STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

WEIGHTED MAJORITY RULE FOR HYBRID CELLULAR AUTOMATA TOPOLOGY AND NEIGHBORHOOD

ANCA ANDREICA[(1)] AND CAMELIA CHIRA[(1)]

Abstract. Evolution rules for Cellular Automata (CAs) able to perform computational tasks which require global coordination highlight an interesting emergent behavior. CAs can generate this complex behavior starting from a simple initial configuration based on the local interaction of simple components that evolve according to some state change rule. However, the detection of rules that exhibit coordinated global information processing is a very challenging task highly important in the study of complex systems. In this paper, we propose a new weighted rule for a cellular automaton with hybrid topology and neighborhood in which the state of a cell changes according to the cell itself and both local and long-distance cells. In the proposed approach, each cell in the neighborhood has a different weight (determined using an evolutionary algorithm) in the decision of changing the state for the current cell. Computational experiments focus on the well-known density classification task for the one-dimensional binary-state CA. Results support a better performance of the proposed weighted rule compared to the standard majority rule applied to the same CA topology.

1. Introduction

Cellular Automata (CAs) represent useful and important tools in the study of complex systems and interactions. A cellular automaton is a system evolving in discrete time steps with a discrete spatial geometry (usually a regular lattice). The CA is specified in terms of rules that define how it changes and evolves in time. A global coordinated behavior results from the local interaction of simple components [25, 15]. The emergent behavior and computational

Received by the editors: March 30, 2013.

2010 *Mathematics Subject Classification.* 68-02.

1998 *CR Categories and Descriptors.* I.2.8 Computing Methodologies [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*.

Key words and phrases. evolutionary algorithms, density classification task, cellular automata.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

complexity of a system can be analyzed and better understood based on CA dynamics [2].

One of the most widely studied CA problems is the density classification task (DCT) [10, 21, 16, 18, 2]. This is a prototypical distributed computational task with the objective of finding the density most present in the initial cellular state. The discovery of rules exhibiting a high degree of global self-organization is not easily achieved. DCT is not a trivial task since coordinated global information processing must rise from the interactions of components with local information and communication.

Most existing studies [6, 3, 15, 9, 16, 20, 14, 11, 18] focus on developing algorithms able to find high performant rules for one-dimensional CAs with fixed neighborhood size. Each iteration, the change of a cell state depends on the cell itself and r local cells neighbors on each side. In this case, the neighborhood size is fixed to $2 * r + 1$ and the method searches for the best rule able to correctly classify potentially any initial configuration. Network-based CAs have also been investigated in the context of DCT [23, 24, 4, 5, 22]. The topology of the CA is given by a general graph and the neighborhood of each cell varies with the number of connecting nodes in the graph while the rule remains fixed (i.e. the majority rule). For network-based CAs, algorithms are designed to search for the graph topology that triggers the best neighborhood to be used in connection with a fixed rule for the DCT. Another topology proposed in [1] is a hybrid one in which the state change of a cell is allowed to be influenced by long-distance cells. Besides the cell itself and the cells in the local neighborhood, fixed distant cells contribute to the way in which the cell state is changed over time. This hybrid neighborhood has a fixed structure combining local and, to a certain extent, global information.

In this paper, we propose a new weighted rule that can be applied for the hybrid topology mentioned above. In the proposed approach, the local neighbors and the long-distance neighbors have different weights when computing the next state of a current cell. An evolutionary algorithm is developed to search for the best performing weighted rule for DCT based on the hybrid neighborhood structure. Computational experiments indicate a better performance obtained by the proposed weighted rule compared to the standard majority rule for hybrid CA topology and neighborhood.

The rest of the paper is structured as follows: section 2 briefly presents the density classification problem, section 3 describes the relevant lattice, network-based and hybrid CA topologies, section 4 presents the proposed weighted majority rule, section 5 includes the computational experiments and results, and section 6 contains the conclusions of the paper and directions for future work.

2. The Density Classification Problem

DCT is a challenging problem extensively studied due to its simple description and potential to generate a variety of complex behaviors.

The aim of DCT is to find a binary one-dimensional CA able to classify the density of 1s (denoted by ρ_0) in the initial configuration. If $\rho_0 > 0.5$ (1 is dominant in the initial configuration) then the CA must reach a fixed point configuration of 1s otherwise it must reach a fixed-point configuration of 0s within a certain number of timesteps. Most studies consider the CA length of $N = 149$ and the CA radius of $r = 3$ (which gives a neighborhood size of 7).

The CA lattice starts with a given binary string called the initial configuration (IC). After a maximum number of iterations (usually set as twice the size of CA), the CA will reach a certain configuration. If this is formed of homogeneous states of all 1s or 0s, it means that the IC has been classified as density class 1, respectively 0. Otherwise, the CA makes by definition a misclassification [20]. It has been shown that there is no rule that can correctly classify all possible ICs [13].

The performance of a rule measures the classification accuracy of a CA based on the fraction of correct classifications over ICs selected from an unbiased distribution (ρ_0 is centered around 0.5).

3. CA Topologies and Results for DCT

This section describes some relevant CA topologies, i.e. lattice, network and hybrid, and presents related existing rule detection methods and their results for DCT.

3.1. **Regular Lattice Topology.** A one-dimensional lattice of N two-state cells is used to represent the CA. The state of each cell changes according to a function depending on the current states in the neighborhood. The neighborhood of a cell is given by the cell itself and its r neighbors on both sides of the cell, where r represents the radius of the CA. The initial configuration of cell states (0s and 1s) for the lattice evolves in discrete time steps updating cells simultaneously according to the CA rule.

The regular lattice topology is engaged in most studies tackling DCT for 1D binary-state CA. Packard [19] developed the first evolutionary approach to detect rules for one dimensional binary state CA with a radius of 3. The potential of genetic algorithms for computational emergence in the density classification task has been extensively investigated by Mitchell et al [6, 3, 15, 9, 16, 20, 14]. The evolutionary approach to DCT proposed by Mitchell et al sets the fitness of a rule table as the fraction of correct classifications made over 100 randomly chosen initial configurations with uniformly

distributed density. The three main strategies discovered are default, block-expanding (a refinement of the default strategy) and particle revealing increasing computational complexity [3]. The best particle rule found by Mitchell et al [15] has a performance of 0.769.

Juille and Polack [11] used coevolutionary learning for the density classification task reporting good results for this problem (performance of 0.86). Oliveira et al [17] present a multiobjective evolutionary approach to DCT based on the non-dominated sorting genetic algorithm (NSGA). The algorithm is implemented in a distributed cooperative computational environment and is able to discover new rules with a performance of 0.8616. Wolz and Oliveira [26] proposed a two-tier evolutionary framework able to detect rules with a performance of 0.889. The approach integrates a basic evolutionary algorithm in a parallel multi-population search algorithm. To the best of our knowledge, the DCT rules presented in [26] are the ones with best performance discovered to date [18].

3.2. **Network Topology.** Some studies [23, 24, 4, 5, 22] consider a network-based topology for the CA where the cells can be connected in any way while the rule is the same for all cells.

Watts [24] studied the small-world graph version of the DCT: the rule is fixed and the performance of different small-world networks for DCT is evaluated. A small-world graph is constructed starting from a regular ring of nodes in which each node has k neighbors followed by a random rewiring procedure [23, 24]. The rule used for small-world network DCT is simple: at each time step, each node takes the state of the majority of its neighbor nodes in the graph (if the number of state 1s equals the number of state 0s in the neighbors list then the node is randomly assigned a state with equal probability between 0 and 1). Small-world networks have a performance of around 0.8 for the DCT with this fixed majority rule for 149 cells CA.

Tomassini et al [4, 5, 22] investigated network-based CAs for the density and synchronization problems. Spatially structured evolutionary algorithms are used to find the best performant network topology for DCT when the rule is fixed to the majority rule. An individual represents a network structure and the fitness is computed based on the fraction of ICs (out of 100 ICs generated anew for each individual) correctly classified by the majority rule based on the neighborhood given by the network. The best evolved network starting from initial regular rings has a performance of 0.823 (for 149 cells) while the result for random graphs as initial population is similar (performance of 0.821 of the best network). Similar results have been obtained by [8] with a simple evolutionary algorithm able to produce network topologies with high performance for DCT based on the majority rule.

3.3. **Hybrid Topology.** A hybrid topology has been proposed in [1] where a mixed induced neighborhood keeps invariable the number of neighbors. This topology works on a lattice based on the CA radius r and a parameter n referring to the number of long-distance cells allowed to contribute to the current hybrid neighborhood. In order to create the new topology of radius r, we start with a regular ring lattice of radius $r - n$. The other $2 * n$ nodes that are part of the hybrid neighborhood of a node i are long distance neighbors randomly chosen from the rest of the nodes, but following some rules that ensure the equilibrium of the neighborhood. This means that i always remains the central node of the neighborhood and the distance between node i and the long distance neighbors places half of them (n nodes) on the left hand side and the other half (n nodes) on the right hand side of i. Using this approach, a new topology which resembles a network topology but it is still very close to a regular ring lattice is obtained. This allows us to consider the same majority understanding as in the classical case of regular ring lattice topology. A detailed description of the proposed hybrid topology can be found in [1].

The potential of this topology with the corresponding notion of hybrid neighborhood is shown by the significantly better performance obtained in fewer CA iterations, compared to the classical approach that uses the regular ring lattice as topology. We should also note that the best performance obtained with the hybrid topology (0.79) is considerably higher than the performance obtained for regular lattice topology (0.64) and is very close to the performance obtained by CA based on small-world topologies (0.82).

4. New Weighted Rule for Cellular Automata with Hybrid Neighborhood

As already mentioned, the state of each cell in CA changes according to a function depending on the current states in the neighborhood. The neighborhood of a cell is given by the cell itself and its r neighbors on both sides of the cell. The initial configuration of cell states (0s and 1s) for the lattice evolves in discrete time steps updating cells simultaneously according to the CA rule.

In current existing approaches, each neighbor (including the cell itself) has the same vote weight when deciding which is the next state of the current cell. Indeed, whether we consider the regular lattice ring or the network topology for CAs, we can not differentiate between neighbors - the vote of each neighbor has the same weight. The hybrid topology described in the previous section allows us to introduce a new concept of majority rule, where different neighbors have different vote weights when deciding the next state of the current cell.

The hybrid topology involves the presence of two kinds of neighbors: local and far neighbors. The proposed rule gives different vote weights to local

neighbors and to far neighbors. Let us denote by w_l the weight assigned to local neighbors and by w_f the weight assigned to far neighbors. We identify two different scenarios that can be applied in this approach:

$$(i) 0 \leq w_l \leq 1, w_f = 1 - w_l$$

$$(ii) 0 \leq w_l \leq 1, 0 \leq w_f \leq 1$$

In the first scenario, the increase of one weight leads to the decrease of the other one, while in the second scenario, there is no relation between the weights of the two kinds of neighbors. In both scenarios, we compute the maximum weighted sum that can be obtained when each neighbor has the value 1. Let us denote by w_{max} the obtained value:

$$w_{max} = n_l * w_l + n_f * w_f,$$

where n_l represents the number of local neighbors and n_f represents the number of far neighbors.

Let s_i^t denote the state of cell i at timestep t. Let i_j, where $j = 1, 2r + 1$, represent the neighbors of cell i. The number of neighbors of a cell is $2 * r + 1$, out of which $n_l = 2*(r-n)$ are local neighbors and $n_f = 2*n$ are far neighbors, according to the hybrid topology.

For each cell i, the weighted sum of the neighbors states at timestep t, denoted by σ_i^t, is defined as follows:

$$\sigma_i^t = \sum_{j=1}^{2r+1} s_{i_j}^t * w_{i_j}$$

As indicated above, $s_{i_j}^t$ represents the state of cell i_j in the hybrid neighborhood at timestep t and w_{i_j} is the weight of neighbor i_j:

$$w_{i_j} = \begin{cases} w_l, & \text{if } i_j \text{ is a local neighbor} \\ w_f, & \text{if } i_j \text{ is a far neighbor} \end{cases}$$

The next state of a cell i is 1 if the weighted sum σ_i exceeds half of the maximum weighted sum:

$$s_i^{t+1} = \begin{cases} 1, & \text{if } \sigma_i^t \geq w_{max}/2 \\ 0, & \text{otherwise} \end{cases}$$

The proposed weighted rule is a very realistic model with many applications in real-world problems where different entities have different voting weights when deciding a next state in a system that could be modeled as a CA.

5. Experimental Results

Computational experiments consider a well studied version of the DCT problem: the one-dimensional binary-state CA of size 149 based on the radius of 3. For the hybrid topology we set $r = 3$ and $n = 1$ (i.e. 2 local neighbors on each side and 1 distant neighbor). The neighborhood size is 7 cells which leads to a rule size of $2^7 = 128$.

A simple evolutionary algorithm has been developed to detect rules for the DCT. A potential solution of the problem is encoded as a one-dimensional array of weights (one weight for the first proposed scenario and two weights for the second one). The initial population is randomly generated. The potential solutions are evaluated by means of a real-valued fitness function $f : X \to [0, 1]$, where X denotes the search space of the problem. The fitness function represents the fraction of correct classification over 100 randomly generated initial configurations. A relative fitness is used, as the set of initial configurations is generated anew for each generation of the algorithm. This way, solutions with high fitness in one generation and which survive in the next generation will be evaluated again using another set of 100 initial configurations. While the fitness is evaluated by using 100 uniformly distributed initial configurations, the performance of a rule is computed as the fraction of correct classifications for 10^4 randomly generated initial configurations. The initial configurations are generated in such a way that each cell has the same probability $\frac{1}{2}$ of being 0 or 1. The CA is iterated until it reaches a fixed-point configuration of 1s or 0s but for no more than $M \approx 2N$ time steps.

The individual resulted after each recombination will be mutated at exactly two randomly chosen positions. A weak mutation is considered, the probability of obtaining a different value for the chosen position being equal to the probability of obtaining the very same value. The algorithm is applied for 100 generations with a population size of 100, roulette selection, one point crossover with probability of 0.8, weak mutation with probability 0.2 and elite size of 10%.

TABLE 1. Performance obtained in 10 runs for $HybCA$, W_1HybCA and W_2HybCA

	$HybCA$	W_1HybCA	W_2HybCA
Best	0.7902	0.8202	0.8083
Average	0.74847	0.79475	0.76573
Std Dev	0.034297815	0.017707202	0.0374052
T-test p-value		0.001870	0.364680

The performance obtained with the new weighted majority rule (both scenarios) is compared to the performance obtained when using the classical rule table. It should be noted that the same evolutionary algorithm with the same parameters is used to evolve both standard and weighted rules in order to allow a direct comparison of results. Both standard and weighted rules use the same hybrid CA topology and neighborhood. The case of standard rules can be viewed as a simplification of weighted rules evolution where no weights need to be determined.

Table 1 presents the results obtained after 10 runs of the algorithm to detect a classical rule ($HybCA$) and for evolving a weighted rule (W_1HybCA for first scenario and W_2HybCA for the second scenario considered). The weighted rule triggers an improvement of performance from 0.7902 ($HybCA$) to 0.8202 for W_1HybCA and 0.8083 for W_2HybCA. The results are compared using the paired t-test with a 95% confidence interval. For the first scenario, the p-value is significantly smaller than 0.05, which indicates that the mean performance obtained when using the proposed weighted rule is notably better than that obtained when using a classical rule. The mean performance obtained based on the second scenario is higher compared to the classical rule but not significantly (according also to the p-value which is higher than 0.05 as shown in Table 1). The best performance obtained in 10 runs of the evolutionary algorithm is indeed more significantly improved when using the weighted rule in the first scenario (the evolved weight for local neighbors was 0.16 when the best performance was obtained). Therefore, we further consider this first scenario to perform a further analysis and report results.

A comparison that shows the efficiency of the proposed weighted rule is the number of iterations needed to correctly classify the initial configurations (for example, for a performance of 0.82, there are 8200 correctly classified initial

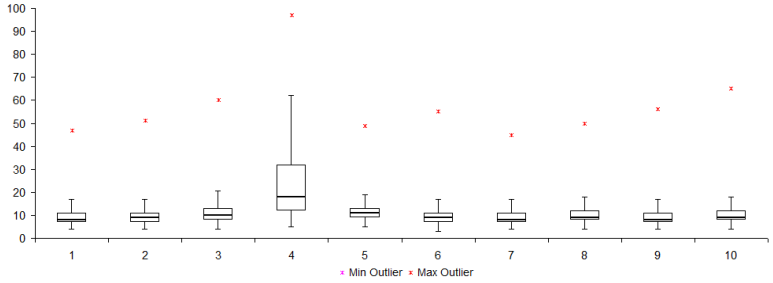

FIGURE 1. Number of iterations needed to correctly classify the initial configurations for HybCA, for each of 10 runs

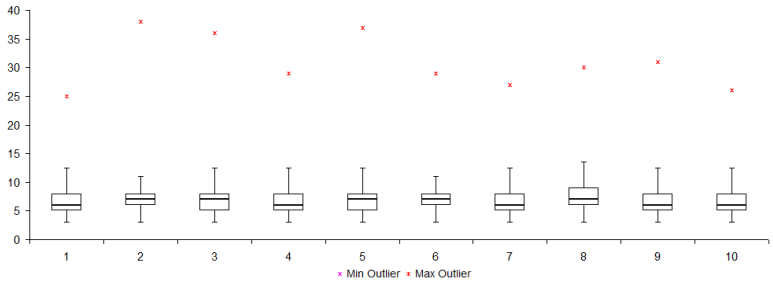

FIGURE 2. Number of iterations needed to correctly classify the initial configurations for W_1HybCA, for each of 10 runs

configurations and therefore 8200 observations of how many iterations were needed for those correctly classified initial configurations). Obtained results for 10 different runs are depicted in Figures 1 and 2. It can be easily observed that in the case of W_1HybCA the number of needed iterations is slightly smaller compared to $HybCA$. This means that the convergence is accelerated when using the proposed rule.

The obtained best rule is further tested against dynamic changes in order to evaluate the robustness of the hybrid topology when using weighted rules. Dynamic changes are understood as replacements of long distance neighbors with other randomly generated long distance neighbors. We apply 1000 random replacements for each neighborhood obtained by the 10 runs. The performance of the resulting CA is evaluated after each change and obtained results for all 10 runs are depicted in Figure 3.

As shown, there are no significant variations of the performance when the nodes neighborhoods are subject to dynamic changes. This is a good indicator of the topology robustness.

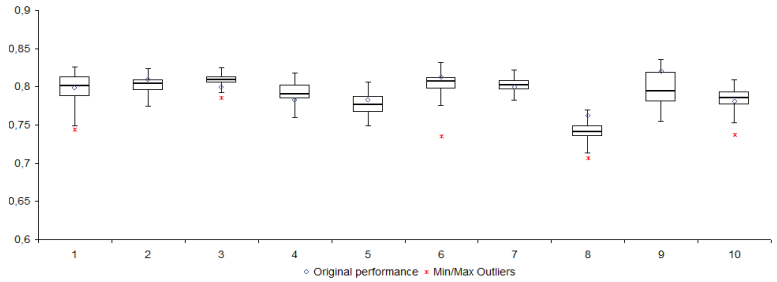

FIGURE 3. Performance obtained in 1000 dynamic steps for 10 runs of the algorithm

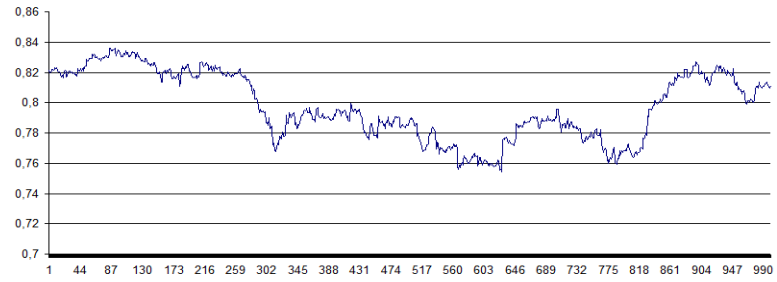

FIGURE 4. Performance obtained in 1000 dynamic steps for one run of the algorithm

Figure 4 presents the results obtained by a typical run of the algorithm (from the considered 10 runs). The original performance of the obtained rule before the topology being subject to dynamic changes was 0.8202. It can be noticed that the performance does not go below 0.7545, we even obtain better performances (the best one is 0.8362) and an average performance of 0.7977.

6. Conclusions and Future Work

A new weighted rule for a hybrid CA topology and neighborhood has been proposed. The cells in the hybrid neighborhood are allowed to have different weights (according to their local or far association to the current cell) in the decision of changing the current cell state. This approach enables a weighted rule application to the hybrid CA topology which has been investigated for the DCT in one-dimensional binary state CAs. Computational experiments emphasize that the weighted rules detected by an evolutionary algorithm have a good performance shown to improve that of standard rules

evolved for the same CA topology. The CA performance remains stable when dynamic changes are introduced in the neighborhood structure.

Future work focuses on investigating weighted schemes in connection with other CA rules and topologies, particularly for network-based CAs where weights associated with edges connecting two cells can be easily used in a weighted majority rule for graph topologies.

Acknowledgment

This research is supported by Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania.

References

[1] Andreica, A., Chira, C., Using a Hybrid Cellular Automata Topology and Neighborhood in Rule Discovery, submitted to IEEE Congress on Evolutionary Computation (CEC 2013).

[2] Chira, C., Gog, A., Lung, R. I., Iclanzan, D., Complex Systems and Cellular Automata Models in the Study of Complexity, Studia Informatica series, Vol. LV, No. 4, pp. 33-49 (2010)

[3] Crutchfield, J.P., Mitchell, M., The evolution of emergent computation, Proceedings of the National Academy of Sciences, USA 92 (23), (1995), pp.10742-10746.

[4] Darabos, C., Giacobini, M., Tomassini, M., Performance and Robustness of Cellular Automata Computation on Irregular Networks. Advances in Complex Systems 10: 85-110 (2007)

[5] Darabos, C., Tomassini, M., Di Cunto, F., Provero, P., Moore, J.H., Giacobini, M., Toward robust network based complex systems: from evolutionary cellular automata to biological models, Intelligenza Artificiale 5(1): 37-47 (2011)

[6] Das, R., Mitchell, M., Crutchfield, J.P., A genetic algorithm discovers particle-based computation in cellular automata, Parallel Problem Solving from Nature Conference (PPSN-III), Springer-Verlag (1994) pp.344-353.

[7] Ferreira, C., Gene Expression Programming: A New Adaptive Algorithm for Solving Problems, Complex Systems, Vol. 13, No. 2 (2001), pp.87-129.

[8] Gog, A., Chira, C., Dynamics of Networks Evolved for Cellular Automata Computation, Proceedings of the 7th International Workshop on Hybrid Artificial Intelligence Systems (HAIS 2012), Salamanca, Spain, Hybrid Artificial Intelligent Systems, Springer-Verlag, vol. 7208-7209 (2012) pp. 359-368.

[9] Hordijk, W., Crutchfield, J.P., Mitchell, M., Mechanisms of Emergent Computation in Cellular Automata, Parallel Problem Solving from Nature-V, Springer-Verlag (1998) pp.613-622.

[10] Juille, H., Pollack, J.B.: Coevolving the 'ideal' trainer: Application to the discovery of cellular automata rules. Genetic Programming 1998: Proceedings of the Third Annual Conference (1998)

[11] Juille, H., Pollack, J.B., Coevolutionary learning and the design of complex systems, Advances in Complex Systems, Vol. 2, No. 4 (2000), pp. 371-394.

[12] Koza, J.R., Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge (1992).
[13] Land, M., Belew, R. K., No perfect two-state cellular automata for density classification exists, Physical Review Letters, 74:25 (1995), pp. 5148-5150.
[14] Marques-Pita, M., Mitchell, M., Rocha, L., The role of conceptual structure in designing cellular automata to perform collective computation, Proceedings of the Conference on Unconventional Computation, UC 2008, Springer (Lecture Notes in Computer Science), 2008.
[15] Mitchell, M., Crutchfield, J.P., Das, R., Evolving cellular automata with genetic algorithms: A review of recent work, In Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96). Russian Academy of Sciences (1996)
[16] Mitchell, M., Thomure, M. D., Williams, N. L.: The role of space in the Success of Coevolutionary Learning. Proceedings of ALIFE X - The Tenth International Conference on the Simulation and Synthesis of Living Systems (2006)
[17] de Oliveira, P.P.B., Bortot, J.C., Oliveira, G., The best currently known class of dynamically equivalent cellular automata rules for density classification, Neurocomputing, 70:1-3 (2006), pp. 35-43.
[18] Oliveira, G.M.B., Martins, L.G.A., de Carvalho, L.B., Fynn, E., Some investigations about synchronization and density classification tasks in one-dimensional and two-dimensional cellular automata rule spaces, Electron. Notes Theor. Comput. Sci., 252 (2009), pp. 121-142.
[19] Packard, N.H., Adaptation toward the edge of chaos, Dynamic Patterns in Complex Systems, World Scientific (1988), pp. 293-301.
[20] Pagie, L., Mitchell, M.: A comparison of evolutionary and coevolutionary search. Int. J. Comput. Intell. Appl., 2, 1, 53-69 (2002)
[21] Tomassini, M., Venzi, M.: Evolution of Asynchronous Cellular Automata for the Density Task. Parallel Problem Solving from Nature - PPSN VII, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2439, 934-943 (2002)
[22] Tomassini, M., Giacobini, M., Darabos, C., Evolution and dynamics of small-world cellular automata, Complex Systems 15 (2005), pp. 261-284.
[23] Watts, D.J., Strogatz, S.H., Collective dynamics of 'smallworld' networks, Nature 393 (1998), pp. 440-442.
[24] Watts, D.J., Small Worlds: The Dynamics of Networks Between Order and Randomness, Princeton University Press, Princeton, NJ (1999)
[25] Wolfram, S., Theory and Applications of Cellular Automata, Advanced series on complex systems, World Scientific Publishing, 9128 (1986)
[26] Wolz, D., de Oliveira, P.P.B., Very effective evolutionary techniques for searching cellular automata rule spaces, Journal of Cellular Automata 3 (2008), pp. 289-312.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: [anca,cchira]@cs.ubbcluj.ro