

Agent-Based Evolutionary Model for Knowledge Acquisition in Dynamical Environments

Wojciech Froelich¹, Marek Kisiel-Dorohinicki², and Edward Nawarecki²

¹ Institute of Computer Science
Silesian University, Sosnowiec, Poland
froelich@konto.pl

² Institute of Computer Science
AGH University of Science and Technology, Kraków, Poland
{doroh, nawar}@agh.edu.pl

Abstract. The basic idea of the approach proposed in this paper is to apply multi-agent paradigm in order to enable the integration and co-operation of different knowledge acquisition and representation techniques. The effective operation of learning process is achieved by evolutionary optimization running at the level of agents' population. In the discussed variant of the model, each agent uses reinforcement learning, and the obtained knowledge is represented as the set of simple decision rules. The approach is illustrated by a particular realization of the system dedicated to the evasive maneuvers problem, together with preliminary experimental results.

1 Introduction

The research in the domain of artificial intelligence has led to the formation of many methods that permit to solve effectively different complex tasks, also in the field of knowledge acquisition. Yet, for many difficult problems, and particularly in case of dynamically changing environments or incomplete and uncertain data, it may be necessary to adjust the method configuration to each specific variant of the task. This leads to changes in the composition and parameters of the algorithm being used (optimization through experimental evaluation). One of possible solutions is the use of hybrid methods, hierarchical composition of algorithms, and also multi-starting methods. Such approach is inspired by the observation of the natural world, where knowledge acquisition processes are realized on different levels of abstraction and the knowledge is represented by different structures (neural networks, evolutionary processes).

The basic idea of the proposed approach is to apply a multi-agent system [1, 8] in order to enable the integration and co-operation of various methods of knowledge acquisition and representation [6]. In the proposed model, the environment represents the problem being solved, and at the same time, enables the mutual agents' observation. Every agent represents a system that learns to achieve complete or partial solution of the considered problem. It is assumed that the agent is situated in the environment and interacts with it, for instance by experimentation with different behavioral strategies. In the environment there can be distinguished objects that are the source of information for learning performed by single agents and the whole population. The task that the every

agent has to perform consists in the identification of the features of objects, relations between them, or even in specifying the effective strategy of objects' control. The objects' characteristics may change during the learning process, also due to the (control) signals obtained from the system. Thus in fact the knowledge acquisition is related to the constant mutual interaction between the environment and system [2].

The issues being considered in this paper apply to the case, when the environment, from which the knowledge is to be derived, is not entirely known. In such a case, the knowledge acquisition takes place by the realization of the sequence of agent's decisions, that by intention are to lead to reaching the goal, while their intermediate effects (environment state changes) give information about its characteristics. So, even if the goal is not reached, agent's activities will give the information resulting from the environment response.

In the following sections the sketch of the model is presented. Then the details of the knowledge acquisition process are discussed. These general considerations are illustrated by a sample realization of the system dedicated for solving the evasive maneuvers problem. Preliminary experimental results conclude the work.

2 Knowledge and Learning in a Multi-agent System

Multi-agent system is composed of the set of agents working in a common environment. We can say about dual nature of such defined system, in which on one hand we have to do with the environment that surrounds the agents, and on the other hand with the population of agents building the knowledge representation, which applies to this environment. Thus in a given moment of time, the multi-agent system state is represented by the 2-tuple:

$$MAS \equiv \langle ES, AG \rangle, \quad (1)$$

where: ES represents the environment state, and AG denotes the set of agents (their states).

In the proposed model, the environment can be observed by agents through signals, and in fact the environment state is represented only by the set of signals ($es \in ES$). The signals can be interpreted as *a priori* unknown, but measurable quantities representing phenomena occurred in it.

The initial agents' population is generated on the strength of the expert knowledge and is equipped with a random set of features. Then the operation of this system is considered in consecutive moments of time. The set of environment signals is the source of information for all agents. Those signals are received by sensors, each agent is equipped with (observation), and the agent undertakes some activities. Agent's activities can mean the realization of single actions or the sequence of them. In consequence the changes of observed signals can appear, which in turn may be observed by the agents in future. The reactive method of agents' activity is assumed, which means, that agents do not perform mathematical calculations and algorithmically controlled analysis of the history of their observations and actions. Planning in the meaning of building and estimation of behavioral patterns is neither realized.

The knowledge acquisition is realized by using reinforcement learning [2] paradigms and a multi-agent evolution [5]. Reinforcement learning leads to the creation and

selection of decision rules that allow for reaching the agent's goal. The multi-agent evolution is to improve the effectiveness of the learning process through optimization of individual agent's features, which can apply among others to its observation abilities (e.g. sensors sensitivity on the environment signals change), and may be the parameters of the decision process. The agents are equipped with knowledge that can be classified according to its nature as a-priori, delivered by the designer, or a-posteriori acquired during the learning process. The a-posteriori knowledge learned by every agent can be considered as:

- phylogenetic knowledge (genotypic characteristics obtained through the
- multi-agent evolution process), ontogenetic knowledge (acquired through the adaptation in the environment).

3 The Structure of a Learning Agent

The agent's internal structure has been defined *a priori* at the stage of model design [2], and is composed of the following functional blocks. By means of **sensors**, agent observes the environment signals. Sensors are characterized among others by sensitivity that describes the minimal change of the environment signals that are being recorded. **Iconic memory** stores successive observations and decisions. In case of incomplete environment observation, an agent may lack sufficient information necessary to make an appropriate decision. This short term memory provides some historical data, which may be used to discover the dynamics of the environment changes. With the use of **effectors**, the agent puts into practice the activities in the environment. Agent activities can be either single reactions (actions) to received stimulus or sequences of actions leading to particular goals. **Knowledge base** is composed of decision rules, which consists of some observation pattern (premise) and assigned agent's decision (conclusion). **Genotype** describes selected elements of the agent's structure, its sensory and morphological features. Examples of the agent's genotype features can be: sensors sensitivity or environment observation range. **Energy** is a consumable and renewable resource that describes agent's usability in the realization of the task. **Decision algorithms** constitute the agent's structure kernel, and make it possible to take decisions and implementation of residual functions connected with agent's activity.

The the actual agent's state is represented by the 6-tuple:

$$ag \equiv \langle s, d, m, e, g, R \rangle \quad (2)$$

where: s denotes observation vector, d – decision vector, m – iconic memory matrix, e – agent's energy, g – genotype vector (phylogenetic knowledge), R – (ontogenetic) knowledge base.

Figure 1 illustrates a single step of the agent's activity. The signals observed from the environment are stored in the observation vector. A knowledge base should provide a correlation of the observation vector with an adequate decision leading to accomplishing the task. Due to the limited volume of memory, an agent cannot remember all observations, and store in the knowledge base only patterns that represent subsets of

the observation space. An agent makes classification (matching) of the obtained observation vector in a set of observation patterns. As a result the related agent’s decision is determined, on the basis of which agent realizes actions in the environment. The observation and decision vectors are then stored in the iconic memory.

The (ontogenetic) knowledge base of an agent consists of simple decision rules (classifiers): $r \equiv \langle w, a, h \rangle \in R$, consisting of condition part w ($w_i \in \mathbf{R}$), decision part a ($a_i \in \mathbf{N}$), and auxiliary attributes h (performance weight, frequency of activations, actual policy gauge). On the basis of the auxiliary attributes the rules can be filtered out after some periods of learning. For example the classifiers which are less frequent or possess lower performance can be deleted from the knowledge base. Thus the amount of classifiers can be adopted to the requirements of the particular task of an agent.

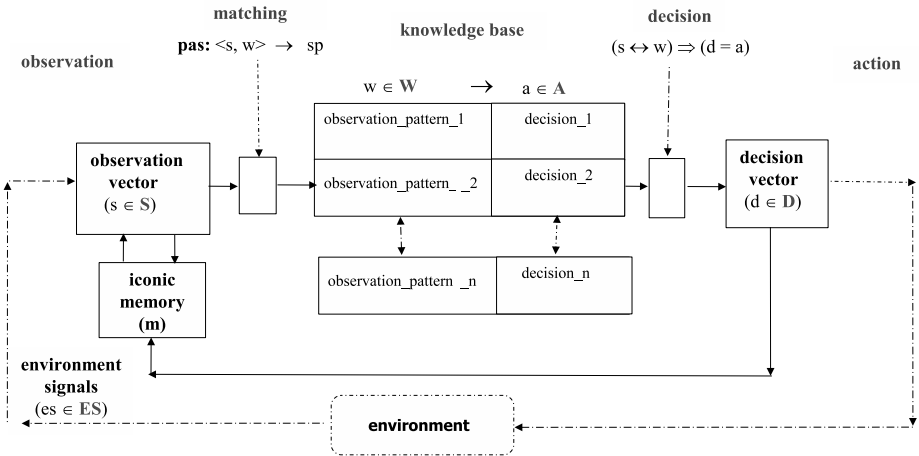
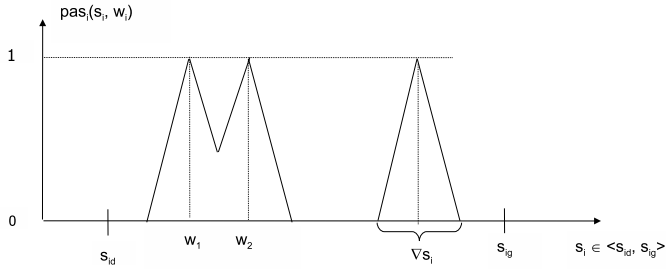


Fig. 1. Agent’s activity diagram

4 Knowledge Acquisition Process

The agent’s decision process is in fact divided into particular steps that it undertakes in order to achieve its goal, which is the reward received from the environment (reinforcement learning). The value of rewarding function is usually known after finishing a total decision sequence in a defined time horizon. In each step an agent observes the environment and gets vector $s \in S$, then it selects action $a \in A$. The environment responds in a way that after finishing the actions by all agents of the population (a learning episode), it informs every agent about the reward granted. Learning consists in finding the optimal strategy $S \rightarrow A$ where S is the set of environment conditions, A – the set of agents’ activities. Considering the sequence of actions, the learning means searching for strategy $S \times A \rightarrow A$.

Let us assume that $pas : S \times W \rightarrow \mathbf{R}$ is the matching function and $s_p \in \mathbf{R}$ is the matching threshold. If the condition $pas(s, w) \geq s_p$ is satisfied, it is assumed that the considered rule is a candidate to activate (fire), which means that its decision part a is a candidate to fullfil the agent’s decision vector d . In general, the function $pas(s, w)$



$$pas_i(s_i, w_i) = \begin{cases} \frac{\nabla s_i - r}{\nabla s_i} & , \quad \text{for } r \leq \nabla s_i \\ 0 & , \quad \text{for } r > \nabla s_i \end{cases} \quad \text{where: } \begin{cases} \nabla s_i - \text{parameter stored in genotype,} \\ r = |w_i - s_i| \end{cases}$$

Fig. 2. Matching function

can be considered in n -dimensional observation space S . For the simplification of the computational complexity, it was assumed that $pas(s, w) = \sum_{i=1}^n w s_i \cdot pas_i(s_i, w_i)$, where s_i, w_i are the components of vectors s and w respectively (fig. 2). The value of $w s_i$ is the weight reflecting the importance of the i -th component of the observation vector for the decision process and for the purpose of the following experiments has been encoded into the agent’s genotype vector.

During creation of a new agent, its **ontogenetic knowledge** does not exist (its knowledge base is empty). Then, an agent makes decisions at random, which applies to the way of activity (exploration or exploitation).

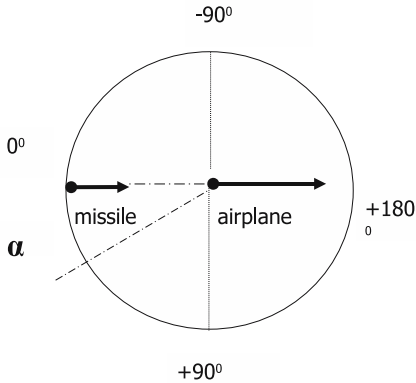
1. In case it choses the exploration, a new rule in the knowledge base is generated. Its conditional part is the observation pattern (obtained on the basis of the observation vector). Decision part is generated at random. A new rule must be unique in the knowledge base.
2. Second possibility is the choice of knowledge exploitation. The agent tries to classify the observation vector according to the previously stored observations. Every unclassified observation generates a new classifier in it’s knoledge base. In this case, the obtained observation vector is the subject to matching in the knowledge base through comparisons with patterns in the conditional parts of rules already existing in the knowledge base.
 - If there is a lack of matching pattern, the complement of the knowledge base is undertaken and a new rule is added just like in the case of exploration.
 - In case of correct matching of more than one pattern in the knowledge base, decision is undertaken at random (the roulette wheel method has been used), the probability of choosing the particular rule depends on its performance weight attribute (the adequate auxiliary attribute of a rule).
3. The final effect of the decision process is the choice of the rule (activation) in the knowledge base. The selected rule is then marked by modification of the value of the usability indicator (another auxiliary attribute). The attribute specifying the number of the rule’s activation is also updated.

4. The agent executes actions as a result of the decision process. The execution of an action by the agent involves spending the energy resource. The decision vector can be stored in the iconic memory.
5. If the agent gains the energetic reward, it modifies the performance weight for all used rules (that led to the success). In order to optimize the parameters of the knowledge base (size, searching speed), the rules with the low performance weight and rarely used (the values of related auxiliary attributes below certain threshold) are removed from the knowledge base.

The acquisition of **phylogentetic knowledge** is realized by the evolutionary processes. Genotype optimization takes place at the population level and takes advantages of the multi-agent evolution paradigm [5]. In this model classical selection mechanism realized on the basis of global fitness function cannot be used, since it leads to the introduction of the order into the phenotype space, which is not always adequate with the space structure of the given problem's solutions. Concerning the above, the energetic selection method observed in biology [7] has been introduced. Every agent of the initial population is provided with a random allocation of the resource, called *life energy*. Having undertaken actions, the agents spend the energy, thus lose some part of energy possessed (energetic cost). Agents, which energy drops below a certain level, are to be eliminated. Agents, which achieve success during the realization of a task, get the energetic reward, which allows for surviving in the environment. Agents, which reach certain energetic level may undergo reproduction. The agent's genotype is the subject of mutation during reproduction. In this way, it is possible to gradually improve the efficiency of agents in the population.

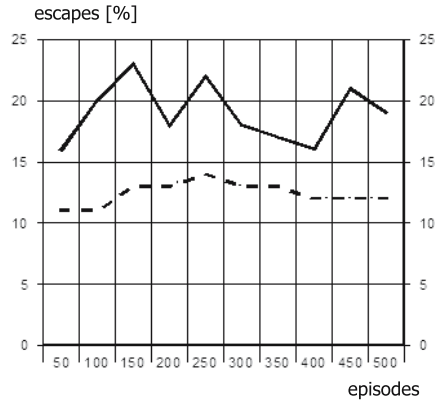
5 Experimental Results

The method of learning an agent described above has been applied to the evasive maneuvers problem (pursue and evasion game) in a two-dimensional space for a single pursuer (rocket) and a single evasive object (plane). In this case, towards the flying plane, the rocket is launched, which goal is to hit the plane. The rocket is led automatically (in a deterministic way) at the target. The plane should be capable of evading the racket by changing the parameters of flight, i.e. the direction and speed. The plane is equipped with measuring devices that inform about the parameters of the approaching rocket flight: the distance, mutual flight angle, rocket's speed. On the basis of the radar readings the agent controlling the plane should be capable of learning how to change the direction and speed of the plane to avoid the hit. Agent learning task has been divided into learning episodes. One of the initial conditions of each episode was the mutual location of the objects: the rocket and the plane (fig. 3a). The episode ends when the plane is hit or manages to escape. The degree of difficulty has been established (by changing dynamical parameters of the objects) in such a way that the trivial strategies, consisting in continuous turning of the plane in one direction, did not allow to escape. The reference point for executed experiments was the case of using by the agent the random strategy of controlling the plane's escape (consisting in random changes of flight direction and speed), which is illustrated in fig. 3b. In this case, the number of escapes achieved, did not exceed 40% of all learning episodes.



$\alpha \in <-90^\circ, +90^\circ>$,
 $\alpha = k \times 180$, $k \in <-5, +5>$

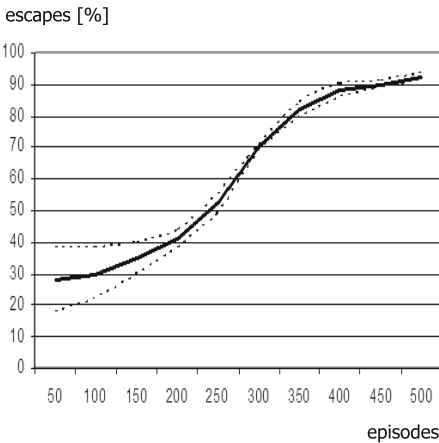
a) Initial position of pursuer and evader



———— The best agent
 - - - - Average of the population

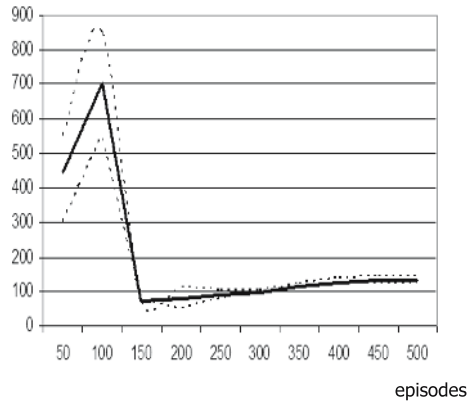
b) Random strategy of the evader

Fig. 3. Preliminary conditions for the simulation



———— the best agent
 - - - - standard deviation

a) The percentage of escapes during learning process



———— the best agent
 - - - - standard deviation

b) The amount of rules stored by the best agent

Fig. 4. Effectiveness of the best agent

Further, the tests had been executed, during which, the agent’s learning based on the suggested model had been performed. Obtained results have shown the effectiveness of the learning method being used. The learning effectiveness for the best population agent, let to achieve over 90% ratio of the plane escapes from among all learning episodes (fig. 4a). The effectiveness of the best agent is comparable to the results

achieved in other systems [4, 3]. The optimization of the agent's knowledge base size also has been achieved, i.e. the reduction of the number of decision rules (fig. 4b).

6 Concluding Remarks

The main idea of the approach was based on the agent paradigm in order to organize the hybrid method of knowledge representation and acquisition [6], together with the evolutionary optimization of the population of co-operating algorithms—agents [5]. A suitable model has been elaborated, supporting analysis of the learning processes that occur on different levels of abstraction.

Based on the presented realization and performed experiments, it may be said that the suggested idea fulfills requirements applying to the effectiveness of the obtained solutions. It is also foreseen to have great universality within the chosen range of applications. Further improvements of the suggested knowledge acquisition model require more experiments concerning other applications, which will be the subject of continued research.

References

1. J. Ferber. *Multi-Agent Systems*. Addison-Wesley, 1999.
2. W. Froelich. Evolutionary multi-agent model for knowledge acquisition. In *Intelligent Information Processing and Web Mining (IIPWM'05)*, Advances In Soft Computing. Springer, 2005.
3. D. Gordon and D. Subramanian. A multistartegy learning scheme for assimilating advice in embedded agents. In *Proc. of the Second Intl. Workshop on Multistrategy Learning*, 1993.
4. J. Grefenstette, C. Ramsey, and A. Schultz. Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5(4), 1990.
5. M. Kisiel-Dorohinicki. Agent-oriented model of simulated evolution. In W. I. Grosky and F. Plasil, editors, *SofSem 2002: Theory and Practice of Informatics, Lecture Notes in Computer Science*. Springer-Verlag, 2002.
6. M. Kisiel-Dorohinicki, G. Dobrowolski, and E. Nawarecki. Agent populations as computational intelligence. In L. Rutkowski and J. Kacprzyk, editors, *Neural Networks and Soft Computing*, Advances in Soft Computing. Physica-Verlag, 2003.
7. S. Wierzchoń. Multimodal optimization with artificial immune systems. In *Intelligent Information Systems*. Physica-Verlag, 2001.
8. M. Wooldridge, editor. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.