Models of Cognitive Evolution: Initial Steps

Names of Authors
(Do not show your name when to be submitted! Should be appeared only for camera-ready)

Affiliation of Authors
Address
e-mail
(Do not show your affiliation either when to be submitted.)

Abstract

The urgency of modeling cognitive evolution that is modeling evolution of animal cognitive abilities is underlined. Three initial models of autonomous agents that have elementary cognitive features are described. The first model describes emergence of action sequences of the single self-leaning agent that exists in two-dimensional cellular environment. The second model is devoted to adaptive behavior of agents that have natural needs (feeding, division and safety). The interaction between evolutionary optimization and learning processes in evolving population of autonomous agents is analyzed in the third model. The models demonstrate the formation of agent adaptive behavior. The simple cognitive features of agents are formed, namely, relations between situations and agent behavior are memorized in agent control systems. Further directions of modeling cognitive evolution are proposed.

1 INTRODUCTION

Investigation of cognitive evolution, evolution of animal cognitive features is very interesting and urgent. Studies of cognitive evolution are related with a very profound epistemological problem: why is *human* mind applicable to cognition of *nature*? In order to investigate the problem seriously, it is reasonable to analyze it by means of mathematical and computer models. Modeling cognitive evolution, we can analyze, why and how did animal and human cognitive features emerge, how did applicability of human mind to cognition of nature origin. So, this modeling is related with foundation of science, cognitive science and epistemological studies. Fortunately, there is a direction of research “Adaptive Behavior” (Meyer and Wilson, 1991; Donnart and Meyer, 1996) that is in close relation with modeling cognitive evolution. Using models of adaptive behavior, it is possible to analyze main steps of cognitive evolutions from simple forms of adaptive behavior to human deductive methods (Red’ko, 2008).

The current work describes three models of initial steps of cognitive evolution studies. Section 2 describes the model of emergence of action sequences of an autonomous agent that exists in two-dimensional cellular environment. The model of adaptive behavior of autonomous agents that have natural needs (feeding, division and safety) is designed and investigated in Section 3. Interaction between evolutionary optimization and learning processes in evolving population of autonomous agents is analyzed in Section 4. In particular, the genetic assimilation of acquired features of agents during a number of generations of Darwinian evolution (the Baldwin effect) is observed at computer simulations in this model. Finally, further steps of modeling cognitive evolution are discussed in Section 5.

2 GENERATION OF CHAINS OF ACTIONS

The computer model of adaptive behavior of the single self-leaning agent in the two-dimensional cellular environment is designed and investigated below. An agent control system is based on sets of logic rules that have the following form «If the situation **S** takes place, then it is necessary to execute the action *A*». The agent control system is optimized by means of reinforcement learning (Sutton and Barto, 1998). The formation of chains of actions leading to the increase of agent resource is demonstrated by computer simulations.

2.1 Description of the model

It is supposed that the autonomous agent “lives” in the two-dimensional cellular environment. The agent has the direction “forward”. In fixed number of cells there are portions of food of the agent. The agent has resource $R(t)$ that is increased at eating of food and is decreased at execution of actions by the agent. The time t is discrete, t = 0,1, … The two-dimensional environment consists of N_xN_y cells.

Each time moment the agent executes one of following five actions: eating food, moving into the forward cell, turning right or left, resting. The control system of the agent is a set of logic rules similar to rules of classifying systems (Holland et al, 1986).

Executing the action “eating”, the agent eats the whole portion of food in its cell. After removing the food portion at this eating, the new portion of food is placed in a random cell.

The agent control system ensures its action selection. The control system of the agent consists of the set of rules that have the following form:

$$\mathbf{S}_k \rightarrow A_k \, , \qquad (1)$$

where $\mathbf{S}_k$ and A_k are the situation and the action corresponding to the rule, k is the number of the rule. Each rule has the weight W_k ; weights of rules are modified at agent learning. Components of the vector $\mathbf{S}_k$ are equal to 0 or 1. Values 0 and 1 correspond to presence and absence of a portion of food in a certain cell of “the field of vision” of the agent. The field of vision of the agent includes four cells: its own cell, the forward cell and two cells to right and to left from the agent.

Each time moment the agent executes one action and is learned too. The action for the execution A^* is selected as follows. If there are rules corresponding to the current situation $\mathbf{S}(t)$ (i.e. $\mathbf{S}_k = \mathbf{S}(t)$), then the action A^* is chosen in accordance with the ε-greedy method (Sutton and Barto, 1998): the action $A^* = A_k$ corresponding to the rule that has maximal W_k is chosen with the probability 1-ε, the arbitrary action A^* is chosen with the probability ε ($0 < \varepsilon < 1$). If there is no rule corresponding to the current situation $\mathbf{S}(t)$, then the arbitrary action A^* is chosen. If the rule $\mathbf{S}(t) \rightarrow A^*$ is absent in the agent control system, then the new rule $\mathbf{S}(t) \rightarrow A^*$ is formed; the initial weight of this rule W is equal to 0. The selected action A^* is executed.

The annealing method (Kirkpatrick et al, 1983) was used at computer simulation: at $t = 0$ it was set $\varepsilon = 1$, then the value ε was exponentially decreased to zero; the characteristic time of ε reduction was 1000 time steps. At initial steps of simulation, rules were formed; at t >> 1000 actions were selected according to rule weights.

The rule weights W_k of the agent are adjusted by means of reinforcement learning (Sutton and Barto, 1998):

$$\Delta W(t\text{-}1) = \alpha\, [R(t) - R(t\text{-}1) + \gamma W(t) - W(t\text{-}1)] \, , \qquad (2)$$

where $W(t$-1) and $W(t)$ are weights of rule that are used at time moments t-1 and t, respectively, $R(t$-1) and $R(t)$ is agent resource at these time moments, α is the learning rate, γ is the discount factor; $0 < \alpha << 1$, $0 < \gamma < 1$, $1\text{-}\gamma << 1$. The rule weights that lead to growth of agent resource are increased during learning.

2.2 Simulation results

The main parameters of simulations were as follows. The environment consisted of 100 cells ($N_x = N_y = 10$); portions of food were distributed in 50 random cells. The increase of agent resource at eating was 1. The decrease of the agent resource at any action was equal to 0.01. The initial value of resource of the agent (at $t = 0$) was $R = 1$.

Simulations demonstrated that initially unknown chains of agent actions, leading to food finding, were formed. The example of time dependence of agent resource $R(t)$ is shown on Figure 1.

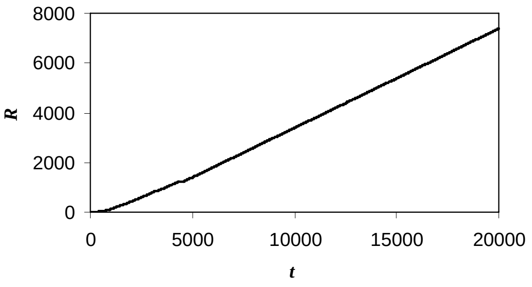

Figure 1: Time dependence of resource $R(t)$ of the single self-learning agent.

The each situation $\mathbf{S}(t)$ is determined by food presence/absence in 4 cells of agent field of vision, so there are 16 possible situations and 5 possible actions; consequently, there are 80 possible rules. In any simulation, the total number of the rules generated by the agent was 80. However, weights of these rules were varied by means of reinforcement learning, and at the end of a simulation only 16 rules were mainly used. The set of these selected rules can be considered as the following agent heuristics: 1) if a portion of food is present in the same cell where the agent is placed, then the agent executes the action “eating”; 2) if there is no food in the agent cell, and there is food in a cell that is forward or right/left with respect to the agent cell, then the agent executes the action “moving forward” or “turning right/left”, respectively; 3) if there is no food in the agent field of vision, then the agent prefers the searching action “moving forward”. We can note that the action “resting” is ignored in all situations. Hence, during learning, the autonomous agent forms quite natural heuristics, that define “reasonable” chains of actions resulting to reaching and eating of food.

Thus, the model demonstrates generation of effective action chains resulting to the increase of agent resource.

3 SEVERAL NEEDS OF AGENTS

The current computer model describes adaptive behavior of autonomous agents that have several natural needs (feeding, reproduction, safety). The model is similar to the described one in the previous section. The time t is discrete, $t = 0,1, \ldots$ The agent control system is a set of rules, characterized by the

Eq. (1). The rule weights W_k are adjusted both by reinforcement learning (in accordance with the Eq. (2)), and in the course of evolutionary optimization.

The main particularizes of the current model are as follows. The external world consists of two cells: one cell is dangerous for agents; other cell is safe. The status of cells is changed with period T_D time steps: the dangerous cell becomes the safe one, and the safe cell becomes the dangerous one. The agent in the dangerous cell loses resource r_D each time moment t. Each time moment the agent executes one of the following of 4 actions: division, eating, moving to another cells, resting. The agent sensory system defines the situation $\mathbf{S}(t)$. Vector $\mathbf{S}(t)$ has 3 binary components (0 or 1) that determine the following: 1) does amount of food in the external world exceed the certain threshold f_{th} , 2) does agent resource $R(t)$ exceed the threshold r_{th} , 3) where is the agent in the moment t: in the safe cell or in the dangerous cell. As the total number of different situations is 8 and the number of actions is 4, the whole number of rules is 32. The initial weights of these rules $\{W_{0k}\}$ constitute the genome of the agent. This genome is received by the agent from its parent (with small mutations). The temporary rule weights $\{W_k\}$, which are used by the agent at action selection, are adjusted by reinforcement learning. So, each agent has two sets of rule weights: initial weights $\{W_{0k}\}$ that are not modified during agent life and temporary adjusted rule weights $\{W_k\}$. At the moment of agent birth the temporary weights are equal to the initial ones: $\{W_k\} = \{W_{0k}\}$. In odder to consider restricted lifetime of agents, it is supposed that any agent dies at small probability P_d (P_d << 1) each time moment. If resource of the agent becomes smaller than R_{min} , then this agent dies.

The decrease of the agent resource $R(t)$ at performance of one of actions “division”, “eating”, “moving to another cell”, and “resting” is equal to r_d , r_e , r_t , and r_r , respectively. The increase of the agent resource $R(t)$ at eating is equal to r_{eat} . Actions “division” and “eating” correspond to needs of reproduction and feeding. The action “moving” corresponds to the need of safety, as it can provide movement of the agent from the dangerous cell into the safe cell. At action selection ε-greedy method is used.

The main simulation parameters were as follows. The resource decrease at any action was equal to 0.01: $r_d = r_e = r_t = r_r = 0.01$. The period of status change of cells (dangerous ↔ safe) was $T_D = 100$. The reduction of agent resource in the dangerous cell was $r_D = 10$. The increase of agent resource at eating was $r_{eat} = 10$. The probability of a random death of the agent was $P_d = 0.001$. Parameters of reinforcement learning were $\alpha = 0.1$ and $\gamma = 0.9$. The parameter of the ε-greedy method was $\varepsilon = 0.1$. Thresholds R_{min} , f_{th} , r_{th} did not influence strongly on agent behavior; in typical simulations these values were: $R_{min} = 0$, $f_{th} = 10$, $r_{th} = 1$. The control system of each agent consisted of 32 possible rules; at the start of simulations weights W_{0k} of all rules were small and random. The variations of these weights at mutations were uniformly distributed in the interval $[-0.5P_m, 0.5P_m]$, where P_m is the intensity of mutations, $P_m = 0.1$.

Using special choice of parameters, following three cases were analyzed:

Case LE (learning + evolution), i.e. full model, with the parameters described above.

Case L (pure learning); in this case the intensity of mutations was zero: $P_m = 0$.

Case E (pure evolution), in this case the intensity of learning and the parameter of greedy method were zero: $\alpha = 0$ and $\varepsilon = 0$.

According to simulations, learning (the case L) ensures quicker finding of asymptotic form of behavior as compared with evolutionary optimization (the case E). The asymptotic behavior was reached during 5000 and 100000 time steps for the cases L and E, respectively. Behavior of agents in the case LE (the full model) was similar to that of in the case L.

In the case L actions of agents in the stationary mode (at $t > 5000$) were distributed as follows. The action “resting” was executed by 25% of agents of the population, the action “eating” was executed by 70% of agents; the action “division” was executed by 3% of agents. Just after changing the danger status of cells (5-10 time steps), the frequency of the action “division” did not vary essentially, and frequencies of actions “resting” and “eating” decreased to 5% and 30%, respectively. The frequency of the action “moving” just after changing the danger status of cells increased from 5% to 60%.

In the case E actions of agents in the stationary mode (at $t \approx 200000$) were distributed as follows. The action “resting” was executed by 5% of agents of the population, the action “eating” was executed by 55% of agents; the action “division” was executed by 40% of agents. Just after changing the danger status of cells, the frequency of the action “division” was decreased to 5%, and frequencies of actions “resting” and “eating” decreased, but only in small amount (about 5%). The frequency of the action “moving” just after changing the danger status of cells was increased from almost zero value to 40%.

So, dynamics of actions of agents in cases L and E were similar each to other. The main difference consisted in relatively large frequency of the action “division” in the case of pure evolution.

In the case LE (the full model) frequencies of actions of agents were approximately the same as in the case L.

Thus, simulations demonstrate formation of rather natural behavior of agents. It is essential that reproduction plays the important role at evolutionary optimization. Evolutionary optimization is slower as compared with learning. When learning and evolutionary optimization function together, learning plays a dominant role and simulation results in case of the full model are close to results in the case of pure learning.

4 INTERACTION BETWEEN LEARNING AND EVOLUTION

The computer model of agents which are similar to the biological organisms adapting to change of temperature T in environment is designed and analyzed in this section. The control system of an agent is based on neural network adaptive critic designs (Prokhorov and Wunsch, 1997). The control system ensures forecasting of T changes and agent movement in accordance with temperature changes. Agent behavior is adjusted by means of reinforcement learning and evolutionary optimization. The interaction between learning and evolution is analyzed. The Baldwin effect is demonstrated: certain acquired features (obtained by means of learning) of agents can be genetically assimilated during several generations of Darwinian evolution.

The Baldwin effect (Baldwin, 1896; Turney et al, 1996) that is the genetic assimilation of acquired features during a number of generations of Darwinian evolution is well known. The operation of this effect includes two stages. At the first stage, evolving organisms obtain (owing to appropriative mutations) a property to learn some useful features. Fitness of such organisms increases; hence, they are distributed in the population. But learning has some disadvantages: it demands energy and time. Therefore the second stage (the genetic assimilation) is possible: useful features can "be reinvented" by evolutionary processes and these features can be directly coded in genomes of organisms.

In the article (Red'ko et al, 2005), the Baldwin effect was demonstrated for the model of agents-brokers. However, the model of agents-brokers is too far from biology. The current model is more close to biological organisms.

4.1 Description of the model

The model is based on the following analogy. Modeled "lizards" that adapt to temperature changes are considered. The adaptation essence consists in the following. There are two places, which lizards can choose: 1) a place on a stone, 2) a place in a burrow. The natural behavior is as follows. At large temperature the lizard heats on the stone, at low temperature it gets into the burrow and keeps its body warm.

A lizard uses its control systems to choose a place. The control systems of agents-lizards are based on neural network adaptive critic design (Prokhorov and Wunsch, 1997). The agent control system is optimized by means of reinforcement learning and Darwinian evolution.

The temperature of environment T_{ext} (the temperature on a stone) is determined by time series $T_{ext}(t)$, t = 0,1,2... The current situation $\mathbf{S}(t)$ is determined by two values $T_{ext}(t)$ and $P(t)$, $\mathbf{S}(t) = \{T_{ext}(t),P(t)\}$, where $P(t)$ is the parameter of the position of a lizard. It is supposed that $P(t) = 0$ if the lizard is in a burrow, and $P(t) = 1$ if the lizard is on a stone. Actions of the lizard consist in a choice of its position $P(t+1)$ in the next time moment.

It is supposed that there is some optimum temperature of lizard body T_0 and when the lizard is in the burrow its temperature is close to T_0 ; though the environment temperature influences slightly on the temperature in the burrow. So, the temperature in burrow $T_{int}(t)$ is the following:

$$T_{int}(t) = T_0 + k_1\ [T_{ext}(t) - T_0]\ , \qquad (3)$$

where k_1 is small positive parameter, $k_1 > 0$, $k_1 <<1$.

The reinforcement $r(t)$, which is received by a lizard at the time moment t, is proportional to the difference $T(t) - T_0$, where $T(t)$ is the current temperature in that place where the lizard is in the moment t:

$$r(t) = k_2\,[T(t) - T_0]\ , \qquad (4)$$

where $k_2 > 0$. For simplicity we suppose that the lizard predicts $T_{ext}(t)$, and $T_{int}(t)$ can be estimated by it according to the Eq. (3).

4.1.1 Control system of the agent-lizard

The control system of the agent-lizard is intended for maximization of the utility function $U(t)$ (Sutton and Barto, 1998):

$$U(t) = \sum_{j=0}^{\infty} \gamma^j r(t+j), \qquad t = 1,2,..., \qquad (5)$$

where $r(t)$ is the current reinforcement determined by the Eq. (4), γ is the discount factor ($0 < \gamma < 1$, $1-\gamma << 1$).

The control system of the agent consists of two neural networks (NNs): the model and the critic. The model NN predicts dynamics of the environment temperature $T_{ext}(t)$. The critic NN estimates the utility function U for the current situation $\mathbf{S}(t)$, predicted situations for two possible positions of the agent in the next time step, and the next situation $\mathbf{S}(t+1)$.

4.1.2 Operation and learning of the agent control system

Inputs of the model NN are m previous values of temperature $T_{ext}(t\text{-}m+1),\ldots,T_{ext}(t)$, this NN predicts the environment temperature in the next time moment $T^{pr}{}_{ext}(t+1)$. The model is the two-layer NN that operates according to formulas:

$$\mathbf{x^M} = \{T_{ext}(t\text{-}m+1),\ldots,T_{ext}(t)\},\ \ y^M{}_j = \tanh\,(\textstyle\sum_i w^M{}_{ij}\, x^M{}_i),$$

$$T^{pr}{}_{ext}\,(t+1) = \textstyle\sum_j v^M{}_j\, y^M{}_j\,,$$

where $\mathbf{x^M}$ is the input vector, $\mathbf{y^M}$ is the vector of outputs of neurons of the hidden layer, $w^M{}_{ij}$ and $v^M{}_j$ are synaptic weights of the model NN.

The critic NN is intended for the estimation of quality of a situation V($\mathbf{S}(t)$), namely, the estimation of the utility function $U(t)$ for the agent in the situation $\mathbf{S}(t)$. The critic is the two-layer NN that operates according to formulas:

$$\mathbf{x^C} = \mathbf{S}(t) = \{T_{ext}(t), P(t)\},\ \ y^C{}_j = \tanh\,(\textstyle\sum_i w^C{}_{ij}\, x^C{}_i),$$

$$V(t) = V(\mathbf{S}(t)) = \textstyle\sum_j v^C{}_j\, y^C{}_j\,,$$

where $\mathbf{x^C}$ is the input vector, $\mathbf{y^C}$ is the vector of outputs of neurons of the hidden layer, $w^C{}_{ij}$ and $v^M{}_j$ are synaptic weights of the critic NN.

Following operations are carried out in the agent control system each time moment t:

1) The model NN predicts the external temperature in the next time moment $T^{pr}{}_{ext}(t+1)$.

2) The critic NN estimates the value V for the current situation $V(t) = V(\mathbf{S}(t))$ and for predicted situations for both possible actions $V^{pr}{}_P(t+1) = V(\mathbf{S}^{\mathrm{pr}}{}_P(t+1))$, where $\mathbf{S}^{\mathrm{pr}}{}_P(t+1) = \{T^{pr}{}_{ext}(t+1),P(t+1)\}$, $P(t+1) = 0$ or $P(t+1) = 1$.

3) The ε-greedy method is applied (Sutton and Barto, 1998): the action corresponding to the maximum value $V^{pr}{}_P(t+1)$ is chosen with probability 1-ε, the alternative action is chosen otherwise ($0 < \varepsilon \ll 1$). The action choice is the selection of the value $P(t+1)$.

4) The chosen action $P(t+1)$ is carried out. The transition to the next time moment $t+1$ occurs. The reinforcement $r(t+1)$ in accordance with the Eq. (4) is obtained by the agent. The real value $T_{ext}(t+1)$ is observed and compared with the prediction $T^{pr}{}_{ext}(t+1)$. The synaptic weights of the model NN are adjusted to minimize the error of the prediction by means of the usual back-propagation method (Rumelhart et al, 1986). The learning rate of the model NN is equal to α_M .

5) The quality of the next situation is estimated by the critic NN: $V(t+1) = V(\mathbf{S}(t+1))$; $\mathbf{S}(t+1) = \{T_{ext}(t+1), P(t+1)\}$. The time difference error $\delta(t)$ is calculated (Sutton and Barto, 1998):

$$\delta(t)\ = r(t+1) + \gamma V(t+1) - V(t)\,. \qquad (6)$$

6) The synaptic weights of the critic NN are adjusted to minimize the time difference error $\delta(t)$; this adjustment is carried out by the gradient method, similar to the back-propagation method. The learning rate of the critic NN is equal to α_C .

4.1.3 The evolution scheme

In addition to agent learning, the evolutionary optimization of control systems of agents takes place. The evolving population consists of n agents. Each agent has its resource $R(t)$ that changes according to reinforcements: $R(t+1) = R(t) + r(t)$, where $r(t)$ is determined by the Eq. (4). Evolution passes through a number of generations, $n_g = 1, 2,\ldots$ Duration of each generation n_g is T_g time steps (T_g is lifetime of the agent). At the beginning of each generation, initial resource of any agent $R(t)$ is zero. At the end of each generation the agent having maximum resource $R_{max}(n_g)$ (the best agent of the generation n_g) is determined. This best agent gives birth to n descendants that constitute the next generation.

Each agent has two sets of synaptic weights of both NNs: **G** and **W**. The set **G** are initial NN synaptic weights that are received by the agent at the moment of its birth from the agent-parent. This set **G** is the agent genome that does not vary during its life. The set **W** are temporary NN synaptic weights that are adjusted during the agent life by means of learning. At the moment of the agent birth **W** = **G**. Descendants of the agent inherit its genome **G** (with small mutations). As the genome **G** is inherited, the evolution process has Darwinian character.

4.2 Simulation results

The main parameters of computer simulations are the following: the discount factor $\gamma = 0.9$; the number of inputs of the model NN $m = 10$; the number of neurons in the hidden layers of the model and critic NNs $N_{hM} = N_{hC} = 10$; the learning rate of the model and critic NNs $\alpha_C = \alpha_C = 0.01$; the parameter of the ε-greedy method $\varepsilon = 0.05$; the intensity of mutations $P_{mut} = 0.1$; the duration of a generation $T_g = 1000$, the population size $n = 10$.

The time dependence of the external temperature is the sinusoid:

$$T_{ext}\,(t) = 0.5\sin(2\pi t/20) + T_0\,,\quad T_0 = 1.5.$$

In order to compare learning and evolutionary optimization the following cases (similar to cases of the previous section) were analyzed:

Case L (pure learning); in this case single self-learning agent was considered;

Case E (pure evolution), i.e. evolving population of agents without learning;

Case LE (learning + evolution), i.e. the full model described above.

The values of resource obtained by agents during 1000 time steps for these three cases are compared. For cases E and LE the generation duration was $T_g = 1000$, and the maximum value of agent resource in the population $R_{max}(n_g)$ at the end of each generation was registered. In the case of L (pure learning) a single agent was analyzed. The resource of this agent was set to be zero every 1000 time steps: $R(T_g(n_g\text{-}1)+1) = 0$. In this case the index n_g was increased by 1 after every T_g time steps, and it was set $R_{max}(n_g) = R(T_g n_g)$.

The plots $R_{max}(n_g)$ are shown in Figure 2 that demonstrates that learning together with evolution (the case LE), ensures more effective growth R_{max} as compared with learning or evolution separately (cases L and E). The curves are averaged over 1000 simulations.

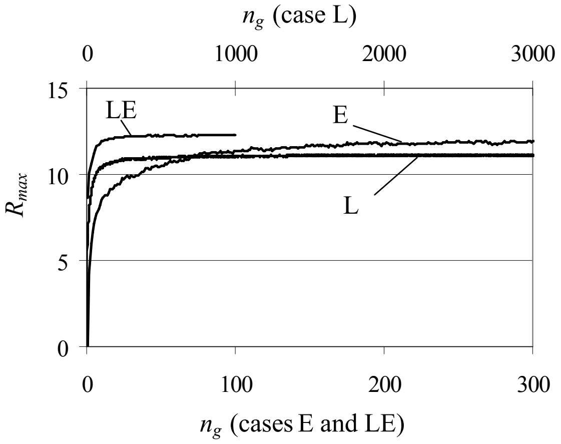

Figure 2. The plots $R_{max}(n_g)$.

The obvious influence of leaning on evolutionary processes was often observed in simulations. The essential growth of resource of the best agent began with certain time delay (200-400 time steps). This means that the agent learnt initially to get satisfactory behavioral policy, and only after several generations the resource growth began from the start of a generation. This phenomenon can be interpreted as the Baldwin effect: initially acquired (via learning) property to obtain resource became inherit during several generations. The example of this phenomenon is shown in Figure 3. This figure demonstrates resource dynamics for the best agent of the population during five generations.

Figure 3 shows that during early generations (generations 1 and 2), any significant increase of agent resource begins only after a lag of 200 to 500 time steps. The best agent optimizes its policy by learning. Subsequently, the best agents find an advantageous policy faster and faster. By the fifth generation, a newborn agent "knows" a decent policy because it is encoded in its genome **G**, and the learning does not improve the policy significantly. Thus, Figure 3 demonstrates that the initially learned policy becomes inherited (the Baldwin effect).

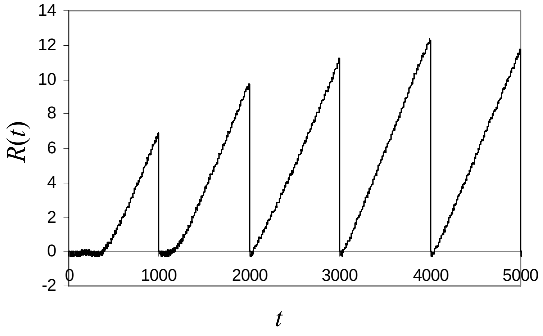

Figure 3. Time dependence of the best agent resource $R(t)$ during five generations. The case LE.

So, simulations show that the strategy initially obtained by means of learning, becomes inherited (the Baldwin effect) though evolution has Darwinian character. It should be underlined that the genetic assimilation of initially acquired features in the current model can take place quickly: during only 3-5 generations of Darwinian evolution.

5 FURTHER STEPS

The described models characterize elementary cognitive features. Autonomous agents memorize relations between situations and useful actions that should be executed in these situations. These relations are stored in the form of set logical rules (in the first and second models) or by means of neural networks (in the third model). What should be further steps of modeling more effective cognitive features? Let us consider several directions of further research.

The most non-trivial property of considered autonomous agents is generation of five simple heuristics by single self-learning agent in the first model. These heuristics generalize sensory information. Using such generalization and certain prediction of action results, it could be possible to form plans of behavior.

The second interesting direction of research is to investigate more powerful models of adaptive agents that have natural needs. The simplest forms of natural needs are analyzed in the second model. Now it is reasonable to develop further this approach.

Along with interesting behavior of evolving population of self-leaning agents in the third model, this model outlines rather intelligent agent control system. In particular, the model block of the control system provides certain prediction of future and some knowledge about interaction of an agent and its environment. More effective control systems could use similar architectures.

Acknowledgments

This work is partially supported by the program of the Presidium of the Russian Academy of Science "Intelligent informational technologies, mathematical modeling, system analysis, and automatics", Project No 2.15.

References

Baldwin, J. M. (1896). "A new factor in evolution." *American Naturalist*. Vol. 30. No. 354. PP. 441-451.

Donnart, J. Y., and J.-A. Meyer (1996). "Learning reactive and planning rules in a motivationally autonomous animat." *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*. V. 26. No. 3. PP. 381-395.

Holland, J. H., K. J. Holyoak, R. E. Nisbett, P. Thagard (1986). *Induction: Processes of Inference, Learning, and Discovery*. Cambridge: MIT Press.

Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi (1983). "Optimization by Simulated Annealing". *Science*. Vol. 220. No. 4598. PP. 671–680.

Meyer, J.-A. and S. W. Wilson (Eds.), (1991). *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*. Cambridge: The MIT Press/Bradford Books.

Prokhorov, D. V. and D. C. Wunsch (1997). "Adaptive critic designs." *IEEE Transactions on Neural Networks*. Vol. 8. No. 5. PP. 997-1007.

Red'ko, V.G. (2008). "Towards modeling cognitive evolution." In Proceedings of the Fifth International Conference on Artificial Intelligence and Neural Networks. PP. 17-21.

Red'ko, V. G., O. P. Mosalov, D. V. Prokhorov (2005). "A model of evolution and learning." *Neural Networks*. Vol. 18. No. 5-6. PP. 738-745.

Rumelhart, D. E., G. E. Hinton, and R. G. Williams (1986). "Learning representation by back-propagating error." *Nature*. Vol. 323. No. 6088. PP. 533-536.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Turney, P., D. Whitley, R. Anderson (Eds.), (1996). *Evolution, Learning, and Instinct: 100 Years of the Baldwin Effect*. Special Issue of Evolutionary Computation on the Baldwin Effect. Vol. 4. No. 3.