

---

01 Jul 1999

## Evolving Neural Networks Applied to Predator-Evader Problem

Shivakumar Viswanathan

Ilker Ersoy

Filiz Bunyak

*Missouri University of Science and Technology*

Cihan H. Dagli

*Missouri University of Science and Technology, dagli@mst.edu*

Follow this and additional works at: [https://scholarsmine.mst.edu/comsci\\_facwork](https://scholarsmine.mst.edu/comsci_facwork)



Part of the [Systems Engineering Commons](#)

---

### Recommended Citation

S. Viswanathan et al., "Evolving Neural Networks Applied to Predator-Evader Problem," *Proceedings of the International Joint Conference on Neural Networks, 1999. IJCNN'99.*, Institute of Electrical and Electronics Engineers (IEEE), Jul 1999.

The definitive version is available at <https://doi.org/10.1109/IJCNN.1999.833442>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# Evolving Neural Networks Applied to Predator-Evader Problem

Shivakumar Viswanathan, Ilker Ersoy, Filiz Bunyak and Cihan Dagli

Smart Engineering Systems Lab.  
Engineering Management Department,  
University of Missouri-Rolla,  
1870 Miner Circle, Rolla MO 65401  
{shiva,bunyak,ersoy,dagli}@umr.edu

## Abstract

*The creation of strategies to meet abstract goals is an important behavior exhibited by natural organisms. A situation requiring the development of such strategies is the predator-evader problem. To study this problem, Khepera robots are chosen as the competing agents. Using computer simulations the evolution of the adaptive behavior is studied in a predator-evader interaction. A bilaterally symmetrical multi-layer perceptron neural network architecture with evolvable weights is used to model the "brains" of the agents. Evolutionary programming is employed to evolve the predator for developing adaptive strategies to meet its goals. To study the effect of learning on evolution, a self-organizing map (SOM) is added to the architecture, it is trained continuously and all the predators can access its weights. The results of these two different approaches are compared.*

## 1. Introduction

The ambiguity in determining clearly defined events to provide rewards and/or punishments in reaching the goal and the large number of strategies available to predator and evader has resulted in the application of evolutionary techniques to evolve the strategies. Organisms having neural networks as "strategy engines" to translate the sensory inputs into actions resulting in the achievement of the goal based on the Braitenberg vehicles [1] has been used extensively by researchers. There have been increasing attempts to understand these adaptive mechanisms by using artificial simulation models to replicate these behaviors. The predator's primary objective is to capture the evader and evader's objective is to avoid becoming the prey. As a result both predator and evader have to develop strategies to defeat each other and thus stay alive. As this goal is considerably more abstract than

simple goals like "don't bump into wall" or "seek light stimulus" the development of a strategy is a very complex process. Due to this reason and the enormous number of strategies involved, evolutionary techniques have been used to develop a large body of possible strategies.

Researchers have investigated different versions of this problem. Cliff et al. [2][3] developed simulated agents undergoing competitive co-evolution to evolve predator-evader strategies where the agents developed their sensory-motor mechanisms by morphogenesis. Reynolds [4] used genetic programming to develop agents that undergo competitive co-evolution to play the game of tag. Nolfi et al. [5][6] presented a neural network based evolution strategy to demonstrate the effects of competitive co-evolution between the predator and evader which they implemented using Khepera robots. Gomez et al. [7] proposed an incremental evolution approach to the problem. This involved evolving neural network architectures with incremental evolution for simple tasks whose complexity is increased till a strategy is evolved. However the aspect of competitive co-evolution is not addressed and the strategy evolution is restricted to the predator.

## 2. Approach

Based on the model of intelligence proposed by Albus [8], the modules that interact in an intelligent system are: a) sensory input, b) sensory processing, c) value judgment, d) world model, e) action generation, f) actuators. Therefore, for the agent the objective is to develop the modules (b) through (e) that translate the sensory inputs into a strategy to capture the evader. Since the agent is assumed to be autonomous it has no *a priori* information about the environment. An attempt to independently model these modules and their interactions would be very complex and more importantly go against the spirit of autonomous

evolution [5]. The approach we used is based on the ideas presented by Braitenberg [1]. The “brain” of the robot consists of a bilaterally symmetrical multi-layer perceptron neural network. In this perspective, evolving neural networks provide a promising improvement opportunity [9]. An aspect of special interest is the effect of learning on the efficiency of the evolution.

In this paper, we compare the strategy evolution of the predator using a pure evolution approach with assisted evolution. To do this a neural network with evolvable weights is developed. A second neural network in the form of a Self-Organizing Map (SOM) is added to the architecture to study this effect of learning on strategy evolution. Results obtained from these two approaches are compared.

### 3. The Robots

Two Khepera robots and Khepera Simulator 2.0 [10] are used for implementation of predator-evader problem. Khepera is a miniature mobile robot of circular shape (Figure 1). Its design is well suited for experiments with autonomous agents. Khepera robot is provided with 8 proximity sensors and its actuator system is composed of two motors. The proximity sensors take on values from 0 in the absence of an object in its proximity and 1023 at maximum excitation. Each motor takes on a speed from -10 to +10. Though not provided in the simulator, the predator is provided with a virtual “eye” that performs a simple function of detecting the absence or presence of the other robot in its line of sight. A sighting is said to have occurred when the condition is met:

$$|\phi - \alpha_{pred}| < \theta$$

where

$$\phi = \tan^{-1}[(Y_{pred} - Y_{prey}) / (X_{pred} - X_{prey})]$$

$\alpha_{pred}$  is the orientation of the robot w.r.t. horizontal axis.  $2\theta$  is the angle of vision and is taken to be 30 degrees. A confined static environment in the form of a simple rectangular area is chosen to conduct the competition. These implementation characteristics have been mainly drawn from [5][6].

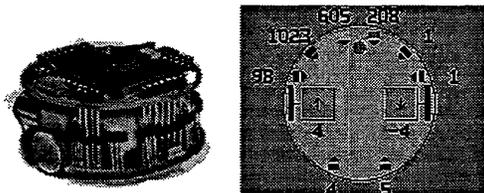


Figure 1: Khepera, the miniature mobile robot.

### 4. Neural Network

The neural network used is as shown in Figure 2. Due to the recognized advantages of having bilateral symmetry in the sensory-motor reaction apparatus [1], the neural networks on the “right” and “left” sides of the robot are identical. The networks are multi-layer perceptrons with one hidden layer and no recurrent connections. Each side is provided with 4 neurons in the hidden layer. As the “eye” is located on the midline it is uniquely connected to a separate hidden layer neuron and also symmetrically connected to both output layer neurons as shown in Figure 2. The output layer of each network has one neuron. The transfer function used is the hypertangent.

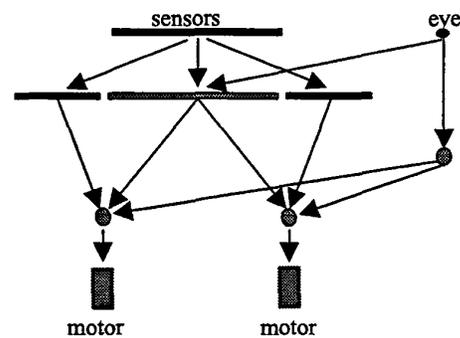


Figure 2: Neural network of predator.

With the exception of the neurons in the output layer, all other neurons are provided with a bias. The architecture is maintained constant for all the predators. The weights and biases of the network can be evolved but do not change during the life of the predator. This basically means that the predator does not possess the ability for phenotypic learning in the first phase of experiments.

In the second phase of experiments, a self-organizing map (SOM) is added to incorporate the effect of learning. SOM is used as an equivalent of a social knowledge development and transfer process. It is used to accumulate knowledge of the environment as the number of generations keeps increasing. This knowledge is more like a description rather than an accumulation of cause-effect experiences i.e. it is based on the sensory inputs not on the actions associated with them. The associated actions for each cluster is an evolved attribute and forms a part of the chromosome. The cluster information is unique and available equally to all the individuals of a generation. The SOM with all its changes is transferred to the next generation without any modification. The outputs of the SOM forms a component of the inputs to the motor actuators (See Figure 3).

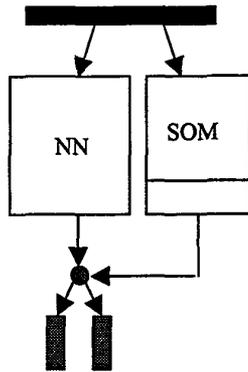


Figure 3: Incorporating learning.

### 5. Evolution Scheme

The weights and biases of multi-layer perceptron network is coded in a chromosome for each individual. The weights and biases have a range value of [-1, +1]. The reproduction is agamic, i.e. the chosen parents give birth the specified number of offspring that are identical copies of themselves that undergo mutation to varying extents. There is no crossover in agamic reproduction. This method of reproduction of chromosomes has been found to be much greater utility than the conventional sexual reproduction for the evolution of neural network [5][11]. Mutation is the only process by which the chromosome can explore the neighboring regions of the fitness landscape. Therefore mutation is not a rare occurrence but takes place every reproduction.

The fitness function consists of three components: the goal satisfaction component (G) is given by:

$$G = (\text{Max\_moves} - \text{Moves}) / \text{Max\_moves}$$

Where Max\_moves is the maximum number of moves for which the competition takes place and Moves is the number of moves after which the goal is met, i.e. prey is captured. This component is to encourage the predator to meet its goal. The directing component (D) is given by:

$$D = (\text{Number of sightings}) / (\text{Max\_moves})$$

This is to give the predator a greater tendency to follow the prey. The directing component (O) is given by:

$$O = \frac{\sum_{i=1}^{\text{Max\_moves}} |V_{(right)i} - V_{(left)i}|}{20 \cdot \text{Max\_moves}}$$

This is to discourage the predator from going around in circles. The fitness function is given by:

$$F = c_1G + c_2D + c_3O$$

where the coefficients,  $c_i$ , change as a function of the number of generations.

### 6. Experiments

Experiments have been carried out in two phases. In the first phase, the weights and biases of the multi-layer perceptron has been kept constant during life-time. These weights and biases have been evolved by a genetic algorithm. A population of 50 predators is created and each individual in the population is pitted against the evader. Each individual is given 5 trials against the evader where start positions and orientations of both are random chosen for each trial. A minimum distance of 300 units is maintained between predator and prey for all starting positions. The top 4 individuals having the highest average fitness values are allowed to reproduce asexually, producing 5 offspring each. Each of these offspring undergo mutation and replace the lesser fit individuals in the population. This new population is again placed in competition with the evaders. The competitions are allowed to occur for 25 generations.

In the second phase, the SOM is added to the architecture and its weights are updated at each move. The rest of the architecture and the evolution process remained same as in the first phase.

### 7. Conclusion

We developed the defined architecture for the predator-evader problem for Khepera robots using Khepera simulator. The approach is a combination of a biological evolution and the equivalent of social knowledge development and transfer. The preliminary results that we obtained are promising. However further experimentation needs to be conducted to prove the full potential of this approach. An aspect that requires further investigation is the application of the developed approach to a competitive evolution scenario.

### References

1. V. Braitenberg, *Vehicles: Experiments in synthetic psychology*, MIT Press, Cambridge MA, 1984.
2. D. Cliff and G.F. Miller, "Co-evolution of Pursuit and Evasion II: Simulation Methods and Results", in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, Falmouth, MA, pp. 506-515, Sept. 1996.

3. D. Cliff and G.F. Miller, "Tracking the Red Queen: Measurement of adaptive progress in co-evolutionary simulations", in *Proceedings of the Third European Conference on Artificial Life: Advances in Artificial Life*, pp. 200-218, Springer-Verlag, Berlin, 1995.
4. C.W. Reynolds, "Competition, co-evolution and the game of tag", in *Proceedings of the First Workshop on Artificial Life*, pp. 59-69, MIT Press, MA 1994.
5. S. Nolfi and D. Floreano, "Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution", To appear in *Artificial Life special issue on "Evolutionary robotics"*.
6. D. Floreano and S. Nolfi, "Adaptive behavior in competing co-evolving species", in *Proceedings of the Fourth Conference on Artificial Life*, pp. 378-387, Cambridge, MA, July 1997.
7. F. Gomez, R. Miikkulainen, "Incremental Evolution of Complex General Behavior", *Adaptive Behavior*, V.5, pp. 317-342, 1997.
8. J.S. Albus, "Outline for a Theory of Intelligence", *IEEE Trans. on Systems Man and Cybernetics*, Vol. 21, no. 3, pp. 473-509, May/June 1991.
9. S. Nolfi, D. Floreano, O. Mignino, F. Mondada, "How to evolve autonomous robots: Different approaches in evolutionary robotics", in *Proceedings of Fourth International Workshop on Synthesis and Simulation of Living Systems*, pp.190-197, MIT Press, Cambridge, MA, July 1994.
10. O. Michel, *Khepera Simulator* Package version 2.0, <http://wwi3s.unice.fr/~om/khep-sim.html>
11. L. Meeden, "An incremental approach to developing intelligent neural network controllers for robots", *IEEE Trans. on Systems Man and Cybernetics, Part B: Cybernetics*, Vol. 26, no. 3, pp. 474-485, June 1996.