Embodied Evolution

Richard A. Watson, Sevan G. Ficici, Jordan B. Pollack
richardw/sevan/pollack@cs.brandeis.edu
http://demo.cs.brandeis.edu

Introduction

The vision of a population of robots evolving autonomously, articulated by Harvey, 1995, (*Vision*) describes a method for evolving robot controllers that we will call Embodied Evolution, EE (*Definition*). There are several advantages for this approach to evolutionary robotics (*Motives/Background*) especially in the study of multi-agent interaction. However, it is immediately apparent that there are several problematic technological requirements for Embodied Evolution (see *Tupperbots, Continuous Untethered Power*), and there is considerable algorithmic detail that must be added to this vision before EE is workable (*Evolutionary Algorithm*). However, this work reports initial experiments (*Task Environment*) that have realized this vision successfully (*Experimental Results*) - providing proof of concept for Embodied Evolution.

Vision

Imagine a warehouse full of robots, clambering over each other, attempting to perform some task - say collecting objects representing food or energy. Imagine, these robots can mate with each other, i.e. exchange genetic material, producing control programs that become resident in other members of the robot population. Naturally, the likelihood of producing offspring is regulated by the ability to perform the task, or the 'energy' collected. Ideally there will be no need for human intervention either to evaluate, breed or reposition the robots for new trials. Thus the population of robots evolve hands-free - an artificial population that evolves autonomously.
c.f. Harvey, 1995.

Definition of EE

We define embodied evolution, EE, as evolution taking place in an embodied population [Brooks, 1990]. Natural evolution is indeed an example of embodied evolution - but in our present context we will be referring to EE that takes place in populations of artificial creatures. Since our definition specifies embodied individuals we exclude simulated creatures, and so for our purposes EE is evolution taking place in a population of robots. We will also stipulate that the evolutionary algorithm is distributed in the population as it is in natural evolution. That is, an architecture that maintains and manipulates the specifications of the individuals in the population in a centralized manner is excluded. Evaluation and reproduction must be carried out by the robots and between the robots in an autonomous manner.

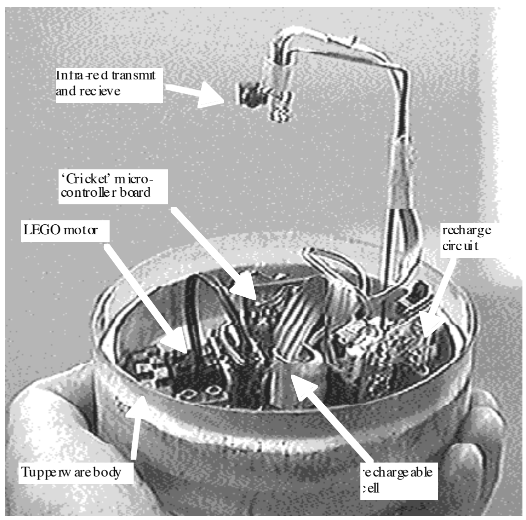
Motives/Background

Other Evolutionary Robotics work has used a single robot that evaluates many controllers in serial (e.g., Harvey, Husbands & Cliff 1993, Floreano and Mondada 1996). This method avoids all problems of transference that evolving controllers in simulation can suffer from (Mataric & Cliff, 1996). But serial evaluations are time consuming even if they can be performed hands-free. It is conceivable that the process could be speeded-up by using many robots evaluated in parallel together with some centralised algorithm coordinating reproduction and download of new controllers (although, this would require the duplication of the task environment). Our EE experiments so far are essentially embodied evaluations carried out by a population of robots in a shared environment (avoiding duplication of the task environment). However, the EA is also fully-decentralised (*Evolutionary Algorithm*) and therefore avoids potential bottleneck problems that may arise if a centralised algorithm were scaled to hundreds of robots.
More importantly, EE provides a platform for evolving group behaviours. EE is ideal for experiments investigating cooperation/competition, coevolution, group formation, self-organisation, etc. in multi-robot domains. For these areas of research it is highly unlikely that simulators could model the physical interactions accurately enough for

successful transference, and it is arguable that the complexities of modeling the environment, especially for high resolution sensory apparatus (e.g. vision), would make simulation slower than real time.

Tupperbots

- minimal design robots
- approx. $100 each
- 'Cricket' microcontroller board (provided by the Media Lab at MIT, [Martin 1998])
  - PIC based microcontroller
  - infra-redcommunications
  - byte arithmetic
  - 2 inputs, 2 outputs
  - max. program size 1536 bytes
  - serial EEPROM, program stays resident
- local range omni-directional infra-red robot-robot communications
- Tupperware body
- power pick-ups (*Continuous untethered power*)

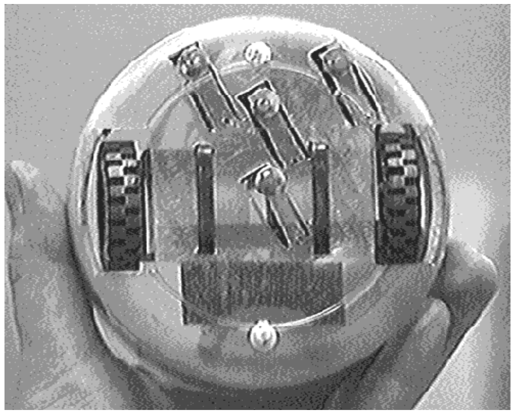


A Tupperbot

Continuous Untethered Power

- stainless steel tape fixed to modular tongue and groove flooring
- strips connected to DC power supply alternate strips to positive and negative
- each robot has four contact points on the underside (made of thumb tacks)
- geometry of contacts provides circuit in any orientation or position
- generalised rectifier provides DC to robot
- Tupperware body and contact springs made in one piece
- rechargeable cell delivers power when contact with floor is lost, and charges from floor when contact is good

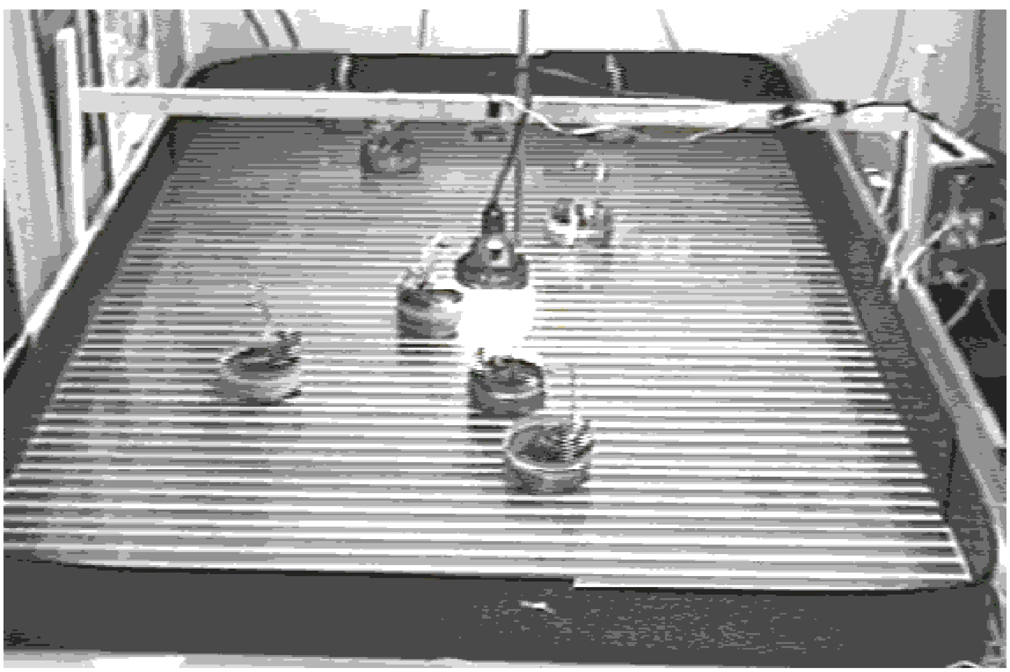
Whilst building our power floor we learned of two other groups that have built floors of a similar construction - Keating 1998, Billeter 1998.



Tupperbot underside showing the four contact points that collect power from the floor.

Task environment

- Braitenberg-style light-seeking behavior [Braitenberg 1984]
- 130cm x 200cm pen with light at center
- task implicitly includes handling interference with other robots and being able to move around enough to reproduce effectively
- an infra-red beacon above the light signals the robot that it has arrived and triggers a built-in reset behavior: i.e. orient to face the light, backs-up to edge of the pen, perform random turn
- random turns are also invoked when the robot appears to be stuck: i.e. when sensors have not changed significantly for several time steps



The robot pen for the phototaxis experiments: showing 8 Tupperbots, the power-floor and the light in the center. Our eight robots are hardly a 'warehouse full' (*Vision*) - but they have been sufficient to illuminate a significant number of issues in this endeavor.

Control architecture

- simple feed-forward neural network
- 4 integer weights [-7,8]
- 1 bit input {1 if left sensor brighter than right sensor, 0 - otherwise}
- output motor speeds [-8, 8] : i.e. 8 speeds forward or reverse

Evaluation

- each time a robot succeeds in finding the light its ‘energy’ is increased by a fixed amount
- energy is consumed by reproduction (*Evolutionary Algorithm*) so energy levels are a reflection of the frequency with which the robot reaches the light

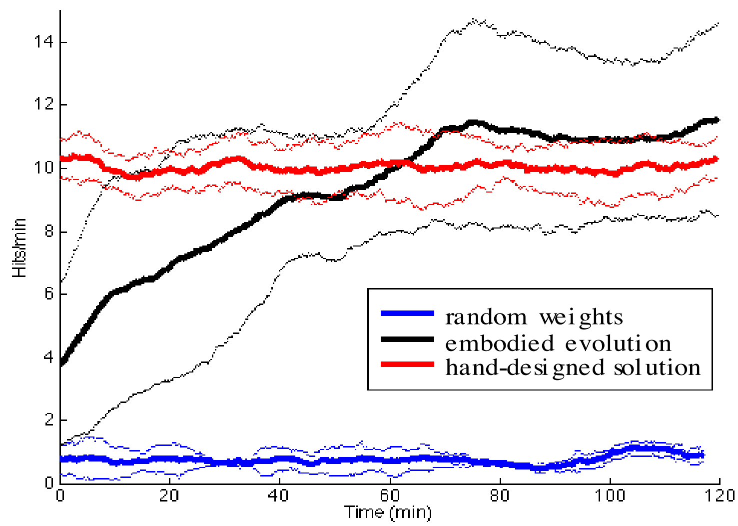
This quite trivial task and control architecture were chosen for our first proof of concept experiments and were sufficient to illustrate the overall architecture of the experiment and technology.

EvolutionaryAlgorithm

- no individual learning, robots only get new weight values from other robots during reproduction
- robots exchange genetic material (the weights of their networks) when they move within infra-red range (i.e. when less than about 4cm apart)
- each robot sends genes on its infra-red probabilistically at a rate proportional to its energy level (see *Evaluation*). The value sent is a mutation of its own weight value (mutated adding {-1,0,1} with equal probability)
- each robot upon receiving a gene from another robot accepts it to overwrite its own with a probability proportional to 1 - its own energy level
- this probabilistic model allows one-byte communications
  - when sending a robot does not need to know if anybody will receive the gene, let alone who or what their fitness is
  - when receiving a robot does not need to know who sent the gene or what their fitness was
  - avoids all problems of cross-talk and unreliability of multi-byte communications
- implements probabilistic version of Microbial GA [Harvey, 1995] cross-over
- when a robot sends a gene its energy level is decremented, and since its rate of sending is proportional to its energy level a robots energy level and rate of reproduction decays exponentially over the time from last reaching the light
- completely decentralised reproductive algorithm

Experimental Results

The figure below shows the frequency with which the light is successfully reached by the robot population (‘hit rate’) over time. All experiments used a population of 8 Tupperbots. The embodied evolution experiment evolves the weights from an initial condition where all weights are zero. We also tested a hand-designed solution which (after considerable ‘tuning’) achieves approximately 10 hits per minute. Embodied evolution produces a population with a performance that exceeds the hand-designed solution after about one hour. Interestingly, the evolved solutions exhibit qualitatively different behavior from our hand-designed solution: we designed a Braitenberg-style [1984] ‘swagger’, but evolution favoured a spiraling solution.



Mean hit rate over time for embodied evolution and two control experiments; hand-designed weights, and random weights. EE and designed data is averaged over 6 runs, random averaged over 2 runs. The dotted lines are +/- 1 standard deviation. A time window of 20 minutes is used to compute the instantaneous hit rate for each data point on the graph.

A 'seeded' embodied evolution demonstration

- Tupperbots and a section of the power-floor
- the task environment
- an evolved light-seeking behaviour
- the propagation of genetic material via inter-robot communication
    - one robot (marked with yellow) is a 'seed'; it is given weights that were evolved in one of our EE experiments
    - the other robots start with weights at zero and receive weights from the seed and each other as the demo progresses
    - to make the demonstration stable mutation is not used
- our real EE experiments were not seeded (*Experimental Results*) - the seed is used here for demonstration only

Further Information

Full implementation details of our experiments and discussion can be found in our forthcoming technical report, "Embodied Evolution," available on the DEMO web site. http://demo.cs.brandeis.edu

References

- Billeter 1998, Jean-Bernard Billeter, personal communication, Laboratoire de microinformatique (LAMI), Switzerland.
- Braitenberg, V. (1984) Vehicles: experiments in synthetic psychology. MIT Press.
- Brooks 1990 R. Brooks. Elephants don't play chess. Robotics and Autonomous Systems, 6(1--2):3--15, 1990.
- Floreano, D. and Mondada, F. Evolution of Homing Navigation in a Real Mobile Robot. IEEE Transactions on Systems, Man, and Cybernetics--Part B: Cybernetics, 26(3), 396-407, 1996.
- Harvey, I. (1995) University of Sussex, U.K., Personal communication.
- Harvey, I: The Microbial Genetic Algorithm, Submitted, under review.

- Harvey, P. Husbands and D. Cliff: "Issues in evolutionary robotics" in: J.-A. Meyer, H. Roitblat and S. Wilson (eds.), From Animals to Animats 2: Proc. of the Second Intl. Conf. on Simulation of Adaptive Behavior, (SAB92), pp. 364--373. MIT Press/Bradford Books, Cambridge MA, 1993.
- Keating 1998, Dave Keating, personal communication, D.A.Keating@reading.ac.uk
- M. Mataric and D. Cliff ``Challenges in Evolving Controllers for Physical Robots'' invited paper for Robotics and Autonomous Systems special issue on ``Evolutional Robotics'', 1996.
- Martin 1998, see http://fredm.www.media.mit.edu/people/fredm/projects/cricket/