Simulating exploratory behavior in evolving Artificial Neural Networks

Richard Walker
Università di Palermo
Viale delle Scienze
Palermo
Italy
rwalker3@csi.com

Orazio Miglino
Consiglio Nazionale delle Ricerche
Viale Marx, 15
00145 Roma
Italy
orazio@caio.irmkant.rm.cnr.it

Abstract

Animals released into unfamiliar environments will often engage in "roaming" behavior, apparently for exploratory purposes. It is likely that this behavior constitutes a "behavioral primitive" which can be used in the construction of more complex behaviors. This paper reports a series of experiments in which a Genetic Algorithm is successfully used to "evolve" efficient exploration strategies in a population of software simulated Khepera robots, controlled by Artificial Neural Networks. Robots based on simple perceptrons with no hidden neurons outperformed those with more complex control networks. These robots tended however to adapt to the specific environments where they had evolved. More robust behavior was obtained from robots where input from the external environment was enriched with data from cyclical "time sensors". It is suggested that control-networks based on this architecture could become a useful component in more complex systems.

1 INTRODUCTION

When a mouse is released into an unfamiliar environment it will "wander" or "roam" in an apparently random fashion. If we track the animal's path during roaming we discover that the exploration is "efficient" - the mouse rapidly covers a large proportion of the available territory (Carr & Watson, 1908). Similar behavior may in fact be found in a broad variety of species (Gallistel, 1990). It seems very likely that the function of this behavior is exploratory: "roaming" or "wandering" enables the animal to learn about the new environment, to locate sources of food or danger, to create an internal map of the terrain or to identify landmarks which can be used in subsequent navigation (Thinus-Blanc, 1996). Wandering may also be a low-level building block used in the construction of more complex behaviors. (e.g. reaching a target while performing a "detour" around an obstacle (Corbacho & Arbib, 1995)). An ability to explore unknown environments could be useful for man-made autonomous or semi-autonomous agents as well as for animals. A robot engaged in planetary exploration could search a terrain for interesting geological material without waiting for commands from earth; a software agent exploring the World Wide Web could automatically decide which site to visit next. Investigation and simulation of animal exploration strategies is thus of engineering as well as scientific interest.

The problem of how to efficiently explore an environment of unknown shape and configuration is computationally equivalent to the task of "filling" a polygon in computer graphics. The problem can be easily solved by the use of recursive techniques (e.g. "flood-fill" or "boundary fill" algorithms). It should however be noted that these algorithms require the computational system to keep track of the areas it has already visited. This approach is memory intensive. Cognitive science models positing the existence of detailed topological maps suffer from the same difficulty. Perhaps more seriously the sensory input available to robots or biological organisms, operating in real-world environments, may not be rich enough to generate high resolution location information. It is interesting therefore to investigate the feasibility of efficient exploration without the use of topological maps or similar representations.

2 RELATED WORK

"Wandering"and "roaming" by animals has been studied extensively by field biologists (Gallistel, 1990). Exploratory behavior has often been reported in labora-

tory experiments designed to explore more complex forms of behavior (Carr & Watson, 1908). It seems likely that "wandering" behavior is closely related to foraging. According to "Optimal foraging theory" animals forage "efficiently", maximizing benefits and minimizing costs (Stephens & Crebs, 1986). Other workers in the field argue, on the other hand, that actual animal behavior may often be severely sub-optimal, owing to cognitive limitations and poor perceptual input (Gould & Lewontin 1979). The debate is an interesting one with implications for general issues in theoretical biology. To the knowledge of the authors there have, however, been relatively few attempts, prior to the work reported in this paper, to model the basic computational mechanisms underlying this kind of behavior - perhaps considered too simple to deserve serious attention.

The use of simulated robots to explore cognitive mechanisms in animals was first introduced (at a conceptual level) by V. Braitenberg (Braitenberg 1984). Arbib (Arbib, 1987) has used simulated (hand-designed) robots to investigate animal exploration based on visual clues. Dorigo and Colombetti (Dorigo & Colombetti, 1994) applied evolutionary techniques in the design of physical robots with the ability to perform simple behavioral tasks (light approaching etc.). Wee Kheng Leow (Kheng Leow, 1998) has used evolutionary techniques to investigate exploratory behavior on simulated robots guided by smell

The general methodology used in our experiments is inspired by the basic techniques developed by Nolfi, Floreano, Miglino and Mondada. (Nolfi, Floreano et al., 1994). The simulator software used in the experiments was developed in previously published work by Miglino, Lund and Nolfi (Miglino, Lund et al., 1996). In this work the robot was trained to perform an obstacle-avoidance task. Other work based on the same basic model has included the modelling of other forms of obstacle avoidance (Nolfi, Miglino et al., 1994) as well as a "garbage collection" task (Nolfi & Parisi 1997) and detour behavior (Miglino, Denaro et al., 1998).

"Wandering" behavior has been investigated by Miglino, Nafasi and Taylor (Miglino, Nafasi et al.,1995) who used techniques derived from evolutionary robotics to develop an ANN-controlled mobile Lego robot trained to explore an open arena.

3 OBJECTIVES AND METHODS

The aim of our investigation was to find the simplest possible computational mechanism capable of generating efficient exploratory behavior, if possible, without resort to detailed tracking information or topological maps. In order to achieve this goal we used a Genetic Algorithm to evolve robots exhibiting the desired behavior.

The genetic algorithm operated on a population of 100 robots, simulated in software. The basic structure and behavior of the robots was based on the Khepera robot, developed at E.P.F.L. Lausanne (Mondada, Franzi et al., 1993).The simulator software was based on careful measurement of the behavior of the physical robot in real life environments (Miglino, Lund et al., 1996).

Khepera has a circular shape with a diameter of 5.5 cm. The robot's basic architecture is based on three main components: a set of sensors, a motor apparatus and a control network. In the experiments reported here the input to the control network comes from 8 infrared proximity sensors, two of which are positioned on the front of the robot, two on the back and two on each side. Each proximity sensor is sensitive to obstacles with a range of 3 cm and has an angle of vision of 20 degrees. Output values are computed as a function of the position of the robot, using data matrices collected from Khepera. In one experiment input from the proximity sensors was enriched with input from two "internal clocks" The activation value of a time sensor at time t is computed using the following algorithm:

$IF\,(activation_t < 1)\,THEN$

$\quad activation_t = activation_{t-1} + k$

$ELSE$

$\quad activation_t = 0$

For the Khepera control network we used 2 different time sensors with $k = 0.002$, $k = 0.004$.

The motor apparatus consists of a left and a right wheel driven by stepping motors which can move both forwards and backwards. With both stepping motors working at full speed the robot is capable of moving 10 cm/cycle of computation. The Control Network is an Artificial Neural Network (ANN) whose input neurons represent the state of the sensors and whose output neurons control the stepping motors. Neuron activation levels are a logistic function of total input to the neuron including input from a permanently active bias neuron.

In our experiments we tested three different network topologies: a fully connected perceptron (Minsky & Pappert 1988) with eight infrared sensors

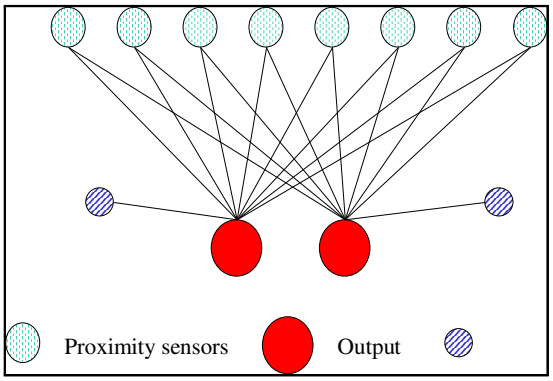

Figure 1: Fully connected perceptron

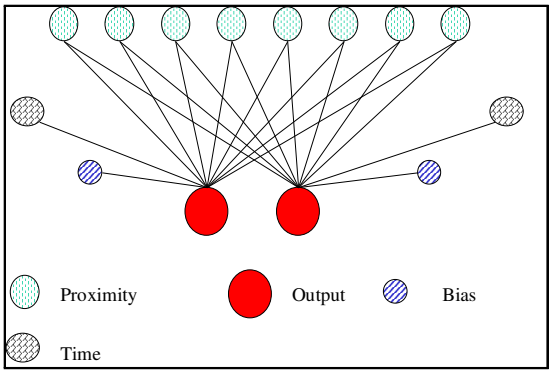

Figure 2: Perceptron with time sensors

(see Figure 1), a second fully connected perceptron with 8 infrared sensors and two internal clocks (Figure 2) and a fully-connected 3 layer feed-forward network with 5 hidden neurons (Figure 3).

Evolution involved "mutations" in the strengths of the connections linking the output neurons to the input and bias neurons. The genome of the organism consisted of a sequence of binary coded numbers (8 bits per number) representing the strengths of individual connections.

In our experiments we evolved efficient exploration strategies by applying artificial selection to the population of control networks. In the initial population connection strengths for individual networks were set to random real values uniformly distributed between -1 and 1. Each robot was placed in an "open field box" (a square or rectangular terrain surrounded by a "fence"). Robots were tested four times in a rectangular 90cm by 40 cm box and four times in a square 60 cm by 60 cm box. On each test the robot started with a randomly chosen position and orientation. For measurement purposes the terrain was divided into square

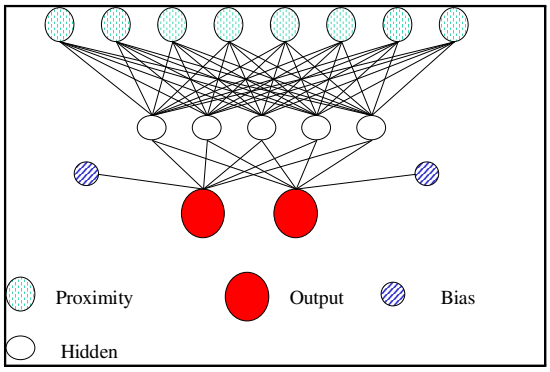

Figure 3: Perceptron with hidden units

10 cm by 10 cm cells. Each test consisted of 1,000 cycles of computation. Efficiency of exploration was measured by a fitness function which rewarded completeness and speed of exploration.

$fit = ncells + (nCycles - cyclesToCompletion)$

Where:

fit is the fitness attributed to the robot

$nCells$ is the number of different cells touched by the robot during exploraiton

$nCycles$ is the duration of the exploration (in cycles)

and

$cyclesToCompletion$ is the number of cycles traversed before the robot has touched every cell in the box. (If the robot never touches all the cells $cyclesToCompletion$ is assigned the same value as $nCycles$.)

It will be noted that the fitness function used is additive rather than multiplicative - this ensures that any increase in the number of cells traversed during exploration will translate into improved fitness even if the robot does not succeed in visiting all the cells in the box.

When all robots had been tested the 20 robots with the highest fitness scores (summed over the eight tests) were selected for "reproduction". Each of the selected robots produced 5 offspring. Reproduction was asexual. During the cloning process "mutations" were introduced by flipping bits in the genome with a probability of 0.04 per bit per generation. This process was iterated for 100 generations by which time no further improvements in fitness were observed.

In order to guarantee the statistical robustness of our results each simulation was repeated five times using

a different random number seed on each occasion.

Finally, the robots produced in each of the simulations were tested in a new environment (a 120 cm by 90 cm box) which they had not encountered during the evolutionary process. This test made it possible to measure the robustness of evolved network configurations with respect to changes in the shape of the environment to be explored. As in previous tests each robot was tested from 4 different starting positions. Given the larger size of the new environment the number of computation cycles was increased to 3,000.

4 RESULTS

4.1 OVERALL FITNESS RESULTS

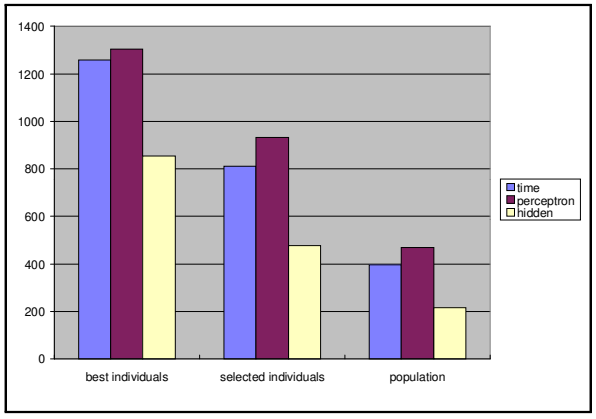

Figure 4: Mean fitness on last generation

Figure 4 shows the mean levels of fitness (over the five simulations) achieved by individuals in the last generation tested. Data is presented for the whole population, the population of individuals selected for reproduction and the best individual in the population. Analysis of variance shows that for each of these groupings the architecture with hidden neurons - the most complex of those tested - achieved significantly lower levels of fitness than the combined results for perceptron and time sensor architectures (Population$F(1498,1)=4.89$, $p<0.05$; Selected:$F(298,1)=4.35$, $p<0.05$; Best:$F(73,1)=4.91$, $p<0.05$).

Limiting the comparison to the Time Sensor and the Perceptron-based architectures the Perceptron achieved significantly higher levels of fitness in the population as a whole ($F(1498,1)=3.96$, $p<0.05$) and in the group selected for reproduction ($F298,1)=4.09$, $p<0.05$). There was however no statistically significant difference between the results achieved by the best organisms with these two architectures ($F(73,1)=3,45$, $p>0.05$).

In terms of robustness (the ability to maintain their efficiency in unfamiliar environments) the fittest robots were again those based on perceptron and time sensor networks. For clarity of presentation we will limit our statistical analysis to a comparison between the average performance for robots with these architectures.

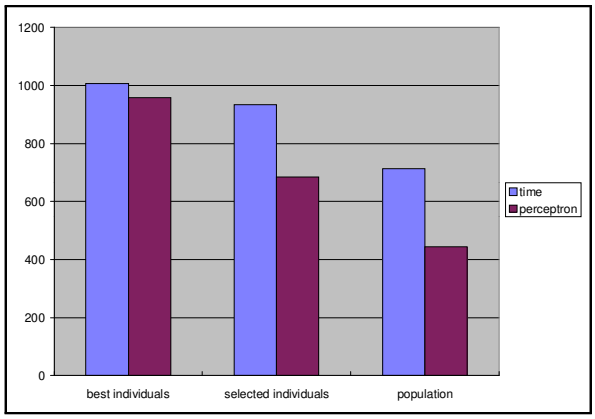

Figure 5: Mean performance in unfamiliar environments

Figure 5 compares the two architectures in terms of average performance for the whole population, robots selected for reproduction and the best performing individual. In each of these comparisons the time-sensor based robots achieve significantly higher performance than the perceptron models. (population: $t(998)=2560$, $p<0.01$; selected: $t(198)=2240$, $p<0.01$; best: $t(48)=2100$, $p<0.01$

ERROR: undefined
OFFENDING COMMAND: r

STACK:

43
132
2025
1720
0
0
0
0