

Rafal DREZEWSKI
Institute of Computer Science
University of Mining and Metallurgy, Poland
drezew@agh.edu.pl

The Application of Escape and Aggregation Evolution Operators to the Evolution Process in Prediction System

Abstract: Classical evolution programs have the following features (limitations), which seem to be very unrealistic from the biological point of view: 1) the process of evolution is centralized, 2) beings taking part in the evolution process are reduced to a chromosome, what implies that they cannot act independently, possibilities of rivalry and cooperation are limited, there are no social relations etc., and 3) only limited number of basic evolution operators such as *reproduction*, *mutation* and *crossover* are used. On the other hand the agent paradigm and its application to the design of decentralized and distributed systems arises. Evolution process realized in the multi-agent systems introduces following new possibilities: 1) avoiding limitations of the classical evolution programming, and 2) adaptation toward changing environments of the population of agents. In this paper the application of *Evolutionary Multi Agent Systems (EMAS)* technology to a 0-1 random time sequence prediction system is presented. In addition to the classical evolution operators such as *reproduction*, *mutation* and *crossover* two new operators: *escape* and *aggregation* are introduced. The *escape* operator enables migration of the agents towards different environments. The *aggregation* operator enables the emergence of social relations between agents. Also, the results of simulation experiments carried out with the SWARM-based evolutionary prediction system are discussed.

Introduction.

Classical (we use word “classical” in order to distinguish our concepts from such as presented in [5, 6]) evolution programs have the proud history of more than thirty years of research and application. Successful applications of genetic algorithms are known in many areas such as: function optimization, game theory, medicine, etc. [5, 6]. However results of applications are not fully satisfactory in some cases, although many improvements and modifications were introduced.

Classical evolution programs have the following features (or rather we should say: limitations) which seem to be very unrealistic from the biological point of view:

1. General common algorithm is used in the process of selection and reproduction of beings taking part in the evolution process. This implies that the whole evolution process is centralized,
2. Beings taking part in the evolution process are reduced to a chromosome, what implies that they cannot act independently, possibilities of rivalry and cooperation are limited, there are no social relations etc., and
3. Only limited number of basic evolution operators such as *reproduction*, *mutation* and *crossover* are used in the evolution process.

These limitations lead to problems with applying such biological mechanism as sexual reproduction and species formation, to take only two more obvious examples. The biological literature on these subjects is very extensive but there is no significant progress in applying such mechanisms to the construction of evolution programs.

Evolution process realized in the multi-agent world offers us new possibilities such as:

1. In the process of developing of our system we can make use of existing models of multi-agent systems. This have effect that beings (now we can call them agents) that participate in the evolution process, environment in which they act, agent-agent and agent-environment relations are well defined,
2. Evolution process is decentralized and is performed with no common cadence.
3. Agents can interact independently with each other and with environment. This cause that emergence of social relations between agents is now possible.
4. Process of selection is decentralized because agents can compete for limited resources of the environment.
5. Last but not least, by introducing evolution process into multi-agent system we provide agents the best mechanism of the adaptation toward changing environment.

The technology of *Evolutionary Multi Agent Systems (EMAS)*, that arises as a result of introducing the evolution process into the multi-agent world, can be treated as a new approach to the construction of evolution programs and a new way of developing multi-agent systems.

The application: Evolutionary multi-agent prediction system.

In this section we will describe sample application of the *EMAS* technology to the 0-1 time sequence prediction system. We use the model of the multi-agent system based on the concept of the M-Agent architecture. Detailed description of this model can be found in [1, 2].

In this system the main goal of the population of agents is to predict the changes of the environment. In the environment a parameter $\alpha \in \{0, 1\}$ is defined. Variations of the parameter α in discrete moments of time may be represented by the binary sequence $x(t)$, where $x(t)$ is the value of α in the time t . Value of the parameter α is available for all the agents acting in the environment. Each agent tries to predict the value that the α will take in the time $t+1$. So the i -th agent generates the binary sequence y_i , such that $y_i(t) = x(t + 1)$.

The structure of the environment.

The whole environment $V = (E, A, C)$, where: E is the space in which agents remain, A is a set of agents that act in E , and C is a set of agent-agent and agent environment relations [1, 2]. The space $E = (R, T)$.

- Resources available for all the agents are parameter α and energy P_e , so in our case $R = \{\alpha, P_e\}$.
- The topology T of the environment is graph. Every node of this graph has connections with its eight neighbors.

The internal structure of the agent.

In the considered system the agent consists of one or more cells. The cell consists of the finite automaton, information about its age and statistics. Each cell makes its predictions with the use of finite automaton, which input/output language consists of symbols 0 and 1. The finite automaton plays the role of the chromosome of particular cell [4, 5, 6].

There are four parameters that show the quality of prediction made by particular agent:

- $\Psi_{ik}^{j(0)}$ - Probability of good prediction connected with the transition through 0 from the j -th state of the k -th cell of i -th agent.
- $\Psi_{ik}^{j(1)}$ - Probability of good prediction connected with the transition through 1 from j -th state of the k -th cell of i -th agent.
- $\Psi_i(t)$ - Probability of good prediction made by the i -th agent.
- Ψ_{\max}^i - Maximal probability of good prediction made by the i -th agent during its lifetime.

The evolution operators.

The evolution process involves *reproduction*, *mutation*, *crossover*, *aggregation*, and *escape* operators:

- In the *reproduction* process an existing one creates a new agent. Chromosomes and other parameters of the new agent are equal to its parent's.
- *Mutation* operator may alter output symbol and transition between particular states.
- *Crossover* operator mixes homologous chromosomes of two agents.
- The *aggregation* operator serves as a mechanism of emergence of social relations between agents. It allows agents to form aggregate that consists of more than one cell.
- The *escape* operator allows agents to migrate toward different environments.

Intellectual profile of an agent.

The agent can be described from two points of view that correspond with two profiles: intellectual and energetic. Intellectual profile of the autonomous agent is [1, 2]: $a = (M, Q, S, I, X, L, m, q, s)$.

The model of the environment of i -th agent in the time t is

$$m_i(t) = \langle (Y_i(t), \Omega_i(t)), (Y_i(t+1), \Omega_i(t+1)), \dots \rangle, \quad (1)$$

where:

$$Y_i(u) = \{y_{i1}(u), \dots, y_{in}(u) : n \text{ is the number of cells}\},$$

$$\Omega_i(u) = \left\{ \Psi_{i1}^{S_1^u(x_1(u))}, \dots, \Psi_{in}^{S_n^u(x_n(u))} : n \text{ is the number of cells} \right\}, \text{ for}$$

$$u = t, t + 1, \dots.$$

$y_{ik}(u) = t_{ik}^{S_k^u}(x_k(u))$ is the output connected with the transition of the k -th finite automaton through $x_k(u)$, S_k^u is the state of the k -th finite automaton in the time u ,

$$x_k(u) = \begin{cases} x(u) & \text{for } u = t \\ y_{ik}(u-1) & \text{for } u > t \end{cases}, \text{ for } k = 1, \dots, n.$$

After applying the strategy realization operator X the agent's model of the environment is

$$m'_i(t) = \langle (y_i(t), \omega_i(t)), (y_i(t+1), \omega_i(t+1)), \dots \rangle, \quad (2)$$

where: $y_i(u) = y_{ik}(u)$ for $u = t, t + 1, \dots$, k is such that

$$\omega_i(u) = \Psi_{ik}^{S_k^u(z(u))} = \max \left\{ \Psi_{i1}^{S_1^u(z(u))}, \dots, \Psi_{in}^{S_n^u(z(u))} : n \text{ is the number of cells} \right\}$$

$$z(u) = \begin{cases} x(u) & \text{for } u = t \\ y_i(u-1) & \text{for } u > t \end{cases}.$$

The goal of this profile is to make correct predictions:

$$q(m_i(t), m'_i(t)) = \begin{cases} 1 & \text{when } y_i(t) = x(t+1) \\ 0 & \text{when } y_i(t) \neq x(t+1) \end{cases}. \quad (3)$$

Now the time is incremented and the observation operator I generates the model $m_i''(t) = \langle (Y_i(t), \Omega_i(t)), (Y_i(t+1), \Omega_i(t+1)), \dots \rangle$ (by getting the value of $x(t)$ from the environment). This model is the initial model for the next iteration.

The adaptation operator $L = \{L_m\}$ (operator L_m changes M - the set of models) is consisted of all the evolution operators.

Energetic profile of an agent.

Energetic profile of i -th agent is represented by the real value $P_i(t)$. The main goal of the agent in this profile is to maximize the value of $P_i(t)$. In order to realize this goal the energetic strategy S_e is realized by applying one of the following operators: *reproduction*, *aggregation*, *escape*. Every action (such as move or reproduction) costs some energy. Energy may be obtained only through correct prediction.

The agent's decision concerning every action (reproduction, aggregation, and direction of the move) depends on actual level of its energy. The energy of the agent

also serves as a mechanism of elimination of the agents with low level of fitness to the environment conditions (i.e. making incorrect predictions) from the system.

The total energy of the system $P(t)$ consists of actual energies of each agent and energy of the environment $P_e(t)$. This parameter prevents excessive grow of the total number of agents.

The simulations' results.

In all these experiments the population of agents had to predict the values of α parameter taken from the 0-1 random sequence. This sequence was repeated periodically giving as a result infinite sequence. Because of the stochastic nature of the experiments all the results are average values of five simulations. The simulation system used in these experiments was implemented with the use of SWARM system [7].

The simulations with the aggregation operator.

First goal of the simulations was to investigate what (if any) improvements the *aggregation* operator would cause in the evolution process.

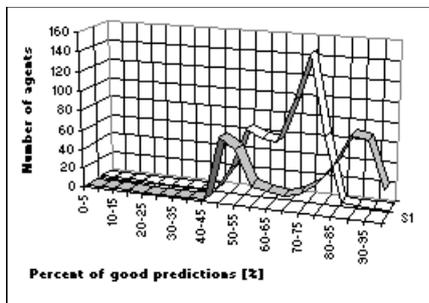


Figure 1 Percent of good predictions' distribution in the last step of simulation. Darker line shows results with *aggregation*.

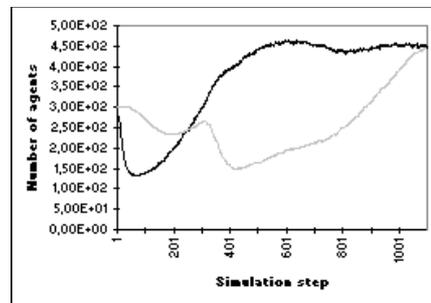


Figure 2 Number of agents. Darker line shows results with *aggregation*.

Figure 1 shows the percentage of correct predictions' distribution in the population of agents in the last step of the simulation. If only classical evolution operators were active there were no agents with 80-100% of correct predictions. Majority of the population of agents had 75-80% of correct predictions. When the *aggregation* operator was active, majority of agents had 90-100% of correct predictions.

Figure 2 shows the changes of the total number of agents during the simulation. We can observe the very fast decline of the number of agents in the case of the simulation made with the *aggregation* activated. The decline is much faster than in the case when only classical operators were active. This effect is caused by very intensive aggregation process in the first stage of simulation. But after short period of time the number of agents rapidly grows and reaches a balance state.

In the case when only classical operators are activated the period of time that is needed to approach a balance state is much longer. Results obtained with the use of the *aggregation* operator are comparable to those presented in [3].

The simulations with the *escape* operator.

The goal of these simulations was to investigate how the *escape* operator could improve the results.

In all these simulations there were three environments. In one of these environments agents had to predict the values from the random sequence that was four times longer than the maximal number of states of the finite automaton. Only in this envi-

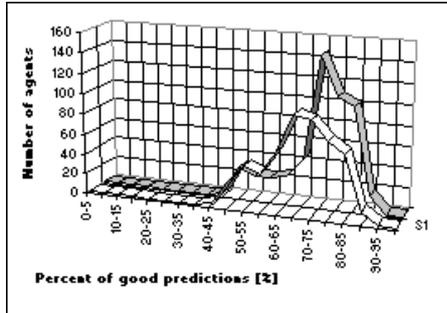


Figure 3 Percent of good predictions' distribution in the last step of simulation. Darker line shows results with *escape*.

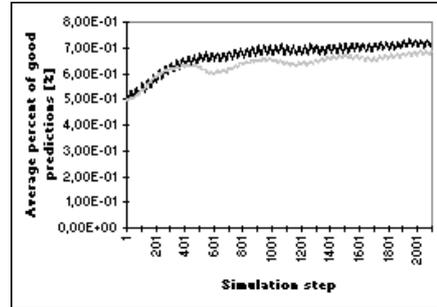


Figure 4 Average percentage of good predictions. Darker line shows results with *escape*.

ronment the *aggregation* operator was activated. Agents that acted in two other environments had to predict values taken from the shorter sequences. Splitting the main sequence into two parts created these sequences.

The first part of the simulation was the process of species formation (agents lived in two environments with shorter sequences). Then, after some time of natural selection, when there were two different species, all the agents were forced to migrate toward the environment with the long sequence. In this environment they could form aggregates consisted of two cells. The results were compared with those obtained when from the beginning of the simulation there was only one environment with the long sequence.

Figure 3 shows the percentage of correct predictions' distribution in the last step of simulations. The *escape* operator caused that there were many more agents with high percentage of correct predictions in the population.

Figure 4 shows that the average percentage of correct predictions made by agents during almost the whole simulation was higher than it was in the case when there was no *escape* operator.

Conclusions.

In this paper the technology of the *Evolutionary Multi-Agent Systems (EMAS)* has been presented. *EMAS* systems may be considered as a new approach to the construction of the evolution programs and a new area of research in the field of *Multi-Agent Systems (MAS)*. Evolution process realized in the multi-agent systems introduces following new possibilities:

- Avoiding limitations of the classical evolution programming,
- Introducing new evolution operators based on the biological evolution. Two such exemplary operators have been introduced in this paper: *aggregation* and *escape*.

- Adaptation toward changing environments of the agents' population which act in the multi-agent world.

Introducing operators like the *aggregation* one to the evolution process makes possible emergence of social relations between agents. The *escape* operator enables migration of the agents towards different environments. It has been shown that these two new operators significantly improved the evolution process in the multi-agent prediction system.

Future research should be focused on introducing new mechanisms based on the biological evolution, and application of the neural network instead of the finite state automaton.

References.

- [1] Cetnarowicz E., Cetnarowicz K., and Nawarecki E. Agent oriented technology of decentralized systems based on the M-agent architecture. Submitted to Sao Paulo, 1997.
- [2] Cetnarowicz K. M-agent architecture based method of development of multi-agent systems. In *Proc. Of the 8th Joint EPS-APS International Conference on Physics Computing*. ACC Cyfronet, Krakow, Poland, 1996.
- [3] Cetnarowicz K., Kisiel-Dorohinicki M., and Nawarecki E.. The application of evolution process in multi-agent world to the prediction system. In *Proceedings of the Second International Conference on Multi-Agent Systems*. AAAI Press, 1996.
- [4] Fogel L.J., Owens A.J., and Walsh M.J. *Artificial Intelligence Through Simulated Evolution*. John Wiley, 1966.
- [5] Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [6] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer -Verlag, 1996.
- [7] *Reference Guide for Swarm 1.4.1*. Swarm Development Group, <http://www.santafe.edu/projects/swarm>.