

Human Breeders for Evolving Robots

Orazio Miglino¹, Onofrio Gigliotta², Michela Ponticorvo³ and Henrik H.Lund⁴

¹ Department of Relational Sciences
University of Naples "Federico II"
Naples, Italy 80133 and
Institute of Cognitive Sciences and Technologies
National Research Council
Rome, Italy 00185
E-mail: orazio.miglino@unina.it

² Department of Psychology
University of Palermo
Palermo, Italy 90100 and
Institute of Cognitive Sciences and Technologies
National Research Council
Rome, Italy 00185
E-mail: onofrio.gigliotta@istc.cnr.it

³ Department of Relational Sciences
University of Naples "Federico II"
Naples, Italy 80133
E-mail: michela.ponticorvo@unina.it

⁴ The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
Odense, Denmark DK-5230
E-mail: hhl@mip.sdu.dk

Abstract

In this paper we describe a new approach in Evolutionary Robotics according to which human breeders are involved in the evolutionary process. While traditionally robots are selected to reproduce automatically according to a fitness formula, a quantitative and strictly defined measure, human breeders can operate a selection basing on qualitative criteria, rewarding behaviors that can slip between the meshes woven by the fitness formula. In authors' opinion this may bring advantages to the Evolutionary Robotics methodology, allowing to produce robots that display more and more multiform behaviors. In order to illustrate this approach, a software, Breedbot, was developed in which human breeders can intervene in evolving robots, complementing the automatic evaluation. After describing the software, some results on sample evolutionary process are reported showing that the joint use of human and artificial selection on an exploration task generates robots with higher performance and in shorter time, compared with the exclusive action of each breeding method. The future work will explore further this hypothesis.

1 Introduction

In classical Evolutionary Robotics (ER), robots are evaluated in terms of their ability to perform a task defined by the researcher. Measurements are based on a so-called "fitness function". The robot's probability of producing offspring is proportional to the fitness measured by this function. Consequently the design of the function plays a crucial role in the success of ER experiments [7]. The problem facing ER designers is similar to a classical problem in behaviorist psychology: how to measure the efficiency of training procedures for experimental animals. The supporters of the so-called molecular approach sought to stabilize micro-behaviors which contributed to the animal's overall performance (cfr. Guthrie [1] and to some extent Pavlov [8]). By contrast, the molar approach [9] preferred to reinforce macro-behaviors leading to a satisfactory end result (e.g. finding a particular target zone, finding a way out of a maze). Experience shows that pragmatically speaking, the two approaches are complementary. Trainers have to consider the peculiarities of particular tasks and particular species of experimental animals and choose the right mix of molecular and molar techniques on this basis. This is exactly the task facing researchers in ER trying to define a fitness function. When Nolfi and Floreano [7] want to teach a robot obstacle avoidance they use a 'molecular' fitness function. In particular this fitness formula rewards, at each step, the robot of a score that is directly proportional to the speed of the wheels and inversely proportional to the differential of the wheels and to the activation value of the most activated infrared sensors. But when they want their robot to find a target zone they are no longer interested in individual actions at each step. In this case they use a 'molar' function [5]. With harder tasks they have to integrate the two approaches [6]. Outside the lab, trainers, breeders and teachers do not usually follow rigid procedures. When they reward their charges, or when they select them for breeding, they use many different criteria. Sometimes they use measurements, but often they base their judgments on qualitative, context-dependent features which are hard to capture in a mathematical formula. Typically they consider several indicators - speed, error rates, exam marks - and then decide heuristically which they are going to reward. Could we adopt a similar approach in ER? More specifically, is there some way we could integrate algorithmically-defined fitness functions with the heuristically-based decisions of a human trainer?

This issue is particularly relevant if we consider the human-robot interaction. Indeed this idea has been described and discussed in robotics in education and edutainment framework. In works by Lund and colleagues [2, 3], authors propose to allow children to select the robots they preferred, thus using what we may call a molar and implicit fitness function. In this work we went on exploring this possibility by developing Breedbot [4], an integrated hardware/software system which allows human 'breeders' to breed a small population of robots. We then put Breedbot in the hands of a small group of users, and studied the way they used the system. Breedbot can be downloaded from <http://gral.istc.cnr.it/gigliotta/breedbot.zip>. In the following pages we describe Breedbot in detail, providing a number of preliminary results.

2 Breedbot: An Environment for Breeding Robots

Breedbot is an integrated hardware/software system that allows users with no technical or computer experience to breed a small population of robots. To achieve this goal, it uses a software environment to simulate a process of artificial evolution. At the beginning of each

simulation, the computer screen shows a first generation of robots in action. After a certain time, some of the robots are selected to produce offspring. Users can let the system select the 'best robots' or make the decision themselves. If the system makes the decision, it rates the robots by their ability to explore the environment, and selects those with the highest scores. Human users, on the other hand, simply choose the robots they think have performed best. Once the selection procedure is over, the system creates clones of the selected robots. During this process it introduces random mutations into their control systems. The robots created in this way constitute a new generation. This selection/cloning/mutation cycle can be iterated until the 'breeder' finds a particularly capable robot. At this point the brain of the simulated robot can be uploaded to a real robot and the user can see how it performs in the real world. Figure 1a shows a robot which has just received a 'brain' developed with Breedbot.

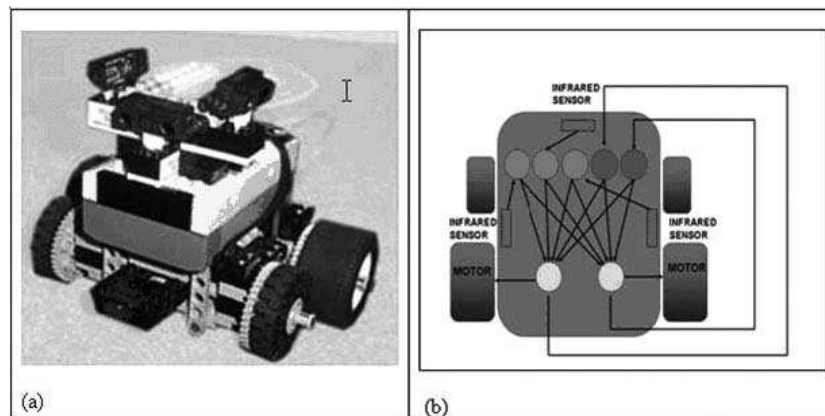


Figure 1: 1a. A picture of the robot built with the Lego MindStorms kit. 1b. A schematic representation of the robot and of its control system.

The robot is built using motors, infra-red sensors, bricks and an on-board computer from the Lego Mindstorms kit. It is rectangular in shape. Its base measures 16*15 cm and it is 10 cm high. To move, it uses two large drive wheels, each controlled by its own small electric motor. Two wheels provide stability. All the wheels are fixed so there is no steering mechanism. The on-board computer (a Lego Mindstorms RCX) and the electric power supply are located on top of the motors. The sensor system - three Mindsensor infrared sensors - is placed above the on-board computer. The first sensor is mounted half way along the robot's short side and points in its direction of motion. The other two are fixed half way along the long sides. Each sensor produces a signal with a strength inversely proportional to its distance from an obstacle. The sensors can detect obstacles up to a maximum range of 15 cm. The on-board computer implements an Artificial Neural Network (the system that controls the robot). The network receives sensory stimuli from the infra-red sensors, processes the data and activates the robot's motors. The system's neural architecture consists of a layer of input neurons and a layer of output neurons (see Figure 1b). The input neurons receive stimuli from the sensors and transmit these signals, through one-directional links ('connections') to the output neurons. Each connection is associated with a transfer value (its 'weight'). In this way, the signal arriving at the output neurons is filtered by the weights of the connections from neurons in the input layer. The input layer is made up of three sensor neurons, two proprioceptor neurons and two

bias neurons. Each infrared sensor is associated with a single sensor neuron which receives its signal and activates the rest of the network. The two proprioceptor neurons have recursive connections from the motor neurons (see Figure 1b). Thus the state of these neurons at time $t+1$ reflects the state of the motor neurons at time t . Finally, the bias neurons are always 'on' (they always have an activation of 1). These neurons, which do not receive any kind of signal from the external environment, play an essential role, ensuring that the robot is always able to move, even when receives no input from the sensors. The output layer consists of two motor neurons: these neurons determine the robot's behavior at any given moment. Each motor neuron controls an electric motor. Its output is regulated by a threshold activation function. If the sum of the inputs to the neuron is equal to or higher than the threshold, the neuron produces an output of '1'. For values below the threshold, the output is 0. When a motor receives a '1' it turns in a clockwise direction for 2 seconds. When it receives a '0' it does nothing. In this way, the robot has three possible behaviors: it can move forward for 3 cm. (when both motors are on); it can turn 10 degrees to the left (when the right motor is on and the left motor is off); it can turn 10 degrees to the right (when the right motor is off and the left motor is on). A software simulator replicates the physical characteristics of the robot and the training arena. Using the simulator we can conduct artificial evolution experiments with a population of 9 simulated robots. In terms of size, sensors, motors, and neural architecture, each individual in the population is identical to all the others. The only thing which distinguishes them is the pattern of the weights of the connections in their control systems. These are stored in their 'genotypes'. When breeding begins (in the first generation of robots), the weights of the connections are extracted randomly from a uniform distribution in the range -1 - 1. For a certain time, the robots are allowed to move freely. Then the 'breeder' (either the system or a human being) selects three robots for reproduction. Each robot's genotype (the values of its connections) is cloned three times, producing three offspring. But the clones are not perfect. During the copying process three per cent of the weights 'mutate'. The choice of which weights to modify and the new value of the modified weight are random. Figure 2 show's BreedBot's graphical interface.

The left hand side of the screen displays the behavior of the nine simulated robots in the arena, which is surrounded by walls. The right hand side provides information about the state of the system (the number of the current generation, a graph showing changes in the mean fitness of the population) along with a number of commands allowing the user to stop the system and to choose between human and artificial selection. The pull-down menu in the top left corner contains system utilities (to change the geometry of the environment, save configurations etc.). Breedbot is designed to be easy to use for breeders of small robots. Breeders can use the system's graphical interface to organize their own experiments in artificial evolution and if they want, they can select the individuals which will be allowed to reproduce. They can stop the program at any time, choose what they consider to be a well-adapted robot, and use the infrared port to upload its control system (its Artificial Neural Network) to a real Lego MindStorms robot (see Figure 3).

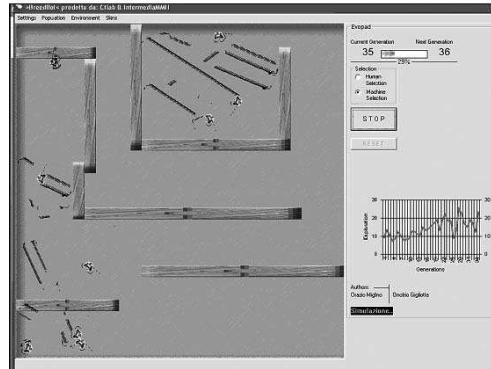


Figure 2: A snapshot representing the Graphical Interface of BreedBot software. On the left side there are the 9 robots while they explore a rectangular arena with walls. On the right side there is a display that indicates the current generation, radio buttons to choose between human and machine selection, STOP and RESET buttons and a graph, updated after each generation, that shows the mean fitness obtained in the exploration task along generations.



Figure 3: The transfer of the control system (an Artificial Neural Network) from the Digital Environment to the real Lego MindStorms robot, through the infrared port.

3 Preliminary Results

To compare the impact of human and artificial selection, we conducted three series of experiments. In the first we left selection to the system, which applied a 'fitness function', in which robot fitness was proportional to the area the robot could explore in a pre-determined period of time. In the second, we asked our human subjects to do the selection judging the robots' exploratory capabilities on purely qualitative criteria. Finally, we conducted a third series of experiments in which we used human selection in some generations and artificial selection in others. Figure 4 shows mean fitness, generation by generation, in the three experimental conditions.

As we can see from the graph, the third condition (Human and Artificial Breeder) leads to higher fitness scores.

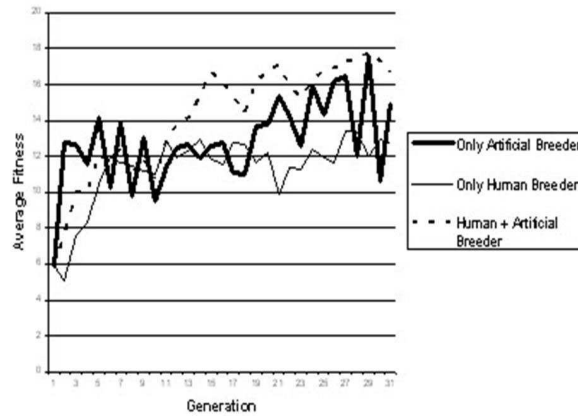


Figure 4: Mean fitness along generations for Only Artificial Breeder, Only Human Breeder and Human and Artificial Breeder.

4 Conclusions and Future Work

Although we are still in the preliminary stages of our work, the results of our experiments suggest that the combination of a pre-defined fitness function and a human selector - applying purely qualitative selection criteria - produces robots with more efficient exploratory capabilities than can be achieved with a purely algorithmic or a purely manual approach. The combined approach is also faster. We now intend to test this new methodology with larger populations, to study if it works with other fitness functions, and to see if it can be applied to more complex tasks. It will be particularly interesting to try the methodology with robots of different shapes (e.g. square robots, round robots) or with richer sensory-motor than the robots described in this paper (e.g. robots with video cameras to detect distant objects or with arms to grasp objects). As far as concerns the technique itself, we intend to investigate the selection behavior of human breeders, identifying the strategies they use. The next version of Breedbot will be designed to address these issues, creating a system that 'beginners' can use as a black box but which more expert users can customize to meet their needs, perhaps through the addition of new modules.

References

- [1] Guthrie, E.R. 1935. *The Psychology of Learning*. New York: Harper.
- [2] Lund, H.H.; Miglino, O.; Pagliarini, L.; Billard, A.; Ijspeert, A. 1998. Evolutionary robotics- a children's game In *Evolutionary Computation Proceedings IEEE World Congress on Computational Intelligence*.
- [3] Lund, H. H. 2001. Adaptive Robotics in Entertainment. In *Applied Soft Computing*, 1:1.
- [4] Miglino, O.; Gigliotta, O. 2004. Allevare robot con BreedBot In *Proceedings of the 2th National Conference of AISC*.
- [5] Miglino, O., Lund, H.H 2001. Do rats need euclidean cognitive maps of the environmental shape?. *Cognitive Processing*, 4 , pp. 1-9.
- [6] Nolfi S. 1997. Evolving non-trivial behaviors on real robots: a garbage collecting robot. *Robotics and Autonomous System*, 22: 187-198.
- [7] Nolfi, S.; Floreano D. 2000. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press/Bradford Books.
- [8] Pavlov, I.P. 1927 *Conditioned Reflexes*. Oxford, England: Humphrey Milford.
- [9] Tolman, E.C. 1922. A new formula for behaviorism. *Psyc. Rev.*, 29, 44-53.