



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect

Electronic Notes in  
Theoretical Computer  
Science

Electronic Notes in Theoretical Computer Science 252 (2009) 121–142

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

# Some Investigations About Synchronization and Density Classification Tasks in One-dimensional and Two-dimensional Cellular Automata Rule Spaces

Gina M. B. Oliveira <sup>1,2</sup> Luiz G. A. Martins  
Laura B. de Carvalho <sup>3</sup> Enrique Fynn <sup>4</sup>

*Artificial Intelligence Laboratory  
Universidade Federal de Uberlândia  
Uberlândia, Brazil*

---

## Abstract

The study of computational aspects of cellular automata (CA) is a recurrent theme being that the investigation of specific tasks to be solved by CA rules a common and widely-known approach. We investigated two of the most-studied computational tasks: synchronization (ST) and density classification (DCT). Different specifications of CA rule space were analyzed for both tasks: one-dimensional rules with radius 1 and 2, and two-dimensional rules with von Neumann and Moore neighborhoods. We also analyzed different lattice sizes when trying to execute these tasks. Several evolutionary experiments were performed to characterize ST and DCT on these different scenarios. Some interesting results have been occurred from these experiments as the adequacy of the tasks to be solved in two-dimensional spaces instead of 1D even using rules with the same length and the dependency to the parity of the lattice size related to good rules for DCT in 1D and 2D spaces.

**Keywords:** Cellular Automata, Evolutionary Algorithms, Synchronization Task, Density Classification Task

---

## 1 Introduction

Although being very simple to implement cellular automata (CA) are able to perform complex computations. The understanding of how they perform these computations is still vague. An efficient approach is the use of evolutionary algorithms,

<sup>1</sup> Thanks for CNPQ and FAPEMIG support

<sup>2</sup> Email: [gina@facom.ufu.br](mailto:gina@facom.ufu.br)

<sup>3</sup> Thanks for CNPQ support

<sup>4</sup> Thanks for FAPEMIG support

as the genetic algorithm (GA), to search for CA able to exhibit a desired computational behavior over their rule space. Two of the most studied computational tasks in this context are the synchronization task (ST) [3] [19] [8] and the density classification task (DCT) [10] [6] [4] [22]. These tasks have been historically studied in 1D cellular automata rule spaces, except for some explorations about DCT in 2D spaces using Moore Neighborhood [22] [12]–[14].

Some results and considerations about investigations we performed in both these computational tasks are presented here. We explore ST and DCT in classical 1D space using radius 2 and radius 3 rules, resulting in rules with 32 and 128 bits respectively. Besides, we investigated these tasks in 2D spaces using von Neumann and Moore neighborhoods, resulting in rules with 32 and 512 bits.

Section 2 gives a brief description about two computational problems widely employed in CA studies. Section 3 describes the previous usage of evolutionary algorithms to find good CA rules that perform a specific task. Section 4 presents a brief historical discussion about experiments involving synchronization task and density classification task in one and two-dimensional CA lattices, including the results and analysis of current experiments. The main conclusions of this work are present in section 5.

## 2 Computational tasks

Cellular Automata have the potential to embody models of complex systems, and to act as abstract machines that perform complex computations with high degree of efficiency and robustness. Various investigations have been carried out on the computational power of CA, with concentrated efforts in the study of one-dimensional CA capable of performing computational tasks [9].

### 2.1 Density classification task

The most widely studied CA task is the density classification task (DCT) [10]. In this task the objective is to find a binary 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.

Figures 1 and 2 present two space-time diagrams of one-dimensional CA rules successfully performing DCT with radius 2 and radius 3, respectively. The rule used to generate the spatial-temporal evolutions in Fig. 1 (radius 2) has an efficacy of 56% in this task, while the one used in Fi. 2 has an efficacy of 88.9%. These efficacies were measured using 10,000 1D lattices with 149 bits.

Although the majority of published works about DCT investigates its original one-dimensional version, Morales and colleagues have proposed to study two-dimensional rules to solve this problem [12]. Figures 3 and 4 present two sequences of two-dimensional lattices formed by the application of 2D CA rules successfully performing DCT; Fig. 3 started from a initial lattice with more 1s than 0s and Fig. 4 with more 0s. The rule used in Fig. 3 uses von Neumann neighborhood and has an

efficacy of 67% while the rule used in Fig. 4 uses Moore neighborhood and has an efficacy of about 83%. These efficacies were measured using 10,000 2D lattices with  $21 \times 21$  bits.

DCT is a nontrivial task for a small-radius CA in any dimension, since they rely only on local interactions. On the other hand, this task is trivial for a system with a central controller or a central storage [9]. Performing this task well for a fixed lattice size requires more powerful computation than can be performed by a single cell or any linear combination of cells. Since the 1s can be distributed throughout the CA lattice, the CA must transfer information over large distances. A kind of global coordination is required to communicate cells that are separated by large distances and that cannot communicate directly. However, it has been proved that no finite-radius, two-state CA with periodic boundary conditions can perform this task perfectly across all lattice sizes [7]. The most studied configuration for DCT is defined by 1D lattice with 149 cells and CA rules with radius 3; the best performance of 1D DCT task in this configuration is about 89% [22].

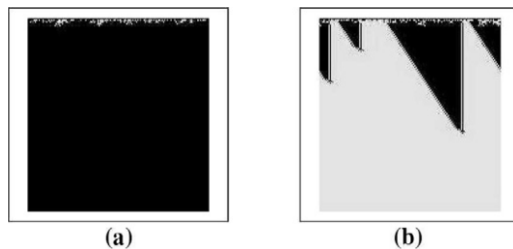


Fig. 1. Density Classification Task using a radius 2 1D rule (rule: 142149B7).

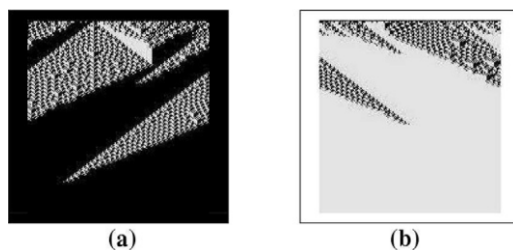


Fig. 2. Density Classification Task using a radius 3 1D rule (rule: 1461077C0F00CEBF1721047F0F333F3BF).

## 2.2 Sincronization task

Another computational task previously studied was the synchronization task, ST for short [3] [19]. In this task, the goal is to find a binary cellular automaton which, given an arbitrary initial configuration of a lattice of  $N$  cells, after  $T$  time steps, it 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.  $T$  is the minimum time required to guarantee synchronization of all cells, a parameter that depends on the lattice size ( $N$ ).

There are several one-dimensional radius-3 CA rules able to solve ST for any arbitrary lattice configuration [3]. Considering one-dimensional radius-2 rules, it

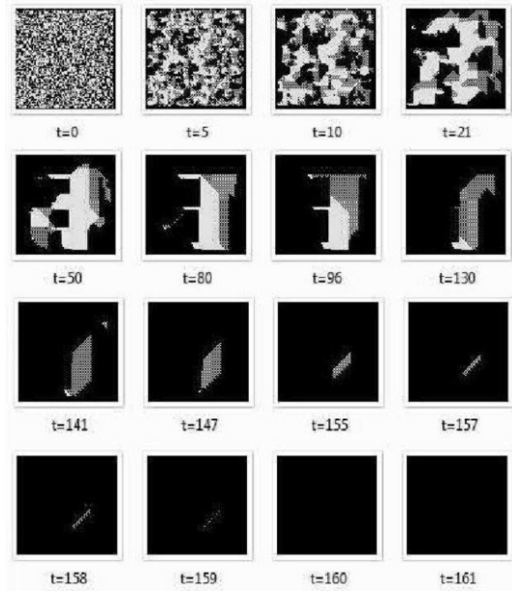


Fig. 3. Snapshots of the evolution of a rule with von Neumann neighborhood solving Density Classification Task in 2D space (rule: 111315FF).

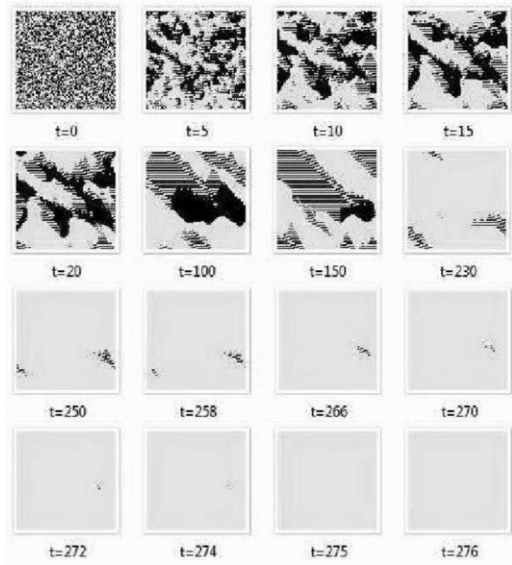


Fig. 4. Snapshots of the evolution of a rule with Moore neighborhood solving Density Classification Task in 2D space (rule: 00000001001101410B1514050D2B77570101130703054557009557D703957FF70B1B455505131345177F475F077F7F77FF57475D1F7F6765BF3F3FBFFFFF7F77).

is known that there are good rules with efficacy above 95% [8]. Figure 5 presents space-time diagrams of two different one-dimensional CA rules solving ST: the first is a radius-3 CA rule and the second is a radius-2 one. The rule used to generate the spatial-temporal evolutions in Fig. 5a (radius 3) has an efficacy of 100% in this task; while the one used in Fig. 5b (radius 2) has an efficacy of about 92%. The efficacy of the radius 2 rule was measured using 10,000 1D lattices with 149 bits.

Synchronization task can also be generalized for the 2D space. Figures 6 and 7 present two sequences of two-dimensional lattices formed by the application of rules successfully performing ST. The rule used in Fig. 6 uses von Neumann neighborhood and it has an efficacy of 100% as so as the rule used in Fig. 7, which uses Moore neighborhood.

ST is also nontrivial since a small-radius CA employs only local interactions while synchronous oscillation is a global property. The locality of interaction can directly lead to regions of local synchrony and it is more difficult to design a CA that will guarantee that spatially distant regions are in phase 0. An effective CA rule must transfer information over large space-time distances.

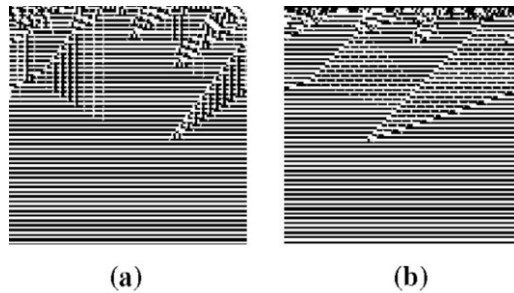


Fig. 5. Synchronization task in 1D space. (a) Radius-2 rule: EAC38AE8 and (b) Radius-3 rule: FEB1C6EAB8E0C4DA6484A5AAF410C8A0.

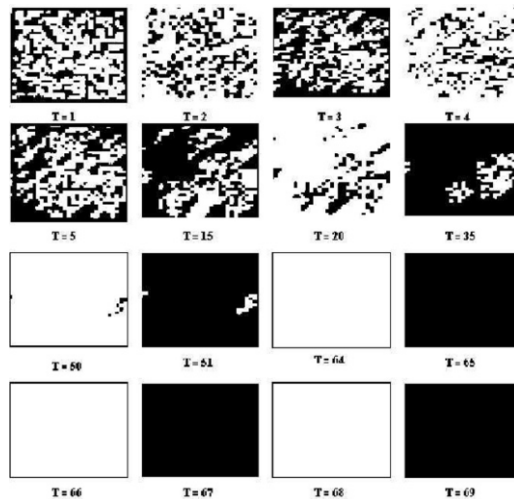


Fig. 6. Snapshots of the evolution of a rule with von Neumann neighborhood solving Synchronization Task in 2D space (rule: F3A9DF98).

### 3 Evolving cellular automata rules

Once a computational task is defined, it is not easy to find a cellular automaton rule that performs it. Manual programming is difficult and costly, and exhaustive search of the rule space becomes impossible, due to its high cardinality. A practical

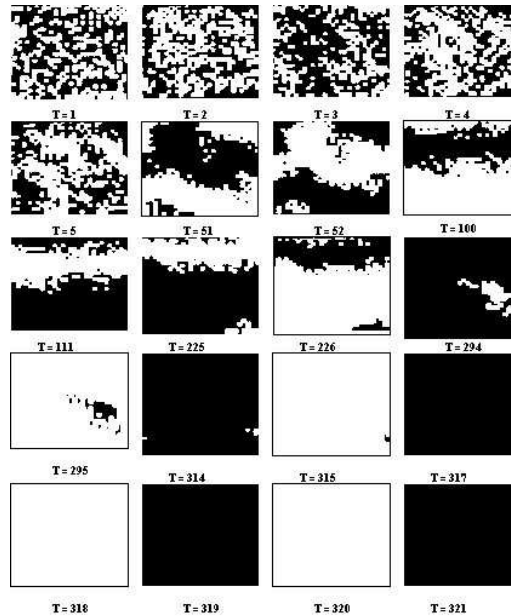


Fig. 7. Snapshots of the evolution of a rule with Moore neighborhood solving Synchronization Task in 2D space (rule: FDEFFCCEEC7EF77FFEEEDFEA7F3FE69DB77FD729BBB97FD9DC46008003888082008200000051020042282D0C0191204080840C08000210202800201A10080000).

alternative has been the use of search and optimisation algorithms, particularly evolutionary computation methods [10] [6] [20] [1] [15] [20].

Packard (1988) was the first to publish results using a genetic algorithm as a tool to find CA rules with a desirable computational behavior [20]. He considered one-dimensional CA rules as individuals in a population and their fitness is defined 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 flip of the output bit of one of the transitions of the rule.

Other evolutionary computation techniques were used to find such kind of CA rules. Genetic programming was also used as a search procedure for CA rules to perform the density classification task [1]. Furthermore, for the same task, an important radius-3 rule was obtained by a coevolutionary approach [6]. It has an efficacy of 86% in performing DCT and it remained as the best-known rule for this task until 2008. Recently, Wolz and de Oliveira (2008) have found a large set of 1D radius 3 rules (more than 100 rules) able to perform this task with an efficacy above 88% [22]. They have used a two-tier evolutionary environment and the best rule found in this work is 1461077C0F00CEBF1721047F0F333FBF (the rules are presented by their hexadecimal code representing the output bits from neighborhood 0000000 to 1111111). It is the best-known rule for this task and it has an efficacy of 88.99%.

Evolutionary methods were also used with synchronization task. ST was proposed by Das and colleagues (1995) [3] and it was also investigated in [19] [8] [21]. In

the experiments reported in [3] a genetic search was used to look for radius-3 rules able to perform the synchronization task. Several radius-3 1D rules that perform ST with an efficacy of 100% were obtained. By other hand, they did not obtain success in CA with radius-2. In this rule space, they only found rules with virtually 0% of efficacy [3]. Later on, Oliveira and collaborators (2001) had evidenced that a simple GA as the employed in [3] was able to find radius-2 rules with efficacy around 80% [19]. Moreover, by introducing a kind of parameter information it was possible to find radius-2 CA rules with more than 90% of efficacy solving ST. Recently, Mariano and Oliveira found the best known rule for this task with 96.4% of efficacy: FD9E9042 [8].

In previous works, a simple GA environment was modified to incorporate a heuristic based on some forecast parameters and it was used to find DCT and ST rules [19] [15]. This approach uses parameter bands where good rules are more likely to occur. Once this information is available, it is used in an active way, as an auxiliary metric to guide the processes underlying the GA search. This approach has been previously used to search for CA rules in: 1D DCT [4] [15], 2D DCT [13] [14] and 1D ST [19] [8].

## 4 Experiments

This section presents results and analysis of evolutionary experiments performed using different CA rule spaces considering synchronization and density classification tasks.

### 4.1 Evolving rules for one-dimensional synchronization task

For this task, it was previously known that there are rules with 100% of efficacy in the space formed by 1D radius 3 CA rules [3]. The rule FEB1C6EAB8E0C4DA6484A5AAF410C8A0 extracted from reference [3] and used to generate Fig. 5b is one example of such kind of rule. These rules can be obtained with a simple GA run as described in [3] with relative small size of GA parameters: population size ( $Pop$ ) = 50, number of initial configurations used to test each rule ( $IC$ ) = 50 and number of generations ( $Ng$ ) = 50. However, considering 1D radius 2 CA rules the good results are not so easy to obtain. Initially, Das and colleagues have reported that their GA has not found good rules for ST using radius 2 CA. Later on, it was confirmed that with a single GA it is possible to find radius 2 rules with efficacy around 80% and a good radius 2 rule was obtained incorporating a parameter-based heuristic to aid the GA [19]. This rule was used to generate Fig. 5a; its code is EAC38AE8 and it has an efficacy of 92% in lattices with 149 bits. Recently, the parameter-based heuristic was refined and two better rules were possible to find by GA runs [8]. Their codes are FD9E9042 and BDF68640; both rules return efficacy around 96.3%. These results were obtained with a relative small GA:  $Pop$  = 100,  $IC$  = 100 and  $Ng$  = 100. Aiming to verify if this efficacy can be improved, we executed an evolutionary environment similar to the described in [8] using  $Pop$  = 500,  $IC$  = 500 and  $Ng$  = 500. It was not possi-



ble to find any rule better than the two rules previously cited. However, we can find a large number of good rules. The best 10 rules are listed in Table 1 each one with its efficacy. The CA specification used in these experiments was the same proposed in [3] and used in [8]: time steps in CA evolution equal to 320, lattice size of 149 bits, number of initial configurations used in the evaluation of the efficacy equal to 10,000. Therefore, we speculate that using these CA specification, the generated rules are the best possible in radius 2 space. However we noted that these rules can exhibit an “almost perfect performance” if a large number of CA steps is adopted. For example, if we evaluate these rules using 1000 time steps in CA evolution instead of 320. Keeping the lattice size with 149 bits the new efficacy of each rule using 10,000 initial configurations and 1000 time steps are presented in Table 1. The efficacies of almost all these rules have returned more than 99% with the increment on the number of CA steps, except for the rule AAE80AC0. However, if we try a high step number, these results do not change as presented in Table 1. It means that some initial lattices will not been synchronized even using a time step number as larger as possible. This situation is due to the existence of few other attractors different from the two-cycle attractor corresponding to the synchronized behavior. If the initial lattice is within the basin of attraction of such kind of attractor the homogeneous lattices will never be reached. Therefore, the perfect rule for ST is apparently not possible in one-dimensional CA rule space.

Table 1  
Top 10 best one-dimensional radius 2 CA rules found in synchronization task.

Rule	Efficacy in $10^5$ (320 time steps)	Efficacy in $10^5$ (1000 time steps)	Efficacy in $10^5$ (10000 time steps)
FD9E9042	96.3%	99.4%	99.4%
BDF68640	96.3%	99.5%	99.5%
E8A63EA8	96.2%	99.4%	99.4%
EA839AE8	96.2%	99.2%	99.2%
AAE80AC0	95.8%	98.0%	98.0%
BD9EB000	95.6%	99.90%	99.90%
E8AFBCA8	95.6%	99.96%	99.96%
BD9EB040	95.5%	99.97%	99.97%
FFF28642	95.5%	99.98%	99.98%
FD9EB002	95.5%	99.94%	99.94%



4.2 Evolving rules for two-dimensional synchronization task

The next step of our investigation was to evaluate the synchronization task in 2D CA spaces. First we investigated Moore neighborhood. In this case the rules are formed by 512 bits and they were tested in 2D lattices formed by 12x12 bits; that is, lattices with 144 bits positioned in a spatial arrangement. The other CA parameters used in these experiments was the same proposed and used in 1D experiments: time steps in CA evolution equal to 320 and number of initial configurations used in the rule evaluation equal to 10,000. It was possible to find rules with 100% of efficacy in the majority of GA runs performed and three examples of these rules are presented in Table 2. Such rules were obtained with a small GA:  $Pop\hat{A}=\hat{A}100$ ,  $IC\hat{A}=\hat{A}100$  and  $Ng\hat{A}=\hat{A}100$ . Subsequently, we performed the same experiment using von Neumann neighborhood. In this case, the rules are formed by 32 bits and they were also tested in 2D lattices formed by 12x12 bits. Rules with 100% of efficacy were possible to find in approximate 33% of the GA runs performed; three examples of these rules are also presented in Table 2.

Since we already knew about the existence of 1D radius 3 rules able to solve this task with 100% of efficacy, formed by 128 bits, the results with Moore neighborhood were not surprising, once they have 512 bits. However, in the case of von Neumann neighborhood, the rules have exactly the same size of radius 2 1D rules: 32 bits. Therefore, the spatial arrangement of the 144 cells has facilitate the communication among cells in a such manner that they were able to solve the problem with a better performance than the linear arrangement of 149 cells used with radius 2 CA rules. This behavior was also observed in DCT experiments reported in next section.

Table 2  
Two-dimensional rules to solve ST

Moore neighborhood	von Neumann neighborhood
FDEFFCECEEC7EF77FFEEDFEA7F3FE69D B77FD729BBB97FD9DC4600800388808200820 0000051020042282D0C0191204080840C0800021 0202800201A10080000	F3A9DF98
FA91F741C26380659C29B284BB3AF183D8F27 4FAC9714EB4E1F648E100804008300040300400 80080A010D80210020808C006040080022880091 040008240000	94A00280
FE77CFFFED7BFA7C001011800500354860050 2835C008002010000804052819080000409200208 800102006840408100400101A0310010002000900 400030000	81C22800

### 4.3 Evolving rules for one-dimensional density classification task

The most-studied specification involving DCT is radius 3 CA rules applied over 149-bits 1D lattices. In 1978, Gacs, Kurdyumov and Levin proposed DCT and presented a radius 3 rule known as GKL which returns 81.6% of efficacy when tested in random 149-bits lattices [5]. Later on, in 1993, Mitchell, Hraber and Crutchfield proposed an evolutionary search for DCT using genetic algorithms that became an important reference to all subsequent works using evolutionary techniques to search for CA rules [10]. However, the best rule found in their experiments returned an efficacy of 76.9%. In 1996, Andre, Bennet and Koza have found for the first time a rule better than GKL, using a genetic programming environment [1]. Its efficacy is about 82.3% in this task. Juillé and Pollack used a coevolutionary environment to find a rule with 86.0% of efficacy in 1998 [6]. Its hexadecimal code is 1451305C0050CE5F1711FF5F0F53CF5F. This rule remained as the best one known for about 10 years until 2008 when Wolz and de Oliveira found a set of more than 7,000 rules better than Juillé and Pollack's one [22]. The evolutionary environment employed in [22] was based on a two level hierarchical architecture where a low level basic evolutionary search is embedded in a global, parallel, multi- population search algorithm. The lower level consists of several basic single population evolutionary algorithms in which initial populations were used ranging from 200 to 500 rules. The best rule in this set returns an efficacy of 88.9%. Its hexadecimal code is 1461077C0F00CEBF1721047F0F333FBF. We will present some tests involving this rule and the best Juillé and Pollack's one in the end of this section.

We have prepared a simple genetic algorithm to find rules for DCT using the following parameters:  $Pop\hat{A}=\hat{A}300$ ,  $IC\hat{A}=\hat{A}300$  and  $Ng\hat{A}=\hat{A}300$  and the niche strategy of resource sharing used during the evaluation of population. No parameter-based heuristic was used in these initial experiments. Using this configuration, the best rule found for radius 3 CA rule has an efficacy about 82%. Subsequently, we used the same environment to search for radius 2 rules able to solve DCT. The best rule found in this search is 142149B7 and it has an efficacy at about 57.3%. It was used to generate the spatial-temporal pattern diagrams of Fig. 1. It is clear that this radius 2 rule does not use a sophisticated communication strategy as the radius 3 rule used to generate Fig. 2. Thus, as pointed by other researches, the 32-bits 1D CA rules do not have local connectivity enough to solve this problem with a reasonable efficacy.

### 4.4 Evolving rules for two-dimensional density classification task

#### 4.4.1 von Neumann neighborhood

Subsequently we turned to investigate the two-dimensional CA rule space formed by 32-bits rules with von Neumann neighborhood in DCT, as we have performed in ST. We started using 2D lattices formed by 12x12 bits and a genetic algorithm using the following parameters:  $Pop\hat{A}=\hat{A}200$ ,  $IC\hat{A}=\hat{A}200$  and  $Ng\hat{A}=\hat{A}200$  and the strategy of resource sharing. The best rule found in this search is 040F35BF and it has an efficacy at about 65.9% in 12x12 lattices. Therefore, comparing with the best

one-dimensional radius 2 rule found in our previous experiments one can see that the two-dimensional arrangement of the rules has facilitated the coordination of the information among the cells and a rule with the same length (32 bits) can have a better performance in this task than the rules using the one-dimensional arrangement. However, some works have investigated rules with Moore neighborhood to solve DCT in other lattice size: 21x21-bits. Therefore, we calculated the performance of the rule 040F35BF applied over 21x21-bits lattices and we obtained an efficacy of about 60.9%. Due to this decay in the performance from 12x12 to 21x21 lattices we decided to evolve GA again to directly find good rules to 21x21 lattices. GA was adjusted with the same configuration established in the previous experiment, except for the lattice size. The best rule found in this search is 111315FF and it has an efficacy of about 67.0% in 21x21 lattices, clearly overcoming the performance of rule 040F35BF in this lattice size. However, when we applied rule 111315FF in 12x12 lattices its efficacy decays to 43.9%. This unstable performance related to the lattice size motivated us to test both rules in different lattice sizes. These results are showed in Table 3, which presents the comparative performance of both rules from 4x4 to 21x21 binary lattices. For the smallest sizes (4x4 and 5x5), the rules were evaluated in all possible lattices, not only in samples of them, completely enumerating all possible initial lattices in each lattice size. For the other sizes, the rules were evaluated in samples of 1,000,000 lattices. An explanation is necessary about the tests using lattices with even sizes. Using such lattices, it is possible to have initial configurations in which the number of 0s is equal to the number of 1s. In this case, DCT is not well defined since it is not possible to decide the majority bit. However, in all other situations, the DCT is possible to be carried out. So, we just excluded the lattices with equal number of 0s and 1s from our tests. Looking over Table 3, except for the smaller sizes, in which rule 040F35BF overcome 111315FF both in 4x4 and 5x5 lattices; the efficacy of these rules oscillates from odd to even lattices. This behavior is specially observed in rule 111315FF, the one evolved in an odd lattice size. In the case of rule 040F35BF, this oscillation is not so severe, but it is also possible to observe in Table 3. Due to this oscillation, the rule evolved in an even lattice size (040F35BF) has a better performance in all the even lattice tested, while the rule evolved in an odd lattice size (111315FF) has a better performance in all the odd lattice tested, except for the smallest one 5x5. Such kind of oscillating behavior related to the parity of the lattice size was never observed, as far as we know, in experiments related to DCT. However, it was also observed by Wolz and de Oliveira in another computational task named parity problem (PP) [22]. PP is a task in which the objective is to get a CA rule to evolve a given initial configuration (IC) to an all-1 global one if the parity of the IC is odd and to an all-0 configuration otherwise. Using the same environment in which was possible to find the best radius 3 1D CA rule to DCT, they try to find good rules for PP and it was possible only for odd-sized lattices. Another interesting result concerning PP task is that for prime-number-sized lattices (in the range from 11 up to 31) they could find perfect rules, evidenced by full space tests (completely enumerating all possible ICs in the pre-specified lattice size). As in 2D regular lattices the prime-number-sized are impossible to obtain, we can not do this parallel with PP task.

Another characteristic noted in Table 3 is that the efficacy of both rules has a decay tendency from small to big sizes. It is explained by the fact that in all the experiments performed it was used the same number of time steps in the CA temporal evolution: 320. It is natural that using smaller lattice sizes this limit on CA time steps will be enough in a high number of tests than using bigger lattice sizes. Thus, this decay tendency is an expected behavior. What is not expected is, for example, the rule 111315FF returning 71% of efficacy using 9x9 lattices and below 44% of efficacy using the next lattice size: 10x10.

#### 4.4.2 Moore neighborhood: Initial experiments

Continuing with our investigation about DCT, we turned to evaluate the CA rule space formed by 512-bits rules with Moore neighborhood. This specification of DCT was first proposed by Morales, Crutchfield and Mitchell [12], which could find a reasonable rule for this task using a GA-based environment. Their best rule is presented in Table 4, named as MCM, and it has an efficacy of about 69% (as reported in [12]). This efficacy was evaluated using the same lattice size in which this rule was evolved: 21x21. Later on, this search was improved by Oliveira and Siqueira using a parameter-based heuristic and a resource sharing strategy and it was possible to find better rules [14]. The best rule found has 70.62% of efficacy (when tested in 12x12 lattices) and it presented in Table 4 named as OS. Recently, Wolz and de Oliveira [22] could find better rules for this specification of DCT using the same two-tier environment, in which they found the best 1D radius 3 CA rule for DCT. They could find several rules with efficacy above 82% in 21x21 lattices and the best one is also presented in Table 4, named as WO. However, due to the incompatible results found by these authors about OS rule, we have discovered that a mistake has been made in reference [13] because the best rule in this experiment (OS) was obtained and tested in 12x12 lattices, although in [13] these results were wrongly presented based on 21x21. Therefore, we decided to evaluate the three rules in both lattice sizes: 12x12 and 21x21. Surprisingly, the best rule in reference [22] (WO) is clearly better in 21x21 lattices but the best rule in reference [13] (OS) is better in 12x12 lattices, as presented in Table 4. The MCM rule has an efficacy similar to OS in both lattice sizes, but the first rule is overcome by the second one in both tests. Thus, we decided to evaluate OS and WO rules in a large number of lattices sizes, including full space tests (completely enumerating all possible ICs in the lattice size) in 4x4 and 5x5 sizes. This comparative analysis is showed in Table 5.

Observing the results of Table 5, one can observe a severe oscillating behavior of WO rule related to the lattice sizes: its performance oscillates between odd and even lattices, being that its efficacy is high in odd-size lattices and significant low in even-size one. The performance of this rule is very similar to what was observed with 32-bits rule 111315FF: both rules oscillate and each one has a better performance when applied in lattices with the same parity in which it has been evolved. The other rule evolved in the even-size lattice (OS) exhibits a lower oscillation as the observed for 32-bits rule 040F35BF.

Subsequently, we performed new experiments using a modest GA ( $Pop=100$ ,  $IC=100$  and  $Ng=100$ ) with resource sharing strategy (and no parameter-based heuristic) only to better investigate this “phenomena” related to the parity of the lattice size used during the evolution. In this new series of experiments a kind of adaptive fitness evaluation was used. It was established in [10] that the initial lattices used to evaluate the individuals should be generated by a uniform distribution for a better performance of the evolutionary search. However, the best rule found is evaluated in a sample of random lattices generated by a Gaussian distribution in the final of each GA run. A strategy that has presented a better performance is starting from the uniform distributed lattices and as the rules becoming better we change to sort a portion of the lattice using Gaussian distribution. As better as the rules turned on, as larger as the portion sorted by Gaussian distribution. This strategy was previously used in [4] and [2] to find rules with high performance on 1D DCT. However, it was not used in [13] to find OS rule in the experiments with 2D DCT. Another kind of adaptive evaluation was also used in [22] but it is not the same used here and in references [4] and [2] as the initial lattices do not start from a totally uniform sample; besides in [22] the lattices started with small sizes and they are smoothly shifted towards larger sizes during the evolution.

Table 3  
Results obtained using 2D CA rules 040F35BF and 111315FF (evolved in lattice size 12x12 and 21x21, respectively) evaluated in different lattice size.

Lattice Size	040F35BF (12x12)	111315FF (21x21)
4x4 (full space)	<b>75.921</b>	35.406
5x5 (full space)	<b>71.914</b>	70.698
6x6	<b>72.072</b>	41.865
7x7	69.157	<b>71.752</b>
8x8	<b>69.805</b>	43.908
9x9	67.167	<b>71.456</b>
10x10	<b>67.770</b>	43.490
11x11	65.559	<b>70.904</b>
12x12	<b>65.901</b>	43.872
13x13	64.115	<b>69.940</b>
14x14	<b>64.375</b>	40.537
15x15	63.389	<b>69.065</b>
16x16	<b>64.084</b>	41.526
17x17	62.076	<b>68.856</b>
18x18	<b>62.955</b>	41.097
19x19	61.796	<b>67.825</b>
20x20	<b>62.332</b>	40.841
21x21	60.913	<b>67.014</b>
BWLR-symmetry	0.875	0.625

We performed a first series of runs using 12x12 lattices. We performed a second experiment using only 21x21 lattices. Finally, a third experiment was performed evaluating two different sizes of lattices: 20x20 and 21x21. That is, each individual (CA rule) of the population is submitted to tests using both 20x20 and 21x21 lattices (50% of each one). Table 6 presents the efficacy of the best rules found in each experiment, evaluating them from 6x6 to 21x21 using 1,000,000 lattices for

each lattice size and using the entire space of 4x4 and 5x5 lattices (excluding the cases in which the number of 0s is equal to the 1s). The results of WO and OS rules were replicated here to aid their comparison with the new evolved rules.

Table 4  
Efficacy of rules WO [22], MCM [12] and OS [13] evolved in lattice sizes 12x12 and 21x21.

	MCM (21x21)	OS (12x12)	WO (21x21)
Hexadecimal code	0003606C00CD40 961117160A872 7284110110225 759E457F29D96 5379F6FBF7F17 3652BE45AB136 9913F88497B7C B3DF00723375 37FFD7FB38F97 FD575FDDD7	020D03110C1914 49230F4B1D85D 35577114900D5 91AF5CB756151 B1D91FFF1FF43 20100B2319357 F2F5C67778E5F 5BBB126E13F65 639718F076B7C F7CED75777	00000001001101 410B1514050D2 B775701011307 0305455700955 7D703957FF70B 1B45550513134 5177F475F077F 7F77FF57475D1 F7F6765BF3F3F BFFFFFF7F77
12x12	71,86%	<b>72,25%</b>	63,95%
21x21	67,99%	69,82%	<b>82,23%</b>

Looking over the results on Table 6 it is possible to observe that WO rule is the best rule in all the odd-size lattices and also in 20x20 lattices. The best rule evolved in 21x21 lattices is clearly inferior to WO in respect to these lattice sizes. It was not a surprise since the evolutionary environment used is much simpler and modest than the one used in reference [22] and we did not believe that it was possible to find a better rule in such situation. However, what have catch our attention was the fact that this rule, even being evolved by a simple and modest GA, overcomes WO rule in all the even-size lattices from 6x6 to 18x18. Besides, although this rule slightly oscillating between even-size and odd-size lattices, its oscillation is not as severe as the observed in WO rule. Comparing the rules evolved based on 12x12 lattices, one can see that OS rule has the best performance only in relation to the 4x4 and 6x6. In all the other sizes OS rule was overcome by the best rule evolved in our experiments using 12x12 lattices, including all the odd-size lattices. As pointed before, GA used in our experiments uses an adaptive evaluation not used in the GA that found OS [13] and it was executed using the same parameters:  $Pop\hat{A}=\hat{A}100$ ,  $IC\hat{A}=\hat{A}100$  and  $Ng\hat{A}=\hat{A}100$ . So, we believe that this new 12x12 rule is better than OS due to the adoption of the adaptive evaluation. Now comparing the performance of the rules evolved using only one lattice size (even or odd size) to the rule evolved using mixed lattices (even and odd-size), one can see that in general the 12x12 evolved rule is better in smaller lattices and the mixed sizes is better in the larger lattices. Thus, it is difficult to conclude which is the best rule between them. The final observation and possibly the most important about the experiments reported in Table 6 is that, although all the evolved rules present at least a slightly oscillation between odd and even size lattices, the unique rule that presents a severe oscillation

is WO rule.

#### 4.4.3 Moore neighborhood: Experiments using BWLR-simmetry

We started to asked ourselves if the specialized behavior observed in the last experiments is really necessary to a rule to have a good performance in this task or if is related to some special characteristic to the evolutionary environment used to obtain WO rule in [22].

Table 5  
Comparative performance using 2D CA rules WO [22] and OS [13]

Lattice Size	OS (12x12)	WO (21x21)
4x4 (full space)	<b>67.91%</b>	44.21%
5x5 (full space)	74.89%	<b>88.76%</b>
6x6	<b>75.32%</b>	52.04%
7x7	73.54%	<b>87.24%</b>
8x8	<b>74.00%</b>	56.87%
9x9	72.86%	<b>85.93%</b>
10x10	<b>72.96%</b>	60.46%
11x11	72.51%	<b>84.87%</b>
12x12	<b>72.25%</b>	63.95%
13x13	72.17%	<b>84.34%</b>
14x14	<b>71.79%</b>	66.95%
15x15	71.13%	<b>84.27%</b>
16x16	<b>72.15%</b>	69.83%
17x17	70.67%	<b>83.58%</b>
18x18	70.42%	<b>72.03%</b>
19x19	70.25%	<b>83.07%</b>
20x20	70.17%	<b>73.95%</b>
21x21	69.82%	<b>82.23%</b>

As pointed before, a complex two-level hierarchical evolutionary environment was used in to obtain both 1D and 2D DCT rules. However, in author's words *a new constraint was used which, in fact, turned out to be the main cause of the achieved success, is defined as the number of state transitions of a rule that respect the joint black-white, left-right transformations. In the case of good DCT rules, this means that the amount of BWLR-symmetry should be maximal.* Any CA rule can have three dynamical equivalent rules obtained using three possible transformations: complementary, reflection and complementary-plus-reflection (or black-white, left-right and joint-black-white-and-left-right transformations). Other works have shown that the symmetry of the bits related to the complementary transformation is related to the success of a rule to perform as specific task. In [4] it was observed that in the best DCT rules known until that moment the bits related to the complementary symmetry has an asymmetrical pattern. This observation was used in the evolutionary experiments reported in [4] to obtain a rule as good as the best rule known at that moment (JP rule discovered in [6]). This symmetry turned out to be important in another context, related to the application of CA chaotic rules to cryptography. It was observed that rules with the bits related to the complementary transformation in a totally symmetrical pattern (resulting in a palindrome



rule) have a high number of Garden-of-Eden states and they are not appropriate to be used in a ciphering method based on the pre-image calculus [17] and [18]. However, in Wolz and de Oliveira’s work the symmetry of the bits related to the complementary-plus-reflection transformation, named as BWLR-symmetry in [4], has revealed as the most significant to the success of a DCT CA rule. In fact, we calculated the value of this symmetry in the best rules published in [4] for 1D and 2D spaces and all the rules have this symmetry equal to 1 (in a range from 0 to 1). That is, these bits are totally symmetrical in these best rules, turning that each one of these rules does not have a complementary-plus-reflection equivalent rule as the same binary code is obtained when the transformation is applied. This information was used as a heuristic to guide the evolutionary search in [4] to the region of the best possible rules for DCT (both in 1D and 2D rule spaces). This heuristic was implemented as a repairing infeasible solutions procedure that was not totally explained in [4] but sure it have caused the discovery of almost all good rules having the BWLR-symmetry equal to 1.

Table 6  
Efficacies obtained using 2D 512-bits CA rules in different lattice size

Lattice Size	Evolved in even-size lattice		Evolved in odd-size lattice		Evolved in mixed-size lattice
	(12x12)	OS(12x12)	21x21	WO (21x21)	(20x20 and 21x21)
4x4 (full space)	54.74%	<b>67.91%</b>	67.40%	44.21%	67.64
5x5 (full space)	77.34%	74.89%	74.31%	<b>88.76%</b>	76.10
6x6	74.91%	75.32%	75.82%	52.04%	<b>76.19</b>
7x7	76.54%	73.54%	74.98%	<b>87.24%</b>	75.83
8x8	<b>76.56%</b>	74.00%	75.35%	56.87%	76.28
9x9	75.95%	72.86%	74.59%	<b>85.93%</b>	75.07
10x10	<b>75.65%</b>	72.96%	74.90%	60.46%	74.86
11x11	74.98%	72.51%	74.04%	<b>84.87%</b>	74.27
12x12	<b>74.97%</b>	72.25%	74.44%	63.95%	74.29
13x13	74.34%	72.17%	73.81%	<b>84.34%</b>	73.68
14x14	73.87%	71.79%	74.01%	66.95%	<b>74.08</b>
15x15	73.85%	71.13%	73.55%	<b>84.27%</b>	73.89
16x16	<b>73.76%</b>	72.15%	73.44%	69.83%	73.10
17x17	73.18%	70.67%	72.87%	<b>83.58%</b>	72.92
18x18	72.51%	70.42%	72.46%	72.03%	<b>72.75</b>
19x19	72.16%	70.25%	72.35%	<b>83.07%</b>	72.44
20x20	71.46%	70.17%	72.23%	<b>73.95%</b>	72.34
21x21	71.32%	69.82%	71.79%	<b>82.23%</b>	71.78
BWLR-symmetry	0.625	0.578	0.578	1.000	0.613

Aiming to investigated the influence of the usage of this parameter to guide the evolutionary search and specially trying to verify if it has any influence in the oscillatory behavior observed in WO rule, we first calculated the value of BWLR-symmetry associated to the six 512-bits rules used in Table 6 and in the two 32-bits rules used in Table 3. As one can see the value of BWLR-symmetry equal to 1 is not natural to find by GA and it was found only in rule WO once this value was pre-specified as desired in the evolutionary search. Thus, we modified our GA environment previously described (standard GA with resource sharing and adaptive

evaluation) to use a heuristic related to BWLR-symmetry to guide the evolutionary search. This information was incorporated not in the same way as in [4] as a repairing infeasible solutions procedure. We incorporated it in a similar way to what was done in the parameter-based heuristic described in [15] and [16]. The parameter BWLR-symmetry was the only one used in this heuristic with a desirable range defined as above 0.9. A series of GA runs was executed using this parameter-guided environment and with larger GA parameters:  $Pop = 300$ ,  $IC = 300$  and  $Ng = 300$ .

Table 7 shows the best rules evolved in one experiment with 12x12 lattices and in another one using 21x21 lattices. Although the 21x21 rule is worst than WO rule, it is clear that this performance in odd-size lattices is better the previous evolved 21x21 rule showed in Table 6. However, in the even-size lattices its performance presented a significant fall when compared to the 21x21 rule of Table 6, except for the 20x20 lattices. Comparing with WO rule, this rule presented better efficacies in the majority of even-size lattices. But what was more evidenced is that this rule also presents a severe oscillatory behavior between even and odd size lattice. Although this oscillation is more moderate than the one observed in WO rule, this rule was the one that returned the most expressive oscillation in our experiments. On the other hand, the best rule evolved using the BWLR-symmetry with 12x12 lattices returned a minor oscillation and a clearly performance improvement in relation to the previous 12x12 rule presented in Table 6: it overcomes the previous rule in all the tested lattice sizes. Besides, when compared with WO rule the new 12x12 evolved rule surpass the previous published one in all even lattice sizes tested while the WO is better in all the odd-size lattices. However due to its oscillatory behavior, the average value returned by WO rule considering all the tested sizes is significant lower than the new rule (73.59% against 81.86%) with a severe standard deviation (13.55 against 2.91). Applying the confidence interval we have 95% of confidence that the efficacy of the new rule in an arbitrary lattice is between 76.09% and 87.64%.

Concluding, the application of the symmetry-based heuristic together with the increase in GA parameters have improved the performance of the rules generated both in even-size than in odd-size lattices. However, this performance improvement was accomplished by an undesirable oscillatory behavior in the case of rules evolved based on odd-size lattice. In the case of rules evolved based on even-size lattice, the rules presents a more stable behavior.

Table 7  
Efficacies obtained using 2D 512-bits CA rules in different lattice size

Lattice Size	(12x12)	(21x21)	WO(21x21)
4x4 (full space)	<b>80.613</b>	26.599	44.21%
5x5 (full space)	85.888	80.925	<b>88.76%</b>
6x6	<b>87.177</b>	47.346	52.04%
7x7	84.863	81.173	<b>87.24%</b>
8x8	<b>85.453</b>	58.793	56.87%
9x9	83.476	80.496	<b>85.93%</b>
10x10	<b>83.849</b>	66.061	60.46%
11x11	82.830	79.764	<b>84.87%</b>
12x12	<b>82.780</b>	70.237	63.95%
13x13	81.969	79.030	<b>84.34%</b>
14x14	<b>81.465</b>	72.146	66.95%
15x15	80.671	78.391	<b>84.27%</b>
16x16	<b>80.458</b>	73.114	69.83%
17x17	79.659	77.338	<b>83.58%</b>
18x18	<b>79.217</b>	73.252	72.03%
19x19	78.409	75.940	<b>83.07%</b>
20x20	<b>77.944</b>	72.896	73.95%
21x21	76.838	74.919	<b>82.23%</b>
BWLR-symmetry	0.984	0.977	1.000

#### 4.5 Parity lattice size dependence in 1D DCT

As the use of even-sized lattice to test DCT is not common (in fact they were never previously cited in the literature as far as we know) due to the existence of lattices with equal number of 0s and 1s in such kind of lattices, we decided to perform a final test to evaluate if the previously published rules for the 1D radius CA present a similar kind of instability in respect to the parity of the sizes used to test them. Table 8 presents the efficacy results of two remarkable published one-dimensional radius-3 rules: JP and WO<sub>1D</sub>. JP rule published in [6] was evolved using a coevolutionary environment which do not use any parameter-based information; its hexadecimal code is 1451305C0050CE5F1711FF5F0F53CF5F and its BWLR-symmetry is 0.810. WO<sub>1D</sub> rule published in [22] was evolved using a two-tier environment in which the parameter BWLR-symmetry and the balancedness level of a rule (in a certain sense can be thought as the Langton's lambda parameter) were used to limit the search space. It is the best currently known rule in this CA specification for DCT (the most widely-studied) and its hexadecimal code is 1461077C0F00CEBF1721047F0F333FBF and its BWLR-symmetry is 1.

Table 8  
Efficacies obtained using 1D 128-bits CA rules in different lattice size

1D Lattice size	JP	WO <sub>1D</sub>
19	77.723%	67.998%
20	54.212%	59.103%
21	76.425%	64.985%
22	60.607%	67.145%
...	...	...
45	89.424%	90.278%
46	83.239%	85.856%
47	89.340%	90.316%
48	83.747%	86.306%
49	89.746%	91.422%
50	84.344%	87.078%
...	...	...
95	89.196%	90.789%
96	87.404%	89.893%
97	89.206%	90.794%
98	87.212%	89.924%
99	89.165%	90.750%
100	87.100%	90.012%
...	...	...
145	86.286%	88.791%
146	85.524%	88.583%
147	86.407%	88.775%
148	85.017%	88.411%
149	86.117%	88.459%
150	85.074%	88.062%
...	...	...
197	84.688%	73.069%
198	84.027%	72.660%
199	84.581%	72.059%
200	83.402%	71.933%
201	86.573%	74.458%

It is possible to point that the oscillatory behavior between even and odd-size lattices observed in some 2D rules can also be observed using the high performance CA rules evolved using the 1D version of the task. This oscillation even/odd is more notable in relative small sizes (for example, examining 19-bits to 22-bits lattices). However, it is a tendency of behavior even in the bigger lattices sizes. Another unexpected result that emerged in this last experiment is that although JP rule has been overcome by WO<sub>1D</sub> rule in the majority of lattice sizes investigated here (including the 149-bits lattices that is commonly used as a pattern to evaluate 1D DCT) it seems that JP rule exhibits a significant better performance when using bigger lattice sizes (above 190). It is important to say that this analysis was performed fixing the CA time steps in 320 and WO<sub>1D</sub> can have a slower communicating strategy to solve this task than JP rule that cause this severe decay.

## 5 Conclusions

In this work, we try to perform a more extensive inquiry over two of the most-studied computational tasks in CA context: synchronization task and density classification task. To do this, we evaluated the two computational tasks applying them to two different neighborhoods in both one-dimensional and two-dimensional cases. Using 1D lattices, we analyzed neighborhoods of radius 2 and 3 generating 32-bits and 128-bits rules, respectively. In 2D lattices, it was analyzed von Neumann and Moore neighborhoods generating rules of 32 bits and 512 bits, respectively.

By analyzing the four neighborhoods for the synchronization task, it was possible to confirm the existence of rules with 100% of efficacy for three of them: the 128-bits 1D radius 3 rules, the 32-bits 2D rules with von Neumann neighborhood and the 512-bits 2D rules with Moore neighborhood. However, for the 32-bits 1D radius 2 rules was not possible to reach 100% of performance and the efficacy limit of approximately 96% obtained in [8] was not overcome. This efficacy was measured in 12x12 bit lattices (resulting in 144 cells) and 320 time steps of CA temporal evolution. However, when the time steps was increased from 320 to 1000, it was possible to obtain 9 rules with efficacy greater than 99%, showing that these rules exhibited a slower communication strategy and they need more time to obtain the task convergence. On the other hand, increasing this time steps still more, it was possible to see that some lattices are not possible to synchronize; therefore, there are other attractors in the basins of attraction field of the evolved rules. A result that has caught our attention was the fact that rules with two-dimensional von Neumann neighborhood are relatively easy to found for this task returning 100% of efficacy. As in both cases (radius 2 1D and von Neumann 2D) the rules are represented by binary sequences of length 32 and the tested lattices have similar sizes (144 and 149) this result only can attributed to a better adaptability of this task to be executed in a two-dimensional arrangement. Although in both cases each cell can “see” only its proper state and the states of 4 neighbors, the two-dimensional rules solve the task with success. This difference of performance should be attributed to the maximum distance between two cells inside the lattice, which is smaller in a two-dimensional arrangement. However, if it was the case, a higher number of time steps would be enough to reach 100% of performance.

In DCT analysis we reached similar results, once it was not possible to find 32-bits 1D radius 2 rules with a behavior minimally driven to solve the task. Although returning a performance above 50% (approximate 57%), the best rule evolved here does not appear to have any particle-spread-based strategy, as observed in other rules found for this task, in the cases that the lattice are dominated by state 0 and the rule perform a correct classification, as the example of Fig. 1a. This behavior causes a high number of failures in lattices clearly dominated by state 0. On the other hand, a simple GA could be used to find 32-bits 2D rules using von Neumann neighborhood with a reasonable efficacy (67% in 21x21 bits) in which a particle-spread-based behavior can be observed. Therefore, the two-dimensional lattice really seems to aid the solution of this task. Another result that caught our attention related to this adaptability of DCT to the 2D space is that the best

516-bits 2D rules known for DCT have a worst performance than the best 128-bits 1D radius 3 rules currently known [22]. Thus, we believe that there is a lot of space to improve the two-dimensional rules using Moore neighborhood.

The analysis of 2D rules for DCT has highlighted a strong dependence of the efficacy of the published and evolved rules with the parity of the lattice size. Although DCT was not commonly studied using even-size lattices due to the existence of configurations in which the task is not well-defined, it is expected that any rule evolved for DCT has also a good performance in the even size lattice configurations in which the majority can be decided. Thus, such parity lattice size dependent behavior is not desirable and probably it was not observed before because in fact the previous studies focused only in odd-size lattices. However this parity dependence was already observed related to other task: parity problem (PP). Besides, this oscillatory behavior was also observed using good 1D radius 3 rules especially in the smallest lattices, although in a lower level. Thus, it is important to discover the reason of such dependence.

Finally, we want to purpose a final discussion about the dependence of the rule efficacy for any task with the CA configuration used to evolve the rule. In several situations, we noted efficacy decay when trying to apply an evolved rule in a different configuration: even/odd sizes, small/medium/larger sizes, low/high time steps, etc. In the last years, the huge effort employed in the search for rules that were better to solve specific configurations for a computational task led us to the development of more complex and efficient evolutionary environments, specific or general evolutionary strategies to improve the performance of the search and heuristics obtained starting from the observation of good rules previously published able to aid the search for better rules. However, does all this effort to improve the previous results would have led us to a situation in which highly specialized rules obtained for the configuration for which they were evolved? Thus, perhaps we have done a slight deviation in our investigations about CA computational ability in the sense that trying to understand how the CA compute, a rule to solve a specific task should be as general as possible and not so dependent on the CA configuration. In future experiments it might be interesting to use different CA configurations during evolution of rules, combining different possible scenarios for the evaluation of a rule solving a proposed task.

## References

- [1] Andre, D., Bennett III, F., Koza, J. *Discovery by Genetic Programming of a Cellular Automata Rule that is Better than any Known Rule for the Majority Classification Problem*, Proceedings of Genetic Programming, Stanford: Stanford University (1996).
- [2] Bortot, J. C., de Oliveira, P. P. B., Oliveira, G. M. B. *Multiobjective, heuristic evolutionary search in a cooperative environment leads to the best cellular automaton rule in the density classification task*, Proceedings of Simpósio Brasileiro de Redes Neurais (SBRN04) (2004), São Luis.
- [3] Das, R., Crutchfield, J., Mitchell, M., Hanson, J. *Evolving Globally Synchronized Cellular Automata*, Proc. of Inter. Conf. on Genetic Algorithms (1995).
- [4] de Oliveira, P.P.B., Bortot, Oliveira, G.M.B. *The best currently known cellular automata rules for density classification and the evolutionary mechanisms that led to them*, Neurocomputing, Amsterdam **70** (2006), 35-43.

- [5] Gacs, P., Kurdyumov, G.L. and Levin, L.A. *One-dimensional uniform arrays that wash out finite islands*, Probl. Perdachi. Inform. **14** (1978), 92-98.
- [6] Juillé, H. and Pollack, J. *Coevolving the Ideal Trainer: Application to the Discovery of Cellular Automata Rules*, Proceedings of Genetic Programming Conference **3** (1998), Madison.
- [7] Land, M. and Belew, R. *No Perfect Two-State Cellular Automata for Density Classification Exists*, Physical Review Letters **74**(25) (1995), 5148.
- [8] Mariano, A.S. and Oliveira, G.M.B. *Evolving one-dimensional radius-2 cellular automata rules for the synchronization task*, Automata-2008: Theory and Applications of Cellular Automata, ed: A. Adamatzky, R. Alonso-Sanz, A. Lawniczak, G. Juarez Martinez, K. Morita, T. Worsch, Luniver Press, (2008) 514-526.
- [9] Mitchell, M. *Computation in Cellular Automata: A Selected Review*, Nonstandard Computation. Weinheim: VCH Verlagsgesellschaft (1996).
- [10] Mitchell, M., Hraber, P. Crutchfield, J. *Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations*, Complex Systems **7** (1993), 89-130.
- [11] Mitchell, M., Crutchfield, J., Das, R. *Evolving Cellular Automata with Genetic Algorithms: a Review of Recent Work*, Proceedings of International Conference on Evolutionary Computation and Its Applications, Russian Academy of Sciences (1996).
- [12] Morales, F., Crutchfield, J., Mitchell, M. *Evolving two-dimensional cellular automata to perform density classification: a report on work in progress*, Parallel Computing **27** (2000), 571-585.
- [13] Oliveira, G.M.B and Siqueira, S.R.C. *Parameter Characterization of Two-Dimensional Cellular Automata Rule Space*, Physica D **217**(1) (2006), 1-6.
- [14] Oliveira, G.M.B and Siqueira, S.R.C. *Using dynamic behavior prediction to guide an evolutionary search for designing two-dimensional cellular automata*, VIIIth European Conference on Artificial Life (ECAL2005). LECTURE NOTES IN ARTIFICIAL INTELLIGENCE (ADVANCES IN ARTIFICIAL LIFE), Canterbury **3630** (2005), 491-499.
- [15] Oliveira, G.M.B., de Oliveira, P.P.B. and Omar, N. *Evolving solutions of the density classification task in 1D cellular automata, guided by parameters that estimate their dynamic behavior*, Proceedings of Artificial Life **VII** (2000), 428-436.
- [16] Oliveira, G.M.B., de Oliveira, P.P.B. e Omar, N. *Definition and applications of a five-parameter characterization of one-dimensional cellular automata rule space* Artificial Life, **7**(3) (2001), 277-301, MIT Press.
- [17] Oliveira, G.M.B., Macedo, H.B., Branquinho, A.A.B. and Lima, M.J.L. *A cryptographic model based on the pre-image calculus of cellular automata*, Automata-2008: Theory and Applications of Cellular Automata, ed: A. Adamatzky, R. Alonso-Sanz, A. Lawniczak, G. Juarez Martinez, K. Morita, T. Worsch, Luniver Press (2008), 139-155.
- [18] Oliveira, G.M.B., Martins, L.G.A., Alt, L.S., Ferreira, G.B. and Martins, M.V. *A Cellular Automata-Based Cryptographic Model with a Variable-Length Ciphertext* (preprint).
- [19] Oliveira, G.M.B., de Oliveira, P.P.B. e Omar, N. *Improving genetic search for 1D cellular automata, using heuristics related to their dynamic behavior forecast*, IEEE Conference on Evolutionary Computation (2001), 348-355.
- [20] Packard, N. "Adaptation toward the Edge of Chaos. Dynamic Patterns in Complex Systems" (1988), 293-301.
- [21] Sipper, M. *Computing with cellular automata: three cases for nonuniformity*, Physical Review E **57** (1998), 3589-3592.
- [22] Wolz, D. and de Oliveira, P.P.B. *Very effective evolutionary techniques for searching cellular automata rule spaces*, Journal of Cellular Automata **3** (2008), 289-312.