THE AGGREGATION MECHANISM FOR AGENT-BASED EVOLUTIONARY COMPUTATION

Krzysztof Cetnarowicz * **Rafał Dreżewski** *

* *Department of Computer Science,*
AGH University of Science and Technology,
Kraków, Poland
{cetnar,drezew}@agh.edu.pl

Abstract:
Evolutionary algorithms have some indispensable features, which limits their applicability in the case of selected problems. The key concept of the agent-based evolutionary computation paradigm is the decentralization of the evolutionary processes. Such decentralized models of evolutionary computation have some interesting features, which are absent in "classical" evolutionary algorithms—among them there are the possibilities of introducing in a very natural and coherent way new mechanisms and operators. One of such operators—the aggregation—is presented in this paper. Also, there are presented results of preliminary experiments with the evolutionary multi-agent system with aggregation mechanism for time series prediction.

Keywords: agent-based evolutionary computation, aggregation operator, prediction

1. INTRODUCTION

Evolutionary algorithms (EAs) are computational intelligence techniques based on principles of Darwinian model of evolutionary processes (Bäck *et al.*, 1997). They have been widely, and with great successes, applied to a wide variety of optimization and adaptation problems. However, there exist some indispensable features of EAs that limits their applicability in the case of some problems. It seems that most of these problems result from the following assumptions on which most of the "classical" evolutionary algorithms models are based:

- The evolutionary processes are centralized and one common algorithm (process) is used to realize the selection and to generate offspring and next generation populations.
- Individuals are simplified to genotypes, what makes it impossible for them to act in the environment, interact with it (including other individuals), and make any decisions that could influence the evolutionary processes.
- The set of evolutionary operators is limited (recombination, mutation, etc.)—this seems to limit the possibilities of modeling some of the important features of real evolutionary processes.
- Mechanisms like species formation, co-evolutionary interactions between species and sexes, social relations are difficult (or even impossible in some cases) to realize.
- There are no resources that individuals could compete for. Instead of this some mechanisms like fitness sharing are introduced,

which only implicitly model the competition for limited resources.
- There are no mechanisms of adapting the number of individuals in the population to the difficulty level of the problem being solved.

All the mentioned above features are the cause of some limitations of EAs in the case of selected problems, like for example multi-modal optimization, multi-objective optimization, adaptation in dynamic environments etc.

Agent-based evolutionary computation paradigm results from the attempts to decentralize the evolutionary processes and to formulate models of evolutionary computation that are closer to the real evolutionary processes observed in nature. This resulted in the formulation of *evolutionary multi-agent system (EMAS)* model. In the case of such system three basic mechanism, which are responsible for initiating and maintaining evolutionary processes, exist: agents are able to reproduce, die, and there exist resources in the environment for which agents compete and which are needed for all their activities (Cetnarowicz *et al.*, 1996). The research on niching and speciation techniques for EMAS model resulted in the formulation of the general model of *co-evolution in multi-agent system (CoEMAS)* (Dreżewski, 2003). This model includes additionally the notions of species, sex and relations between species and sexes in evolutionary multi-agent system. These mechanisms can serve as a basis for creating techniques of maintaining useful population diversity and speciation in systems based on CoEMAS model. Computational systems based on CoEMAS model has already been applied with promising results to multi-modal optimization (Dreżewski, 2006), and multi-objective optimization (Dreżewski and Siwik, 2006).

Such decentralized models of evolutionary computation have some very interesting features which are absent in "classical" EAs:

- Individuals (agents) are no more the passive objects of the evolutionary processes, but they are also involved in these processes, thus we can easily model and introduce co-evolutionary interactions and social relations between agents of different types, species, and sexes.
- Agent-oriented architecture of such systems allows for designing hybrid systems (using many different computational intelligence techniques within the framework of one coherent model), and including user preferences in the computational systems—agents representing user preferences can be introduced into such systems.
- The process of evolution is completely decentralized (agents as the evolving individuals, the absence of centralized selection mechanism) what results in the relaxation of synchronization constraints of the computations, and, as the effect of this, decreasing the overhead resulting from the synchronization mechanisms.
- The possibility of introducing—in a very natural way—such mechanisms and evolutionary operators, which are very difficult to be introduced and used in the case of "classical" EAs. These include for example the aggregation operator (which is the main subject of this paper), escape (migration) mechanism, mechanism of resource sharing which can be used instead of fitness sharing mechanism, which model only in implicit way the competition for limited resources in EAs.
- The possibility of modeling allopatric speciation on the basis of spatial structure of the *EMAS* system, and the sympatric speciation on the basis of co-evolutionary interactions between species and sexes in the *CoEMAS* system.
- The mechanism of auto-adaptation of the system to the type of problem being solved and the difficulty level of the problem—for example the population size can adapt to the problem or to the changing environment conditions.

In this paper the agent-based evolutionary system with aggregation operator is presented. The proposed operator is preliminary assessed with the use of prediction in the changing environment problem. Presented results show that the application of the aggregation operator leads to the emergence of social relations between agent and to the improvement of obtained results of the problem being solved.

2. AGGREGATION AND ESCAPE OPERATIONS

The agent–environment relation is the basic reason that forces agents to participate in the evolution process. If state of this relation is not satisfied for particular agent, it can choose one of following actions:

- agent may change itself—adapting itself to the conditions of the environment with the use of mutation and crossover operations,
- agent may change the environment with, among others, the use of aggregation and escape operations.

The evolution process in which autonomous agent take part, may be realized with the use of following operators that affect single agent or fixed group of agents. In the agent environment mutation, crossover and reproduction operators have the same form as presented in (Fogel *et al.*, 1966; Goldberg, 1989; Michalewicz, 1996) with such an exception that the evolution process of each agent has its own (individual for each agent) characteristics, it depends on agent's decisions, and it takes place according to its own independent cycle.

The idea of aggregation operation may be considered as a creation (by agents) of new environment in which agents act. A group of agents (such that conditions of actual environment do not suit them) make an agreement, which goal is to take over the control over the part of the existing environment. Thus agents change the parameters of the controlled environment, which we mentioned above, to make it better adapted to their requirements. After the creation of new environment a group of agents, mentioned above, owing to specialization and cooperation, maintain desirable parameters in the environment created by them. This group (when it is seen from outside) act like a new agent with new, characteristic features that arose owing to the aggregation operation. To sum up, the aggregation operator makes it possible to change the relation agents-environment with the change of environment's parameters.

The second operator that makes it possible to change the relation agents-environment with the change of environment's parameters is the escape operator. Let us make an assumption that the evolution process takes place in several environments, and that agents can migrate among these environments. The evolution processes that take place in each of these environments differ in some range of their parameters from each other. If in one of the mentioned environments the agent is created as a result of mutation, crossover or aggregation operation that is not well adapted to this environment, it may migrate toward different environment (with different characteristic parameters), where it can act better than in the previous environment. Then it may start there a new population of agents with valuable characteristic features.

2.1 Evolution centers as a method of evolution process control

The introduction of aggregation and escape operators makes possible to control the evolution process, among others, through the mediation of organizing of, so called, evolution centers.

Let us consider the process of realization of aggregation operator (fig. 1), which consists of the following stages:

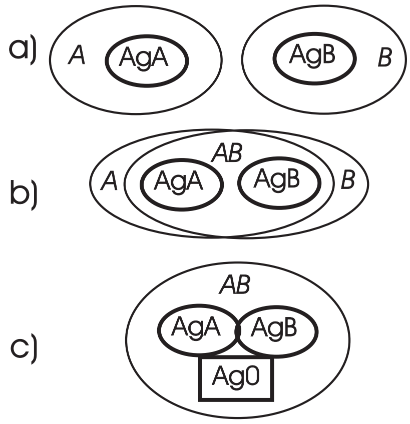

Fig. 1. The principle of functioning of the aggregation operator

(1) Let two parameters characterize the particular environment: A and B (values of A and B $\in \{0, 1\}$). There are agents AgA and AgB in this environment. Agent AgA (agent AgB) has the ability of such influence its environment that it can maintain the value of parameter A (B) equal to one in its neighborhood. Agents A and B require (prefer) the values of both parameters (A and B) equal to one in their neighborhood.
(2) Owing to the ability to move within the environment agent AgA may stay near the agent AgB (and similarly agent AgB may stay near the agent AgA).
(3) Agents AgA and AgB make an agreement and decide to aggregate together and create a new environment (with values of parameters satisfying requirements of both agents — $A = 1$ and $B = 1$) by taking control over the part of existing environment.
(4) As a result of aggregation the new agent $Ag0$ is created, which maintains the information about the configuration of agents A and B. The group of agents AgA, AgB, $Ag0$, and the part of environment that they control (with values of parameters A and B equal to one) constitute the new agent. Agent $Ag0$ keeps (for example encoded in its genes) information that is required to the reproduction of this aggregated agent.

With the use of aggregation and escape operators we can consider the organization of "evolution center", in which it is possible to control the values of environment parameters what implies that we can supervise the course of the evolution process. Let us consider the example presented above, in which the realization of aggregation operator is completed with the application of remaining evolution operators and the control of environment parameters. Let us make an assumption that there exist such agents of type X in the environment that they can not set the environment parameters

A and B in their neighborhood (i.e. set values $A = 1$ and $B = 1$). At the same time agents of type X prefer the environment which parameters A and B are set ($A = 1$ and $B = 1$). Agent of type X may obtain (as a result of mutation and crossover operations) the ability to set either the parameter A ($A = 1$) or B ($B = 1$) in its neighborhood (but not the ability to set both of them). Then it becomes the agent of type A or B. Let us make an assumption that the part of environment is selected — the evolution center. We can control this evolution center from outside and change the values of parameters A and B (setting their values to 0 or 1). The example of supervising the evolution center may be considered as following steps:

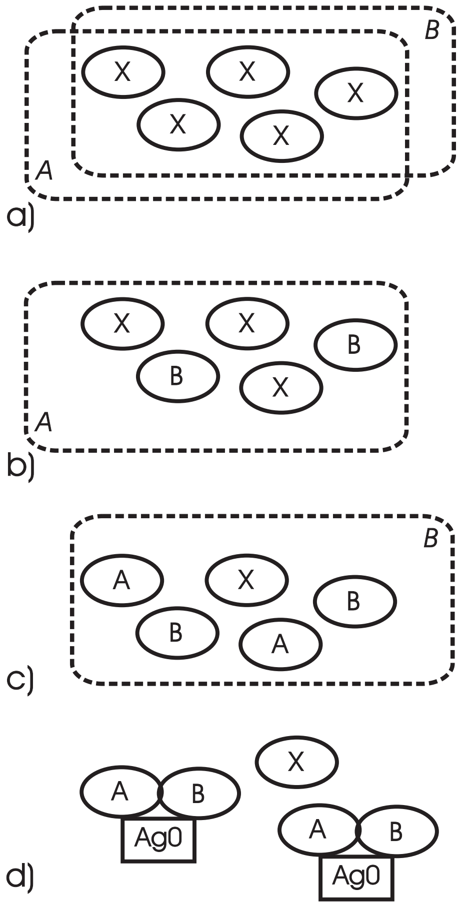

Fig. 2. The principle of functioning of the evolution center

- Parameters A and B are set ($A = 1$ and $B = 1$) in selected areas of evolution center. Agents of type X gather in this area (fig. 2a).
- Parameter A is turned off periodically. This causes that some agents of type X obtain (as a result of mutation and crossover operations) the ability to set the parameter A ($A = 1$) in their neighborhood. Thus they became type A agents (fig. 2b).
- After some time the parameter A is turned on ($A = 1$), and parameter B is turned off ($B = 0$) periodically. This causes that some agents of type X obtain (as a result of mutation and crossover operations) the ability to set the parameter B ($B = 1$) in their neighborhood. Thus they became type B agents (fig. 2c).
- After some time parameters A and B are both turned off ($A = 0$ and $B = 0$). Owing to the fact that agents of type A and agents of type B remain within the same area and they neighbor each other they can form the aggregates AB (as a result of aggregation operation). These aggregates do not depend on the fact that parameters A and B are turned on or off, what implies that they can live in any environment described above (fig. 2d).

Owing to the application of the idea of evolution centers (and the introduction of aggregation and escape operators) we may obtain the possibility of supervising the evolution process by changing the values of environment parameters, or rather, to say it more precise, by selecting the local sets of parameters.

3. EVOLUTIONARY MULTI-AGENT SYSTEM WITH AGGREGATION OPERATOR

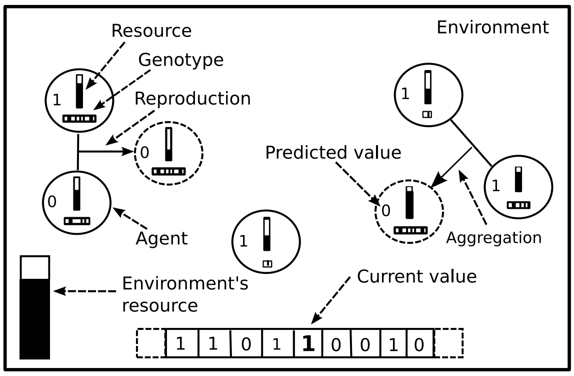

Fig. 3. EMAS with aggregation mechanism

The evolutionary multi-agent system with aggregation operator is presented in fig. 3. The system is based on the general model of evolution in multi-agent system—see (Cetnarowicz *et al.*, 1996; Dreżewski, 2003). In such systems evolving agents are located within the environment and can reproduce and die. There exists resource within the system which is possessed by agents and the environment. Each time step the environment gives agents some of the resource in such a way that better fitted agents are given more resource than the others.

In the particular system described in this paper the agents try to predict the next value of the pseudo-random time series consisted of zeros and ones. The current fitness of the agent is the result of the latest prediction. The agents which predicted correctly are given some of the resource (which is renewable) and those which did not are not given the resource at all. Each time step the agents lose some of their resources. When the agent runs out of resources it dies and is removed from the system.

Each gene of the agent's genotype is responsible for making prediction. The actual prediction of the agent is the value proposed by the majority of its genes. The i-th gene takes the i-th value (let us denote this p) from the history of values that appeared in the time series and on its basis tries to predict the value that will appear in the next time step. There are ten types of genes that utilize different mechanisms in order to predict the next value:

(1) returns NOT p;
(2) returns p;
(3) returns alternately 0 and 1;
(4) always returns 1;
(5) always returns 0;
(6) returns pseudo-random number (0 or 1);
(7) returns the values from the series consisted of k zeros and k ones alternately;
(8) returns the values from the series consisted of k zeros and m ones alternately;
(9) returns the values from the series like: 010011 00011100001111..., the number of zeros and ones in sub-series is increased until it reaches n and then the number of zeros and ones in sub-series is decreased.
(10) returns the values from the series like above but after the number of elements in sub-series reaches the maximal value n it is not decreased but starts again from one.

When the agent has enough resources to perform reproduction it tries to find the partner for reproduction. When it succeeds the new agent is created with the use of clone, recombination (one point crossover is used) and mutation (in which the worst gene is mutated—replaced by different type of gene selected randomly) operators. Each parent gives the child some of its resources.

When the agent predictions are not good enough, the amount of its resource decreases and it tries to aggregate with another agent. The agent which initiates the aggregation process takes genes from another agent and all its resources. These genes are then connected with the genotype of the agent which initiates the process of aggregation. Also the opposite process is allowed—agent may decide to split into two agents. All these operations are performed with some probability and require the appropriate amount of resource.

4. EXPERIMENTAL RESULTS

In this section some results of preliminary experiments with aggregation operator are presented. Because of space limitations only selected results are shown. In the fig. 4 results of typical experiment are presented. The values of parameters in this experiment were as follows: the length of history buffer was set to 16, the time series had finite length and was cyclically repeated, the maximal number of genes was set to 16, the probability of mutation was 0.01, the probability of recombination was 0.001, the probability of reproduction was 0.01, the probability of aggregation was set to 0.01, the probability of splitting the aggregate was 0.001. The decisions within the aggregate were made via voting of agents (within each agent being part of aggregate the decisions were made on the basis of the predictions made by individual genes—the final prediction was the prediction generated by majority of genes). It can be seen in the fig. 4 that with such configuration the application of the aggregation operator led to the increasing of the correct prediction probability from 66% to over 90%.

Table 1. Average results for different types of pseudo-random time series

Type of pseudo-random time series	Av. number of correct predictions	Av. number of correct predictions for best aggregates	Av. number of correct predictions for single agents
Cyclic	78.3%	81%	60.33%
Non-cyclic	59.12%	60.8%	49.2%

In the table 1 average results from experiments with different types of pseudo-random time series are presented. These are average results from five (in the case of non-cyclic time series) and three (in the case of cyclic time series) experiments carried out with the different values of the mentioned above parameters. Here it can also be seen that the aggregates had generally higher rate of correct predictions than single agents. These advantages of aggregates are more distinct in the case of cyclic time series, but they also appear in the case of non-cyclic time series.

5. CONCLUSIONS

In this paper the aggregation operator for agent-based evolutionary computation was presented. The aggregation operator is one the examples of new operators which can be easily introduced in

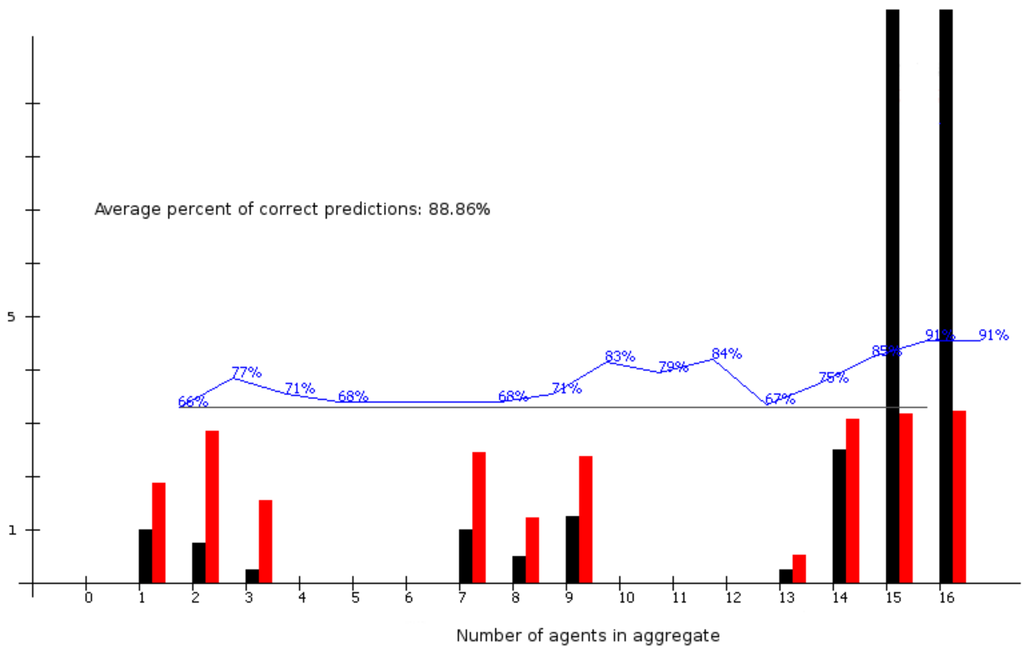

Fig. 4. Results of typical experiment. X axis shows the number of agents within the aggregate, the left bar shows the number of aggregates with the given number of agents, and the right one shows the amount of resources possessed by the given group of aggregates. The lower line shows the average percent of correct prediction made by single agents, and the upper one shows the average percent of correct predictions made by aggregates with the given number of agents

the evolutionary multi-agent systems. The aggregation processes lead to the emergence of social relations between agents and can be be very useful especially in the case of dynamic environments and problems.

The system presented was applied to the prediction of time series values problem. As the presented preliminary results show, the aggregation operator caused that agents were cooperating in the process of predicting, teams (aggregates) were formed and the results obtained were generally better than in the case of single predicting agents.

Future research will be focused on the application of evolutionary multi-agent systems with aggregation operator to dynamic problems. The future plans also include the artificial life simulations focused on the emergence of complex social relations resulting from simple interactions between agents, and the formation of new species through aggregation and co-evolutionary interactions.

REFERENCES

Bäck, T., Fogel, D. and Michalewicz, Z., Eds. (1997). *Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press.

Cetnarowicz, K., M. Kisiel-Dorohinicki and E. Nawarecki (1996). The application of evolution process in multi-agent world to the prediction system. In: *Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS 1996)* (M. Tokoro, Ed.). AAAI Press. Menlo Park, CA.

Dreżewski, R. (2003). A model of co-evolution in multi-agent system. In: *Multi-Agent Systems and Applications III* (V. Mařík, et al., Ed.). Vol. 2691 of *LNCS*. Springer-Verlag. Berlin, Heidelberg.

Dreżewski, R. (2006). Co-evolutionary multi-agent system with speciation and resource sharing mechanisms. *Computing and Informatics* **25**(4), 305–331.

Dreżewski, R. and L. Siwik (2006). Multi-objective optimization using co-evolutionary multi-agent system with host-parasite mechanism. In: *Computational Science — ICCS 2006* (V. N. Alexandrov, et al., Ed.). Vol. 3993 of *Lecture Notes in Computer Science*. Springer-Verlag. Berlin, Heidelberg.

Fogel, L. J., A. J. Owens and M. J. Walsh (1966). *Artificial Intelligence Through Simulated Evolution*. John Wiley. Chichester, UK.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer -Verlag.