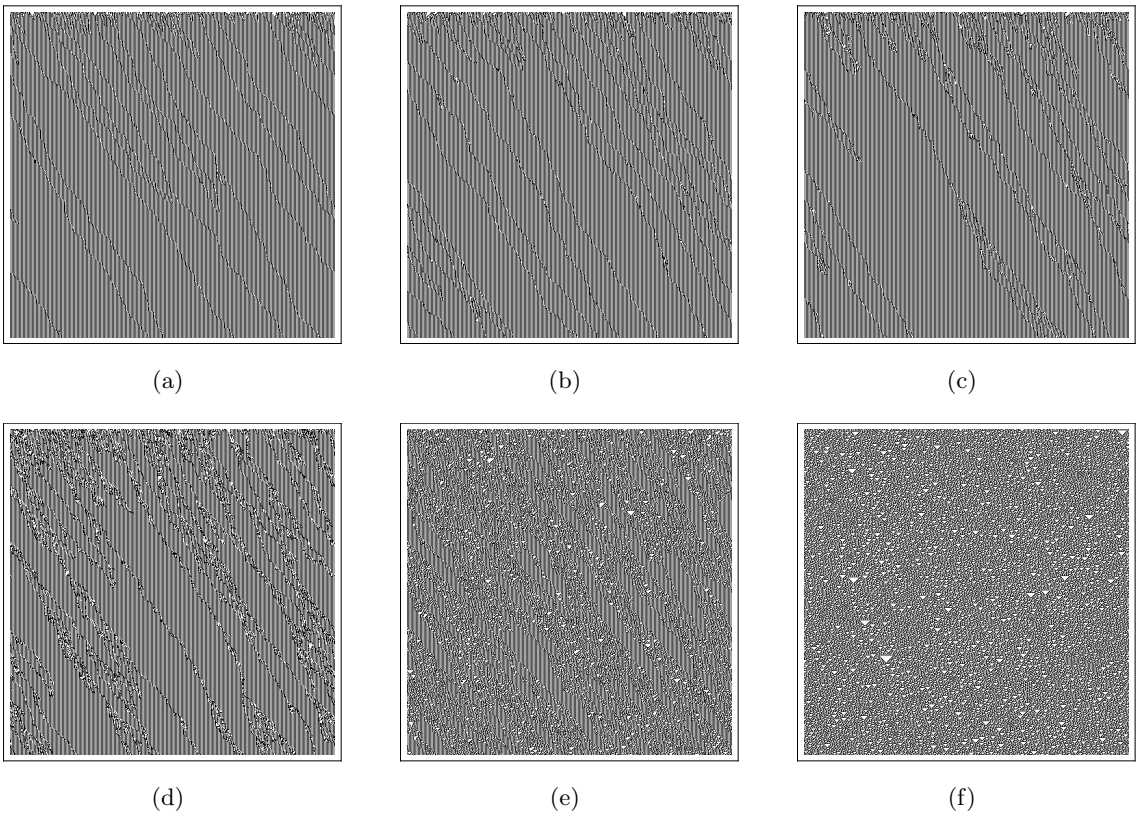


(a) (b) (c)

Figure 4.2: Evolution of SECA 100 from one random initial condition, with p respectively 0.05 (a), 0.5 (b) and 1 (c)

4.2 Stability of SECAs

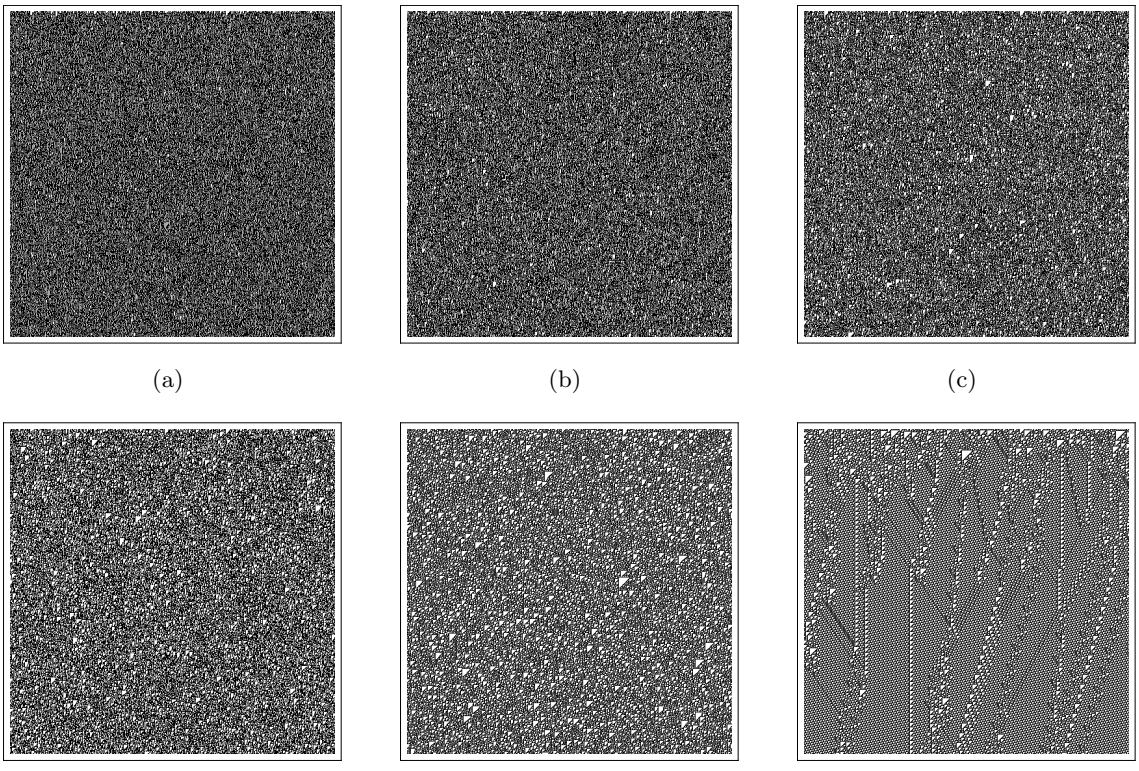
Many different behaviours were encountered, which makes it not easy to find some sort of classification as was found for deterministic CAs and asynchronous CAs. Of course, for every value of p the rules could be classified according to the classification system of Baetens and De Baets (2010), but the goal here is to find some kind of classification that is based on the entire set of Lyapunov exponents for the different values of p.



(a) (b) (c)

(d) (e) (f)

Figure 4.3: Evolution of SECA 30 from one random initial condition, with p respectively 0.05 (a), 0.2 (b), 0.4 (c), 0.6 (d), 0.8 (e) and 1 (f)



(a) (b) (c)

(d) (e) (f)

Figure 4.4: Evolution of SECA 110 from one random initial condition, with p respectively 0.05 (a), 0.2 (b), 0.4 (c), 0.6 (d), 0.8 (e) and 1 (f)

4.2.1 Normalisation

The first thing that was done to enable a comparison between the MLEs of SECAs and deterministic ECAs was a normalisation. This was done by dividing the Lyapunov exponents in both cases by the maximum of their respective upper bounds. This maximum is found for $\bar{\mu} = 1$. According to Eq. (2.5), the MLE of deterministic CAs can be at most $\log(3)$
For a SCA no upper bound has been reported in literature. However, the same equation is applicable, but the input sensitivity ($\bar{\mu}$) now depends on the probability p. In the remainder, the input sensitivity of SCAs will be denoted as $\bar{\mu}_s$, while $\bar{\mu}_d$ will be used to denote the input sensitivity in the deterministic case. In theory, $\bar{\mu}_s$ can be found as the p-weighted average of $\bar{\mu}_d$, on the one hand, and the input sensitivity that would be obtained if cells were not updated, on the other hand. The latter equals $1/\overline{V}$ because the state of a cell at time step $t+1$ remains the same as its state at the time step t, such that the following expression for $\bar{\mu}_s$ is found:

$$\bar{\mu}_s = \frac{(1-p)|\mathcal{T}|\frac{1}{\overline{V}} + p\,|\mathcal{T}|\bar{\mu}_d}{|\mathcal{T}|}. \tag{4.1}$$

Using this expression leads to a mean-field estimate of the upper bound on the MLE for SCA ($\lambda_{m,s}$):

$$\lambda_{m,s}(\bar{\mu}, p) = \log(\bar{\mu}_s\,\overline{V}) = \log(1 - p + p\,\overline{V}\bar{\mu}_d). \tag{4.2}$$

The maximum value of this upper bound for a 1D SECA is found for $\bar{\mu}_d = 1$, and equals $\log(1+2p)$, which is then used to normalise the numerically obtained MLEs of SECAs.
For the sake of completeness, it should be mentioned that this normalisation is not a perfect one, because the dependence of the MLE on $\bar{\mu}_d$ is not identical in the stochastic and deterministic case (see Figure 4.5). This can easily be seen when the exponential of $\lambda_{m,s}$ is considered.

$$e^{\lambda_{m,s}} = 1 - p + p\,\overline{V}\bar{\mu}_d.$$

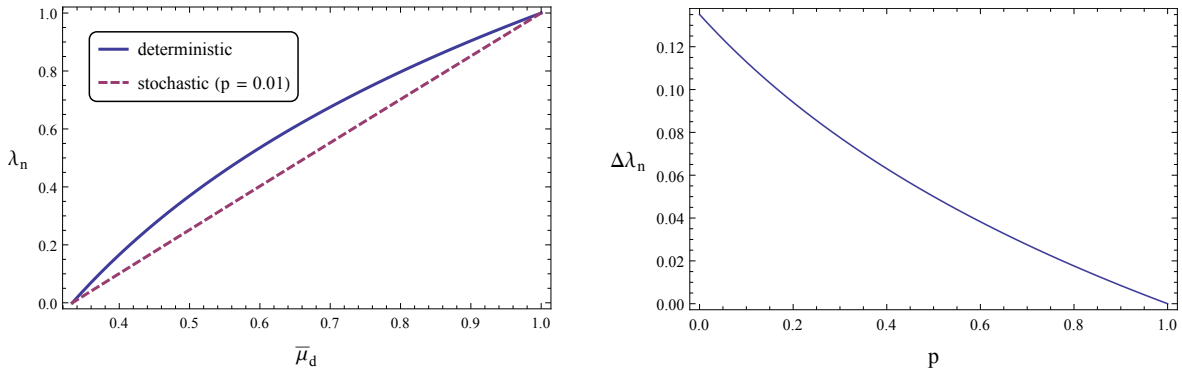
For the deterministic case ($p = 1$), a linear dependence, with intercept equal to zero, is found, whereas for $p < 1$ an extra constant appears. This extra constant leads to a difference when the natural logarithm is taken. This difference increases as p decreases, so for the limit case where $p = 0$, the following normalised MLE is found for deterministic CAs:

$$\begin{aligned}\lambda_{n,d} &= \frac{\log\left(\overline{V}\bar{\mu}_d\right)}{\log\left(\overline{V}\right)} \\ &= \frac{\log\left(\overline{V}\right) + \log\left(\bar{\mu}_d\right)}{\log\left(\overline{V}\right)} \\ &= 1 + \frac{\log\left(\bar{\mu}_d\right)}{\log\left(\overline{V}\right)} \quad ,\end{aligned} \tag{4.3}$$

and analogously for SCAs:

$$\begin{aligned}\lim_{p\to 0}\lambda_{n,s} &= \lim_{p\to 0}\frac{\log\left(1-p+p\,\overline{V}\bar{\mu}_d\right)}{\log\left(1-p+p\,\overline{V}\right)}\\ &\overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{=}} \lim_{p\to 0}\frac{\frac{-1+\overline{V}\bar{\mu}_d}{1-p+p\,\overline{V}\bar{\mu}_d}}{\frac{-1+\overline{V}}{1-p+p\,\overline{V}}}\\ &= \frac{-1+\overline{V}\bar{\mu}_d}{-1+\overline{V}}\\ &= \frac{1}{1-\overline{V}}+\frac{\overline{V}}{\overline{V}-1}\bar{\mu}_d \quad . \end{aligned} \tag{4.4}$$

One could argue that perhaps it would then be better to normalise with $\bar{\mu}_d$ equal to the one found for every rule. However, normalisation is preferably done on the basis of a constant reference value, therefore, $\bar{\mu}_d$ is not varied in the normalisation, but it is put at its maximal value. Another problem with $\bar{\mu}_d$-dependent normalisation can easily be seen when $\bar{\mu}_d = 1/\overline{V}$ is inserted in Eq. (4.2). In the remainder the normalised MLE (nMLE) will be denoted as λ_n.



(a) Normalised MLEs for SCAs with p equal to 1 (deterministic case) and 0.01

(b) The maximum of the error between a SCA and a deterministic CA as function of p

Figure 4.5: Theoretical error between the Lyapunov exponents of deterministic and stochastic CAs after normalisation

An overview of all rules before normalisation along with the upper bound (cfr. Eq. (4.2)) is given in Figure 4.6. The data points represent the averages of the finite MLEs and the averages of the input sensitivity $\bar{\mu}_s$ of the 88 minimal SECAs as a function of the probability p. The surface represents the upper bound on the MLE, which is given by Eq. (4.2).
The situation on the front pane, p=1, is exactly the same as in Figure 2.6. When p decreases, the Lyapunov exponent decreases as well, because fewer cells are updated every time step, leading to a decreased speed of defect propagation. Similar to Figure 2.6, a number of rules do not fit the upper bound. Those are in general rules with an input sensitivity less than 1/3 or rules with an input sensitivity larger than 1/3 and low MLE values. Theoretically, the former should lead to a negative MLE, but CAs are spatially heterogeneous and therefore the local input sensitivity can be larger than 1/3 while the global one is smaller. A similar reasoning applies to those rules with an input sensitivity larger than 1/3. Something else that causes some SECAs to deviate from the theoretical upper bound is the fact that, due to the absence of a second spatial dimension, a defect that reaches a position from where it cannot further propagate in a certain direction does

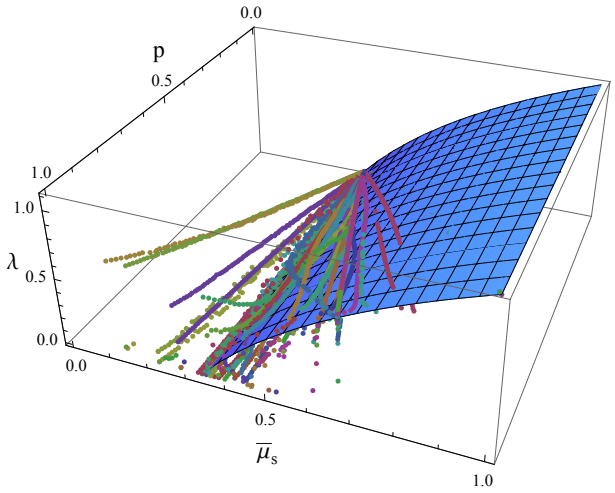


Figure 4.6: Plot of the average finite Lyapunov exponents (λ) and mean input sensitivity ($\bar{\mu}_s$) of the 88 minimal rules for different update probabilities (p), along with the upper bound.

not have the possibility to avoid this obstacle. In 2D SCAs this possibility does exist, which leads to defect propagation rates that are much closer to the theoretical upper bound. In the limit case where $p = 0$, the MLE is $^{\log(1)}/_{t} = 0$ and the input sensitivity is $^{1}/_{\overline{V}}$. This point is, however, never included because it is straightforward and because normalisation is impossible.

4.2.2 Behavioural classes

After normalising the numerically determined exponents, the rules were grouped according to their nMLEs. To avoid confusion, the newly formed classes will be given letters instead of numbers. Eventually, the following classification is proposed. First, the rules that belonged to Class 1 or Class 3 of the classification by Baetens and De Baets (2010) for every p, were grouped into two classes. These are the rules that are unconditionally superstable or unconditionally unstable, respectively, irrespective of the update probability p. They will be denoted as Class A and Class B, respectively. Upon forming these groups, 71 more rules out of 88 remain to be classified.

The 71 remaining rules were first divided into two groups: rules that for some values of p lead converging phase space trajectories (Class 1) and rules that never display such behaviour, irrespective of the update probability. The rules in these groups were then clustered using the FindClusters algorithm of Mathematica (Wolfram Research inc., 2008) on the basis of their MLE sequences. This algorithm uses by default the Euclidean distance between corresponding points as a dissimilarity function. Since the distance to infinity cannot be calculated, $-\infty$ was replaced by -1. Some post-processing was done to get clusters of rules with similar behaviour. Finally, the first group, being the one containing rules that for some values of p give rise to converging trajectories, was divided into three subgroups. Two of them are referred to as Class C and D. The third subgroup contains rules with properties of Classes C and D, and is referred to as Class E. These kind of results were to be expected, since classifications are artificial.

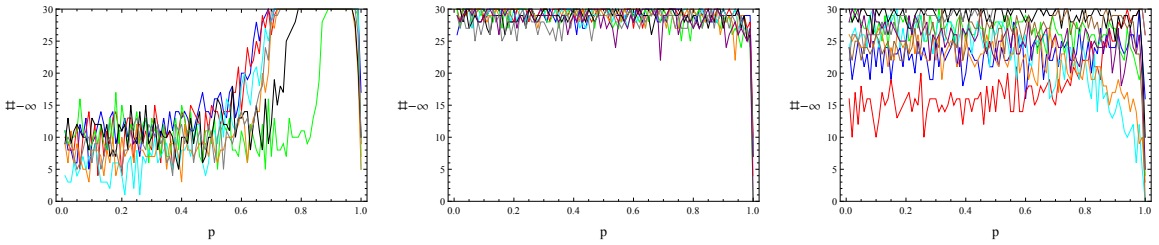
For what concerns the subgroup enclosing rules that never give rise to $\lambda = -\infty$, irrespective of the update probability, one rule is found, namely rule 74, whose behaviour is almost identical to that of rule 56, which belongs to Class E. Therefore, rule 74 was added to Class E. Other

subgroups were not as easily distinguishable, since all rules behave more or less similar. There are several rules that exhibit specific behaviour that looks like other rules, but never are there larger subgroups to be found. Therefore these remaining rules were grouped in one class, referred to as Class F.

Now that different behavioural classes have been defined, it is worthwhile to characterize these classes in more detail. As mentioned earlier, Classes A and B enclose those rules that are unconditionally superstable and unconditionally unstable, respectively, irrespective of p. Therefore, for Class A rules, the number of MLEs that is $-\infty$ is the same as the ensemble size, irrespective of p, whereas it is zero for all rules belonging to Class B.
The behaviour of the rules belonging to Class C is threefold, as can be seen in Figure 4.7(a) (and in Appendix Figure A.2). More precisely, for $p = 1$ or close to 1, conditionally unstable behaviour is observed. But, as p decreases, a range of p-values exists for which the MLE is $-\infty$ for all members of the ensemble. Finally, when p decreases even further the MLE becomes again positive and the number of MLEs that equals $-\infty$ decreases with decreasing p.
The behaviour of rules belonging to Class D is twofold, as can be seen in Figure 4.7(b) (and in Appendix Figure A.3). The distinction in behaviour becomes apparent from investigating the number of Lyapunov exponents that are $-\infty$, rather than from studying the magnitude of the Lyapunov exponents. The finite nMLEs are always very close to zero, and would probably become zero, if an infinite number of time steps could be considered. For $p = 1$, Class D rules have more than five members of the ensemble that are unstable (MLE $\neq -\infty$). For lower values of p, roughly less than five members of the ensemble are unstable. Therefore, we can say that Class D rules are close to being unconditionally superstable.
Since Class E is the intermediary class between Class C and Class D, the behaviours of the rules belonging to Class E are internally different and have properties of both classes. The general trend that can be discerned among the rules in Class F is that the number of members of the ensemble that lead to converging phase space trajectories is more or less independent of the update probability p. The nMLE itself is also largely independent of p, leaving some minor exceptions aside.

When comparing the Lyapunov exponents of Class B and Class F, given in Figures A.1 and A.5, respectively, it can be seen that they exhibit similar behaviour. In fact it is likely that there are only four types of behaviour (Classes A, B, C and D), since Class E and Class F are intermediary classes. The question might arise whether it would not be better to put the rules in these intermediary classes in one of the four other classes.

The reason why stability in Class C rules decreases again for low values of p is not yet fully understood, but, the following observation was made when analysing the space-time diagrams of rules in this class and of the multiplicity of defects. Essentially, it appeared that the initial defect introduced in SECAs belonging to both Class C and Class D stays very localised in the case of the deterministic scenario ($p = 1$). For Class D rules this is still the case when p decreases, meaning that there is no need for an immediate stabilisation of the system. In contrast, for Class C rules, defects start to propagate to the left and to the right very quickly, therefore, stabilisation needs to happen at the start, because otherwise too many cells contain a defect, so



(a) Class C (b) Class D (c) Class E

Figure 4.7: Number of infinite MLEs as a function of the update probability p for Classes C, D and E of the SECAs.

that stabilisation is no longer possible.

Table 4.1: Classification of the 88 minimal 1D SECAs.

Class	Rule number
A	0, 8, 32, 40, 128, 136, 160, 168
B	51, 54, 57, 60, 105, 108, 150, 156, 204
C	1, 3, 7, 19, 23, 50, 178
D	2, 10, 15, 34, 42, 130, 162, 170
E	5, 14, 24, 56, 138, 142, 152, 184
F	4, 6, 9, 11, 12, 13, 18, 22, 25, 26, 27, 28, 29, 30, 33, 35, 36, 37, 38, 41, 43, 44, 45, 46, 58, 62, 72, 73, 74, 76, 77, 78, 90, 94, 104, 106, 110, 122, 126, 132, 134, 140, 146, 154, 164, 172, 200, 232

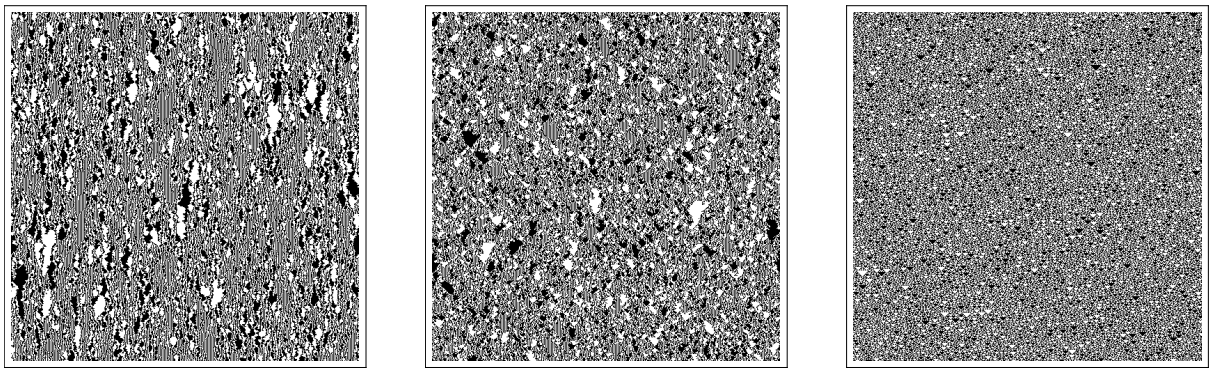
4.3 Detailed examination

Now that different classes have been found, an exemplary rule from each class will be examined in detail. This will be done by looking at their space-time diagram for different values of p, the standard deviation on the MLE across the members of the ensemble, the final density as a function of p and a measure for determining the complexity of the configuration at the final time step. The latter measure is the Lempel-Ziv complexity (LZ) (Lempel and Ziv, 1976), which will be explained in Section 4.3.4. Besides, for the rules belonging to Class C a close-up of the range of values of p between 0.99 and 1 will be presented because the stability of SECAs changes drastically across this range. The rules for which these measures are determined are 150, 23, 130, 152, 110 and 30. Class A rules will not be examined in detail, since their behaviour is trivial, irrespective of the update probability. More precisely, such rules reach a uniform configuration after a certain number of time steps and cannot escape from this configuration anymore, leading to straightforward values of the measures under consideration. It should also be noted that the MLEs and the detailed measures were calculated in two different runs. This might lead to small discrepancies due to chance. The space-time diagrams were calculated on the desktop and only for one combination of IC and IP.

4.3.1 Space-time diagram

The space-time diagrams of rules 150 (Class B), 23 (Class C), 130 (Class D) and 152 (Class E) are given in Figures 4.8, 4.9, 4.10 and 4.11, respectively. The space-time diagram of rule 110 (Class F) has already been shown in Figure 4.4. Also rule 30, whose space-time diagram is given in Figure 4.3, belongs to class F.

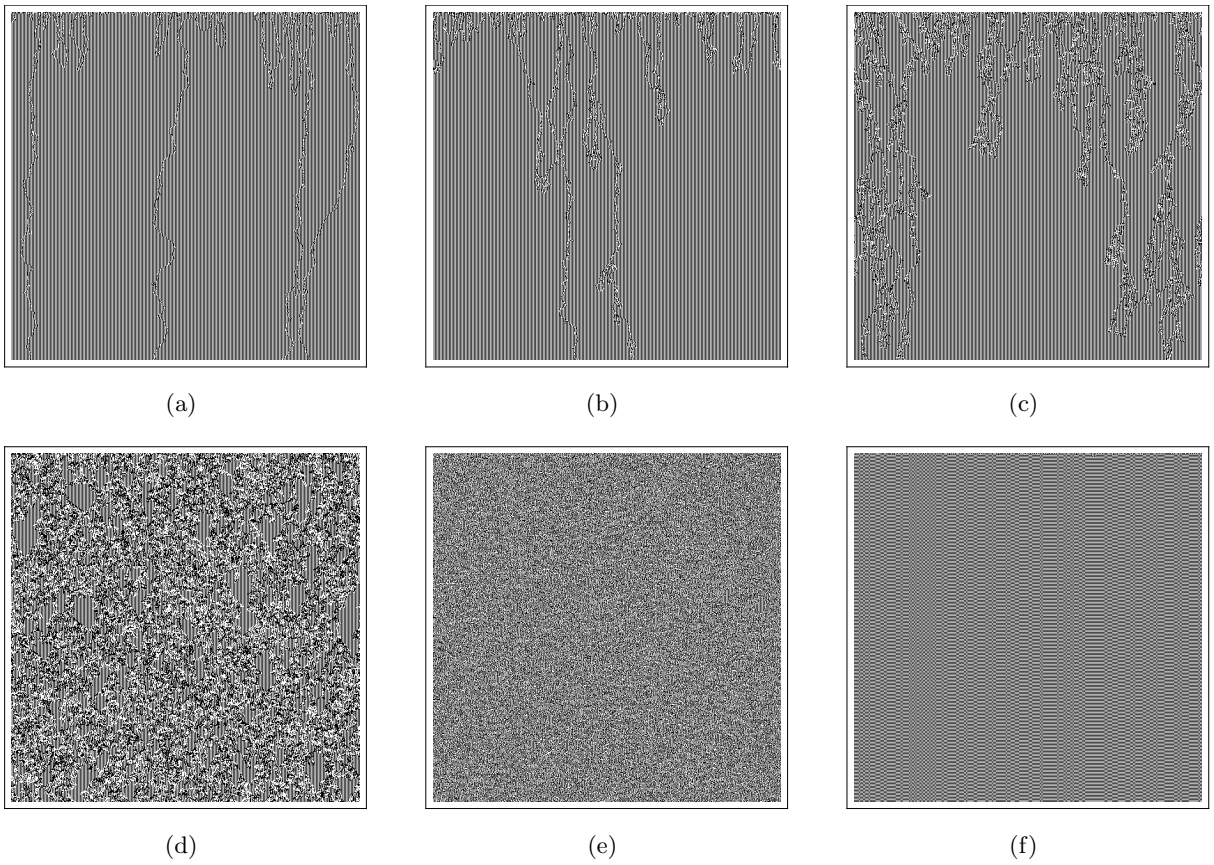
Rule 150 displays chaotic behaviour irrespective of the update probability p. For $p = 1$ triangular patterns can be discerned, which disappear when p decreases. A shift is observed towards other short lasting structures, such as black areas, white areas and areas in a striped pattern.



(a) (b) (c)

Figure 4.8: Evolution of SECA 150 from a random initial condition, with p respectively 0.05 (a), 0.5 (b) and 1 (c).

For Class C rules it would be expected to observe some sort of stable pattern for those p that lead to infinite MLEs. Observing the space-time diagram of rule 23 in Figure 4.9, however, it can be concluded that this is not the case. So, although this pattern seems chaotic, it is stable. One could look at this as if the pattern at every time step is so random that one defect has no effect. When p is decreased beyond the range of values that give rise to infinite MLEs, patterns (repeats of 01) emerge that seem to be stable. Yet, the MLE indicates that, although these patterns seem stable, they can be easily disturbed by inserting a defect. This illustrates the difference between phase space and configuration space. The inherent instability of a series of 01 is observed when this pattern is tested for the range of values of p where random initial conditions lead to stability. This test consists of determining the MLE for the following setting: $p = 0.8$, the initial condition consists of 50 repeats of 01, the 50th cell is perturbed, 100 time steps are considered and 30 repetitions are performed to account for the stochastic nature of the SECA. This leads to only two out of 30 MLEs being $-\infty$ and an average nMLE of 0.615.

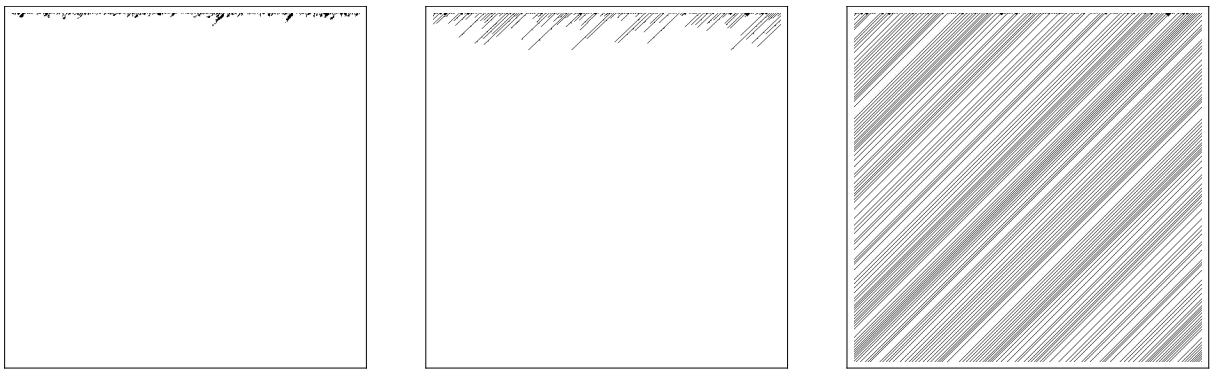


(a) (b) (c)

(d) (e) (f)

Figure 4.9: Evolution of SECA 23 from a random initial condition, with p respectively 0.05 (a), 0.2 (b), 0.4 (c), 0.6 (d), 0.8 (e) and 1 (f).

The space-time diagrams of rule 130 are consistent with its MLE graphs. The graphs representing the number of infinite MLEs suggest that in some cases a defect can remain, but the low magnitude of the remaining positive MLEs indicates that these defects do remain localised. The development of a uniform configuration in combination with a MLE that can be positive indicates that in phase space, where single defects are considered, defects are not always cancelled, whereas in configuration space two or three defects may cancel out each other until all defects are cancelled.



(a) (b) (c)

Figure 4.10: Evolution of SECA 130 from a random initial condition, with p respectively 0.05 (a), 0.95 (b) and 1 (c).

Rule 152 is a Class E rule. It differs from rule 130 in that it has fewer MLEs that equal $-\infty$ and that when p decreases beyond 0.1, the finite nMLEs increase towards 0.5. This range of update probabilities is also the one where the number of infinite MLEs reaches its maximum. This behaviour is reflected in the space-time diagrams. The situation at $p = 0.05$ shows that it takes longer for the SECA to reach a uniform configuration. This means that the chance of a defect not being cancelled, in configuration space (see space-time diagram) and in phase space (see MLEs), increases. It is possible that the few positive MLEs at low values of p would also go to $-\infty$ if t approaches ∞.

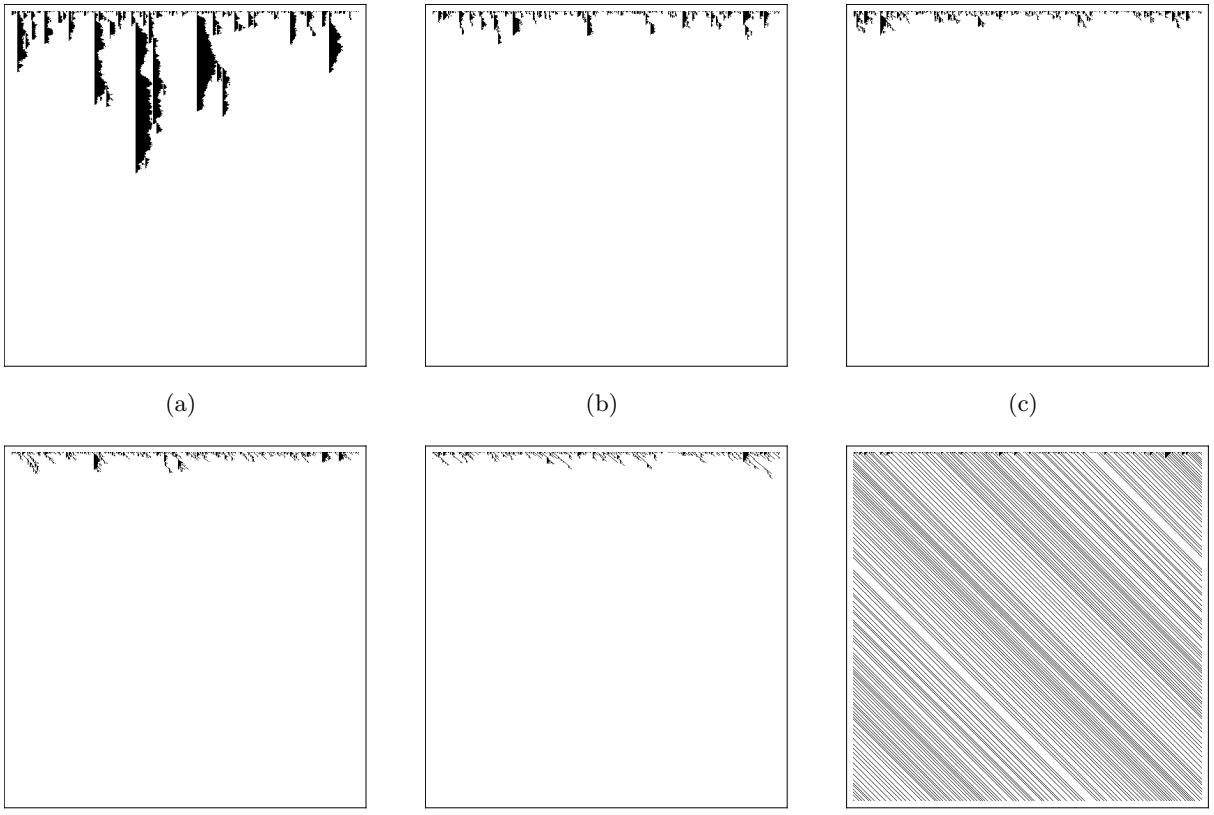


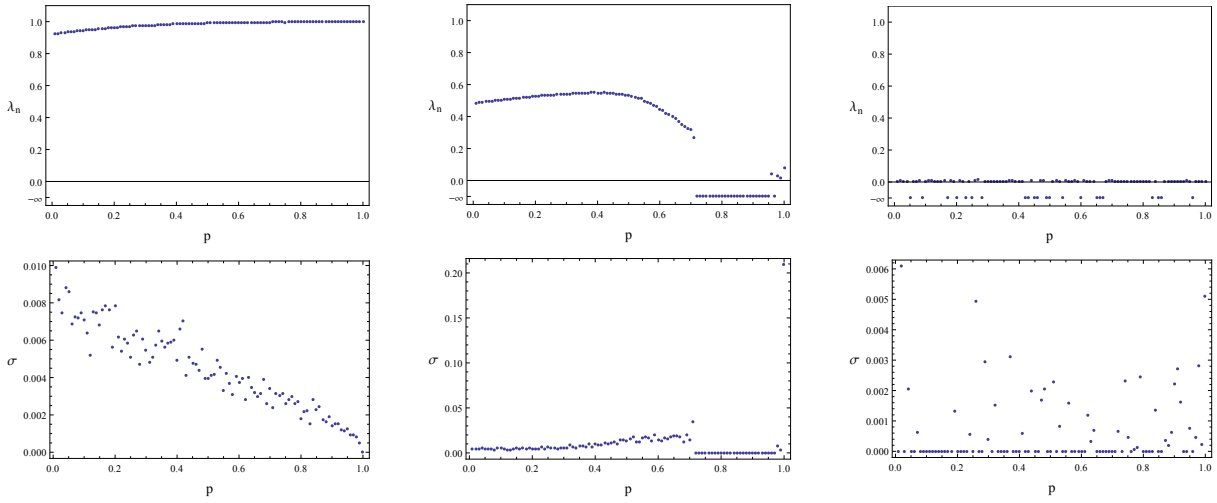
Figure 4.11: Evolution of SECA 152 from a random initial condition, with p respectively 0.05 (a), 0.2 (b), 0.4 (c), 0.6 (d), 0.8 (e) and 1 (f).

The Class F rules 30 and 110 have already been discussed in Section 4.1. The space-time diagrams of rule 30 (see Figure 4.3) resemble the ones of rule 23, however, without the situation where the chaos had increased. Therefore, it is not surprising that this rule is unstable. In the space-time diagrams of rule 110 (see Figure 4.4) the long-lived patterns that are so typical for a Class 4 rule, as introduced by Wolfram (1984), are replaced with more chaotic patterns. This change turns out to have not much effect on the stability.

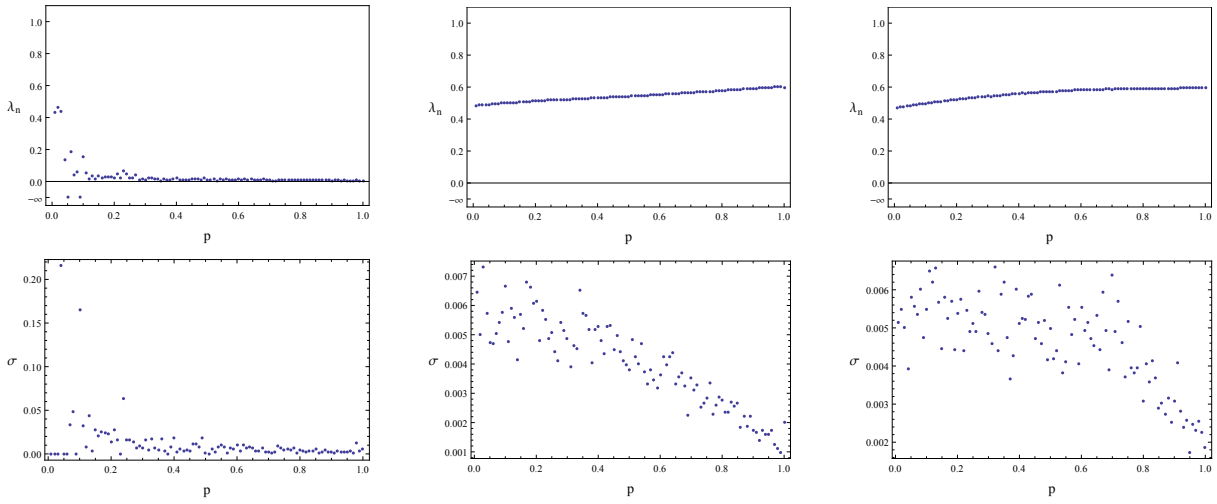
4.3.2 Standard deviation

The standard deviation (std) of the nMLE across the members of the ensemble of the exemplary rules is given in Figure 4.12. It is important to notice that the scale of the vertical axis differs between the six graphs. For the sake of clarity, the nMLEs of the different rules are also given.

At first, it could be expected that the standard deviation would increase with decreasing p, since it is to be expected that a lower update probability would introduce additional variability between the members of the ensemble. This is the case for Class B and Class F rules.



(a) rule 150 (b) rule 23 (c) rule 130



(d) rule 152 (e) rule 110 (f) rule 30

Figure 4.12: Average and standard deviation of the finite nMLEs of rules 150, 23, 130, 152, 110 and 30 versus the update probability.

For rule 130, and even more pronounced for rule 23, the highest standard deviation is found in the deterministic case. This can be explained by looking at the difference in the approaches for determining the MLE between Bagnoli et al. (1992) and Baetens and De Baets (2010). Bagnoli avoided problems with the incapability of some CAs for certain IP and IC to reach their most stable configuration by swapping positions of a few cells at every time step. This extra push makes sure that the CA does not get stuck in an unstable configuration if there exists a more stable one. On the other hand, the approach used in this thesis does not push CAs to their most stable configuration, but the stability of a CA is observed across an ensemble of IC, leading to unstability for some IC. This might explain the big drop in standard deviation, and MLE, for

rule 23 for the transition region between the deterministic and stochastic settings. A cell that is not updated can be seen as a push to move away from the configuration in which the CA got stuck. This reasoning also explains the increase in stability for these two rules, as p becomes smaller than 1. Another increase in standard deviation for rule 23 is found in the range of values of p where rule 23 leaps from unconditionally to conditionally stable behaviour. This is to be expected, since the stochasticity causes a transition to happen around a certain value, but not exactly at a constant value.
The std plot of rule 152 looks like the one of rule 130. This makes sense, since the MLE plots of both rules are also very similar. The MLE of rule 152 increases again for very low values of p, giving rise to a transition zone that brings along an increase in standard deviation here.

From the plots of rules 150, 110 and 30 the magnitude of the standard deviation due to stochasticity can be deduced. The plots also confirms the hypothesis that rules belonging to Class F are very alike those of Class B.

4.3.3 Density

In Chapter 2 it was mentioned that the density is not a good measure since it does not converge as t becomes large. The low value of the standard deviation observed in Figure 4.13 indicates that although this measure does not approach a constant value, it fluctuates around the same value, leading to a small standard deviation. This does, however, not mean that it is a good measure for stability, as is clearly illustrated by rule 23, which becomes stable for a certain range of values of p whereas the density remains constant. The density gives an indication of the configuration, but it is not very distinctive. The link between the density and the space-time diagram can easily be seen. When comparing both, it must be kept in mind that the densities in Figure 4.13 are not based on the space-time diagrams depicted in this dissertation and that the densities represent the averages over 30 combinations of IC and IP, whereas the space-time diagrams are based upon one combination of IC and IP.

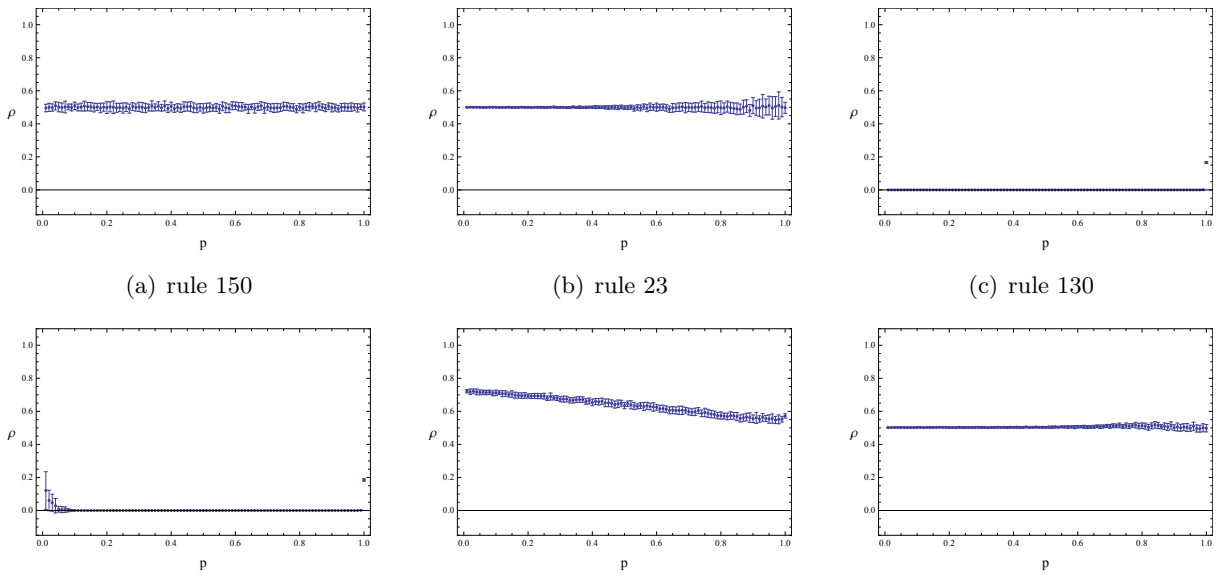
4.3.4 Lempel-Ziv complexity

The Lempel-Ziv complexity (Lempel and Ziv, 1976), denoted in the remainder as ℓ, can be defined as follows.

Definition 4.1 *Let $v = v_1v_2...v_k$ and $w = w_1w_2...w_n$ be binary strings. v is a prefix of w if $v_i = w_i$ for $1 \leq i \leq k$. If $k < n$, then v is said to be a proper prefix of w.*
Now, $y_1|...|y_r$ is called the Lempel-Ziv partition of w, if

- *$y_i \neq y_j$ for all $i = 1, \ldots, r-1$ and $j = 1, \ldots, i-1$*
- *$w = y_1y_2...y_r$*
- *every proper prefix of y_i is equal to a y_j for all $i = 1, \ldots, r$ and $j = 1, \ldots, i-1$*

The Lempel-Ziv complexity of w, $\ell(w)$, is then defined as the number of patterns, r, in the Lempel-Ziv partition of w.



(a) rule 150 (b) rule 23 (c) rule 130

(d) rule 152 (e) rule 110 (f) rule 30

Figure 4.13: Average and standard deviation of the density (ρ) on the final time step of rules 150, 23, 130, 152, 110 and 30.

According to this definition y_r could satisfy $y_r = y_i$ for some $i = 1, \ldots, r-1$. If it does satisfy this condition, w is called an open sequence. Otherwise, w is called a closed sequence. In this dissertation the Lempel-Ziv complexity of an open sequence will be adjusted in such a way that it equals $r - 1$, rather than r, making it equal to the maximal number of unique substrings.
For example, consider the binary string 111000111110100. The Lempel-Ziv partition of this string is given by 1|11|0|00|111|110|10|0. According to Definition 4.1, the Lempel-Ziv complexity would be eight, but since this is an open sequence, as the last substring is not unique, the Lempel-Ziv complexity will be considered seven in this dissertation.
It can easily be seen that the Lempel-Ziv complexity depends on the length of the string. Therefore, a normalisation will be applied according to:

$$\ell_n = \frac{\ell - \ell_{min}}{\ell_{max} - \ell_{min}}, \tag{4.5}$$

where ℓ_n, ℓ_{min} and ℓ_{max} represent the normalised LZ (nLZ) and the minimum and maximum LZ for a string of a given length, respectively. The nLZ has a value between zero and one and may be used to quantify the complexity of a certain configuration of a (S)ECA at a certain time step. The higher the complexity, the closer to one the nLZ will be. Uniform configurations will have a nLZ equal to zero. The average values of the nLZ of the SECA configuration at the last time step, as a function of p, can be found in Figure 4.15. Before discussing the results, the convergence of the LZ will be examined. Figure 4.14 shows the nLZ as a function of the time step for rule 23 with update probability 0.5 for a random initial condition. It can be seen that no convergence to a fixed value is reached. The results of the exemplary rules show a smooth line, indicating that, similar to the density, the LZ exhibits some sort of convergence.

Rule 150 behaves as was to be expected. It leads to complex configurations in the deterministic case and continues to do so when stochasticity is introduced.

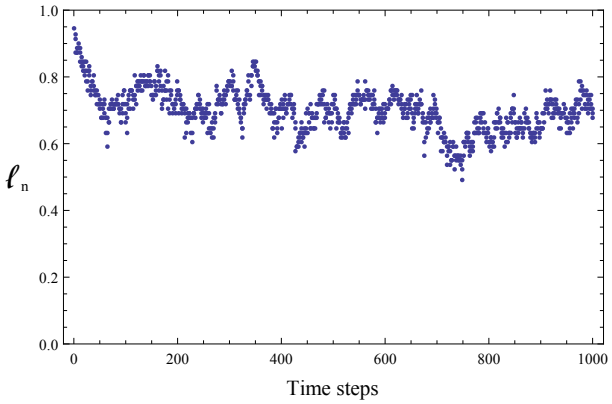
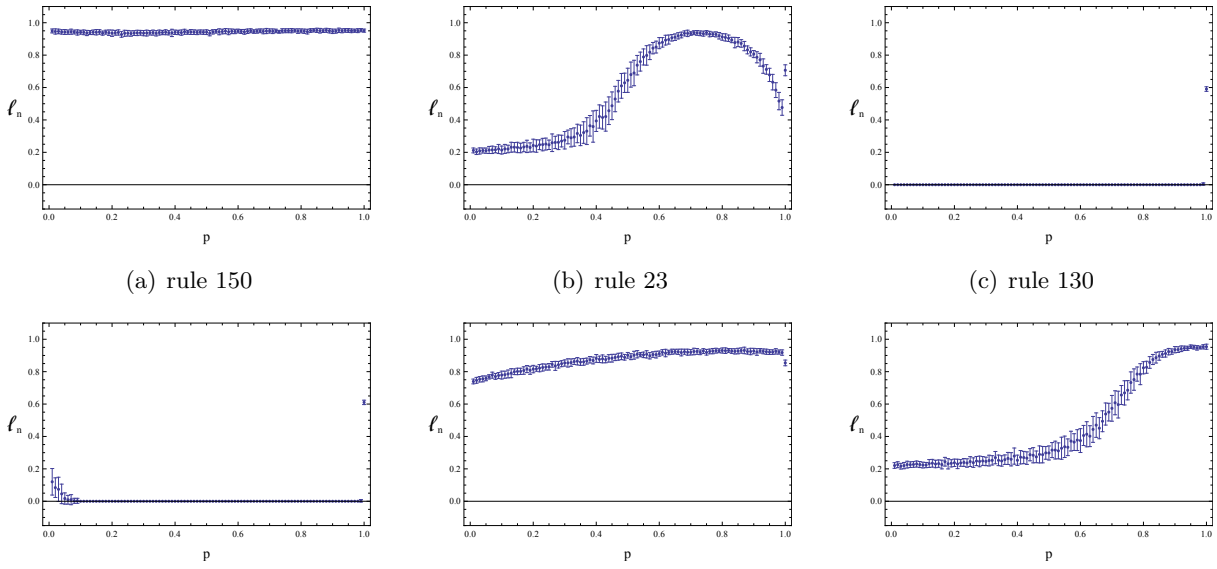


Figure 4.14: The normalised Lempel-Ziv complexity for rule 23 with update probability being 0.5 and starting from a random initial condition.

The Class C example (Rule 23) seems to have a nLZ that is in concordance with its space-time diagram (see Figure 4.9). At first, the complexity of the final configuration increases with decreasing p to reach a maximum and then decreases again with decreasing p.
Generally speaking, the LZ of a CA seems in accordance with its space-time diagram. Therefore, the LZ can be used as a measure for characterizing the configuration of a CA. A remark can be made, namely, the LZ suggests that the exemplary rules of Class B and Class F are not as alike as was indicated in Section 4.2.2. However, this conclusion is only based on one example from Class B (rule 150) and two examples from Class F (rules 30 and 110). Maybe even Class B rules exhibit different LZ within their own class. One might also argue that classification was done on the basis of the rules' stability in phase space, whereas the LZ is based on configuration space. To address the problem that only one or two examples are selected, the LZ of all minimal rules was computed (see Appendix A.1.2). From this data it could be concluded that the LZ profile of Class B rules is quite uniform, with only minor exceptions. Both Classes C and D exhibit two different kinds of profiles, which are also mutually different. The LZ profiles of Class E rules can be divided into four different types, similar as those in Classes C and D. In Class F, a variety of profiles emerge, both similar to profiles found in other classes and completely new profiles. This means that the proposed classification based on the MLEs is still more or less valid if the LZ measure is considered. This is along the line of the expectations, since defect propagation in configuration and phase space are related, but not identical.

4.3.5 Class C close-up

Recall that Class C rules are rules that are unconditionally stable for a certain range of values of p. Since the transition between conditional and unconditional stability occurs quickly for Class C rules, a close-up is considered between $p = 0.99$ and $p = 1$ to determine whether this transition really occurs abruptly or more gradually. The close-ups of the number of infinite MLEs for Class C rules can be found in Figure 4.16. This graph shows that the transition from conditional to unconditional stability happens gradually, even for those rules for which it seems a discrete transition if steps of 0.01 are considered (see Figure A.2) for the update probability.



(a) rule 150 (b) rule 23 (c) rule 130

(d) rule 152 (e) rule 110 (f) rule 30

Figure 4.15: Average and standard deviation of the normalised Lempel-Ziv complexity (ℓ_n) on the final time step of rules 150, 23, 130, 152, 110 and 30.

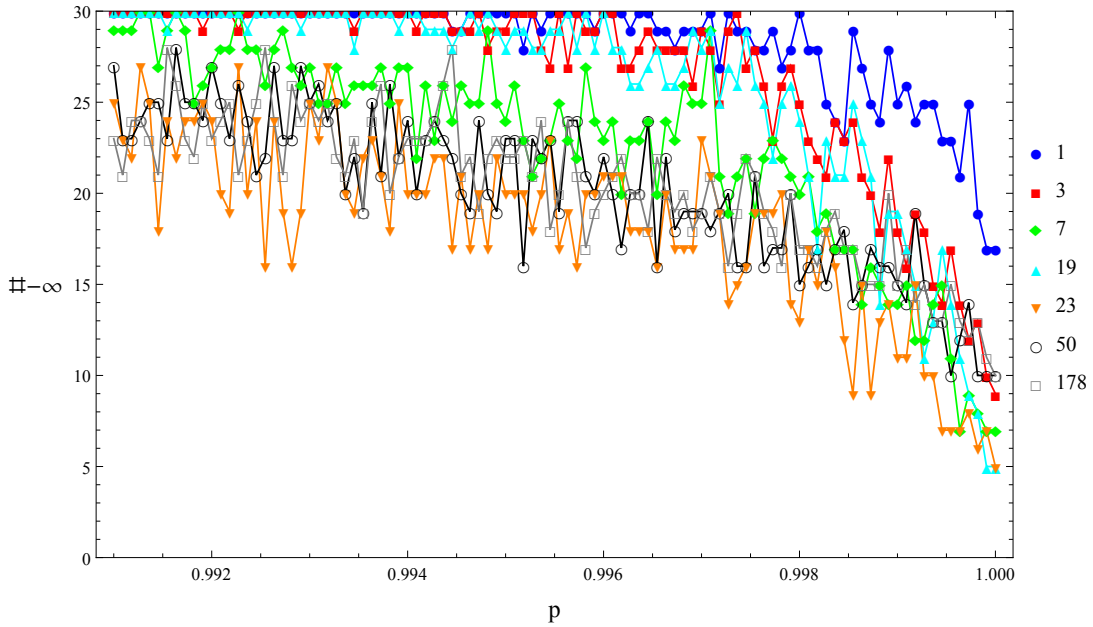


Figure 4.16: Close-up of the number of infinite MLEs for Class C rules for values of p between 0.99 and 1.

4.4 Two-dimensional cellular automata

The 2D SCAs that are examined in the framework of this thesis are the (2,7) totalistic SCAs (see Section 2.2.3) (Baetens and De Baets, 2010). This means that there are 256 rules to be examined.

4.4.1 Normalisation

For the normalisation of 2D (S)CAs the same equations as the ones used in the case of 1D (S)CAs are applicable, but the value of the average connectivity ($\overline{V}$) should be adjusted appropriately. For the tessellation used in this dissertation, $\overline{V}$ equals 6.97. The average MLE as a function of the input sensitivity and the update probability, as well as the theoretical upper bounds are given in Figure 4.17.

4.4.2 Behavioural classes

Upon classifying the rules in this family in a similar way as was done in Section 4.2.2 for the SECAs, a few issues arise. First of all, Class B, enclosing the unconditionally unstable SCAs, turns out to be empty. This might be due to the totalistic nature of the CAs in combination with their irregularity. The irregularity leads to the introduction of a threshold on the sum of states that is taken into account above which an increase in the sum of the states of the neighbouring cells has no further effect (see Section 2.2.3). For (2,7) totalistic CAs, this threshold is seven. Analysing the number of neighbours of the tessellation learns that around twelve percent of the cells have more than seven neighbours. Because Class B is empty, a little flexibility regarding unconditionally unstable behaviour is applied. More precisely, if 28 or more members of the ensemble have a positive MLE, for a certain value of p, the rule is considered to be unconditionally unstable, and similarly for unconditionally stable SCAs. By doing this, Class A (unconditionally stable) and Class B (unconditionally unstable) contain, respectively, twenty and nineteen rules. The remaining rules are divided into two groups, those that are for some values of p unconditionally stable and those that are not. The former can be divided into two distinct classes (C and D, see Figure 4.18) and one class that is an intermediary one between Class A and Class C or Class D. This last class is not named yet, since it will be merged with other classes. The second group is not split any further and will be referred to as Class F. Although there are some rules (for example, rules 65, 161 and 228) with distinct behaviour in Class F, they are not assigned to a separate class. Moreover, rules 23 and 95 are moved from Class F to Class D. Finally, the rules in the intermediary class located in the first group are merged with Class F. One could also argue that the intermediary class of the first group should be kept as a separate class. However, Class F contains several rules that exhibit behaviour related to the unnamed intermediary class, leading to difficulties in deciding which rule of Class F to consider as related with Class C or Class D. The graphs of the number of infinite MLEs, the averages of the finite nMLEs and the averages of the input sensitivity as a function of the update probability p can be found in Appendix A.2.

As a conclusion to this section it can be seen that Classes A, B and C for 2D SCAs are more or less similar to the corresponding 1D SECAs classes. The behaviour of these classes, however, is

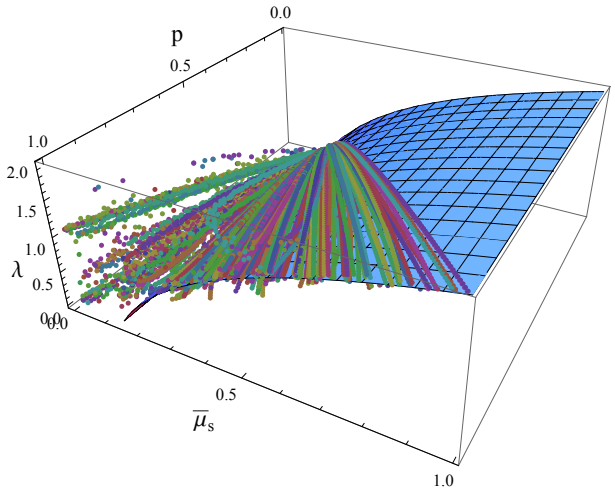


Figure 4.17: Plot of the average finite Lyapunov exponents (λ) and average input sensitivity ($\bar{\mu}$) of the 256 (2,7) totalistic SCAs for different probabilities (p), along with the upper bound

less pronounced for 2D SCAs. This is clear from the fact that a certain amount of flexibility was necessary to have Class B rules and from the fact that the range of values of p for which Class C rules become unconditionally unstable is not as clear-cut. Class F of 2D SCAs corresponds with Class F of 1D SECAs as well, but it contains some extra rules, being those that are for 1D SECAs classified into Classes D and E.

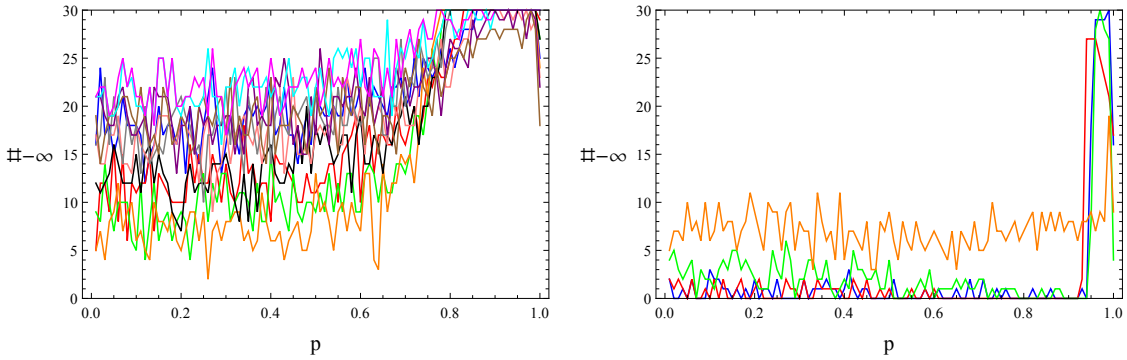
Table 4.2: Classification of the 256 2D (2,7) totalistic SCAs.

Class	Rule number
A	0, 16, 32, 64, 80, 96, 128, 144, 160, 192, 208, 224, 248-255
B	20, 21, 42, 43, 74, 84-86, 106, 107, 148, 149, 170, 171, 202, 203, 212-214
C	1, 3, 7, 15, 31, 62, 63, 124-127
D	11, 23, 47, 95
F	2, 4-6, 8-10, 12-14, 17-19, 22, 24-30, 33-41, 44-46, 48-61, 65-73, 75-79, 81-83, 87-94, 97-105, 108-123, 129-143, 145-147, 150-159, 161-169, 172-191, 193-201, 204-207, 209-211, 215-223, 225-247

4.5 Conclusion

Overall, it can be concluded that the update probability does affect the stability, determined by the Lyapunov exponent, of some CAs. However, there are different ways of interference between the update probability and stability. For some rules one of more bifurcations arise. These bifurcations could be called stochastic bifurcation, analogous to topological bifurcations when the effect of topology is investigated (see Section 2.3.3).

As indicated in the above section, some 2D CA classes are not found in 1DCAs and vice versa. This indicates that there is need for a unified classification of SCAs. Furthermore, Class F, which now contains the rules that could not be classified in any of the other classes, might also be divided into more classes. One possibility to do this is based on whether or not a rule displays



(a) Class C (b) Class D

Figure 4.18: Number of infinite MLEs as a function of the update probability p for Classes C and D of the 2D stochastic (2,7) totalistic CAs.

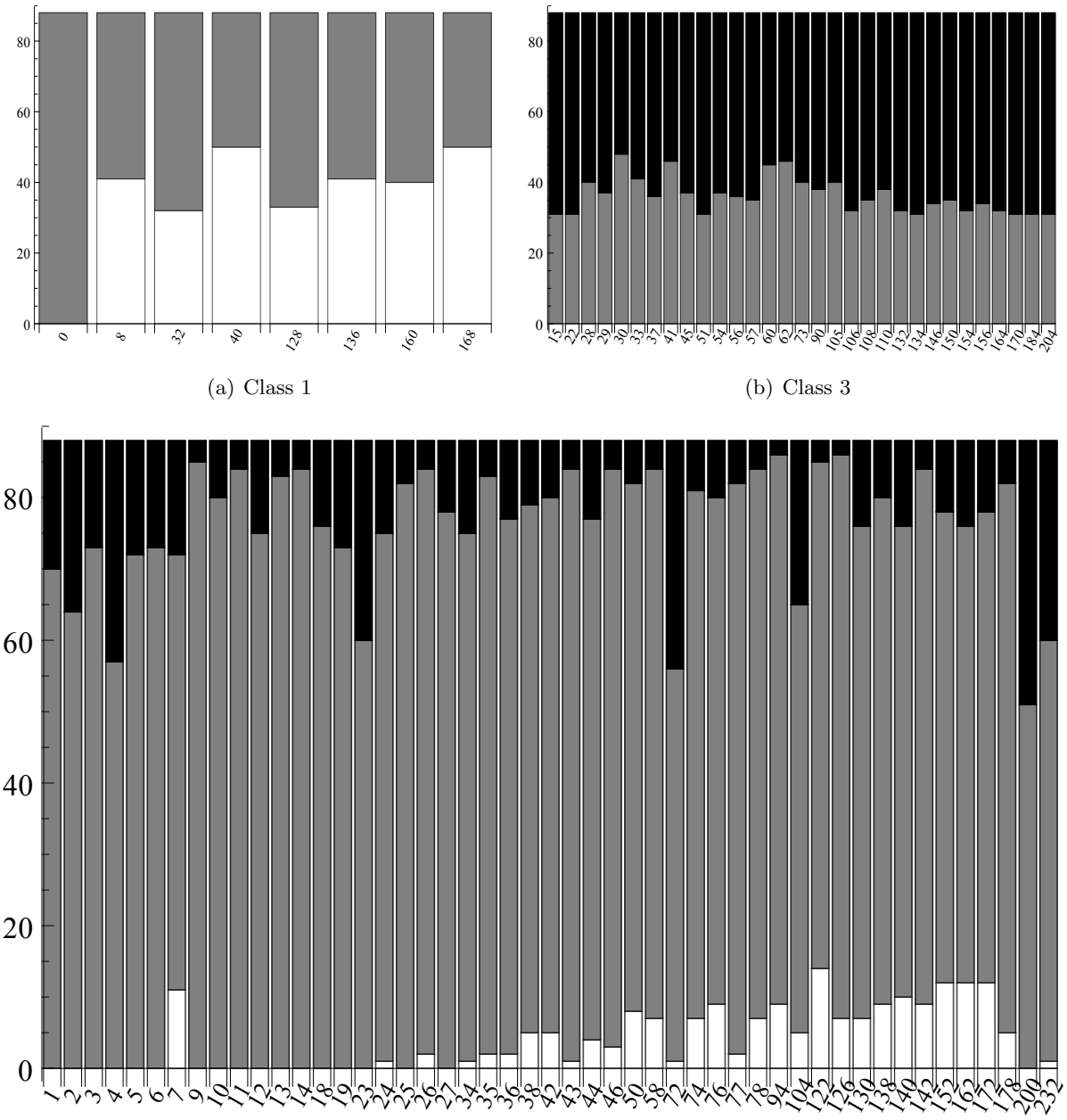
the typical behaviour as a function of the update probability. This typical behaviour consists of $\bar{\mu}_s$ having linear dependence on the update probability and the MLE being a horizontal line that bends down a little for small values of p.

5. Alternating cellular automata

For the sake of curiosity, another family of CAs will be examined briefly, namely the family of so-called alternating cellular automata (ACAs) (Buchholz et al., 2000). In this dissertation, ACAs are CAs that evolve according to two ECA rules, each of them applied at every other time step. The effect on the stability of combining a given rule with every other rule will be examined. In Figure 5.1 the bar charts summarize the change in stability. A change in stability is defined here as a transition between one of the behavioural three classes as proposed by Baetens and De Baets (2010) (see Section 2.3.2). Therefore, rules that remain Class 2 are considered as being equally stable, irrespective of the possible change in number of members in the ensemble that give rise to MLEs equal to minus infinity.

The results for the Class 1 (unconditionally superstable) rules are a bit surprising. It could have been expected that the stability of these rules does not change when combining them with other rules, since they lead to a uniform configuration, as such cancelling every defect. With respect to rule 0, for which the uniform configuration is reached after one time step, this is indeed the case. But other rules that normally need a few more time steps to reach a uniform configuration seem to leave, in some cases, enough time for the unstable rule to create so many defects that they cannot be cancelled by the stable rule. However, most Class 1 rules whose stability decreases move to Class 2. Only rule 40 combined with rule 29 and rule 168 combined with rules 57, 105 and 156 become unconditionally unstable (Class 3).

The results of Class 2 rules are also not as obvious as expected. The results clearly show that combining a Class 2 rule (conditionally (un)stable) with rules other than Class 1 rules can increase the stability. This is shown in more detail in Figure 5.2. This figure clearly proves that combining a conditionally (un)stable rule with an unconditionally unstable rule can lead to Class 1 (unconditionally stable) behaviour. The question now arises whether combining two unconditionally unstable rules could also lead to Class 1 behaviour. This turned out to be not the case.



(a) Class 1

(b) Class 3

(c) Class 2

Figure 5.1: Bar chart of the change in stability of Class 1, 2 and 3 ECA rules, representing the number of rules that lead to decreased (white), equal (gray) or increased (black) stability when combined with the rule at hand. On the horizontal axis the rule number of the rule at hand is given.

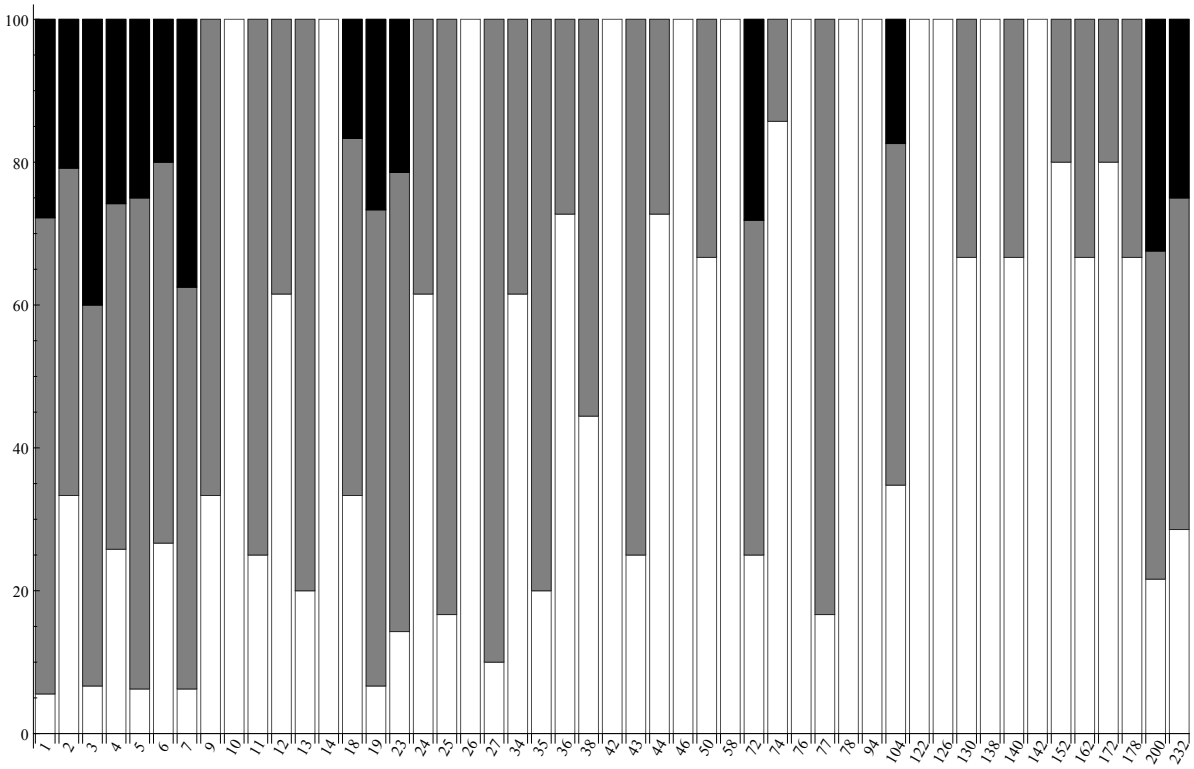


Figure 5.2: Distribution of the classes where rules that increase stability for class 2 rules come from. White (Class 1), gray (Class 2), black (Class 3).

6. Practical model

6.1 Rationale

Now that several theoretical SCAs have been examined, it is time to use the insights gained throughout Chapter 4 to unravel the dynamics of a SCA mimicking a real-world process. An attempt was made to find a model that is as alike the theoretical cases as possible. This means that the model should have the following characteristics:

1. it should have only two states,
2. be stochastic,
3. be two-dimensional,
4. and be constituted of a CA senso stricto.

Especially the first and the last requirement are hard to fulfil, since two-state CAs either model only one substance/organism or empty spaces do not exist. These simple two-state CAs are often of a mainly theoretical nature or require extra information, mostly in the form of a coupled-map lattice. Aside from these four characteristics, two additional properties are preferred, being: that the CA should be homogeneous and the model should be governed by only one parameter. These two additional properties are mainly practical, since a heterogeneous CA would significantly increase the complexity and, therefore, the required computing time, whereas multiple parameters would require either multi-dimensional graphs of the MLE as a function of the parameters or assessing the influence of one parameter on the stability while keeping the other parameters fixed.

Most of the models given in Section 2.2.6 fail to meet the first requirement. The paper in which a CA is used to represent interaction between amino acids (Diao et al., 2008) does not meet the third requirement. The only model that fulfils all four requirements is the model for forest spatial dynamics (Kubo, 1996). Yet, this model has multiple update probabilities, which would significantly complicate the stability analysis.

6.2 Bacterial competition with uniform update probability

Since no suitable models were found in literature, a new model was constructed, based on the model of Reichenbach et al. (2008). At first, a model for bacterial growth was conceived, in which the two states were: a living cell, represented by 1 and an empty space or a dead cell, represented

by 0 (see Appendix B.1). However, this model was quickly put aside and a competition model, for competition between two species, was developed. Since no cyclic domination can be used when dealing with only two species, another competition mechanism had to be used. Also, no empty spaces were allowed, to limit the number of possible states to two.

6.2.1 Model kinetics

The rule that was used to model the competition was the majority rule, meaning that competition leads to victory of the species that has the most congeners in the considered neighbourhood. On a regular square grid the majority rule is the (32, 5) totalistic rule for a Moore neighbourhood and the (56, 5) totalistic rule for a von Neumann neighbourhood. A regular grid is used because otherwise the majority rule would require a heterogeneous CA. The most straightforward implementation of the rate of competition is by using a uniform update probability. This means that, at every consecutive time step, every cell has a certain probability of being updated. In this way this model is very similar to the theoretical models in Section 4.4.

6.2.2 Results

The stability was tested for an ensemble of ten random ICs and with each a random IP. The averages of the finite Lyapunov exponents, as well as the standard deviation, are shown in Figure 6.1(a) and Figure 6.1(c), for a Moore and von Neumann neighbourhood, respectively. The corresponding numbers of infinite Lyapunov exponents are given in Figure 6.1(b) and Figure 6.1(d).

These graphs indicate that the competition model with a homogeneous update probability can be classified as a Class F SCA. This was to be expected, since the one-dimensional form of the majority rule, being rule 232, also belongs to Class F (see Table 4.1). However, when a von Neumann neighbourhood is used, the degree of instability appears to be more pronounced than when a Moore neighbourhood is used. This is to be expected, since the size of a von Neumann neighbourhood is markedly smaller than that of a Moore neighbourhood, implying that the effect of a single defect is more pronounced. The dependence of the stability on the topology was already mentioned in Section 2.3.3 and is observed here in practise. This means that although this thesis is about assessing the effect of the update probability on the stability, other influential factors should not be discarded in a thorough stability analysis.

6.3 Bacterial competition with varying update probability

In the previous section the update probability was assumed to be the same for every cell. It would, however, make sense that competition would happen faster as one species is increasingly outnumbered by the other in its neighbourhood. The dependence of processes on the local population density is a well-established concept in population ecology (Rockwood, 2009). A similar dependence of the update probability on the neighbourhood is used in the gap dynamics model of Kubo (1996). In the field of microbiology, a well-known example of density dependence of physiological processes is quorum sensing (Swift et al., 2001).

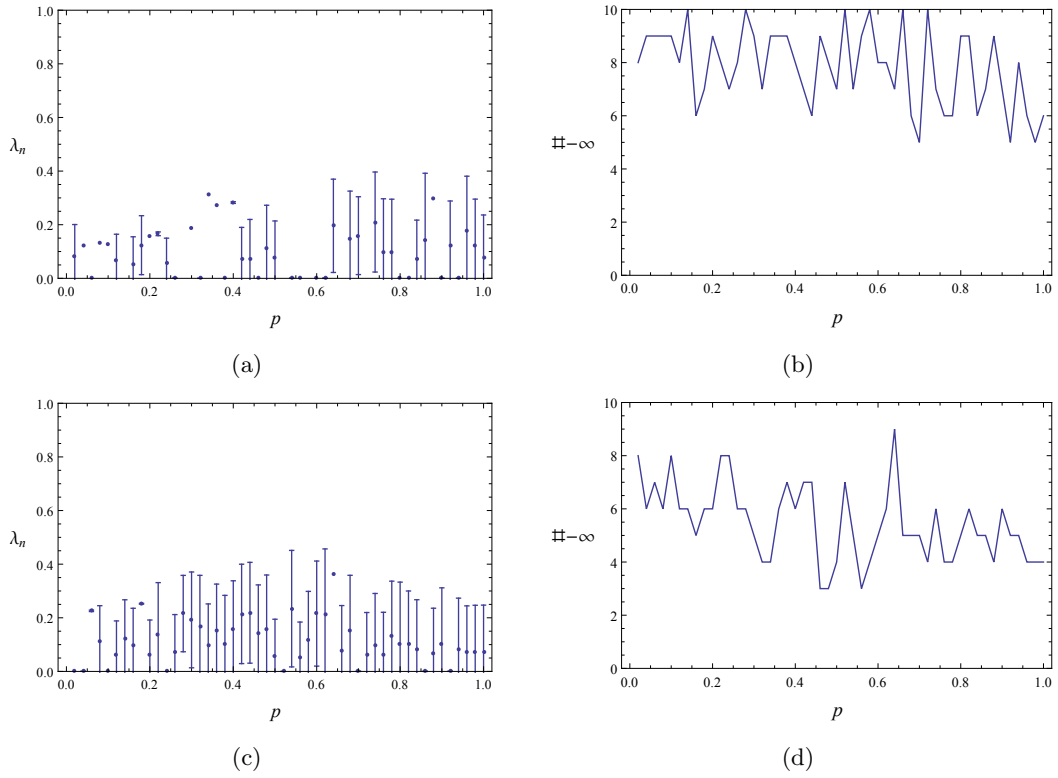


Figure 6.1: Average normalised Lyapunov exponents and number of infinite Lyapunov exponents for the competition model with uniform update probability for a Moore neighbourhood (a, b) and for a von Neumann neighbourhood (c, d).

6.3.1 Model kinetics

The model is similar to the previous model, except for the fact that that the update probability is no longer homogeneous. Instead, the update probability depends on the minimal update probability, denoted as p_m, which is the update probability when neither of the species in a cell's surroundings has the majority. The surroundings of a cell is defined here as the neighbourhood without the cell itself and the number of organisms of species A in the surroundings is denoted as z. The maximum rate of competition is assumed to correspond with an update at every time step (update probability $p = 1$), and occurs when the surroundings only contains cells of one species ($z = 0$ or $z = |N| - 1$). The only parameter in this model is the minimal update probability p_m. The dependence of the update probability on the surroundings is illustrated in Figure 6.2.

6.3.2 Results

This alteration of the place in the model where stochasticity is included has two consequences. Firstly, a minimal update probability equal to zero is not a trivial case, and secondly, normalisation is not possible, since the update probability is spatially heterogeneous. The results (see Figure 6.3) appear to be consistent with the results of the model with a homogeneous update probability. Besides the effect of the neighbourhood on the stability, also an effect of the chosen combination of IC and IP seems to have an effect. This can be seen by looking at the number of infinite MLEs across all minimal update probabilities of each member of the ensemble, instead

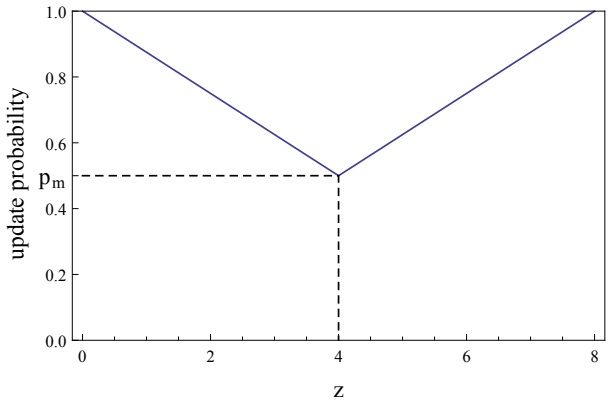


Figure 6.2: Update probability as a function of the number of species A in a cell's surroundings (z).

of looking at the number of infinite MLEs across all members of the ensemble for each minimal update probability. For the Moore neighbourhood, the number of infinite MLEs, across the different values of the minimal update probability p_m, varies between 24 and 50 out of the 51 considered minimal update probabilities. For the von Neumann neighbourhood, it varies between 7 and 46. This indicates that for the SCA at hand, there are regions with more stability and regions with less stability. This is also reflected in the configuration at equilibrium (see Figure B.2).

6.4 Continuous model

Reichenbach et al. (2008) developed a method for transforming a SCA into a set of PDEs. This method will be used here to get the mean-field counterpart of the model with uniform update probability. Since this model only has two states, one PDE suffices. If a von Neumann neighbourhood is considered and the density of species A and B at time t and position (x, y) is represented by $X(x, y, t)$ and $Y(x, y, t)$, respectively, the change of X can be described by the following equation:

$$\begin{aligned}\frac{\partial X}{\partial t} = -p\,X(x,y,t)\Big(&Y(x+\Delta x,y,t)\,Y(x-\Delta x,y,t)\,Y(x,y+\Delta y,t)\\ &+Y(x+\Delta x,y,t)\,Y(x-\Delta x,y,t)\,Y(x,y-\Delta y,t)\\ &+Y(x-\Delta x,y,t)\,Y(x,y+\Delta y,t)\,Y(x,y-\Delta y,t)\\ &+Y(x+\Delta x,y,t)\,Y(x,y+\Delta y,t)\,Y(x,y-\Delta y,t)\Big)\\ +p\,Y(x,y,t)\Big(&X(x+\Delta x,y,t)\,X(x-\Delta x,y,t)\,X(x,y+\Delta y,t)\\ &+X(x+\Delta x,y,t)\,X(x-\Delta x,y,t)\,X(x,y-\Delta y,t)\\ &+X(x-\Delta x,y,t)\,X(x,y+\Delta y,t)\,X(x,y-\Delta y,t)\\ &+X(x+\Delta x,y,t)\,X(x,y+\Delta y,t)\,X(x,y-\Delta y,t)\Big).\end{aligned} \tag{6.1}$$

This equation was found by following this reasoning. If the central cell is occupied by species A, it will switch to species B if three out of four cells of the surroundings are of species B, leading to a negative contribution. The speed of this transition is p. The positive contribution is found when the central cell is of species B and three or four out of four cells in the surroundings

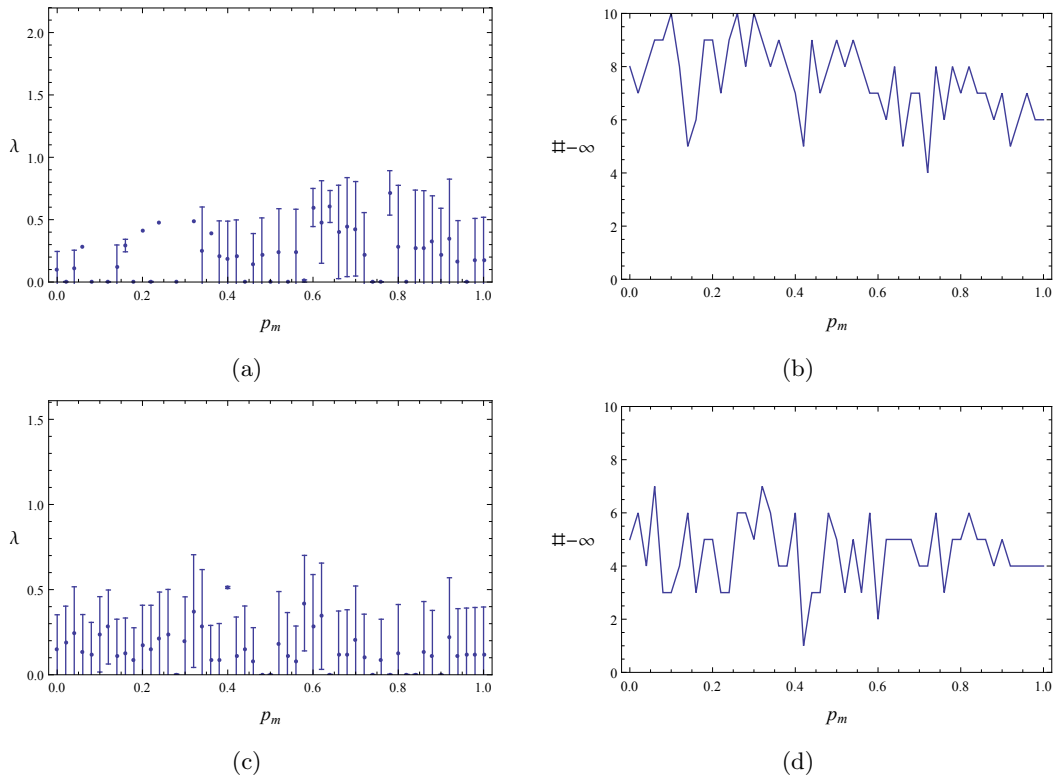


Figure 6.3: Average normalised Lyapunov exponents and number of infinite Lyapunov exponents for the competition model with varying update probability for a Moore neighbourhood (a, b) and for a von Neumann neighbourhood (c, d).

are of species A. In this equation all terms that are evaluated at a position adjacent to the central position are replaced by their Taylor series, dropping third and higher order terms. Subsequently, Δx and Δy are replaced by Δr. Also, $X(x, y, t)$ and $Y(x, y, t)$ are replaced by X and Y, respectively, for the sake of brevity. The following equation is now obtained for the change of X:

$$\begin{aligned}\frac{\partial X}{\partial t} = {} & \frac{p}{4} Y \left[16X^3 + 12\Delta r^2 X^2 \nabla^2 X \right. \\ & \left. + \Delta r^4 \left(-4 \left(\frac{\partial X}{\partial y} \right)^2 \frac{\partial^2 X}{\partial x^2} + \frac{\partial^2 X}{\partial y^2} \left(-4 \left(\frac{\partial X}{\partial x} \right)^2 + \Delta r^2 \frac{\partial^2 X}{\partial x^2} \nabla^2 X \right) \right) \right] \\ & - \frac{p}{4} X \left[16Y^3 + 12\Delta r^2 Y^2 \nabla^2 Y \right. \\ & \left. + \Delta r^4 \left(-4 \left(\frac{\partial Y}{\partial y} \right)^2 \frac{\partial^2 Y}{\partial x^2} + \frac{\partial^2 Y}{\partial y^2} \left(-4 \left(\frac{\partial Y}{\partial x} \right)^2 + \Delta r^2 \frac{\partial^2 Y}{\partial x^2} \nabla^2 Y \right) \right) \right] \\ & + \frac{p}{2} \Delta r^2 XY \left[-4 \left(\frac{\partial X}{\partial y} \right)^2 + 4 \left(\frac{\partial Y}{\partial y} \right)^2 - 4 \left(\frac{\partial X}{\partial x} \right)^2 + 4 \left(\frac{\partial Y}{\partial x} \right)^2 \right. \\ & \left. + \Delta r^2 \left(\left(\frac{\partial^2 X}{\partial y^2} \right)^2 - \left(\frac{\partial^2 Y}{\partial y^2} \right)^2 + 4 \frac{\partial^2 X}{\partial y^2} \frac{\partial^2 X}{\partial x^2} - 4 \frac{\partial^2 Y}{\partial x^2} \frac{\partial^2 Y}{\partial y^2} + \left(\frac{\partial^2 X}{\partial x^2} \right)^2 - \left(\frac{\partial^2 Y}{\partial x^2} \right)^2 \right) \right]\end{aligned} \tag{6.2}$$

Using the fact that $\Delta r^6 \approx 0$ and $\Delta r^4 \approx 0$, the following simplified equation is found:

$$\frac{\partial X}{\partial t} = pXY\left[4X^2 - 4Y^2 + 2\Delta r^2\left(-\left(\frac{\partial X}{\partial y}\right)^2 + \left(\frac{\partial Y}{\partial y}\right)^2 - \left(\frac{\partial X}{\partial x}\right)^2 + \left(\frac{\partial Y}{\partial x}\right)^2\right)\right. \\ \left. +3\Delta r^2 X\,\nabla^2 X - 3\Delta r^2 Y\,\nabla^2 Y\right]. \tag{6.3}$$

Since no empty spaces are allowed, it may be said that $X = 1 - Y$. Inserting this into Eq. (6.3) leads to:

$$\frac{\partial X}{\partial t} = p(1-X)X(2x-1)\left[4 + 3\Delta r^2\,\nabla^2 X\right]. \tag{6.4}$$

For the numerical solution of this equation the Mathematica function NDSolve (Wolfram Research inc., 2008) was used. The method of lines was chosen to solve the PDE and the spatial discretisation was done by using the tensor product grid method with a minimum of 101 points. The derivatives were approximated pseudospectrally. Dirichlet boundary conditions were imposed, consisting of the initial values at the boundaries. The initial condition that was used for solving the PDE can be found in Figure 6.4 and is described by

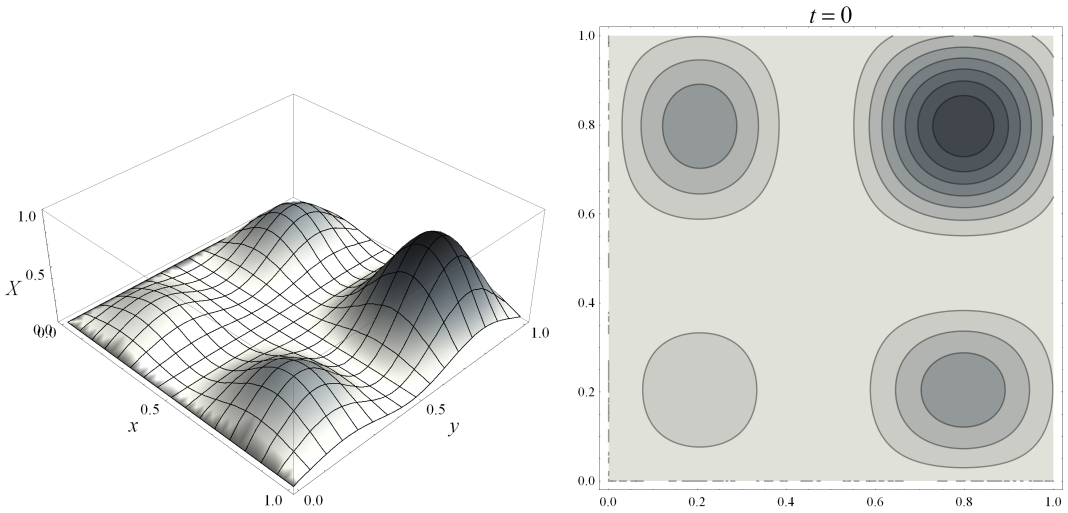
$$X(x, y, 0) = \frac{\sqrt{xy}}{4}\left(\sin(10x) + 1\right)\left(\sin(10y) + 1\right).$$

However, since for the Taylor series, $\Delta r \to 0$, Eq. (6.3) can even further be simplified to:

$$\frac{\partial X}{\partial t} = 4p(1-X)X(2X-1). \tag{6.5}$$

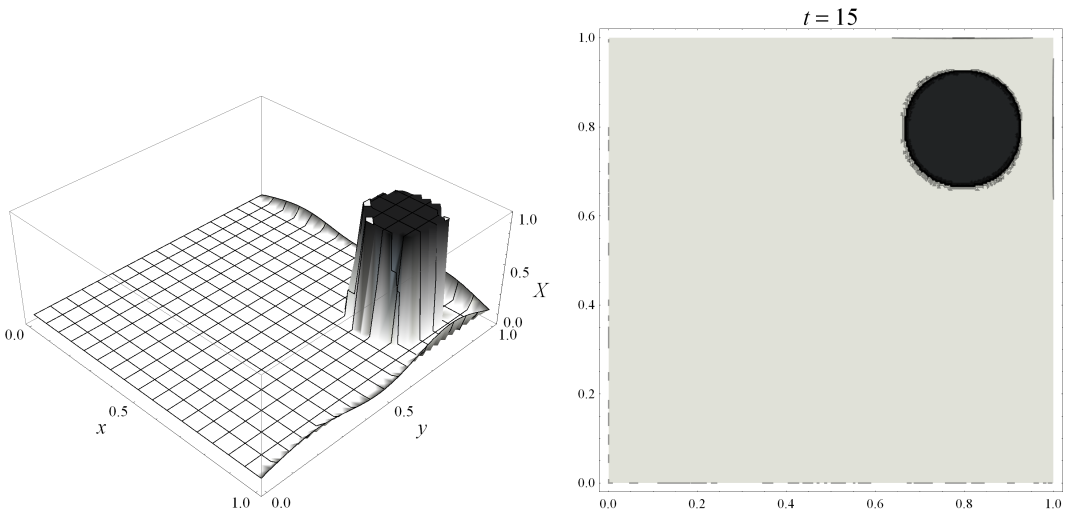
The solution of this equation was not much different from the one of Eq. (6.3) and due to its less complicated structure the minimum number of points considered by the tensor grid discretisation could be raised to 401, leading to the solution as shown in Figure 6.5. This solution was found for $p = 0.9$ and $t = 15$, starting from the initial condition given in Figure 6.4. At the transitions between density 1 to 0, discretisation errors can be observed. Since Eq. (6.5) has no spatial derivatives, it can be solved for a single point in space. Figure 6.6 shows the solution for an initial condition of $X = 0.4$ for several values of the update probability. This clearly illustrates the influence of the update probability on the rate of change in density. The steady state solutions of Eq. (6.5) can also be easily found by setting the time derivative to zero: $X = 1$, $X = 0$ and $X = 0.5$.

The solution that was found lives up to the expectations, because at the positions where species A has the majority the density of species A reaches the maximum density and in all other cases it goes to zero. The problem is that there are no spatial derivatives present in Eq. (6.5), which means that there is no possibility of colony formation if the initial condition would be composed of very small, very close colonies. This grouping into larger colonies is observed in the SCA model. Therefore, like in the paper of Van der Weeën et al. (2011), it might be necessary to include a diffusion/motility term, even though this is not explicitly included in the SCA model, because motility requires empty spaces to go to.



(a) (b)

Figure 6.4: 3D-plot (a) and contour plot (b) of the initial condition that was used to solve the PDE representing the continuous counterpart of the majority rule in the case of a von Neumann neighbourhood.



(a) (b)

Figure 6.5: 3D-plot (a) and contour plot (b) of the solution of the most simplified continuous approximation of the competition model for $t = 15$, for $p = 0.9$ and for a von Neumann neighbourhood.

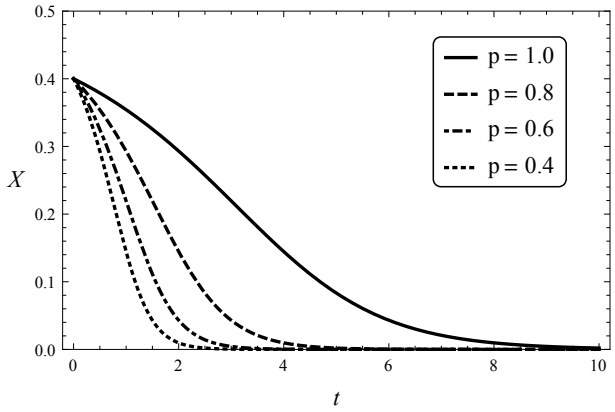


Figure 6.6: Plot of the solution of the most simplified continuous approximation of the competition model for several update probabilities.

7. Conclusions

Generally speaking, it may be concluded that based on the stochasticity, different stability regimes can be distinguished. A few behavioural classes seem to emerge only in either one-dimensional or in two-dimensional SCAs. This indicates the need for a unified classification. Furthermore, it might be appropriate to discard the intermediary Class E of the 1D SCAs and classify the rules it contains in either Class C or Class D. For Class F, which is now considered as some sort of rest class, a division could be performed. Rules which, to a certain extent, comply with the general behaviour of SCAs, described in Section 4.5, can be put in one class, denoted here as Class F', and the other rules can be put in an other class, denoted here as Class F". One could then argue that Class A and Class B also comply with this general behaviour and that the only difference lies in the number of members of the ensemble that lead to infinite MLEs. It follows then that these two classes combined with Class F' represent the expected behaviours of SCA and that Class C, Class D and Class F" represent eccentric behaviours.

It can be said that most SCAs exhibit, when considering the MLE, the same behaviour across all update probabilities. However, some SCAs have one or more bifurcation points. This indicates that for some rules the influence of stochasticity is not limited to the rate at which the CA evolves, but that it can also influence the Lyapunovian stability. The Lempel-Ziv complexity indicated that, besides effects on the stability, also changes in the emerging configurations occur. This was already demonstrated by Bouré et al. (2012) by observing the density, which holds slightly less information than the Lempel-Ziv complexity. However, the latter only applies to 1D CAs and, therefore, nothing can be said about configurational changes in 2D CAs.

The results of the competition model indicate that besides bifurcations that can be traced back to the stochasticity, topological bifurcations also have to be accounted for in a stability analysis.

8. Future developments

The classification set forth in this thesis was mainly based on the Lyapunov exponents. It might be useful to use the input sensitivity $\bar{\mu}_s$ as the basis for classification. The input sensitivity is related to the MLE, but there are important differences. For instance, the input sensitivity is always a value between zero and one and is therefore never $-\infty$, which makes the processing of data easier. The input sensitivity is also linked more directly to the configuration than the MLE.

On a related topic, it might be useful to express $\bar{\mu}_s$ differently. It is now based on the Boolean derivative of every cell, leading to $1/\overline{V}$ for an update probability $p = 0$. If for the calculation of $\bar{\mu}_s$ only the Boolean derivatives of the updated cells would be used, the theoretical curve of the input sensitivity as a function of the update probability would be a horizontal line. This way shifts in average input sensitivity and, therefore, configuration can be noticed easier. An extra advantage is that classification could be done based on the size of the area included between the line of the input sensitivity as function of the update probability and the line representing $\bar{\mu}_s = 1/\overline{V}$.

Since the value obtained for the MLE does not only depend on the update probability, but also on the chosen IC and IP, it might be useful to assess the stability of every cell for a certain IC. The stability for a certain cell could be determined by calculating the MLE a certain number of times. The extent of intra- or inter-cellular variation of the MLE is an indication of the relative influence of the stochasticity and the topology on the stability, respectively. This could be repeated for several IC.

Bibliography

D.C. Armstead and M.A. Karls. Does the wave equation really work? *Problems, Resources, and Issues in Mathematics Undergraduate Studies*, 16:162–177, 2006.

J.M. Baetens. *Discrete spatio-temporal modelling paradigms for environmental processes: SWOT analysis, application portfolio and Lyapunovian stability analysis*. PhD thesis, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Ghent, Belgium, 2012.

J.M. Baetens and B. De Baets. Phenomenological study of irregular cellular automata based on Lyapunov exponents and Jacobians. *Chaos*, 20:033112, 2010.

J.M. Baetens and B. De Baets. Topology-induced phase transitions in totalistic cellular automata. *Physica D: Nonlinear Phenomena*, 249:16–24, 2013a.

J.M. Baetens and B. De Baets. A Lyapunov view on the stability of two-state cellular automata. In H. Zenil, editor, *Irreducibility and Computational Equivalence*, pages 25–33. Springer, 2013b.

J.M. Baetens, P. Van der Weeën, and B. De Baets. Effect of asynchronous updating on the stability of cellular automata. *Chaos Solitons & Fractals*, 45:383–394, 2012.

J.M. Baetens, K. De Loof, and B. De Baets. Influence of the topology of a cellular automaton on its dynamical properties. *Communications in Nonlinear Science and Numerical Simulation*, 18:651–668, 2013.

F. Bagnoli, R. Rechtman, and S. Ruffo. Damage spreading and Lyapunov exponents in cellular automata. *Physics Letters A*, 172:34–38, 1992.

S. Bandini, A. Bonomi, and G. Vizzari. What do we mean by asynchronous CA? A reflection on types and effects of asynchronicity. In S. Bandini, S. Manzoni, H. Umeo, and G. Vizzari, editors, *Proceedings of the 9th international conference on cellular automata for research and industry*, pages 385–394, 2010.

L. Berec. Techniques of spatially explicit individual-based models: construction, simulation, and mean-field analysis. *Ecological Modelling*, 150:55–81, 2002.

O. Bouré, V. Chevrier, and N. Fatès. Probing robustness of cellular automata through variations of asynchronous updating. *Natural Computing*, 20:1–20, 2012.

T. Buchholz, A. Klein, and M. Kutrib. Real-time language recognition by alternating cellular automata. In *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*, pages 213–225. Springer Berlin Heidelberg, 2000.

E. Chi Fru, I.D. Ofiţeru, V. Lavric, and D.W. Graham. Non-linear population dynamics in chemostats associated with live-dead cell cycling in *Escherichia coli* strain k12-mg1655. *Applied Microbiology and Biotechnology*, 89:791–798, 2011.

Y. Diao, D. Ma, Z. Wen, J. Yin, J. Xiang, and M. Li. Using pseudo amino acid composition to predict transmembrane regions in protein: cellular automata and Lempel-Ziv complexity. *Amino Acids*, 34:111–117, 2008.

J.-P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, 57:617–656, 1985.

G.B. Ermentrout and L. Edelstein-Keshet. Cellular automata approaches to biological modeling. *Journal of Theoretical Biology*, 160:97–133, 1993.

K. Fehlhaber and G. Krüger. The study of *Salmonella enteritidis* growth kinetics using rapid automated bacterial impedance technique. *Journal of Applied Microbiology*, 84:945–949, 1998.

V.K. Juneja, B.S. Eblen, and G.M. Ransom. Thermal inactivation of *Salmonella spp.* in chicken broth, beef, pork, turkey, and chicken: Determination of d- and z-values. *Journal of Food Science*, 66:146–152, 2001.

V.J. Katz. *A History of Mathematics: An Introduction.* Addison Wesley Publishing Company Incorporated, 2009.

T Kubo. Forest spatial dynamics with gap expansion: Total gap area and gap size distribution. *Journal of Theoretical Biology*, 180:229–246, 1996.

V. Lavric and D.W. Graham. Birth, growth and death as structuring operators in bacterial population dynamics. *Journal of Theoretical Biology*, 264:45–54, 2010.

A. Lempel and J. Ziv. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, 22:75–81, 1976.

T. Nyström. Aging in bacteria. *Current Opinion in Microbiology*, 5:596 – 601, 2002.

V.I. Oseledec. A multiplicative ergodic theorem. Lyapunov characteristic numbers for dynamical systems. *Transactions of the Moscow Mathematical Society*, 19:197–231, 1968.

T. Reichenbach, M. Mobilia, and E. Frey. Mobility promotes and jeopardizes biodiversity in rock-paper-scissors games. *Nature*, 448:1046–1049, 2007.

T. Reichenbach, M. Mobilia, and E. Frey. Self-organization of mobile populations in cyclic competition. *Journal of Theoretical Biology*, 254:368–83, 2008.

L.L. Rockwood. *Introduction to Population Ecology.* Wiley, 2009.

M.A. Shereshevsky. Lyapunov exponents for one-dimensional cellular automata. *Journal of Nonlinear Science*, 2:1–8, 1992.

S. Swift, J. Allan Downie, N.A. Whitehead, A.M.L. Barnard, G.P.C. Salmond, and P. Williams. Quorum sensing as a population-density-dependent determinant of bacterial physiology. *Advances in Microbial Physiology*, 45:199–270, 2001.

T. Toffoli. Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D: Nonlinear Phenomena*, 10:117–127, 1984.

P. Van der Weeën, J.M. Baetens, and B. De Baets. Design and parameterization of a stochastic cellular automaton describing a chemical reaction. *Journal of Computational Chemistry*, 32: 1952–1961, 2011.

P. Van der Weeën, N. De Clercq, J.M. Baetens, C. Delbaere, K. Dewettinck, and B. De Baets. A discrete stochastic model for oil migration in chocolate-coated confectionery. *Journal of Food Engineering*, 119:602–610, 2013.

G.Y. Vichniac. Boolean derivatives on cellular automata. *Physica D*, 45:63–74, 1990.

J.W.T. Wimpenny and R. Colasanti. A unifying hypothesis for the structure of microbial biofilms based on cellular automaton models. *FEMS Microbiology Ecology*, 22:1–16, 1997.

S. Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

Mathematics and Algorithms. Wolfram Research inc. and Wolfram Media inc., 2008.