Multiobjective Evolutionary Search for One-Dimensional Cellular Automata in the Density Classification Task

Gina M. B. Oliveira, José C. Bortot and Pedro P. B. de Oliveira
Universidade Presbiteriana Mackenzie
Rua da Consolação 896, 6° andar — Consolação
01302-907 São Paulo — Brazil
{gina, jbortot, pedrob}@mackenzie.br

Abstract

A key concern in artificial-life-oriented research in complex systems has been the relationship between the dynamical behaviour of cellular automata (CA) and their computational ability. Along this line, evolutionary methods have been used to look for CA with predefined computational behaviours, the most widely studied task having been the Density Classification Task (DCT). It has recently been showed that the use of an heuristic guided by parameters that estimate the dynamical behaviour of CA, can improve evolutionary search. On the other hand, an approach that has been successfully applied to several kinds of problems is the Evolutionary Multiobjective Optimization (EMOO). Here, the EMOO technique called Non-Dominated Sorting Genetic Algorithm is combined with the parameter-based heuristic, and successfullly applied to the DCT, suggesting a positive synergy out of using the two techniques in the search for CA.

Introduction

One of the greatest motivations for studying cellular automata (CA) is their ability to perform computations (Mitchell 1996). However, the comprehension about how these computations are carried out is still extremely vague, what has entailed various studies on methods to make CA programming possible. One of these possibilities is the use of evolutionary techniques to design CA that perform a predefined computation. The most widely studied CA task is the Density Classification Task (DCT) (Andre, Bennett, & Koza 1997; Mitchell, Hraber, & Crutchfield 1993; Juillé & Pollack 1998; Oliveira, de Oliveira, & Omar 2000). Even with evolutionary techniques, the high cardinality of CA rule spaces may become a serious obstacle to be skipped over in order to find CA that perform the desired computational task. This aspect turns the search slow and sometimes unproductive.

In (Oliveira, de Oliveira, & Omar 2000) a set of static parameters (i.e., directly derived from the CA rule table) was proposed, aiming at the reduction of the latter problem, as they are used as an auxiliary metric to guide the processes underlying the genetic search. Based on this set, it was possible to build a guide to a standard genetic algorithm (GA) to find CA rules for DCT. The parameter-based heuristic was incorporated into the GA in two aspects: in the fitness function, and in the genetic operators (of crossover and mutation). The fitness function of a cellular automaton rule was made by the weighed sum between a fitness component due to its efficacy in a sample of initial configurations (ICs), with a second fitness component that represents the bias due to the parameter-based heuristic (Oliveira, de Oliveira, & Omar 2000). Although this solution yielded an improvement on the genetic search in all the tasks studied, the weight of the heuristic in the rule evaluation interferes on this performance.

Instead of using any weighted sum to evaluate the rule, new experiments are reported here, where an evolutionary multiobjective approach is used. Accordingly, the heuristic and the efficacy in the IC sample are kept separate, as independent objectives to be followed by the genetic search.

This work is totally related to the idea of understanding the impact of the inherently local information processing of CA on their ability to perform a coordinated computation at the global level, as mediated by an evolutionary process. It is very much in tune with one of the alife open problems in (Bedau *et al.* 2000), namely, that we should be able to "*develop a theory of information processing, information flow and information generation for evolving systems*".

Evolutionary Multiobjective Methods

Several real-world problems involve simultaneous optimization of multiple objectives, so that it is not always possible to achieve an optimum solution in respect to all objectives, individually considered. In this kind of problem, there is a set of solutions better than all the other solutions in the search space (Srinivas & Deb 1994). This solution set is called the *Pareto optimum* or non-dominated solutions. Multiobjective evolutionary methods try to find this solution set by using each objective separately, without aggregating them as an unique objective (Coello 1999).

The Nondominated Sorting Genetic Algorithms (NSGA) method was proposed by Srinivas and Deb (1994) and is based on the concept of non-domination. Suppose that there are N objectives $f_1, f_2, \ldots f_N$ to be simultaneously optimised. A solution A is said to be dominated by another solution B, if B is better than A in relation to at least one of the objectives f_i, and is better or equal to A in relation to the other objectives. Two solutions A and B are non-dominated in relation to each other if A does not dominate B and B does not dominate A. The Pareto optimum is the set of non-dominated solutions considering the entire search space.

The basic difference of NSGA in relation to a simple GA is the way in which the individuals are evaluated: in order to obtain the fitness value of an individual, the fitness components associated with each objective involved in the problem are used to rank the individuals according to their degree of domination over the others in the population. A classification is performed based on several layers of non-domination. Initially, all individuals in the population that are non-dominated are separated as the first, outer layer, and a (dummy) fitness value is assigned to them whose role is simply to characterise their degree of domination over the others in the population, as mentioned above; for the outer layer, the highest fitness value is assigned as the individuals in it exhibit the highest degree domination (the actual value assigned is proportional to the population size, but other details do exist which are being omitted here). Then, the remaining individuals are classified again, also based on the non-domination criterion, and the second layer is formed with a (dummy) fitness lower than the first one. This process continues until all individuals are classified in their respective layers. A stochastic remainder proportional selection is used so that the outer the layer an individual is in, the likelier its chance to reproduce. The other steps of the NSGA are very similar to the simple GA.

CA Dynamics and Parameterisation

Through the analysis of the dynamic behaviour exhibited by CA, Wolfram (1984) proposed a qualitative behaviour classification. Later on, Li and Packard (1990) proposed a refinement in the original Wolfram classification that divides the rule space into six classes: Null, Fixed Point, Two-Cycle, Periodic, Complex and Chaotic.

The dynamics of a cellular automaton is associated with its transition rule. In order to help forecast the dynamic behaviour of CA, several parameters have been proposed, directly calculated from their transition table (Langton 1990; Li & Packard 1990; Binder 1993; Oliveira, de Oliveira, & Omar 2001). A set of five parameters was used in (Oliveira, de Oliveira, & Omar 2001): two of them were chosen from among those already published, and three new ones. In the experiments involving the DCT task described in the next section, four parameters from that set were used: Sensitivity (Binder 1993), Absolute Activity, Neighborhood Dominance and Activity Propagation (Oliveira, de Oliveira, & Omar 2001). All of them have been normalized between 0 and 1, for one-dimensional CA with any radius.

Computational Tasks and CA Evolving

CA have the potential of executing computations in a non-standard fashion. Various investigations have been carried out on the their computational power, with concentrated efforts in the study of 1D CA capable of performing computational tasks. The most widely studied CA task is the Density Classification Task (DCT) (Packard 1988). In this task the objective is to find a binary 1D CA that can classify the density of 1s in the Initial Configuration: if the initial lattice has more 1s than 0s, the automaton should converge to a null configuration of 1s; otherwise, it should converge to a null configuration of 0s. Once a computational task is defined, manual programming is difficult and costly, and exhaustive search of the rule space becomes impossible, due to its high cardinality. A solution is the use of search methods, particularly evolutionary methods (Mitchell, Hraber, & Crutchfield 1993; Oliveira, de Oliveira, & Omar 2000). Packard (1988) was the first to publish results using a GA as a tool to find CA rules. Other evolutionary techniques were used to find such kind of rules (Juillé & Pollack 1998; Andre, Bennett, & Koza 1997). In a recent work (Oliveira, de Oliveira, & Omar 2000), a simple GA environment was modified so as to incorporate an heuristic based on the forecast parameters selected and then used to find DCT rules. First, parameter value regions where good rules should be more likely to occur were obtained by calculating the parameter values for some published CA rules. This information was used as an auxiliary metric to guide the processes underlying the GA search. A simple GA was adapted so as to incorporate the parameter-based heuristic in two aspects. First, the fitness function F of a rule was defined as a weighed composition between the heuristic-based component (Hp) and the fitness component derived from the actual performance of the rule (F_{IC}) in the attempt to solve the DCT as in Equation (1).

$$F = F_{IC} + \rho \times Hp \quad (1)$$

The parameter-based heuristic is coded as a function that returns a value between 0 and 100 for each rule, depending on the values of the its parameters: Hp returns 100 if all parameter values match the ranges of the published rules; otherwise, the value returned decreases linearly as the parameter values became increasingly away from those ranges. All parameters contribute equally in the calculation of Hp. The function F_{IC} also returns a value between 0 and 100, acccording to the rule efficacy in solving 100 different initial configurations. Finally — and crucially, to the point being made in this paper — ρ is the weight that establishes the influence of the heuristic component in the overall rule fitness. The second aspect in which the heuristic information was used was in that it allowed the definition of biased genetic operators of reproduction and mutation: in order to select the crossover point and the rule table bits to be mutated, N_{CM} attempts were made; among them, those that generated rules with high Hp value were selected. In (Oliveira, de Oliveira, & Omar 2000), it was used $\rho = 40\%$, $N_{CM} = 10$ and the insertion of the parameter information managed to improve the performance of the rules found for the DCT.

Number of individuals	100
Number of generations	100
Number of ICs per generation	100
Number of ICs at the final evaluation of the run	10,000
Number of GA runs	500
Elitism rate	20%
Crossover selection	Random selection out of the elite
Mutation rate	1% per bit
CA radius	3
Number of CA cells	149
Number of CA steps	300

Table 1: Specification of the simple GA experiments.

DCT Experiments

Simple GA experiments: ρ influence

Since an analysis of the effect of weight ρ has not been done in (Oliveira et al., 2000), this is what we go about now, in the context of DCT. For this matter, a series of experiments were performed, with varying values of ρ from 0% (corresponding to Mitchell *et al.*, (1993)), up to 100%. The specification of the GA and CA environment is presented in Table 1, and the results of the experiments are shown in Table 2, where each row corresponds to one experiment. The efficacy of the GA run was measured by testing the performance of the best rule found, at the end of the run, in the classification of 10^4 random initial configurations. The columns 2 through 7 in Table 2 display the percentage of runs in which the efficacy of the best rule found was within the corresponding interval. The last column shows the average efficacy of the ten best rules found in each experiment. One can also observe that all experiments with the parameter-based heuristic yielded superior results in comparison with the experiment without the heuristic; however, it is clear that an adequate value of ρ is necessary to improve the search. Within the values tested, the best experiment relied on $\rho = 40\%$, as used in (Oliveira, de Oliveira, & Omar 2000). Nevertheless, a question remains of whether there may be another ρ value, which might yield better results and whether that value would be adequate for other parameter ranges and other computational tasks.

Multiobjective experiments

In a second stage, instead of using the weighed sum of Equation (1), a multiobjective approach was used; the parameter-based heuristic is still introduced in crossover and mutation in the same way as in (Oliveira, de Oliveira, & Omar 2000). However, the fitness is no longer given by the composition of the two separated objectives F_{IC} and Hp from Equation (1): instead of establishing any *a priori* heuristic weight, the multiobjective dynamics is expected to define the relative importance of the objectives during the search.

A multiobjective environment was implemented after the NSGA method, with a major modification referring to the selection method: instead of the stochastic remainder proportional selection, elitism of 20% was used, as in Mitchell *et al.*, (1993), with the crossover pairs being randomly selected directly from the elite. The elite classification was based on the dummy fitness of each individual, after the non-domination classification. The results of the multiobjective experiment are presented in Table 3, referred to as MO, also obtained out of 500 runs. It replicates some results from Table 2: the GA experiment without the heuristic, called WH, as well as the experiment with $\rho = 40\%$, called $\rho40$. The efficacy of the best rule found in all the runs is presented for each experiment. Figure 1 shows the ten best rules efficacies. It should be observed that the MO experiment behaved in a similar way to the $\rho40$, which was the best experiment with the simple GA. In fact, MO is only slightly better than $\rho40$.

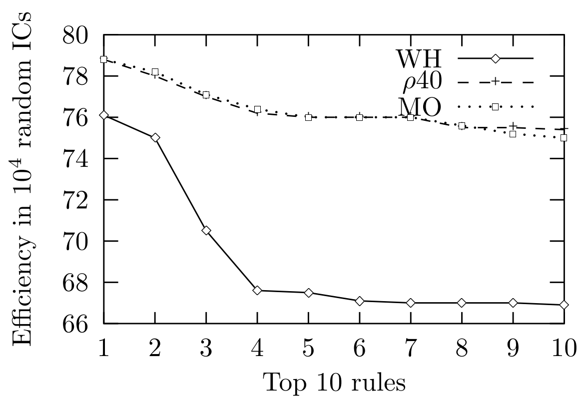

Figure 1: Efficacy of the top 10 rules in simple GA and NSGA experiments, with population of 100 individuals.

ρ(%)	(0,55)	(55,60)	(60,65)	(65,70)	(70,75)	(75,80)	Top10
0	8.2	0.2	49.8	41.2	0.4	0.2	69.26
10	0.0	0.0	18.0	79.6	2.2	0.2	74.12
20	0.0	0.2	16.4	81.4	1.4	0.6	73.94
30	0.0	0.0	12.6	86.4	0.8	0.2	71.27
40	0.0	0.2	12.8	83.8	0.8	2.4	76.49
50	0.0	0.0	10.4	88.2	0.8	0.6	73.02
60	0.0	0.2	17.0	81.4	0.8	0.6	72.96
70	0.0	0.4	17.8	79.4	1.2	1.2	75.49
80	0.0	0.2	16.2	80.6	1.4	1.6	76.26
90	0.0	0.2	13.4	83.4	1.6	1.4	76.20
100	0.0	0.0	13.2	85.0	0.8	1.0	74.42

Table 2: Experiments with the simple GA.

	Best Rule	(0,60)	(60,65)	(65,70)	(70,75)	(75,80)	Top10
WH	75.98	8.4	49.8	41.2	0.4	0.2	69.26
ρ40	78.65	0.2	12.8	83.8	0.8	2.4	76.49
MO	78.94	0.4	17.8	79.2	1.0	1.6	76.57

Table 3: Results obtained with the simple GA and NSGA experiments, with 100 individuals in the population

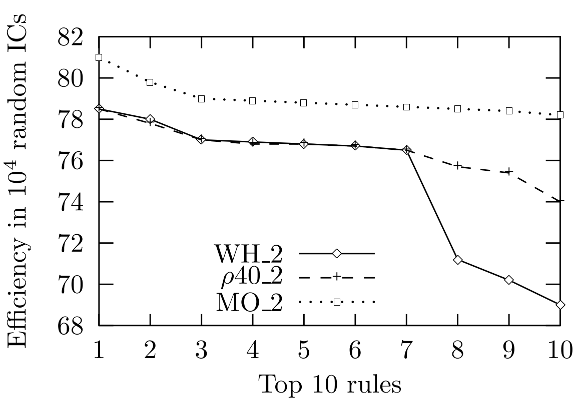

Figure 2: Efficacy of the top 10 rules in simple GA and NSGA experiments, with population of 200 individuals.

Subsequently, additional experiments were performed, so that the search could run freer, the idea being to check whether the NSGA could better explore the search space in less stricter conditions. The following parameters of the GA were used: 200 individuals in the populations, 200 initial configurations for testing the rules at each generation, and 1000 generations per run. One experiment was performed with the GA without the parameter-based heuristic, named WH_2. A GA was also used but with the parameter-based heuristic incorporated as in Equation (1) with $\rho = 40\%$; this was called ρ40_2. Finally, the experiment MO_2 was performed with the NSGA adapted with the heuristic information. All experiments were composed of 200 GA runs. Table 4 and Figure 2 show the results. Comparing with the corresponding experiments with 100 individuals in Table 3, all experiments improved with the more flexible parameter settings. The multiobjective experiment is also better than the others. But notice that now, it is more evident that the multiobjective experiment performed much better in relation to the non-multiobjective approach with $\rho = 40\%$.

Final Remarks

The experiments have shown that the multiobjective solution is a good approach for the incorporation of the parameter-based heuristic we have used in previous works, as way to help automatic CA programming. The results obtained with the multiobjective approach are clearly at least as good as the one derived from the adequate choice of a weight to balance the role of heuristic, with the advantage that no actual choice has to be made.

The lesson the experiments discussed here are bringing forth is that the notion of Pareto dominance seems to be prevailing also in the context of the parameter-based heuristic in CA search. However, it is still premature to draw a final conlusion on this issue. In fact, although the heuristic information was considered as an independent objective, it is composed of four different parameters, each of them contributing equally for the fitness component Hp (through a simple average). However, it is possible that in other tasks the individual guidance role of each parameter may have distinct effectiveness over the search. The next step in our investigation is to break the heuristic in different objectives, associated with each individual parameter, thus allowing the evolutionary multiobjective algorithm to directly handle the

	Best Rule	(0,60)	(60,65)	(65,70)	(70,75)	(75,80)	Top10
WH_2	78.73	17.6	77.8	0.9	2.8	0.0	71.76
ρ40_2	78.86	10.5	83.0	2.0	4.5	0.0	76.62
MO_2	81.16	4.0	83.8	5.2	6.4	0.6	78.76

Table 4: Results obtained with the simple GA and NSGA experiments, with 200 individuals in the population

resulting multi-directed search.

Acknowledgments

G.M.B.O. and P.P.B.O. are grateful to Univ. Mackenzie for their support, and CAPES, FAPESP and MACK-PESQUISA for funding.

References

Andre, D.; Bennett, F.; and Koza, J. 1997. Evolution of intricate long-distance communication signals in cellular automata using programming. In Langton, C., and Shimohara, K., eds., *Artificial Life V*. Cambridge, Mass.: MIT Press.

Bedau, M.; McCaskill, J.; Packard, N.H. Rasmussen, S.; Adami, C.; Green, D.; Ikegami, T.; Kaneko, K.; and S., R. T. 2000. Open problems in artificial life. *Artificial Life* 6:363–376.

Binder, P. 1993. A phase diagram for elementary cellular automata. *Complex Systems* 7:241–247.

Coello, C. 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems* 1(3):269–308.

Juillé, H., and Pollack, J. 1998. Coevolving the "ideal" trainer: Application to the discovery of cellular automata rules. In *Proceedings of Genetic Programming Conference 3*.

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

Li, W., and Packard, N. 1990. The structure of elementary cellular automata rule space. *Complex Systems* 4:281–297.

Mitchell, M.; Hraber, P.; and Crutchfield, J. 1993. Evolving cellular automata to perform computations. *Physica D* 75:361–391.

Mitchell, M. 1996. Computation cellular automata: a selected review. In Gramss, T., ed., *Nonstandard Computation*. Weinheim: VCH Verlagsgesellschaft.

Oliveira, G.; de Oliveira, P.; and Omar, N. 2000. volving solutions of the density classification task in 1D cellular automata, guided by parameters that estimate their dynamic behavior. In Bedau, M. A.; McCaskill, J. S.; Packard, N. H.; and Rasmussen, S., eds., *Alife VII*, 428–436. Cambridge, Mass.: MIT Press.

Oliveira, G.; de Oliveira, P.; and Omar, N. 2001. Definition and applications of a five-parameter characterization of one-dimensional cellular automata rule space. *Artificial Life* 7(3):277–301.

Packard, N. 1988. Adaptation towards the edge of the chaos. In Kelso, J.; Mandell, A.; and Shlesinger, M., eds., *Dynamic Patterns in Complex Systems*. World Scientific. 293–301.

Srinivas, N., and Deb, K. 1994. Multiojective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 221–248.

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