Definition and Application of a Five-Parameter Characterization of One-Dimensional Cellular Automata Rule Space

Gina M. B. Oliveira
Pedro P. B. de Oliveira
Universidade Presbiteriana Mackenzie
Rua da Consolação 896, 6° andar
Consolação
01302-907 São Paulo, Brazil
gina@mackenzie.com.br
pedrob@mackenzie.com.br

Nizam Omar
Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gomes 50,
Vila das Acácias
12228-901 São José dos Campos, Brazil
omar@comp.ita.cta.br

Abstract Cellular automata (CA) are important as prototypical, spatially extended, discrete dynamical systems. Because the problem of forecasting dynamic behavior of CA is undecidable, various parameter-based approximations have been developed to address the problem. Out of the analysis of the most important parameters available to this end we proposed some guidelines that should be followed when defining a parameter of that kind. Based upon the guidelines, new parameters were proposed and a set of five parameters was selected; two of them were drawn from the literature and three are new ones, defined here. This article presents all of them and makes their qualities evident. Then, two results are described, related to the use of the parameter set in the Elementary Rule Space: a phase transition diagram, and some general heuristics for forecasting the dynamics of one-dimensional CA. Finally, as an example of the application of the selected parameters in high cardinality spaces, results are presented from experiments involving the evolution of radius-3 CA in the Density Classification Task, and radius-2 CA in the Synchronization Task.

Keywords
cellular automata, elementary rule space, dynamic behavior, static parameters, density classification task, synchronization task

1 Introduction

Cellular automata (CA) are prototypical, spatially-extended, discrete dynamical systems, formed by a great number of identical and simple components, with local connectivity and processing. In spite of the simplicity of their basic components, CA exhibit a variety of dynamic behaviors. This is a common characteristic of complex systems, of which CA are genuine examples. Several researchers have studied CA dynamics as a way to understand the mechanisms involved in complex systems dynamics. However, it has already been proved that the decision problem associated with forecasting CA dynamics is undecidable [6].

The direct way to analyze the dynamics of a cellular automaton is to observe its behavior through the spatial-temporal patterns it generates out of several random initial conditions, and then to use statistical metrics to quantify the observed behavior [16]. Such an approach is adequate for characterizing the various dynamic regimes of the automaton, but does not help forecast its behavior.

Another approach is the static analysis of the automaton's transition table. Various analytical procedures have been developed, which allow for various partial aspects of the dynamics to be precisely forecast, but not the full dynamic regime; in addition,

the applicability of these methods is still very constrained, not yet constituting robust methods [10]. Still, in tune with the static analysis of the automaton's transition table, various studies on CA dynamics have been carried out, based upon parameterizations of their rule space, and using these parameters as indicators of the dynamic features at study [3, 13–16]. Although less precise than the analytical methods, they have shown much more robustness and wider applicability, particularly in the issue of helping forecast the dynamic behavior.

The present work stresses precisely the latter approach. Our work began with studies of the dynamic behavior of elementary CA, because they are the simplest nontrivial CA, and, because the cardinality of the elementary space is very small, it can be subjected to exhaustive analysis from the perspective of the parameter values. As background for the core of the article, Section 2 provides a brief review of important pieces of research in the analysis of CA dynamics.

From the analysis of the strengths and weaknesses of several parameters—many defined by us, as well as those already published—we conceived guidelines that should be followed in the establishment of a forecast parameter, and which were used to support the critical analysis of the main published parameters in [23]. The guidelines and the main conclusions of the critical analysis of the parameters are mentioned in Section 3.

Supported by these guidelines, we then selected a set of five forecast parameters with distinct and complementary characteristics, two out of the literature, and three new ones. Section 4 is devoted to the parameter selection; parameters are presented and their main characteristics and qualities are individually pointed out. The definition of each parameter in the elementary space and in their generalized expression for one-dimensional binary CA of any radius is also presented.

In Section 5, two results of using the selected set in the Elementary Space are discussed: the derivation of a neat phase transition diagram from the space, and the establishment of forecast heuristics of the CA behavior.

Section 6 is devoted to presenting some results of the application of the selected set in larger rule spaces than the elementary space. First, some experiments are described, which gave us evidence that the selected parameters may be effectively used in larger spaces. Then, our recent work in the evolution of computational tasks in CA guided by the selected forecast parameters is discussed; results are shown in the evolution of radius-3 CA in the Density Classification Task [24], and radius-2 CA in the Synchronization Task [7]. Finally, in Section 7 some concluding remarks are made, in particular on the relation of this work to the field of artificial life.

2 Cellular Automata

For what follows, the notation adopted is drawn from [17]. Basically, a cellular automaton consists of two parts: the cellular space and the transition rule. Cellular space is a regular lattice of N cells, each one with an identical pattern of local connections to other cells, and subject to some boundary conditions. The set of states in a cell is denoted by Σ and the number of states in the set by k. Each cell is referred to by an index i, and its state at time t by S_i^t, where $S_i^t \in \Sigma$. State S_i^t of cell i, together with the states of the cells connected to cell i, is named the neighborhood η_i^t of cell i.

The transition rule, represented by $\Phi(\eta_i^t)$, yields the next state for each cell i, as a function of η_i^t. At each time step, all cells synchronously update their states according to $\Phi(\eta_i^t)$.

In computational terms, a cellular automaton is, therefore, an array of finite automata, where the state of each automaton depends on the state of its neighbors.

Table 1. Equivalent rules classes of the elementary space.

Class	Other Rules	Dynamics	Class	Other Rules	Dynamics	Class	Other Rules	Dynamics
0	255	Null	**35**	49,59,115	Two-Cycle	**108**	201	Two-Cycle
1	127	Two-Cycle	**36**	219	Fixed-Point	**110**	124,137,193	Complex
2	16,191,247	Fixed-Point	**37**	91	Two-Cycle	**122**	161	Chaotic
3	17,63,119	Two-Cycle	**38**	52,155,211	Two-Cycle	**126**	129	Chaotic
4	223	Fixed-Point	**40**	96,235,249	Null	**128**	254	Null
5	95	Two-Cycle	**41**	97,107,121	Periodic	**130**	144,190,246	Fixed-Point
6	20,159,215	Two-Cycle	**42**	112,171,241	Fixed-Point	**132**	222	Fixed-Point
7	21,31,87	Two-Cycle	**43**	113	Two-Cycle	**134**	148,158,214	Two-Cycle
8	64,239,253	Null	**44**	100,203,217	Fixed-Point	**136**	192,238,252	Null
9	65,111,125	Two-Cycle	**45**	75,89,101	Chaotic	**138**	174,208,244	Fixed-Point
10	80,175,245	Fixed-Point	**46**	116,139,209	Fixed-Point	**140**	196,206,220	Fixed-Point
11	47,81,117	Two-Cycle	**50**	179	Two-Cycle	**142**	212	Two-Cycle
12	68,207,221	Fixed-Point	**51**		Two-Cycle	**146**	182	Chaotic
13	69,79,93	Fixed-Point	**54**	147	Complex	**150**		Chaotic
14	84,143,213	Two-Cycle	**56**	98,185,227	Fixed-Point	**152**	188,194,230	Fixed-Point
15	85	Two-Cycle	**57**	99	Fixed-Point	**154**	166,180,210	Periodic
18	183	Chaotic	**58**	114,163,177	Fixed-Point	**156**	198	Two-Cycle
19	55	Two-Cycle	**60**	102,153,195	Chaotic	**160**	250	Null
22	151	Chaotic	**62**	118,131,145	Periodic	**162**	176,186,242	Fixed-Point
23		Two-Cycle	**72**	237	Fixed-Point	**164**	218	Fixed-Point
24	66,189,231	Fixed-Point	**73**	109	Chaotic	**168**	224,234,248	Null
25	61,67,103	Two-Cycle	**74**	88,173,229	Two-Cycle	**170**	240	Fixed-Point
26	82,167,181	Periodic	**76**	205	Fixed-Point	**172**	202,216,228	Fixed-Point
27	39,53,83	Two-Cycle	**77**		Fixed-Point	**178**		Two-Cycle
28	70,157,199	Two-Cycle	**78**	92,141,197	Fixed-Point	**184**	226	Fixed-Point
29	71	Two-Cycle	**90**	165	Chaotic	**200**	236	Fixed-Point
30	86,135,149	Chaotic	**94**	133	Periodic	**204**		Fixed-Point
32	251	Null	**104**	233	Fixed-Point	**232**		Fixed-Point
33	123	Two-Cycle	**105**		Chaotic			
34	48,187,243	Fixed-Point	**106**	120,169,225	Chaotic			

For one-dimensional CA, the size m of the neighborhood is usually written as $m = 2r + 1$, where r is called the *radius* of the automaton. In the case of binary-state CA, the transition rule is given by a state transition table, which lists each possible neighborhood together with its output bit, that is, the updated value for the state of the central cell in the neighborhood.

2.1 The Elementary Rule Space

The 256 one-dimensional, $k = 2$, $r = 1$ CA are known as the elementary cellular automata (ECA). Wolfram proposed a numbering scheme for ECA, in which the output bits are lexicographically ordered, and read right to left, to form a binary integer between 0 and 255 [28]. The 256 ECA form the Elementary Rule Space [15].

Due to symmetry properties of the state transitions, each elementary rule can be equivalent to other elementary rules obtained by simple transformations [30]. For example, elementary rule 2 is equivalent to elementary rules 16, 191, and 247. There are 88 equivalent rule classes for the 256 elementary rules and they are presented in Table 1. Each class is represented in the table by a rule number, shown in the first column. The rule chosen to be the representative of the class is the one with the smallest number (in Wolfram's lexicographical ordering scheme) of all the rules in the class. The other members of each class, if any, are shown in the second column. The third column is explained in the Section 2.2.

2.2 Rule Space Classifications

Analysis of the dynamic behavior exhibited by CA verified that they could be grouped into classes. More than one rule space classification scheme has been used in the literature; for instance, Wolfram proposed a qualitative behavior classification in four dynamic classes, which is widely known [28–30]. Later on, Li and Packard proposed a series of refinements in the original Wolfram classification [13–16]. The following is one version [14] of their classification scheme, which divides the rule space into six classes; it is the version adopted in this article, and is referred to as Li-Packard classification scheme.

- *Null (or Homogeneous Fixed-Point)*: the limiting configuration (i.e., the automaton's state of the entire lattice) is formed only by 0s or 1s.
- *(Heterogeneous) Fixed-Point*: the limiting configuration is invariant, with possibly a spatial shift, by applying the cellular automaton rule, the null configurations being excluded.
- *Two-Cycle*: the limiting configuration is invariant, with possibly a spatial shift, by applying the rule twice (i.e., the state of the entire lattice cycles through two configurations).
- *Periodic*: the limiting configuration is invariant by applying the automaton rule L times, with the cycle length L either independent or weakly dependent on the system size.
- *Edge of Chaos, or Complex*: although their limiting dynamics may be periodic, the convergence time can be extremely long and they typically increase more than linearly with the system size.
- *Chaotic*: they produce nonperiodic dynamics. These rules are characterized by the exponential divergence of its cycle length with the system size, and for the instability with respect to perturbations.

In the third column of Table 1, the dynamic class of each rule in the Elementary Space is given according to the Li-Packard classification scheme.

2.3 Rule Space Parameterizations

Before proceeding further, it should be remarked that, although the objective here is the conception of a forecast parameter set, it is not possible to precisely forecast a generic cellular automaton from an arbitrary initial configuration. It has been proved, in [6], that the decision problem associated with the latter generic proposition is undecidable. Hence, all we can expect is really a parameter set that can *help* forecast the dynamic behavior of a cellular automaton; in particular, for present purposes (but in tune with the literature on the topic), our analysis is focused on one-dimensional, two-state CA.

The dynamics of a cellular automaton are associated with its transition rule. In order to help forecast the dynamic behavior of CA, several parameters have been proposed, directly calculated from the transition table. CA rule spaces with high cardinality make their parameterization a difficult task, so the case has been made for using more than a single parameter in order to achieve a better characterization [3, 13].

In [21] and [23] we reviewed and analyzed four of the main published parameters, namely, the λ parameter used by Langton [12, 13]; the mean field parameters, by Li and Packard [13, 14]; the sensitivity parameter (μ), as defined and published by Binder [3, 4], but originally proposed by the second author of this paper [13] and the Z parameter, by Wuensche [31–33]. In [23] we also considered three further parameters: λ-

ratio [33], the obstruction parameter θ [27], and the structural parameter σ [34]. The Sensitivity and Z parameters will be detailed in Section 4.

3 Analysis of the Rule Space Parameterizations

Out of the study of the parameters published in the literature and others proposed by us, an analysis of the main problems found was made, which led us to propose eight desired guidelines for the establishment of a parameter to help forecast CA dynamic behavior, as well as for the systematic analysis of the main published parameters (see [23]). The quantity of guidelines came out of the empirical/conceptual analysis we carried out. In light of these guidelines, in [23] we presented a critical analysis of the published parameters cited in Section 2.3. One of the conclusions of the analysis is that λ, the most widely used parameter, has some conceptual problems that limit its efficacy as a dynamic behavior forecast parameter (such as the fact that rules belonging to the same equivalence class of dynamical behavior have distinct λ values; see [23]). This point, in addition to others made in the literature [3, 13, 17], leads to the conclusion that λ alone is not sufficient to capture all the singularities of the rule space, since other parameters are also required. In spite of agreeing with the necessity of using a set of parameters, we have argued that λ should be kept out of that set, and that only parameters more aligned with the proposed guidelines should be preserved.

Another conclusion was that, just as those guidelines were good to evaluate already proposed parameters, they could also be used to guide the creation of new parameters, which we then proceeded to do. Together with quantitative and qualitative criteria toward how to select the parameters, the guideline provided us a parameter set that exhibited good performance in helping forecast the CA dynamic behavior.

4 The Selected Parameters

In tune with what the proposed guidelines establish, we selected two out of the published parameters: Sensitivity [3] and Z [32]. In addition, we conceived and tested several others, eventually selecting three more. These three new parameters have been referred to as Absolute Activity, Neighborhood Dominance, and Activity Propagation [21]. The resulting set of five parameters composes the set to be used herein.

Before defining the parameters, it is worth pointing out that they are not completely independent from each other. In fact, Sensitivity and Z do share common features, as do Absolute Activity and Neighborhood Dominance. But in each case, one parameter can be used to refine the other. Furthermore, Activity Propagation even combines concepts related to both Sensitivity and Neighborhood Dominance, thus acting as a link between the two groups. Finally, both Absolute Activity and Neighborhood Dominance can be regarded as refinements of the λ parameter, to the extent that all of them are measures of the activity level of a CA rule. However, the simplicity of λ does not allow taking into account the actual neighborhoods, which the new parameters do. So, while Absolute Activity considers where exactly in the neighborhood the activity happens, Neighborhood Dominance accounts for the activity in relation to the neighborhood as a whole.

Each of the five parameters is introduced below, together with its performance in the characterization of the elementary rule space.

4.1 Sensitivity

Binder proposed the sensitivity parameter, or μ [3, 4], which had been previously proposed by the second author of this paper, with the name of "Context Dependence" [13].

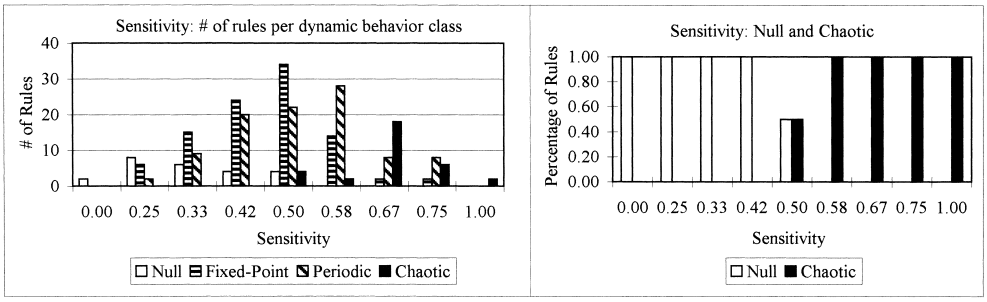

Figure 1. a) Number of occurrences of ECA rules for each value of the Sensitivity parameter, for each class of dynamic behavior. b) Relative occurrences of the null and chaotic ECA rules for each value of the Sensitivity parameter.

Its notion is motivated by the numeric observation that the Wolfram classes are characterized by its sensitivity to changes in the state of a unique cell of the neighborhood of the CA transition rule. Sensitivity is defined as the number of changes in the outputs of the transition rule, caused by changing the state of each cell of the neighborhood, one cell at a time, over all possible neighborhoods of the rule being considered. The following expression summarizes it:

$$\mu = \frac{1}{nm} \sum_{(v_1 v_2 \ldots v_m)} \sum_{q=1}^{m} \frac{\delta \Phi}{\delta v_q} \tag{1}$$

where n is the number of possible neighborhoods $(v_1 v_2 \ldots v_m)$ and m is the number of cells in the neighborhood $(n = 2^m)$. The CA Boolean derivative is $\frac{\delta \Phi}{\delta v_q} = 1$ if $\Phi(v_1 \cdots v_q \cdots v_m) \neq \Phi(v_1 \cdots \bar{v}_q \cdots v_m)$, that is, if the value of μ is sensitive to the value of the bit v_q, and zero, otherwise.

Additional explanation on μ can be found in [3], [4], and [21].

The sensitivity parameter (normalized between 0 and 1) was calculated for all 256 rules of the elementary space, which is presented in Figure 1a. The average value of the sensitivity parameter in the elementary space is 0.5, with a standard deviation of 0.14.

Figure 1a presents the number of rules of each dynamic class (null, fixed-point, periodic, and chaotic) for each possible value of the parameter in the elementary space. The chart, and others in this section, do in fact display the parameter value for only 250 rules; this was done because the six complex rules (which form only two equivalent classes) were left out, due to their lack of representativeness in the elementary space. Furthermore, the class referred to as "Periodic" is indeed the merging of the Two-Cycle and Periodic classes from the Li-Packard classification (that is, the Periodic class of the figure refers to rules with periodic behavior of cycle 2 or more).

Notice that the sensitivity parameter takes on nine different values: 0, 0.25, 0.33, 0.42, 0.50, 0.58, 0.67, 0.75, and 1. The null behavior happens in rules with low sensitivity and the chaotic behavior happens in rules with high sensitivity. Fixed-point and periodic behaviors are concentrated around 0.50. Due to this characteristic, the sensitivity parameter helps to relatively discriminate null and chaotic behaviors. Figure 1b makes this fact evident by plotting just null and chaotic ECA rules as a function of their relative occurrences. It is clear that sensitivity works as an almost perfect discriminator between these two dynamic behavior classes.

4.2 Z

The definition of the Z parameter given by Wuensche is derived from a pre-image calculation algorithm in the state space [31–33]. The pre-image of a cellular automaton is the configuration it was in for the previous iteration. The Z parameter is a measure of the pre-imaging degree imposed by the automaton's rule, that is, it indicates if the number of possible pre-images is high (low Z) or low (high Z) for an arbitrary lattice configuration.

Let us assume that part of a pre-image of an arbitrary configuration is known, and that we want to infer the missing cell states, successively, from left to right. Z_{LEFT} is defined as the probability that the next cell to the right in the partial pre-image has a unique value, and is directly calculated from the transition table of the automaton by counting the deterministic neighborhoods. Z_{RIGHT} is the converse, from right to left. The Z parameter is then taken as the largest value between Z_{LEFT} and Z_{RIGHT}.

The procedure for calculating Z is the following: Let d_m be the count of rule-table entries belonging to deterministic pairs of neighborhoods such that:

$$\begin{aligned} &v_1 v_2 \cdots v_{m-1} 1 \rightarrow T \\ &v_1 v_2 \cdots v_{m-1} 0 \rightarrow \bar{T} \end{aligned} \tag{2}$$

where T is the output bit of the transition. The probability that the next cell is determined on the basis of deterministic pairs is given by $P_m = \frac{d_m}{2^m}$.

Let d_{m-1} be the count of rule-table entries belonging to deterministic quadruples of neighborhoods. The probability that the next cell is determined on the basis of deterministic quadruples is given by $P_{m-1} = \frac{d_{m-1}}{2^m}$. This procedure is repeated up to the counting of the table configurations that belong to the deterministic 8-tuple neighborhoods, and so on, up to the special case when just one deterministic 2^m-tuple neighborhood occupies the whole transition table. The probability that the next cell of the partial pre-image is determined on the basis of at least one of the deterministic m-tuple neighborhoods is given by the expression:

$$Z_{\text{LEFT}} = P_m + \sum_{q=1}^{m-1} P_{m-q} \prod_{t=m-q+1}^{m} (1 - P_t) \tag{3}$$

where $P_q = \frac{d_q}{2^m}\mathrm{W}$ and d_q is the count of rule-table entries belonging to deterministic 2^{m-q+1}-tuple neighborhoods. An analogous, converse procedure yields Z_{RIGHT}. Additional explanation about the calculation of Z can be found in [31–33].

The normalized Z parameter was calculated for 250 rules of the elementary space, as mentioned above, and is displayed in Figure 2a. The average value of Z in the elementary space is 0.66, with a standard deviation of 0.2. Similarly to the sensitivity parameter μ, Z also helps to relatively discriminate null and chaotic behaviors, although not as clearly as μ. A disadvantage of Z in relation to μ is that Z can only take on six possible values: 0, 0.25, 0.50, 0.62, 0.75, and 1. As a consequence, the classes are more concentrated in Figure 2a than in Figure 1a, with a less sharp separation among them. Even then, Z remains as an option to relatively discriminate null and chaotic behaviors, as can be observed in Figure 2b, where just null and chaotic ECA rules are plotted as a function of their relative occurrences. Thus, the Z parameter can be used to reinforce the null/chaotic discrimination of the sensitivity parameter.

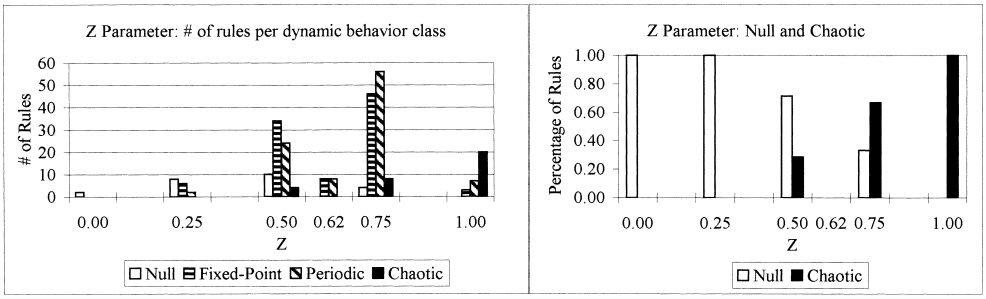

Figure 2. a) Number of occurrences of ECA rules for each value of the Z parameter, for each class of dynamic behavior. b) Relative occurrences of the null and chaotic ECA rules for each value of the Z parameter.

4.3 Absolute Activity

The motivation underlying the definition of Absolute Activity comes from the analysis of parameter λ (also known as activity) proposed by Langton. In spite of parameter λ intending to measure the activity level of a CA rule, what it does is simply to quantify the number of state transitions that maps onto state 1, regardless of the actual cell states of the neighborhood of concern [23]. The definition of Absolute Activity was conceived to amend the latter, its expression in the elementary space being as follows: *Number of state transitions that change the state of the central cell of the neighborhood + Number of state transitions that map the state of the central cell onto a state that is either different from the left-hand cell or from the right-hand cell of the neighborhood* −6.

The factor "6", subtracted in the end of the previous definition, is due to the fact that only two of the eight possible neighborhoods of the transitions are homogeneous (000 and 111), and in the others, the heterogeneous neighborhoods (001, 010, 011, 100, 101, 110), the new state of the central cell is always different from at least one of the three cells of the neighborhood. Therefore, every elementary rule has at least six transitions to be counted in the calculation of the absolute activity.

The normalized absolute activity was worked out for the 250 elementary rules of reference (as done before), and the result is shown in Figure 3a. The average value of the parameter is found to be 0.50, with a standard deviation of 0.22. The parameter has nine different possible values in the elementary space, 0, 0.12, 0.25, 0.38, 0.50, 0.62, 0.75, 0.88, and 1. Notice that absolute activity helps the relative discrimination between fixed-point and periodic behaviors, while null rules take on low values and

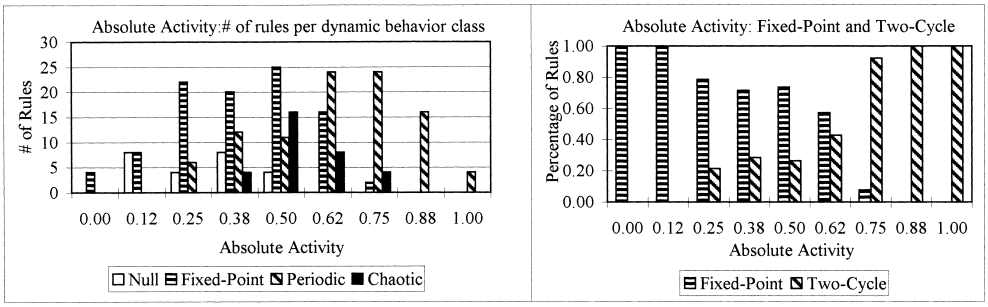

Figure 3. a) Number of occurrences of ECA rules for each value of the Absolute Activity parameter, for each class of dynamic behavior. b) Relative occurrences of the fixed-point and two-cycle ECA rules for each value of Absolute Activity.

chaotic rules, medium values. The relative discrimination can be more clearly realized by plotting just fixed-point and two-cycle ECA rules as a function of their relative occurrences, as displayed in Figure 3b. Two-cycle rules were selected for the plot because they represent the periodic behavior that is the closest to fixed-point. Such a discrimination is not as good as that obtained for null-chaotic rules, in respect to the sensitivity parameter. However, we should bear in mind that the number of rules involved in fixed-point/two-cycle relative discrimination (176) is larger than the number of rules involved in the null/chaotic situation (56). Besides, null and chaotic behaviors are conceptually extremes, while fixed-point and two-cycle behaviors are closer (indeed, fixed-point behavior is also periodic, with unitary cycle). Therefore, fixed-point/two-cycle relative discrimination is undoubtedly more difficult to be obtained than null/chaotic discrimination.

Once the results for this parameter were obtained in the elementary space, a generalization of its definition was made for 1D binary CA rules with arbitrary radius. This generalization was also necessary for the Neighborhood Dominance and Activity Propagation parameters that will be presented in the next sections. It is worth remarking that there is no single way to perform such a generalization, so that the one presented below is an attempt to capture the same rationale that led to the parameter definition in the elementary space. The generalized definition of the Absolute Activity parameter (not normalized) is given by:

$$A = \sum_{(v_1 v_2 \ldots v_m)} \left([\Phi(v_1 \cdots v_c \cdots v_m) \neq v_c] + \sum_{q=1}^{c-1} [\Phi(v_1 \cdots v_q \cdots v_m) \neq v_q) \,\Big\|\, \Phi(v_1 \cdots v_{m-q+1} \cdots v_m) \neq v_{m-q+1}] \right) \quad (5)$$

where m is the number of neighborhood cells and c is the position of the central cell, that is, $c = \frac{(m+1)}{2}$.

The symbol $\|$ used in Equation 4 (and elsewhere in the paper) represents the logical operator OR. The notation *[expression]* adopted in the equation is the Iverson convention (as in [9]), meaning that, if *expression* is true, the result of *[expression]* is 1, otherwise, the result of *[expression]* is 0. This convention is used throughout the paper.

The normalized value of the parameter is given by:

$$a = \frac{A - MIN}{MAX - MIN} \quad (6)$$

where:

$$MIN = \sum_{(v_1 v_2 \ldots v_m)} (\min(m_0, m_1)) \quad (7)$$

$$MAX = \sum_{(v_1 v_2 \ldots v_m)} (\max(m_0, m_1)) \quad (8)$$

with m_0 and m_1 being calculated for each possible neighborhood $(v_1 v_2 \ldots v_m)$ and being given by:

$$m_0 = \sum_{q=1}^{c} ([v_q = 0] \| [v_{m-q+1} = 0]) \quad (9)$$

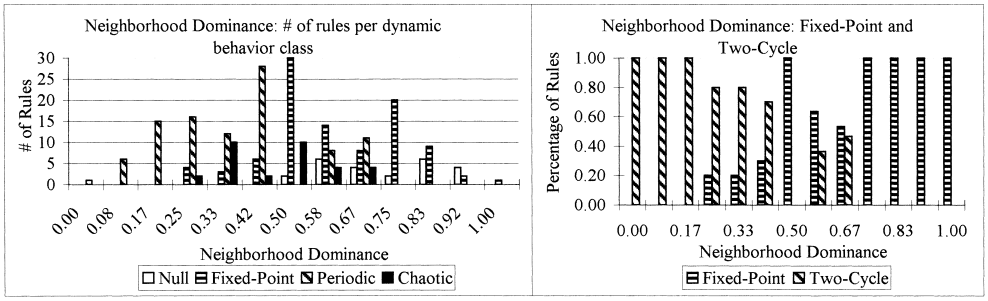

Figure 4. a) Number of occurrences of ECA rules for each value of the Neighborhood Dominance parameter, for each class of dynamic behavior. b) Relative occurrences of the fixed-point and two-cycle ECA rules for each value of the Neighborhood Dominance parameter.

$$m_1 = \sum_{q=1}^{c}([v_q = 1]\|[v_{m-q+1} = 1]) \tag{10}$$

The MIN portion represents the minimum absolute activity value for any CA rule of the rule space under consideration, due to the formation law of the neighborhoods. Analogously, the MAX portion represents the maximum value the parameter can take on in this rule space.

4.4 Neighborhood Dominance

This parameter also seeks to measure the changes caused by the state transitions of the CA, but it accounts for the effect of a change considering the neighborhood as a whole, with no special privilege to the central cell (as in Absolute Activity) or to any other cell. The parameter was named Neighborhood Dominance, as it verifies whether the new value of the central cell "follows" the state that dominates (i.e., that appears the most in) the neighborhood. Its definition for the elementary space is: *3 × (Number of homogeneous rule transitions that establish the next state of the central cell as the state that appears the most in the neighborhood) + (Number of heterogeneous rule transitions that establish the next state of the central cell as the state that appears the most in the neighborhood).*

We empirically observed in the elementary space that dominance from homogeneous neighborhoods (000 and 111) seems to have a larger effect on the rule's dynamic behavior than dominance from heterogeneous neighborhoods. These observations agree with Li's findings in [16], where the output bits of the transitions from neighborhoods 000 and 111 were named *hot bits*. It was because of this that we imposed a higher weight on homogeneous neighborhoods' dominance. By empirically testing the most appropriate value for this weight we found the value 3. Besides, we observed that there are three heterogeneous neighborhoods for each homogeneous neighborhood in the ECA. So, we expected that the best relative weight to be established among the neighborhoods in the calculation of the Neighborhood Dominance for any rule space can be given as a function of the number of occurrences of each neighborhood type. The more homogeneous the neighborhood, the smaller is the number of occurrences of this type of neighborhood and the larger is the relative weight that we should give to it.

Once again, the normalized neighborhood dominance was calculated for the 250 elementary rules of reference, as depicted in Figure 4a. The figure shows that, in the elementary space, the average value of the parameter is 0.50, its standard deviation is 0.20, and it has 13 different possible values: 0, 0.08, 0.17, 0.25, 0.33, 0.42, 0.50, 0.58,

0.67, 0.75, 0.83, 0.92, and 1. It is clear from the figure that neighborhood dominance also helps to relatively discriminate fixed-point and periodic behaviors, but in a sharper way than absolute activity. Unlike the latter, neighborhood dominance has high values for fixed-point rules while periodic rules have low values; null rules have high values and chaotic rules have medium values. This inversion of the positions of the dynamic classes in the axes of the two parameters is explained by the fact that the two parameters have a tendency to be inversely correlated: Rules that have a high neighborhood dominance tend to have a low absolute activity.

For a better visualization of this relative discrimination, only fixed-point and two-cycle ECA rules were plotted in Figure 4b (likewise Figure 3b) as a function of their relative occurrences. It is clear that the result is better than the one obtained for absolute activity. In fact, it was the best result obtained in the fixed-point/two-cycle relative discrimination over all tested parameters.

The generalized definition of Neighborhood Dominance is given by the following non-normalized expression:

$$D = \sum_{(v_1 v_2 \cdots v_m)} \binom{m}{V+c} \left[V < c \wedge \Phi(v_1 v_2 \cdots v_m) = 0 \right] + \sum_{(v_1 v_2 \cdots v_m)} \binom{m}{V-c} \left[V \geq c \wedge \Phi(v_1 v_2 \cdots v_m) = 1 \right] \tag{11}$$

where $V = \sum_{q=1}^{m} v_q$ and $c = \frac{(m+1)}{2}$. The symbol $\wedge$ used in Equation 10, and in others herein, represents the logical operator AND. The normalized version of the parameter is given by:

$$d = \frac{D}{2 \times \sum_{q=0}^{c-1} \binom{m}{q} \binom{m}{c+q}} \tag{12}$$

The denominator that appears in Equation 11 is the maximum value the parameter can have in the rule space under consideration.

4.5 Activity Propagation

Based on concepts related to the parameters mentioned above, new parameters were conceived and tested. One of them, Activity Propagation, was defined from two concepts related to the definitions of Neighborhood Dominance and Sensitivity: the possibility of a transition "following" (or not) the state that dominates the neighborhood, and the possibility of a transition being sensitive to a minimal change (a single state flip) in the neighborhood. Defined as such, it acts as an additional measure of context dependence. Its definition for the elementary space is given below.

First, let us consider the situation in which a neighborhood has the state of each of its cells individually flipped (or complemented). According to the cell being considered—left, right, or central—we define the resulting neighborhood as, respectively, the Left Complemented Neighborhood (LCN), the Right Complemented Neighborhood (RCN), and the Central Complemented Neighborhood (CCN).

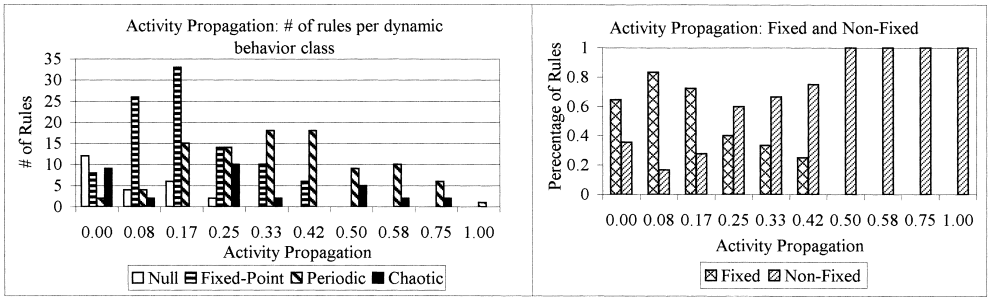

Figure 5. a) Number of occurrences of ECA rules for each value of the Activity Propagation parameter, for each class of dynamic behavior. b) Number of occurrences of the fixed and non-fixed ECA rules for each value of the Activity Propagation parameter.

The Activity Propagation parameter counts three types of transitions:

1. Number of transitions where the next state of the central cell is different from the state that dominates the neighborhood under consideration, and the next state of the central cell of the corresponding LCN is also different from the state that dominates the LCN.
2. Number of transitions where the next state of the central cell is different from the state that dominates the neighborhood under consideration, and the next state of the central cell of the corresponding CCN is also different from the state that dominates the CCN.
3. Number of transitions where the next state of the central cell is different from the state that dominates the neighborhood under consideration, and the next state of the central cell of the corresponding RCN is also different from the state that dominates the RCN.

The value of the Activity Propagation parameter is the sum of these three counts, across all neighborhoods of the rule transition, divided by 2, the latter being due to the fact that each neighborhood is counted twice in the parameter calculation.

The normalized activity propagation was calculated for the 250 elementary rules of reference, the result being displayed in Figure 5a. The average value of the parameter in the elementary space is 0.25, with a standard deviation of 0.19. The parameter has 10 different possible values in the elementary space: 0, 0.08, 0.17, 0.25, 0.33, 0.42, 0.50, 0.58, 0.75, and 1. The definition of the chaotic region is bad, because they are distributed along the entire parameter axis with no particular pattern of concentration. While periodic rules are characteristic of the central region of the parameter axis, fixed-point and null rules get clustered at the low end of the values, this being the strong point of the parameter (as will be discussed below).

For a better visualization of this aspect of the parameter (the clustering of fixed-point and null rules) Figure 5b was plotted, where the 250 elementary rules are grouped into two types, Fixed and Non-Fixed, the former referring to the ensemble of null and fixed-point rules, and the latter being the set of chaotic and periodic rules. Note that the activity propagation parameter does indeed concentrate fixed rules on the low values of its axis.

The generalized, normalized version of Activity Propagation is given by the expression:

$$P = \frac{1}{nm} \sum_{(v_1 v_2 \cdots v_m)} \sum_{q=1}^{m} [((V < c \wedge \Phi(..v_q..) = 1) \| (V \geq c \wedge \Phi(..v_q..) = 0)) \wedge ((\bar{V}_q < c \wedge \Phi(..\bar{v}_q..) = 1) \| (\bar{V}_q \geq c \wedge \Phi(..\bar{v}_q..) = 0))] \quad (13)$$

where $V = \sum_{q=1}^{m} v_q$, $\bar{v}_q$ is the binary complement of v_q, $\bar{V}_q = V - v_q + \bar{v}_q$, $c = \frac{(m+1)}{2}$, m is the size of the neighborhood, and n is the number of possible neighborhoods $(v_1 v_2 \ldots v_m)$.

5 Combination of the Selected Parameters in the Elementary Space

Interesting results were obtained in the characterization of the Elementary Space with the complete set of parameters [21]. Some of them are presented in this section.

5.1 Phase Diagram

In the published work relating the use of parameters for forecasting CA dynamics, the rule space is usually characterized as a function of the parameter variation, showing areas that clearly characterize different dynamic behaviors. The first attempt at this kind of characterization was made with the λ parameter. But, since it presented some problems, other authors suggested that using a pair of parameters was likelier to yield a better result; in this way, they drew a diagram, similar to the phase diagram in thermodynamics [3, 4].

Binder has published a phase diagram using the λ and Sensitivity parameters. His diagram presents a reasonable characterization; however, in order to obtain it, the author used the artifice of excluding several rules that were "out of the expected behavior" [3, 4].

Out of all the combinations of pairs of our selected parameter set, the best characterization of a phase diagram was obtained by using Sensitivity and Neighborhood Dominance. This result is shown in Plate 1, where all 256 ECA rules are represented.

Five dynamic behavior regions can be observed in the figure: null, fixed-point, periodic, complex, and chaotic. The null region is located in the fourth quadrant of the chart, whereas fixed-point rules are mostly in the upper quadrants, with a smaller part in the bottom quadrants. The periodic region is basically located in the surroundings of the diagonal that goes along the first and the third quadrants, and the chaotic region is located in the first and second quadrants, around the axis that divides them. In spite of the complex "area" being reduced to a straight line, it is interesting to notice that it happens exactly where there is an overlap among fixed-point, periodic, and chaotic regions (only two occurrences of complex behavior exist in the elementary space). Even though the regions do not present a perfect characterization (with no overlap), they still allow the identification of the typical kinds of dynamic behavior exhibited by complex systems.

By tracing a diagonal line that crosses the diagram at the fourth and second quadrants, and then going along it, all dynamic behavior regions mentioned previously are traversed. Plate 1 depicts this diagonal as a red line. If we go from the fourth quadrant towards the second quadrant—that is, decreasing the neighborhood dominance of the rules and increasing their sensitivity—a null region is initially traversed, followed by a small area with overlapping fixed-point rules. Then, a pure fixed-point region is found, from which point periodic rules start appearing. Then, there is a region with overlap-

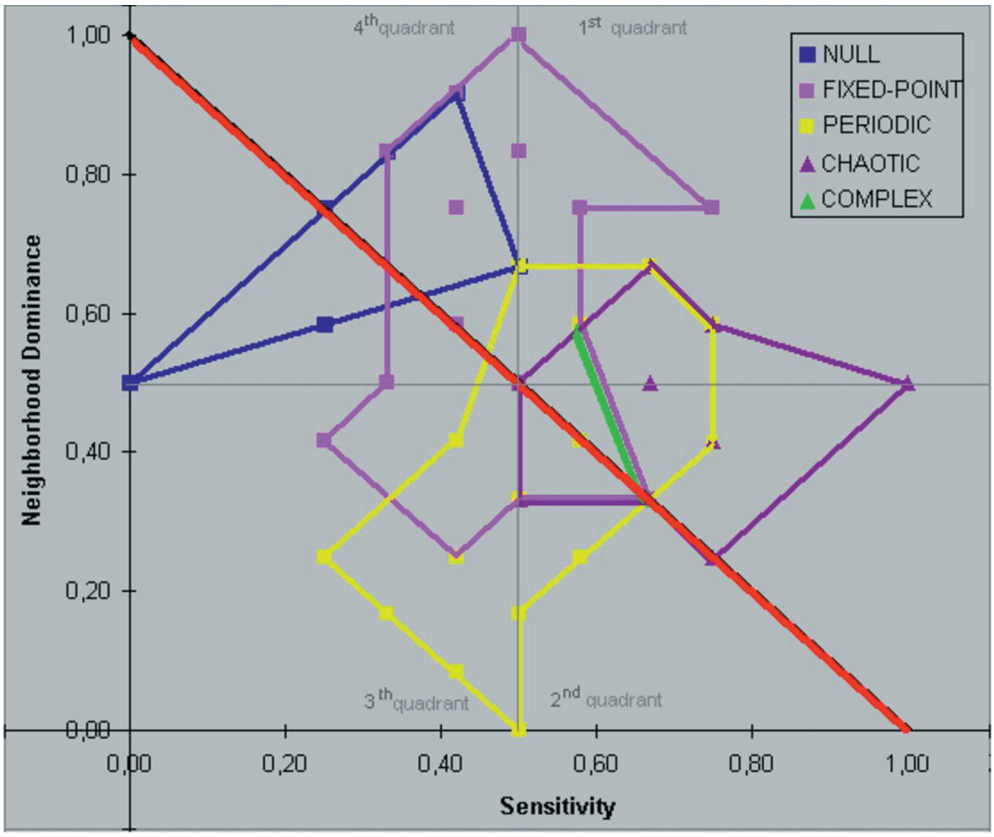

Plate 1. Phase diagram generated after Sensitivity and Neighborhood Dominance parameters.

ping chaotic, fixed-point, and periodic rules. Afterwards, only chaotic rules remain, up to the end of the diagonal. Exactly at the point where the overlap among fixed-point, periodic, and chaotic rules finishes, there is the occurrence of complex rules.

The preceding description is quite interesting, because some phenomena can be observed that are usually associated with complex dynamics. First, in the direction the diagram is traversed an increase in rule activity takes place: To decrease neighborhood dominance means that the rule transitions are imposing new states that are opposed to the states that predominated in the cells; similarly, to increase sensitivity means that the rules become more sensitive to any disturbance in the state of some of the neighborhood cells. Second, the sequence of behaviors found is compatible with the latter activity increase because, in an approximate way, a progression of dynamic regimes NULL → FIXED → PERIODIC → CHAOTIC takes place. Third, complex rules appear exactly when activity becomes high enough so as to guarantee chaoticity, that is, complex behavior can be observed "at the edge of chaos."

5.2 General Heuristics to Forecast Cellular Automata Dynamic Behavior

Even though the parameter set is not able to provide a precise description of the several dynamic behaviors in the rule space, it is possible to extract heuristics that can aid in forecasting a determined behavior type.

One application is to use these forecast heuristics to help the discovery of CA that exhibit particular implicit computation [17] abilities (in other words, that perform spe-

Table 2. Average values of the parameter set in the elementary space.

	Sensitivity	Neighborhood Dominance	Absolute Activity	Activity Propagation	Z
Null Rules	0.32	0.72	0.29	0.08	0.42
Fixed-Point Rules	0.46	0.60	0.39	0.18	0.63
Complex Rules	0.61	0.50	0.50	0.17	0.75
Two-Cycle Rules	0.49	0.33	0.67	0.39	0.66
Chaotic Rules	0.68	0.46	0.55	0.27	0.88

Table 3. Variation range of the parameter set in the elementary space.

	Sensitivity	Neighborhood Dominance	Absolute Activity	Activity Propagation	Z
Null Rules	0 to 0.50	0.50 to 0.92	0.12 to 0.50	0 to 0.25	0 to 0.75
Fixed-Point Rules	0.25 to 0.75	0.25 to 1	0 to 0.75	0 to 0.42	0.25 to 1
Complex Rules	0.58 to 0.67	0.33 to 0.58	0.38 to 0.75	0 to 0.50	0.75
Two-Cycle Rules	0.25 to 0.75	0 to 0.67	0.25 to 1	0 to 1	0.25 to 1
Chaotic Rules	0.50 to 1	0.25 to 0.67	0.38 to 0.75	0 to 0.75	0.78 to 0.98

cific computational tasks). Although we believe that such kinds of heuristics should be established in agreement with the task at issue, it is also possible to establish general heuristics for each behavior class, in agreement with the characterization of the parameters in the elementary space. Table 2 shows the average value of each parameter for each dynamic behavior class in the elementary space. Table 3 presents the range of each parameter where each dynamic behavior happens. For example, the average sensitivity for null ECA rules is 0.32, and it varies in the range 0 to 0.50.

Dynamic behavior heuristics can be built based on the data in Tables 2 and 3, as a function of the probability of occurrence of the parameter value. For each of the five parameters, the probability of the occurrence of some behavior is maximum at parameter values close to its average value in the elementary space (Table 2), and minimum outside the variation range of the parameter in the elementary space (Table 3).

6 Using the Parameter Set in High-Cardinality Rule Spaces

In this section some results are presented of the application of the parameter set, selected in the elementary space, in spaces of larger cardinality, especially in the resolution of computational tasks in radius-3 CA.

6.1 Evidence of the Efficacy of the Parameter Set in High-Cardinality Rule Spaces

Because the parameter set was selected in the elementary space, evidence is required to indicate that it could also be used in rule spaces of one-dimensional, binary CA with radius larger than 1. Naturally, due to the high cardinality of these rule spaces, an exhaustive search of the space is just not feasible. However, a few results of classification

Table 4. One-dimensional CA rules with radius 3, published for the DCT.

Rule	Hexadecimal Rule	Authors	Year	Method	Efficacy
GKL	005F005F005F005F005FFF5F005FFF5F	Gacs, Kurdyumov, Levin	1978	Manual	81.6%
MHC	0504058705000F77037755837BFFB77F	Mitchell, Hraber, Crutchfield	1993	Genetic Algorithm	76.9%
DAV	002F035F001FCF1F002FFC5F001FFF1F	Davis	1995	Manual	81.8%
DAS	070007FF0F000FFF0F0007FF0F310FFF	Das	1995	Manual	82.2%
ABK	0500550505005505555FF55FF55FF55FF	Andre, Bennet, Koza	1996	Genetic Programming	82.3%
CRA	005500550055005571F55FF57FF55FF57	Crany	1998	Numerical Methods	82.5%
JP1	011430D7110F395705B4FF17F13DF957	Juillé, Pollack	1998	Coevolutionary G.A.	85.1%
JP2	1451305C0050CE5F1711FF5F0F53CF5F	Juillé, Pollack	1998	Coevolutionary G.A.	86.0%

of dynamic regimes in these spaces do exist, which can then be used as a small test set, under the light of the selected parameters.

6.1.1 Elementary Subspaces of CA with Radius 2 and 3

In any rule space of one-dimensional binary CA with radius r, larger than 1, there is a set of 256 rules that are dynamically equivalent to the elementary rules, whose classifications are known. These rules are those whose transitions depend only on the states of the three centermost cells of the neighborhood, and which can be referred to as the Elementary Subspace with radius r [21].

We calculated the selected parameters for the Elementary Subspaces with radius 2 and 3, and the main conclusion is that, in spaces with small radius, the calculated values for the equivalent subspaces maintain the general characteristics pointed out in Section 5. The parameters are distributed for each dynamic behavior in value ranges that closely match those obtained for the elementary space.

6.1.2 Published Rules for the Density Classification Task of CA with Radius 3

One of the computational tasks we studied was the Density Classification Task [20, 24], DCT for short. In this task the goal is to find a binary one-dimensional cellular automaton that can classify the density of 1s in the initial configuration of the lattice, such that if the initial lattice has more 1s than 0s, the automaton should converge to a null configuration of 1s after a transient period; otherwise, it should converge to a null configuration of 0s. Various techniques have been described in the literature to find one-dimensional binary CA with radius 3 with such ability.

First we calculated the parameters for the best DCT published rules, eight in total, extracted from [1], [5], [11], and [18]; they are summarized in Table 4. In the first column of the table, an acronym is defined for each rule (from their discoverers' initials). In the other columns the following information is provided: the lexicographic specification of the rule in hexadecimal numbers; the authors of the corresponding piece of work; the year in which the rule was published; the method employed to find the rule (Manual, Genetic Algorithm, Numerical Method, Genetic Programming, or Coevolutionary Genetic Algorithm); and its efficacy for this task. The efficacy of the rule is usually measured in samples of 10^4 randomly generated initial configurations (of 149 cells), which are the most difficult ones to classify, as they have practically the same density of 0s and 1s. The efficacy is the percentage of initial configurations that the rule is able to correctly classify.

Table 5 presents the values of the five selected parameters (Sensitivity, Neighborhood Dominance, Activity Propagation, Absolute Activity, and Z) for the eight rules from Table 4. By analyzing Table 5, one can observe that, except for the Z parameter, all the other parameter values of all the rules are distributed in narrow bands. Table 6 presents these bands in its first row.

Table 5. Parameters of the DCT published rules.

Rule	Sensitivity	Neighborhood Dominance	Activity Propagation	Absolute Activity	Z
GKL	0.23	0.91	0.07	0.10	0.25
MHC	0.37	0.91	0.08	0.18	0.54
DAV	0.30	0.88	0.09	0.16	0.24
DAS	0.25	0.87	0.1	0.22	0.38
ABK	0.23	0.88	0.09	0.20	0.50
CRA	0.25	0.88	0.09	0.21	0.47
JP1	0.40	0.85	0.11	0.25	0.36
JP2	0.33	0.84	0.11	0.26	0.43

Table 6. Parameter ranges of the DCT published rules, elementary null rules, Synchronization Task (ST) published rules, elementary two-cycle rules, and for more than 95% of the randomly generated rules.

Rule	Sensitivity	Neighborhood Dominance	Activity Propagation	Absolute Activity	Z
DCT Published Rules (Radius 3)	0.23 to 0.40	0.84 to 0.91	0.07 to 0.11	0.10 to 0.26	0.24 to 0.54
Elementary Null Rules (Radius 1)	0 to 0.50	0.50 to 0.92	0 to 0.25	0.12 to 0.50	0 to 0.75
ST Published Rules (Radius 3)	0.44 to 0.46	0.19 to 0.26	0.35 to 0.41	0.69 to 0.73	0.52 to 0.56
Elementary Two-Cycle Rules (Radius 1)	0.25 to 0.75	0 to 0.67	0.25 to 1	0 to 1	0.25 to 1
More than 95% Random Rules (Radius 3)	0.45 to 0.55	0.35 to 0.65	0.15 to 0.35	0.4 to 0.6	0.5 to 0.7

In order for a rule to feature a good performance in the DCT, first of all it has to exhibit a null behavior. That is why the second row from Table 6 shows the parameter value ranges of the null rules of the elementary space, which can then be compared to the parameter values of the published rules. It can be observed that nearly all the latter values do in fact occur in the expected range for the null rules. There is one exception, Absolute Activity; but even for this case, its value range closely matches that of the expected band.

Therefore, the information from Table 3 leads to the conclusion that the parameter values of published rules for the Density Classification Task are really in the regions where our parameter set forecasts null dynamic behavior.

6.1.3 Published Rules for the Synchronization Task of CA with Radius 3

Another computational task we studied was the Synchronization Task, ST for short [7, 26]. In this task, the goal is to find a one-dimensional binary cellular automaton that, given an arbitrary initial configuration, after T time steps should reach a configuration that cycles through two lattice types: all cells in state 0 in one time step, and all cells in state 1 in the next time step; that is to say, $\Phi(0^N) = 1^N$ and $\Phi(1^N) = 0^N$. T is the minimum required time to guarantee synchronization of all cells, a parameter that depends on the lattice size (N).

Table 7 presents two CA rules with radius 3 that perform ST with 100% efficacy, both extracted from [18]. The efficacy for this task is the percentage of randomly generated initial configurations that the rule is able to correctly synchronize. We calculated the five selected parameters for them and the results are shown in Table 7; the parameter ranges for these rules are displayed in the third row of Table 6. As the dynamic

Table 7. Parameters of the ST published rules.

Rule	Sensitivity	Neighborhood Dominance	Activity Propagation	Absolute Activity	Z
FEB1C6EAB8E0C4DA6484A5AAF410C8A0	0.44	0.19	0.41	0.73	0.56
CEB2EF28C68D2A04E341FAE2E7187AE8	0.46	0.26	0.35	0.69	0.52

behavior for the Synchronization Task is the two-cycle behavior, we reproduce the parameter bands for this behavior class in the elementary space in the fourth row of Table 6.

Even though the number of rules analyzed here is not as significant as in the previous task, one can notice that the calculated parameter values for the two rules are relatively close, and fit within the ranges forecast by the five parameters for the two-cycle behavior. Besides, it is interesting to realize that the parameter ranges found for the ST (two-cycle behavior) are very different from those found for the DCT (null behavior), as expected.

6.1.4 Sampled Rules with Radius 3

In order to evaluate how specific the parameter ranges of the previous tasks would be for the published rules, we confronted those values with a sample of 50,000 randomly generated CA rules with radius 3. The fifth row of Table 6 shows the parameter value ranges of most of the generated samples (at least 95% of the 50,000 rules). Furthermore, it is worth noticing that the rules of the random sample have approximately the same density of 1s and of 0s (≈ 0.5) due to the random nature of their generation. In spite of this fact (which makes the sample a qualitatively particular one), notice that, quantitatively, they do represent the largest portion of the rule space.

It should also be observed that the parameter bands of the DCT published rules are not characteristic of the largest portion of the rule space, except the values of the Z parameter. The parameter bands of the ST published rules are also not characteristic of the largest portion of the rule space, especially the values of the Neighborhood Dominance, Absolute Activity, and Activity Propagation.

Therefore, the information from Table 6 leads to the conclusions that first, the parameter values of published rules for the two computational tasks are really in the regions where our parameter set forecasts the expected dynamic behaviors; and second, these bands are not trivially found for one-dimensional two-state CA with radius 3.

By calculating the parameter values of the selected set for the groups of rules previously mentioned, the values obtained can be shown to fit in the parameter ranges found in the elementary space for each class of behavior [21]. Therefore, they show there is various evidence that the parameters defined in the elementary space can indeed be used in higher-cardinality spaces. However, we believe that the largest evidence of the effectiveness of the selected parameters is the results obtained in the evolution of computational tasks made by CA. In the next section some of these results will be shown for binary CA with radius 2 and radius 3.

6.2 Using a Parameter-Based Heuristic to Guide the Evolution of Computational Tasks

There is a wide interest in the relationships between the generic dynamic behavior of a cellular automaton and its computational abilities as part of the more encompassing theme of the relationships between dynamic systems and computational theories [29]. Various investigations have been carried out on the computational power of CA, with concentrated efforts in the study of one-dimensional CA able to perform computational

tasks [1, 17–19, 24]. One of the approaches in this kind of research is the use of genetic algorithms (GA) [8] as a search procedure to find CA with the predefined computational behavior. Our approach is related to the latter, in that the selected parameter set mentioned earlier was used as an auxiliary metric to guide the processes underlying the GA search.

Once a computational task is defined, it is far from trivial finding CA that perform it. Manual programming is difficult and costly, and an exhaustive search of the rule space becomes impossible, due to its high cardinality. A solution is the use of search and optimization methods, particularly evolutionary computation methods.

Packard [24] was the first to publish results using a genetic algorithm as a tool to find CA rules with a specific computational behavior. He considered one-dimensional CA rules as individuals in a population and defined their fitness according to their ability to perform the specified task. In this way, the genotype of the automaton was given by its transition rule, and the phenotype by its ability to perform the required task. Crossover among two CA was defined by the creation of two new transition rules out of segments of two other rules; mutation was achieved by the random flipping of the output bit of one of the transitions of the rule.

Having obtained the interesting empirical results described in Section 6.1, and once information is available on parameter ranges where good rules should be more likely to occur, it is appealing to ask whether it would be possible to use this information in an active way when searching for CA of a predefined kind. Some experiments in the Density Classification and Synchronization tasks are presented as follows, where the selected parameter set was used as an auxiliary metric to guide the processes underlying the GA search.

6.2.1 Guided Evolution of the Density Classification Task

This section was adapted from [22]. We replicated one of the published experiments in which a GA was used to search CA for the DCT [19, 20]. In their experiment, Mitchell and collaborators used radius-3 binary CA, with a one-dimensional lattice formed by 149 cells, and evolved a population of 100 individuals during 100 generations. Each individual of the population was, therefore, a rule with 128 bits. At each generation, each individual evaluation was obtained out of testing the efficacy of the automaton in 100 different initial configurations (IC). Additionally, elitism [8] was used at a rate of 20% (that is, the 20 best rules of the population at each generation were always preserved to the next); the other 80 individuals were obtained through crossover and mutation. Parent selection for the crossover was made directly from the elite, with uniform probability for all. Standard one-point crossover was used at a rate of 80%. Mutation was applied after crossover, in each new generated individual, at a rate of 2% per bit. The results found for our replicated experiment (a series of 100 GA runs) are presented in Table 8.

The efficacy of the GA run was measured by testing the efficacy of the best rule found, at the end of the run, in the classification of 10^4 random initial configurations. Each row of Table 8 shows the number of runs in which the efficacy of the best rule found was within the corresponding range; the replicated experiment is referred to as "MHC". Table 9 displays the efficacy of the two best rules found in the experiment. The results we found are compatible with those reported in [19] and [20].

Subsequently, experiments were performed in which the selected parameter set was used as an auxiliary heuristic in evolutionary searching for CA. The idea was that the introduction of the parameter information would entail improvement in the results obtained, thus providing a clear metric so as to evaluate the ability of the parameters in helping forecast the dynamic behavior of CA, and, hence, to gather evidence so as to validate the efficacy of the selected set.

Table 8. Efficacy achieved for the "MHC" and "Parameters" experiments.

Efficacy Bands (%)	# of rules found in "MHC"	# of rules found in "Parameters"
≤ 50	6	0
(50, 55]	4	0
(55, 60]	0	0
(60, 65]	16	14
(65, 70]	69	77
(70, 75]	3	7
>75	2	2
TOTAL	100	100
Average	65%	67.3%

Table 9. Efficacy of the two best rules in the "MHC" and "Parameters" experiments.

	"MHC"	"Parameters"
First Rule	76.1%	80.4%
Second Rule	75.6%	77.2%

The parameter-based heuristic was coded as a function (referred to as F_p), which returns a value between 0 and 100 for each transition rule, depending on the rule's parameter values. In the present experiment, the parameter ranges of the DCT published rules presented in Table 6 were used, and function F_p was defined so as to return 100 if all the parameters of the cellular automaton rule matched those of the ranges of the published rules; otherwise, the value returned would decrease as the parameter values became increasingly further away from those ranges.

The GA was modified so as to incorporate the parameter-based heuristic in two aspects:

- The fitness function of a cellular automaton rule was made by the weighted average of the original fitness function (efficacy in 100 different ICs) and the function F_p.
- Biased reproduction and mutation: In order to select the crossover point and the rule table bits to be mutated, various attempts were made; among them, only those that generated rules with high F_p value were selected.

The parameter-based results are presented under the "Parameters" experiment label in Tables 9 and 10. Looking at the absolute values found after the insertion of the parameter information, two rules were found with higher efficacy than the best ones found by the basic experiment ("MHC"). It is also to be remarked that these results are better than those found by the researchers in the original experiment (76.9% in 300 runs, as reported in [20]). In relative terms Table 8 makes it evident that the rules found using the parameter information had a higher efficacy than those found without the information. And finally, the average efficacy of the "MHC" experiment was 65%, while the average of the "Parameters" experiment was 67.3%.

6.2.2 Guided Evolution of the Synchronization Task

In the experiments reported in [7], Das and collaborators used a genetic search, trying to find CA rules with radius 3 that could perform the Synchronization Task (ST). They actually found radius-3 rules with 100% efficacy, but they did not obtain success in CA

Table 10. Efficacy of the two best rules in the "Simple" and "R3 Guided" experiments.

Efficacy Band (%)	# of Rules found in "Simple"	# of Rules found in "R3 Guided"
≤ 30	1	1
(30, 40]	5	1
(40, 50]	12	3
(50, 60]	4	3
(60, 70]	5	11
(70, 80]	11	18
(80, 90]	2	3
>90	0	0
TOTAL OF RUNS	40	40
Average Efficacy	56.7%	66.7%

Table 11. Best efficacy obtained in the Synchronization Task experiments.

	"Simple"	"R3 Guided"
First Rule	82.8%	85.1%
Second Rule	80.8%	83.0%

with radius 2. In the latter space, they only found rules with virtually 0% efficacy. With the insertion of the parameter heuristics we would expect to find radius-2 CA rules of better efficacy.

So, after altering our working environment so as to allow it to handle a population of radius-2 CA (that is, each rule represented as 32 bits), we went about doing the experiments. The first one was accomplished without any information about the parameters. Table 10 shows the efficacy results of these runs (40 in total), which are referred to as "Simple"; Table 11 presents the two best rules found in this experiment. As we can observe, two rules were found with efficacy above 80%, and more than 10 rules with efficacy above 70%. Only in one run was a rule with efficacy below 30% found. The best result was 82.8%. These results were significantly better than in [7], where the authors reported that all rules found had about 0% efficacy (which led them to conclude it had been due to a limitation of the GA search). However, one possible explanation for the difference is that the results obtained by Das et al. [7] were due to some implementation mishap, possibly when they made the modification in their programming environment so as to allow it to handle individuals with 32 bits, instead of individuals with 128 bits.

Subsequently, a new experiment was performed, where the values of the selected parameters of the published radius-3 CA rules, discussed in Section 6.1, were used as an auxiliary heuristic in the evolutionary process. The method of incorporating this heuristic was the same one used in the DCT. Table 10 shows the efficacy results of this sequence of (also 40) runs, referred to as "R3 Guided," and Table 11 shows the two best rules found in the experiment. As a result, not only was the best efficacy found higher (85.1%), but also there was a clear shift in the number of rules found toward the higher efficacy ranges. Clearly, the introduction of the parameter-based heuristic entailed a substantial improvement in the average efficacy of the rules, as well as in the best rule found.

An interesting point to be remarked is that in the latter experiment, a radius-3-CA–based heuristic was used in the search of radius-2 CA rules and, even then, this entailed

a significant improvement. Such a result endorses, once again, the suggestion that a coherence really does exist among the parameter values calculated in spaces of distinct cardinalities.

Another experiment was then performed, where the general heuristic of the elementary two-cycle rules (described in Section 5.2) was used in the genetic search of the radius-2 rules. The results obtained (omitted here, but available in [21]) are similar to those found in the "R3 Guided" experiment. Once again, results from a rule space (radius 1) were used with success in the genetic search in another rule space (radius 2).

The best results were found in yet another experiment, where the best radius-2 rules found in the previous experiments were used as the heuristic of the genetic search (see [23]). The best rule found out in all experiments has 94.2% efficacy.

7 Conclusions

Elementary cellular automata provide a simple testing ground where dynamical behavior of locally interacting systems can be studied. Predicting the dynamical behavior from the rule table itself is an important issue. If this prediction is possible, then we can design cellular automata to achieve a specific dynamical behavior in an easier way. In tune with that, using the dynamic behavior prediction to guide an evolutionary search for designing CA is one of the possible applications, and, in fact, one that strongly links this work to the entire field of artificial life.

This work also relates to (at least) the two following proposed challenges to the field of artificial life, out of a set of 14, as presented in [2]. (a) *Explain How Rules and Symbols are Generated from Physical Dynamics:* In this respect, we quote a couple of self-explanatory sentences directly from [2], which seem to be very much in tune with the motivations of the work: "In fact, there are two issues here: . . . The first part of the challenge is to provide a theory explaining how dynamical systems can generate phenomena best understood as novel symbolic and rule-based behavior, . . . The second part is to specify under what conditions a natural discrete classification of dynamical states is recognized and reinforced by a dynamical system itself to structure its future evolution." (b) *Develop a Theory of Information Processing, Information Flow, and Information Generation for Evolving Systems:* Understanding the ability of CA to perform computations—such as Density Classification Task or Synchronization Task—is strictly linked to understanding how information is processed by and transmitted through its components (the set of cells). What we have done is definitely along this line, and hence, along with the proposed challenge, even though not yet having a theory for that. More specifically, the work is totally related to the idea of understanding the impact of the inherently local information processing of CA, and their ability to perform a coordinated computation at the global level, as mediated by an evolutionary process.

This article is the main summary of [21], a doctoral thesis written in Portuguese, and supplements our previous publications [22, 23] on the topic. In [22] we dealt with the application of the parameter set to the DCT problem only (which is one of the experiments described in Section 6.2), the parameters having only been mentioned and their general idea commented on. [23] is an in-depth account on the guidelines mentioned above, as well on the critical analysis of the well-known published parameters. Here, we formally define the parameter set, especially the actual new parameters Neighborhood Dominance, Absolute Activity, and Activity Propagation; also, the results of the analyses of the parameter set in the elementary space, like the phase diagram and the general heuristics, have all remained unpublished so far.

The complexity and the diversity of the dynamic behaviors of cellular automata make the characterization of the rule space very difficult to accomplish by a single parameter. All studied parameters were not individually sufficient to yield a reasonable characteri-

zation of the elementary rule space, even considering it is the smallest nontrivial space. The use of a set of parameters provided a much better description of the observed dynamics.

The tests performed for the study of the parameter set in rule spaces of high cardinality indicate that their definitions allow us to use the results obtained in the elementary space as heuristics in those spaces.

The effectiveness of the parameters has been demonstrated by means of the results obtained in evolving one-dimensional CA that perform predefined computational tasks, where the parameter set acts as a working heuristic for the search being made. As shown in Section 6.2, the insertion of the parameter information managed to improve the efficacy of the rules found for the Density Classification and Synchronization tasks, both on average and with respect to the best rules found.

Although the differences in efficacy shown in Tables 8, 9, 10, and 11 are small individually, it is important to notice that they were achieved in each and every experiment, jointly reflecting the consistency and robustness of the method and the ideas underlying it. If we ever want to use CA as models of complex, lifelike systems, evolving CA should be as important an issue as forecasting their dynamic behavior; the various routes taken here show that our parameter set can indeed help in both. Summing up, it is the consistency of the results, all of them together, in both fronts—the forecast and their use—that we think remains significant for the artificial life community, as it has been so far.

It should be remarked that only two experiments in the parameter-guided CA search have been reported here; but others have also been performed. In all of them, several aspects have been varied, such as the function F_p, the relative weight of F_p in the total fitness of a rule, the number of attempts of crossover and mutation, and the genetic algorithm parameters. All in all, the results clearly demonstrated an efficacy gain of the rules found when comparing them to the corresponding evolution without the parameter information, further emphasizing the robustness of the parameter-based heuristic.

We observe in the experiments involving the Synchronization Task that the analysis of the results of one experiment—obtaining the parameters of the best rules—could be used as a heuristic for a new experiment, where there was a significant improvement in the efficacy of the rules in relation to the original experiment. This demonstrates that the approach of using the parameters in the CA "programming" can be used in a adaptive way, through a continuous refinement of the results.

In addition to the Density Classification and Synchronization tasks presented in this article, runs were also carried out in a task we proposed, the Grouping Task [21], derived from the Ordering Task described in [25].

Acknowledgments

G. M. B. O. and N. O. are grateful to ITA and Universidade Presbiteriana Mackenzie for their support, and CAPES and MACKPESQUISA for funding. P. P. B. O. is grateful to Andy Wuensche and Wentian Li for various conversations, Tim Cranny for helpful information, and CNPq (ProTem-CC: CEVAL project) for funding. This work was performed while P. P. B. O. was at Universidade do Vale do Paraíba (UNIVAP-IP&D, Brazil).

References

1. Andre, D., Bennett, F., III, & Koza, J. (1996). Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. *Genetic Programming 96*. Palo Alto, CA: Stanford University.

2. Bedau, M., McCaskill, J., Packard, N., Rasmussen, S., Adami, C., Green, D., Ikegami, T., Kaneko, K., & Ray, T. (2000). Open problems in artificial life. *Artificial Life, 6*, 363–376.

3. Binder, P. M. (1993). A phase diagram for elementary cellular automata. *Complex Systems, 7*, 241–247.

4. Binder, P. M. (1994). Parametric ordering of complex systems. *Physics Rev. E.*

5. Cranny, T. (1998). Personal communication.

6. Culik II, K., Hurd, L. P., & Yu, S. (1990). Computation theoretic aspects of cellular automata. *Physica D, 45*, 357–378.

7. Das, R., Crutchfield, J. P., Mitchell, M., & Hanson, J. E. (1995). Evolving globally synchronized cellular automata. In L. J. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms* (pp. 336–343). San Francisco: Morgan Kaufmann.

8. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning.* Boston, MA: Boston: Addison-Wesley.

9. Graham, R., Knuth, D., & Patashnik, O. (1995). *Concrete mathematics—A foundation of computer science.* Addison-Wesley.

10. Jen, E. (1990). Aperiodicity in one-dimensional cellular automata. *Physica D, 45*, 3–18.

11. Juillé, H., & Pollack, J. B. (1998). Coevolving the "ideal" trainer: Application to the discovery of cellular automata rules. In *Proceedings of Genetic Programming Conference.* Madison, 3.

12. Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D, 42*, 12–37.

13. Li, W. (1991). *Parameterizations of cellular automata rule space.* Santa Fe Institute Tech. Report: Preprints, Santa Fe, NM.

14. Li, W. (1992). Phenomenology of nonlocal cellular automata. *Journal of Statistical Physics, 68*, 829–882.

15. Li, W., & Packard, N. (1990). The structure of elementary cellular automata rule space. *Complex Systems, 4*, 281–297.

16. Li, W., Packard, N., & Langton, C. G. (1990). Transition phenomena in cellular automata rule space. *Physica D, 45*, 77–94.

17. Mitchell, M. (1996). Computation in cellular automata: A selected review. In *Nonstandard Computation.* Weinheim: VCH Verlagsgesellschaft.

18. Mitchell, M., Crutchfield, J., & Das, R. (1996). Evolving cellular automata with genetic algorithms: A review of recent work. In *Proceedings of International Conference on Evolutionary Computation and Its Applications.* Moscow, 5.

19. Mitchell, M., Hraber, P., & Crutchfield, J. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems, 7*, 89–130.

20. Mitchell, M., Hraber, P., & Crutchfield, J. (1994). Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D, 75*, 361–391.

21. Oliveira, G. M. B. (1999). *Dynamics and evolution of one-dimensional cellular automata.* Ph.D. thesis. Aeronautics Institute of Technology, São José dos Campos, Brazil. (In Portuguese).

22. Oliveira, G. M. B., de Oliveira, P. P. B., & Omar, N. (2000). Evolving solutions of the density classification task in 1D cellular automata, guided by a parameter-based heuristic that forecast their dynamic behavior. In M. A. Bedau, J. S. McCaskill, N. H. Packard, & S. Rasmussen (Eds.), *Artificial Life VII* (pp. 428–436). Cambridge, MA: MIT Press.

23. Oliveira, G. M. B., de Oliveira, P. P. B., & Omar, N. (2000). Guidelines for dynamics-based parameterization of one-dimensional cellular automata rule spaces. *Complexity, 6*, 63–71.

24. Packard, N. (1988). Adaptation toward the edge of chaos. In *Dynamic Patterns in Complex Systems* (pp. 293–301). Singapore: World Scientific.

25. Sipper, M. (1998). A simple cellular automata that solves the density and ordering problems. *International Journal of Modern Physics, 9* (7).

26. Sipper, M. (1996). Co-evolving non-uniform cellular automata to perform computations. *Physica D, 92*, 193–208.

27. Voorhees, B. H. (2000). Some parameters characterizing cellular automata rules. *Complexity*. To appear.

28. Wolfram, S. (1983). Cellular automata. *Los Alamos Science, 9*, 2–21.

29. Wolfram, S. (1984). Computation theory of cellular automata. In *Communication in Mathematical Physics, 96*, 15–57.

30. Wolfram, S. (1994). *Cellular automata and complexity*. Boston: Addison-Wesley.

31. Wuensche, A. (1994). Complexity in one-D cellular automata: Gliders, basins of attraction and the Z parameter. *Cognitive Science Research Papers: CSRP 321*. Brighton, UK: University of Sussex.

32. Wuensche, A. (1999). Classifying cellular automata automatically: Finding gliders, filtering, and relating space-time patterns, attractor basins and the Z parameter. *Complexity, 4*, 47–66.

33. Wuensche, A., & Lesser, M. J. (1992). *The global dynamics of cellular automata*. Boston: Addison-Wesley.

34. Zwick, M., & Shu, H. (1995). Set-theoretic reconstructability of elementary cellular automata. *Advances in Systems Science and Applications*. Special Issue I:31–36.