Behaviour Patterns Evolution on Individual and Group Level

Stanislav Slušný, Roman Neruda, Petra Vidnerová
Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, Prague 8, Czech Republic

Abstract: In this paper we compare the evolution of simple behaviour patterns for both an individual and a group of simulated physical robots. An evolutionary algorithm with quite general objective function is used to study the ability to develop behaviour patterns such as the maze exploring ability. The group experiments demonstrate the development of collective behaviour where the group members follow the leader who is exploring the maze. Although controlled by identical simple neural network, the group members demonstrate a level of specialization. The experiments have been verified with the real physical Khepera robots.

Key–Words: Robotic agents, Evolutionary algorithms, Neural networks.

1 Introduction

In contrast to traditional systems, reactive and behavior based systems have placed agents with low levels of cognitive complexity into complex, noisy and uncertain environments. One of the many characteristics of intelligence is that it arises as a result of an agent's interaction with complex environments. The Evolutionary robotics (ER) approach attacks the problem through a self-organization process based on artificial evolution. Its main advantage is that it is an ideal framework for synthesizing agents whose behavior emerge from a large number of interactions among their constituent parts [8].

In recent years, ER is gathering increasing attention, focusing mainly on group of robots. The results are rather encouraging and suggest that an evolutionary method might be successfully applied to the synthesis of cooperative collective behavior. In this paper we present a set of experiments in which both the individual robots and a group of simulated robots were evolved to perform different tasks within a middle-sized arena. For the individual robots we observe the emergence of higher behaviour patterns such as following the wall (either left or right) which happens to be an efficient strategy for general maze exploration. During the group experiments we deal with several robots endowed by identical neural control mechanism, one of which carries a light source that other robots can sense. The ability to follow the team leader is gradually developed during the course of evolutionary algorithm. Team members do not know their position, the only way to sense another robot is by eight infrared sensors. It has been shown, that this way it is hard for a robot to distinguish between a wall and another robot. We show, that despite the difficult conditions, collective behavior can emerge and group of robots can find circular "home" area in the unknown environment.

The organization of the paper is as follows. In the next section we review the related literature. In section 2 and 3 we briefly present multilayer perceptron networks and evolutionary algorithms. The next section describes our experimental framework with the emphasis on the evolutionary algorithm fitness function which is a key element for the behaviour formation. The results obtained are presented in section 5. Finally, in section 6 we discuss the implications of the obtained results and future directions of our work.

2 Related work

The book [8] provides comprehensive introduction to the ER, with focus on robot systems. Recently, effort is made to study emergence of intelligent behavior within the group of the robots. Pioneering work was done by Martinoli [7]. He solved the task, in which group of simulated Khepera robots were asked to find "food items" randomly distributed on an arena. The control system was developed by the artificial evolution. First example of the use of artificial evolution to design coordinated, cooperative behavior for real robots is the work [9]. Artificial evolution was employed to design neural network controllers for small, homogeneous teams of mobile autonomous robots. The robots were evolved to perform a formation movement task from random starting positions, equipped only with infrared sensors.

The work [1] presents a set of experiments in which a group of simulated robots were evolved for the ability to aggregate and to move together toward a light target. Evolved individuals displayed interesting behavioral patterns in which groups of robots acted as a single unit. Moreover, groups of identical evolved individuals displayed primitive forms of ”situated” specializations in which different individuals played different behavioral functions according to the circumstances. Neural networks used in the experiments were simple feed forward neural networks without hidden neurons.

As noted in [2], distributed coordination of groups of individuals accomplishing a common task without leaders, with little communication, and on the basis of self-organising principles, is an important research issue within the study of collective behavior of animals, humans and robots. The paper shows how distributed coordination allows a group of evolved physically-linked simulated robots (inspired by a robot under construction) to display a variety of highly coordinated basic behaviors such as collective motion, collective obstacle avoidance, and collective light approaching.

3 Multilayer perceptron networks

Feedforward neural networks has become a widely used tool in robotics for various reasons. They provide straightforward mapping from input signals to output signals and several levels of adaptation.

A multilayer feedforward neural network is an interconnected network of simple computing units called neurons which are ordered in layers, starting from the input layer and ending with the output layer [5]. Between these two layers there can be a number of hidden layers. Connections in this kind of networks only go forward from one layer to the next. The output $y(x_0, \ldots, x_n)$ of a neuron is defined as:

$$y(x_0, \ldots, x_n) = g\left(\sum_{i=1}^{n} w_i x_i - w_0\right), \qquad (1)$$

where x is the neuron with n inputs $(x_1, \ldots, x_n)$, one output $y(x)$, $(w_0, \ldots, w_n)$ are weights and $g : \Re \to \Re$ is the activation function. We have used one of the most common activation functions, the logistic sigmoid function (2):

$$\sigma(\xi) = 1/(1 + e^{-\xi t}), \qquad (2)$$

where t determines its steepness.

In our approach, the evolutionary algorithm is responsible for weights modification, the architecture of the network is determined in advance and does not undergo the evolutionary process.

4 Evolutionary Learning

The evolutionary algorithms (EA) [6, 4] represent a stochastic search technique used to find approximate solutions to optimization and search problems. They use approaches inspired by evolutionary biology such as mutation, selection, and crossover operations. The EA typically works with a population of *individuals* representing abstract representations of feasible solutions. Each individual is assigned a *fitness* that is a measure of how good solution it represents. The better the solution is, the higher the fitness value it gets. The population evolves towards better solutions. It starts with a population of completely random individuals and iterates in generations. In each generation, the fitness of each individual is evaluated. Individuals are stochastically selected from the current population (based on their fitness), and modified by means of *mutation* and *crossover* operators to form a new population. The new population is then used in the next iteration of the algorithm.

Feed forward neural networks used as robot controllers are encoded in order to use them in the evolutionary algorithm. The encoded vector is represented as a floating-point encoded vector of real parameters determining the network weights. Thus, the encoded network is a vector $(P_1, \ldots, P_N)$ where N is a total number of neurons in the network, and each P_j represents an encoded neuron parameters, i.e. $P_j = (w_{j0}, \ldots, w_{jN_j})$ where N_j determines the number of j-th neuron inputs (it is generally different for different layers).

Typical evolutionary operators for this case have been used, namely the uniform and arithmetic crossover and the mutation which performs a slight additive change in the parameter value [10]. The rate of these operators is quite big, ensuring the exploration capabilities of the evolutionary learning. A standard roulette-wheel selection is used together with a small elitist rate parameter. Detailed discussions about the fitness function are presented in the next section.

5 Experiments

Although evolution on real robots is feasible, serial evaluation of individuals on a single physical robot might require quite a long time. Therefore we used the Yaks simulator, one of the widely used simulation software is the Yaks simulator [3], which is freely available. Simulation consists of predefined number of discrete steps, each single step corresponds to 100 ms. Yaks works with the model of Khepera

Figure 1: The Khepera robot. A schema of sensors and motors within the robot body.

robot (Fig. 1). Khepera is a miniature mobile robot with a diameter of 70 mm and a weight of 80 g. The robot is supported by two lateral wheels that can rotate in both directions and two rigid pivots in the front and in the back. The sensory system employs eight active infrared light sensors distributed around the body, six on one side and two on other side. In passive mode, they measure the amount of infrared light in the environment, which is roughly proportional to the amount of visible light. In active mode these sensors emit a ray of infrared light and measure the amount of reflected light. The closer they are to a surface, the higher is the amount of infrared light measured. The Khepera sensors can detect a white paper at a maximum distance of approximately 5 cm.

To evaluate the individual, simulation is launched several times. Individual runs are called "trials". In each trial, the environment is initialized and the starting location of the robots is chosen randomly, The neural network is constructed from the chromosome, inputs of neural network are interconnected with robot's sensors and outputs with robot's motors. The robots are then left to "live" in the simulated environment for some (fixed) time period, fully controlled by neural network. As soon as any robot hits the wall or another robot, simulation is stopped. Depending on how well the robots were performing, the individual is evaluated by value, which we call "trial score". The higher the trial score, the more successful robots in executing the task in a particular trial. Finally, the fitness value is computed based on trial scores.

5.1 Experiment 1: Individual exploration

In the first experiment, the single robot is put in the maze of 60x30 cm. The robot's task is to fully explore the maze. Fitness evaluation consists of four trials, individual trials differ by agent's starting location. Robot is left to live in the environment for 250 simulation steps. As the robot is able to use eight infrared sensors, feed forward neural with 8 input units, 2 hidden units and 2 output units is used as robot controller.

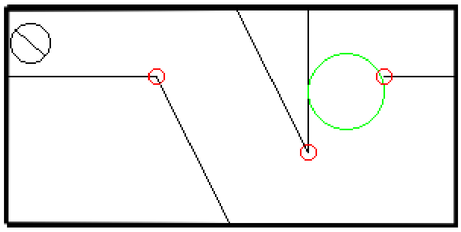

Figure 2: In the maze exploration task, agent is rewarded for passing through the zone, which can not be sensed. The zone is drawn as the bigger circle, the smaller circle represents the Khepera robot. The training environment is of 60x30 cm.

The three component $T_{k,j}$ motivates agent to learn to move and to avoid obstacles.

$$T_{k,j} = V_{k,j}(1 - \sqrt{\Delta V_{k,j}})(1 - i_{k,j}) \qquad (3)$$

First component $V_{k,j}$ is computed by summing absolute values of motor speed in k-th simulation step and j-th trial, generating value between 0 and 1. The second component $(1 - \sqrt{\Delta V_{k,j}})$ encourages the two wheels to rotate in the same direction. The last component $(1 - i_{k,j})$ encourage obstacle avoidance. The value $i_{k,j}$ of the most active sensor in k-th simulation step and j-th trial provides a conservative measure of how close the robot is to an object. The closer it is to an object, the higher the measured value in range from 0 to 1. Thus, $T_{k,j}$ is in range from 0 to 1, too.

In j-th trial, score S_j is computed by summing normalized trial gains $T_{k,j}$ in each simulation step.

$$S_j = \sum_{k=1}^{250} \frac{T_{k,j}}{250} \qquad (4)$$

To stimulate maze exploration, agent is rewarded, when it passes through the zone. The zone is randomly located area, which can not be sensed by an agent. The reward Δ_j is 1, if agent passed through the zone in j-th trial and 0 otherwise. The fitness value is then computed as follows:

$$Fitness = \sum_{j=1}^{4} (S_j + \Delta_j) \qquad (5)$$

Successful individuals, which pass through the zone in each trial, will have fitness value in range from 4 to 5. The fractional part of the fitness value reflects the speed of the agent and it's ability to avoid obstacles.

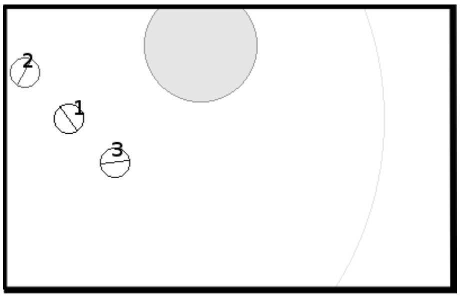

Figure 3: The environment and three robots. Thick lines represent the walls surrounding the arena of size 80x50 cm. The full circle represents the target "home" arena. The arc indicates the light intensity, but meaningful values could be measured up to 20 cm only by robot's sensors. The robot marked with "1" is the team leader, carrying the light source.

5.2 Experiment 2: Collective exploration

In the second experiment, team of three robots looks for circular arena, which is randomly located in the environment. Robots can sense the arena by the simulated ground sensor: sensor returns value 1, if robot is located in the arena and 0 otherwise.

In each of ten trials, the environment is initialized and the starting location of the team leader is chosen randomly, remaining robots are situated not far from him. Each trial took not more than 500 simulation steps.

The team leader is equipped with simulated light bulb. This way, team members can use information from light sensors to produce "following behavior". The emitted light can be detected in surroundings of the team leader up to the distance 20 cm.

All robots are guided by the same neural network, constructed from the chromosome. The robot is able to use eight infrared sensors in active and passive mode and ground sensor. Therefore, feed forward neural with 17 input units, 2 hidden units and 2 output units is used as the robot controller.

Let's denote by $T_{k,j}$ the trial score in k-th simulation step and j-th trial, generating value between 0 and 2.

$$T_{k,j} = L_{k,j} M_{1,k,j} M_{2,k,j} \tag{6}$$

$T_{k,j}$ is computed as a product of evaluation of leader($L_{k,j}$) and team members ($M_{1,k,j}$ and $M_{2,k,j}$) performance. $L_{k,j}$ is responsible for exploration behavior, while $M_{1,k,j}$ is responsible for grouping behavior.

$$L_{k,j} = V_{k,j}(1 - \sqrt{\Delta V_{k,j}})(1 - i_{k,j}) + Z_{k,j} \tag{7}$$

$Z_{k,j}$ has value 1, if the team leader is in k-th simulation step and j-th trial in the target area, and 0 otherwise. Remaining components are computed similarly to previous experiment: $V_{k,j}$ is computed by summing absolute values of motor speed in k-th simulation step and j-th trial, generating value between 0 and 1, $(1 - \sqrt{\Delta V_{k,j}})$ encourages the two wheels to rotate in the same direction and the last component $(1 - i_{k,j})$ encourage obstacle avoidance. Thus, $T_{k,j}$ is in range from 0 to 2.

$$M_{i,k,j} = (1 - D_{k,j}(i, 0)) \tag{8}$$

The value $D_{k,j}(i, 0)$ is a normalized distance of robot i to the team leader, computed as follows:

$$D_{k,j}(i,j) = \begin{cases} 1 & : \ dist(i,j) > 200mm \\ \frac{dist(i,j)}{200} & : \ otherwise \end{cases} \tag{9}$$

In the j-th trial, the trial score S_j is computed by adding trial gains $T_{k,j}$ in each simulation step.

$$S_j = \sum_{k=1}^{500} T_{k,j} \tag{10}$$

The fitness value is then computed by summing trial scores:

$$Fitness = \sum_{j=1}^{10} S_j \tag{11}$$

6 Results and discussion

6.1 Experiment 1: Individual exploration

In the first experiment, robot is rewarded for passing through the zone, which can not be sensed by its sensors. The zone is randomly located circular arena in a maze.

The results are encouraging: typical behavioral patterns, like following the left wall has been developed, which in turn resulted in the very efficient exploration of an unknown maze. The best individuals from the last generation successfully found the randomly located zone in all of 30 trials.

The important thing is to test the quality of the obtained solution in a different arena, where a bigger maze is utilized (Fig. 4). However, the agent is capable of efficient space exploration behavior that has

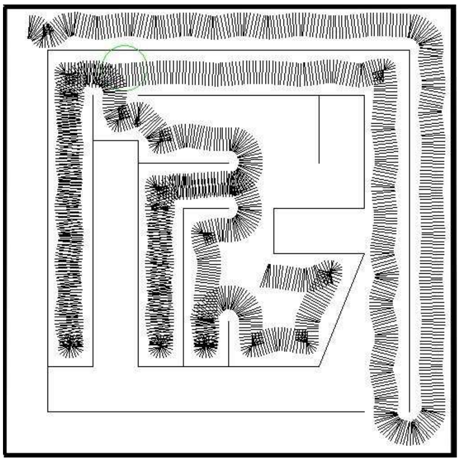

Figure 4: The agent is put in the bigger maze of 100x100 cm. Agent's strategy is to follow wall on it's left side.

emerged during the learning to find random zone positions. The figure shows that the robot trained in a quite simple arena and endowed by relatively small network units is capable to navigate in a very complex environment.

In this experiment, the evolutionary algorithm has chosen reasonable strategy (follow the left wall) based only on basic action concepts (move, avoid walls).

6.2 Experiment 2: Collective exploration

In the second experiment, the team of robots was trained to search for the target circular arena. The team was guided by the team leader equipped with the light source. Robots did not know their relative positions, the team leader could be recognized only by the light bulb. However, as the robots used only infrared sensors, they were not able to to discriminate between wall and another robot. The robots were guided by the single neural network.

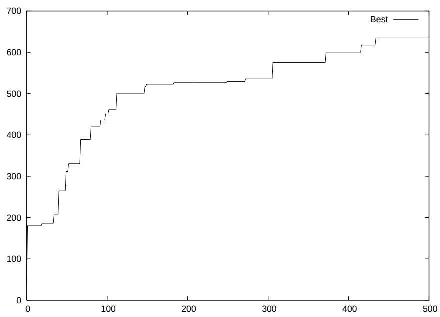

Figure 5: The fitness of the best individual.

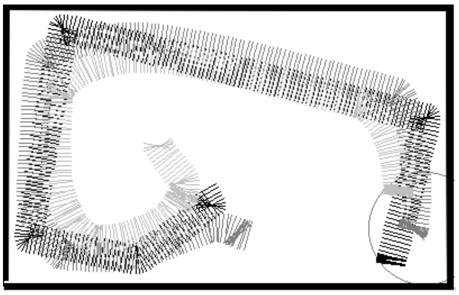

Figure 6: Typical run of the best individual from the 500. generation. Three robots aggregate together and are guided by the team leader (the robot in the middle), which carries the light source. After reaching the target area, they do not leave it anymore.

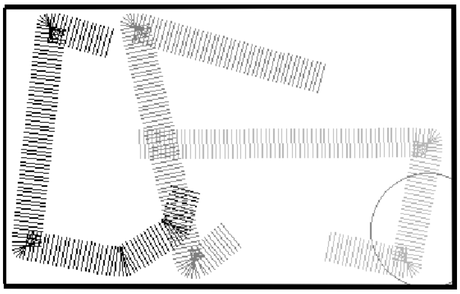

Figure 7: Testing trained behavior when no light source is in the environment. No group behavior is present, robots are exploring the environment searching for the light.

The evolutionary process took 500 generations (Fig. 5). The robot team, corresponding to the best individual, clearly showed grouping behavior - the distance between team members did not get over the 200 cm and the team leader was clearly guiding the group. Remaining robots followed the team leader, keeping the safe distance (Fig. 6). If the distance between robots was too small, the robots crashed or team members lost the trace of the team leader (obviously, rapid change of direction by team leader caused the problems for remaining team members). Starting from approximately 300th generation, individuals were able to find target area in all ten trials. To prove the ability to follow the team leader, robots were put to the environment without light source. In these conditions, the grouping behavior was not present: each robot was exploring the environment on its own.

Whenever the light bulb was switched off during the experiment (Fig. 7), robots scattered. What's more, robots even ignored the target arena. After switching the bulb on again, the group formation was slowly reformed.

7 Discussion

We have demonstrated how a more complicated behavioural pattern can emerge from rather simple setup by self-organization via artificial evolution. It is interesting to note that the control mechanism realized by a neural network does not require a large number of parameters to fulfill the described tasks. The number of adjusted parameters (weights and biases in the network) was around 30 (or 60 for the second experiment). We have performed several tests with larger networks and alternative architectures including RBF and recurrent networks [10], but the results were quite similar for this type of task.

The results reported above represent just a few steps in the journey toward intelligent team coordination and collective behavior. The next would be to extend this approach for more complicated tasks and compound behaviors. Obviously, task specialization was present in the experiment, as it was hard-coded into the fitness function. In the next experiments, we would like to emerge specialization and labour division algorithms. The adaptive process should learn conditions, in which it is useful (from the team's perspective) to leave the group by it's team members, or in which it is useful to act as a single unit.

8 Acknowledgments

This work was supported by the Grant Agency of the Charles University under the project number GA UK 7637/2007. The experiments have been performed partly in the EPCC under the project HPC-EUROPA (RII3-CT-2003-506079) with the support of the European Community - Research Infrastructure Action under the FP6 "Structuring the European Research Area" Programme.

References:

[1] G. Baldassarre, S. Nolfi, and D. Parisi. Evolving mobile robots able to display collective behaviour. In Hemelrijk C. K, editor, *International Workshop on Self-Organisation and Evolution of Social Behaviour*, pages 11–22, 2002.

[2] G. Baldassarre, S. Nolfi, and D. Parisi. Coordination of simulated robots based on self-organisation. *Artificial Life*, (3):289–311, 2006.

[3] J. Carlsson and T. Ziemke. Yaks - yet another khepera simulator. In Witkowski Ruckert, S., editor, *Autonomous Minirobots for Research and Entertainment Proceedings of the Fifth International Heinz Nixdorf Symposium*, 2001.

[4] D. B. Fogel. *Evolutionary Computation: The Fossil Record*. MIT-IEEE Press, 1998.

[5] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1998.

[6] J. Holland. *Adaptation In Natural and Artificial Systems*. MIT Press, 2nd edition, 1992.

[7] A. Martinoli. *Swarm Intelligence in Autonomous Collective Robotics: From tools to the Analysis and Synthesis of Distributed Control Strategies*. Computer Science Department, EPFL, Lausanne, Switzerland, 1999.

[8] S. Nolfi and D. Floreano. *Evolutionary Robotics — The Biology, Intelligence and Techology of Self-Organizing Machines*. MIT Press, 2000.

[9] M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving teamwork and role-allocation with real robots. *Artificial Life*, 8:302–311, 2002.

[10] Stanislav Slusny and Roman Neruda. Autonomous robot control by means of three evolutionary neural architectures. In *Proceedings of the 1st European AROB Workshop*, Wien, 2007.