Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots

Richard A. Watson
richardw@cs.brandeis.edu

Sevan G. Ficici
sevan@cs.brandeis.edu

Jordan B. Pollack
pollack@cs.brandeis.edu

Dynamical and Evolutionary Machine Organization
Volen Center for Complex Systems, Brandeis University
Waltham Massachusetts USA
www.demo.cs.brandeis.edu

Abstract- We introduce Embodied Evolution (EE) as a methodology for the automatic design of robotic controllers. EE is an evolutionary robotics (ER) technique that avoids the pitfalls of the simulate-and-transfer method, allows the speed-up of evaluation time by utilizing parallelism, and is particularly suited to future work on multi-agent behaviors. In EE, an evolutionary algorithm is distributed amongst and embodied within a population of physical robots that reproduce with one another while situated in the task environment. We have built a population of eight robots and successfully implemented our first experiments. The controllers evolved by EE compare favorably to hand-designed solutions for a simple task. We detail our methodology, report our initial results, and discuss the application of EE to more advanced and distributed robotics tasks.

1 Introduction

Our work is inspired by the following vision. A large number of robots freely interact with each other in a shared environment, attempting to perform some task—say the collection of objects representing food or energy. The robots mate with each other, i.e., exchange genetic material, producing (offspring) control programs that become resident in other members of the robot population. Naturally, the likelihood of a robot producing offspring is regulated by its ability to perform the task or collect 'energy.' Further, there is no need for human intervention either to evaluate, breed, or reposition the robots for new trials—the population of robots evolves hands-free. Many substantial technological demands are made by this vision, and considerable algorithmic detail must be added before it is workable.

We have developed this vision (to our knowledge first described in [Harvey, 1995]) into a methodology we call embodied evolution (EE). We define embodied evolution as evolution that takes place in a population of real robots, and we stipulate that the evolutionary algorithm is to execute in a distributed and asynchronous manner within that population. Thus, we distinguish EE from methods that serially evaluate candidate controllers on a single robot as well as algorithms that maintain and manipulate the specifications of individual agents in a centralized manner. We wish to create a population of physical robots that evolve autonomously as well as perform their tasks autonomously. This paper introduces our implementation of embodied evolution and reports results of initial experiments that provide the first proof-of-concept.

2 Motives and Related Work

The EE methodology is motivated by three different research areas. We view EE as an artificial life experiment, as an evolutionary robotics (ER) tool, and, in particular, as a substrate for the evolution of collective robotics behaviors.

2.1 Artificial Life

The adaptive mechanism of natural evolution is completely decentralized and distributed. Evaluation is implicit and reproduction is carried out autonomously by the agents in the population—not at the bequest of some centralized authority. The artificial life literature provides several examples of simulated systems where agent behavior and reproductive activity are integrated [Werner and Dyer, 1991, Fontana, 1991, Ray, 1991, Ventrella, 1998]. In these systems, agent behavior either impacts reproduction directly, or, in some cases, is synonymous with reproduction. These experiments enable researchers to explore the critical effects that result from the merging of reproductive behavior with other behaviors. In contrast, experiments that use physical robots have not been able to integrate reproduction with other autonomous behaviors. Although some evolutionary robotics has used real robots for evaluation of individuals, the evolving population is virtual—a set of controllers centrally stored either off-board or on-board—and so reproduction can not occur between two robots. A significant motive for our EE research is to implement, in a population of real robots, artificial evolution using the distributed and autonomous properties of natural evolution. We wish to employ the ideals of autonomy and distributed control not only in the task behavior of robots, but in their adaptive mechanism as well.

2.2 Evolutionary Robotics

Evolutionary Robotics (ER) seeks to offer an alternative to the hand-design of robotic controllers [Cliff et al., 1993, Husbands and Harvey, 1992]. ER sometimes uses real robots

(typically one or a small number) to evaluate all the controllers that arise during evolution [Harvey et al., 1993, Floreano and Mondada, 1994, Floreano and Mondada, 1996, Nolfi, 1997]. But, evaluating controllers serially on real robots is time consuming, even if the evaluations can be performed without human supervision. Accordingly, the large number of evaluations required for evolutionary algorithms makes simulation an attractive method for the evaluation of candidate controllers. Unfortunately, a lack of fidelity in the simulator can lead to problems of *transference*; that is, controllers evolved in simulation do not account for the subtleties in the physical characteristics of the robots or the task environment and fail when transferred to real robots [Brooks, 1992, Mataric and Cliff, 1996].

Transference problems can provably be eliminated through careful design of the simulator [Jakobi, 1997a, Jakobi, 1997b], but only by the assumption that the environmental factors critical for the task are known. Distributed robotics applications are particularly problematic in this regard because such critical environmental factors may be difficult to ascertain due to the complexity of the environment and the tightly-coupled interactions of a large number of robots. Even when known, the complexity of modeling these environmental factors, especially for high resolution sensory apparatus (e.g., vision), may make simulation slower than real time. Yet, without the help of simulation the large numbers of evaluations required for evolutionary techniques seems prohibitive. EE is our response to the dilemma between fidelity and speed. Embodied evolution does not use simulation and therefore avoids transference completely, and EE uses a large number of robots to parallelize the evaluation process, thereby providing speedup.

2.3 Collective Robotics

Distributed robotics systems pose serious challenges to established controller-design methods. Distributed control is easy to achieve if the decomposition of a problem is known and the problem sub-parts are neatly separable into independent tasks; in such a case, we build an independent autonomous agent for each sub-problem (using either hand design or machine learning). The structures of most real-world problems, however, are neither known *a priori*, nor composed of neatly separable sub-parts. As a result, much work to-date in collective robotics focuses on restricted cases, such as systems that are composed of homogeneous and independent subsystems, for example flocking and foraging. Typically, agents in such experiments use hand-built (non-learning) controller architectures [Beckers et al., 1994, Balch and Arkin, 1995, Rus et al., 1995, Donald et al., 1997]. Work that does involve learning typically occurs in simulation [Tan, 1993, Littman, 1994, Saunders and Pollack, 1996, Balch, 1997], or in relatively simple physical domains/environments [Mahadevan and Connell, 1991, Mataric, 1994a, Mataric, 1994b, Parker, 1997, Uchibe et al., 1998].

The difficulties of accomplishing highly coordinated multi-robot behavior in complex interactive domains provide the third area of motivation for EE. To date, evolutionary robotics has not addressed collective tasks in real robots (nor, for that matter, in simulation) because of the many technical and engineering challenges involved, such as the need for continuous power and the difficulty of coordinating multiple robots. As robot populations become larger (on the order of hundreds or thousands) and deployed in more complex environments, the less tenable a centralized evolutionary algorithm becomes; communication bottlenecks arise with a centralized evolutionary algorithm and synchronized evaluation and reproduction become difficult.

However, EE does not use a centralized evolutionary algorithm. Our definition of EE stipulates that the adaptive mechanism must be distributed. This distinguishes embodied evolution from the mere parallelization of embodied evaluations using a large number of robots (which would have no algorithmic distinction from existing work in ER). As an intrinsically population-based method where robots adapt in the task environment, embodied evolution potentially offers an ideal substrate with which to study emergent group behavior and explore mechanisms that adaptively discover problem decomposition. As well as providing a substrate for studying distributed behavior, the distributed architecture of EE ensures that the adaptive mechanism also adheres to the ideals of scalability and robustness. Finally, EE has the potential to be used where agents must evolve while deployed "in the field"—an issue not usually included in ER goals, but an important consideration for the long term.

2.4 Unifying ALife, ER, and Collective Robotics

Embodied evolution provides a framework that begins to unify artificial life, evolutionary robotics, and collective robotics. Each of these areas provide motives for embodied evolution, and together formulate a long-term goal for their integration.

In summary, several issues are problematic for current ER methods when applied to multi-agent domains:

- We are interested in the interaction of many agents, but current ER methods scale poorly, and
- We need to evaluate a large number of candidate controllers, and it takes too long to perform these evaluations serially on a real robot, yet
- We need to carry out evaluations in real robots to avoid transference problems.

These apparent difficulties can be turned to our advantage by embodying an evolutionary algorithm in a population of robots that are situated in a single, shared environment:

- EE is a population-based method, which provides a large number of agents, and its distributed architecture scales well.
- By using a large number of robots we perform a large number of evaluations in parallel.

- Because we use real robots, there is no transference to cause problems. The interaction between agents occurs without the computational overhead of simulation and with perfect fidelity. We use the real world to act as "its own best model" [Brooks, 1991].

3 Implementing Embodied Evolution

Our first experiments in embodied evolution require that we construct a population of robots, a continuous power delivery system, and a distributed evolutionary algorithm. Here, we review each of these in turn. We also note the revised role that simulation takes in our work.

3.1 A Population of Robots

Embodied evolution requires a larger number of robots than that used in any evolutionary robotics work to-date. The short-term proof-of-concept experiments (described in the next section) require only minimal capabilities of each robot. Similarly, the long-term objectives of EE emphasize the interaction of robots rather than the sophistication of individual robots. Accordingly, we have built a population of simple robots of our own design that are quite minimal in their individual capacity yet have the necessary capabilities for EE. Our robots employ the "Cricket" micro-controller board, supplied by the MIT Media Laboratory [Resnick et al., 1997], which uses a PIC micro-controller. Shown in Figure 1, each robot measures 12cm in diameter and has two light sensor inputs and two motor outputs as well as local-range omnidirectional infra-red communication.

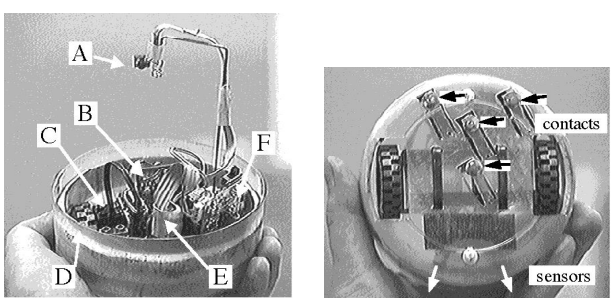

Figure 1: (Left) The robot design used in our initial EE experiments. The directional infra-red diodes are directed vertically downwards and use reflectance off the floor to achieve local omnidirectional communication. A: Infra-red transmit/receive; B: PIC micro-controller; C: Lego motor; D: Tupperware body; E: Rechargeable cell; F: Recharge circuit. (Right) Robot underside showing the two light sensors and four contact points that collect power from the floor.

3.2 Continuous Power Technology

The power requirements for embodied evolution demand a novel power delivery system. Battery power is able to sustain a robot only for a period on the order of hours, often no more than two or three [Brooks, 1992]. Longer periods of uninterrupted power can be achieved by either tethering a robot directly to a power source [Mondada and Floreano, 1996], or by providing battery recharge stations for the robot to visit periodically. Nevertheless, tethers easily tangle with only a few robots, and recharge stations can not be made transparent with respect to the robotic task, as they force robots to interrupt their activity for non-trivial amounts of time. We have developed and refined an alternative method that transparently provides continuous, untethered power.

Our robots run on a powered floor that is constructed with modular interlocking panels. Each panel has a number of strips of stainless-steel tape that alternately connect to the positive and negative poles of a DC power supply. Each robot has four contact points on the underside of its body, shown in Figure 1 (right). The geometry of the contacts guarantees that at least one point can make contact with each pole of the DC supply, regardless of the rotation or translation of the robot on the floor. The power drawn from the robot's contact points is rectified and delivered to the robot's controller and motors. Power is also sent to a circuit that maintains a small rechargeable cell, which is used only in the event of momentary loss of contact with the floor. While building our powered floor, we learned of two other research groups that have built floors of similar construction [Martinoli et al., 1997, Keating, 1998]. These parallel efforts attest to the viability and utility of this power supply approach. Other approaches [AAIS, 1998], like earlier prototypes of our own, use a floor and ceiling "bumper-car" style set-up.

3.3 A Distributed Evolutionary Algorithm

The principal components of any evolutionary algorithm are evaluation and reproduction, and both of these must be carried out autonomously by and between the robots in a distributed fashion for EE to scale effectively. Because the process of evaluation is carried out autonomously by each robot, some metric must be programmed in. This can be quite implicit, for example, where failing to maintain adequate power results in "death" [Mondada and Floreano, 1996]. Or, it can be explicitly hard-coded, for example, where fitness is a function of objects collected and time. Whatever metric is used, performance must be monitored by the robot itself, as no external observer exists to measure a robots ability explicitly.

Reproduction in EE must also be both distributed and asynchronous. Assuming that we cannot really create new robots spontaneously, the offspring must be implemented using (other) robots of the same population. And, if the robots do not have structurally reconfigurable bodies, reproduction must simply mean the exchange of control program code.

In general, selection in an evolutionary algorithm may be realized by having more-fit individuals supply genes (i.e., be parents) or by having less-fit individuals lose genes (i.e., be replaced by the offspring) or by a combination of both. The Microbial GA [Harvey, 1996] uses this observation to simplify the steady-state genetic algorithm; rather than pick

two (above-average fitness) parents and produce an offspring from the combination of their genes to replace a (below-average) third, the Microbial GA selects two individuals at random and overwrites some of the genes of the less fit (of the two) with those from the more fit. In effect, the less fit of the two becomes the offspring.

3.3.1 Probabilistic Gene Transfer Algorithm

We have developed a decentralized and probabilistic version of the Microbial GA for use in EE that we call the Probabilistic Gene Transfer Algorithm (PGTA). This method of reproduction is particularly valuable for evolutionary robotics because it requires that only two robots meet for a reproduction event to occur. Genetic information thus travels via local reproduction events, according to the locations and movements of the robots. In the PGTA, each robot pursues reproductive activity concurrently with its task behavior—there is no "reproduction mode" as such.

Each robot maintains a virtual energy level that reflects the robot's performance at its task and each robot probabilistically broadcasts genetic information on its local-range communication channel at a rate proportional to this energy level. Each broadcast contains a mutated version of one randomly-selected gene from the robot's genome. If another robot receives the broadcast, that robot may allow the received gene value to overwrite its own corresponding gene. The receiving robot accepts the broadcast gene with a probability inversely related to its own energy level. Robots with higher energy thus attempt to reproduce, and resist the reproductive attempts of others, more frequently than robots with lower energy. But, because sending and receiving is probabilistic, and genes are picked at random, the PGTA does not guarantee that a fitter robot will transfer all its genes to a less fit robot. On average each is left with a mixture of genes in proportion to their relative energy levels. This approximates a fitness-proportionate recombinative evolutionary algorithm.

In the PGTA, a broadcasting robot is unaware of who, if anyone, is within range of the broadcast—there is no need to coordinate a reproduction event between two robots. Notice also that each robot's reproductive actions are modulated only by their own energy levels—the robots do not need to know each other's energy levels. The only data broadcast are genes—no robot identifiers or energy values are exchanged. Each reproductive event involves only minimal unidirectional communication, making the algorithm very resilient to genes "dropped" in communication. Overall, the PGTA (summarized in Figure 4) provides a parsimonious algorithm suitably robust for implementation in a population of robots.

3.4 The Role of Simulation in EE

One of the primary benefits of EE is that it eliminates the difficulties of the simulate-and-transfer method, frequently used in ER. Nevertheless, we acknowledge that simulation is a valuable tool, used by researchers in many disciplines to gain insight and understanding of complex systems. In this spirit, we have built a simulator of the EE system. The aim of our simulator is not to provide a high-fidelity simulation of our robots and their environment; it is not part of our methodology to transfer solutions from the simulator to the actual robots. Rather, the simulator serves as a testbed for our evolutionary algorithm and the setup of our experiments. Our simulations of the nascent EE system provided the first indications of its viability, and helped identify critical factors in our approach. We investigate the setup of our experiments with the aid of simulation and then re-implement all experiments from scratch in the real robot population.

4 Experiments and Results

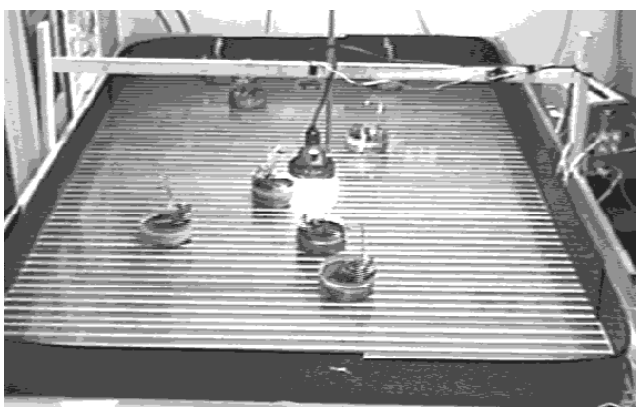

Figure 2: The robot pen for the phototaxis experiments. Eight robots, the power-floor, and the light in the center are shown. The unique ID of a robot is collected when it reaches the light (via infra red receivers on the overhead beam). This data is time-stamped and stored for monitoring experiment progress.

4.1 A Phototaxis Task

Our first embodied evolution environment employs eight of our robots. The behavior of a robot is controlled by a simple artificial neural-network architecture, the weights of which are evolved to perform phototaxis similar to that described in [Braitenberg, 1984]. The task environment consists of a 130cm by 200cm pen with a lamp located in the middle, visible from all positions on the floor plane, as seen in Figure 2. The robot task is to reach the light from any starting point in the pen. An infra-red beacon mounted above the light signals a robot when it reaches the light source and triggers a built-in reset behavior that moves the robot to a random position and orientation along the periphery of the pen, from where the robot recommences its light-seeking behavior. A second built-in behavior, which turns the robot in-place by a random angle, is invoked by a robot when its sensors indicate that it might be physically stuck, i.e., when its sensor readings have not changed significantly for several time steps. These two built-in behaviors operate independently of the evolving neural-network controller. Because

the pen contains a multitude of robots, the *de facto* environment also includes some amount of robot-to-robot interference [Schneider-Fontán and Mataric, 1996]; therefore, the task implicitly requires that each robot also successfully overcome this interference.

4.2 Control Architecture

Our initial experiments use a simple artificial neural-network control architecture to serve as the evolving substrate, depicted in Figure 3. The weights of the network are evolved. The network consists of two output nodes, one for each of the two motors, one binary-valued input node, which indicates which of the robot's two light sensors is receiving more light, and one bias node. Being a fully-connected feed-forward architecture, there are four weights. Each weight has an integer value in the range [-8, 7]. The values sent to the output nodes (controlling motor speed and direction) are the weighted sum of the input nodes; no sigmoid function is used. This network is simple enough to be computable by the PIC microcontroller in real time, yet provides a non-trivial search space of 16^4 network weight configurations.

As no individual learning takes place in our experiments, robots only get new weight values from other robots during reproduction, which is performed via local broadcasts on the robots' infra-red communications channel. The range of a broadcast is such that a robot may communicate with any other robot when the peripheries of their bodies are less than about 4cm apart.

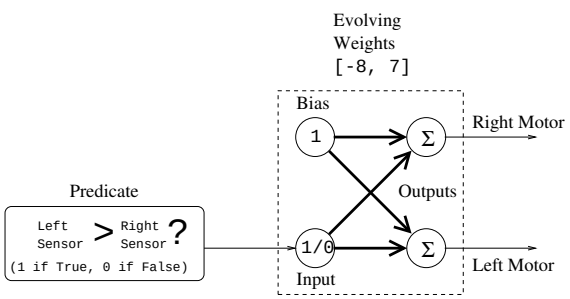

Figure 3: Control architecture for phototaxis experiment. The one-bit input is 1 if left sensor is brighter than right sensor, 0 otherwise; the bias node has constant activation of 1.

4.3 Maintaining Reproductive Energy Levels

Energy levels regulate reproduction events and should reflect the robots performance at the task. In our experiments, a robot's energy is increased only when it reaches the light and is decreased only when it broadcasts a gene. Since the robot's rate of sending genes is proportional to its energy level and decrements occur with each send, the rate of broadcasting decays exponentially over the time from its most recent visit to the light. The more frequently a robot reaches the light, the higher its energy level is likely to be at any instant (up to the saturation point defined by the maximal allowed energy). The energy level thus approximates a leaky integral of the robot's performance at its task (i.e., the frequency with which it reaches the light). Figure 4 provides an overview of how the reproductive energy levels are maintained in our experiments and how the PGTA is integrated with the robots' other behaviors.

```
define embodied_evolve
  initialize_genes[]
  energy = min_energy
  repeat forever
    if (excited?)
      send(genes[random(num_genes)] + mutation)
    if (receptive? and received?)
      genes[indexof(received)] = valof(received)
    do_task_specific_behavior
    energy = limit(energy + reward - penalty)
  endrepeat
enddefine
```

Figure 4: Pseudocode of control program that implements the Probabilistic Gene Transfer Algorithm (PGTA). This code is run on every robot. No methods for synchronizing or coordinating the robots, nor any centralized elements, are used in the PGTA. The predicates **excited?** and **receptive?** are probabilistic functions of **energy**. **send** takes a gene value and broadcasts it on local infrared (wrapped with gene locus). **received?** is *true* if any gene received on infrared. **indexof** and **valueof** return the locus and value of received gene, respectively. **limit** bounds the energy value between **min_energy** and **max_energy**. **random** returns an integer in the range of its argument. **task_specific_behaviour** includes monitoring performance at the task and setting the values of **reward** and **penalty**. In our phototaxis experiments, **min_energy** is 10; **max_energy** is 255. **excited?** returns *true* if **energy** $>$ **random(max_energy)**, *false* otherwise; **receptive** returns *true* if **energy** $<$ **random(max_energy)**, *false* otherwise. Each gene, **genes[1..4]**, is a weight value for the network. **initialize_genes** sets all genes to 0. **mutation** returns $\{0, 1, -1\}$ with uniform probability. **task_specific_behavior** includes reading sensor values, updating network outputs, setting motor speeds/directions accordingly, monitoring sensor readings and performing random turn if robot appears to be stuck, and monitoring for arrival at the beacon. **reward** is set to 127, if the robot detects the beacon, 0 otherwise, and **penalty** is set to 1 whenever the robot broadcasts a gene, 0 otherwise.

4.4 Experimental Results

Figure 5 shows the frequency with which the light is successfully reached by the robot population over time in each of three experiments. The main experiment evolves the neural-network weights to perform the light-seeking task. The initial condition of the networks in the evolution experiment is that all weights have a value of zero (this configuration produces no output to the motors and provides a neutral starting point). The other two experiments are controls where the robots do not evolve; in one case the robots' weights are random values,

and in the other the robots use weights of a hand-designed solution. As Figure 5 shows, the two controls show a broad range of possible performance levels and provide useful references against which to judge the success of the trials where evolution takes place. We see that embodied evolution allows the population of robots to achieve performance favorably comparable to that of our hand-designed solution. Though the robots learn to approach the light in a multi-robot environment, they are able to perform effectively in isolation, as well. These results provide the first evidence that a fully decentralized, asynchronous evolutionary algorithm can operate effectively in a population of physical robots and provide high-quality control programs. Moreover, these results are achieved using a crude measure of performance that does not average over many trials. In fact, the energy level is an odd representation of performance compared to the usual meaning of "fitness." A robot's energy level is not reset when the robot receives a new specification during a reproduction event and is therefore a measure of the performance of the various controllers that have been resident on that robot.

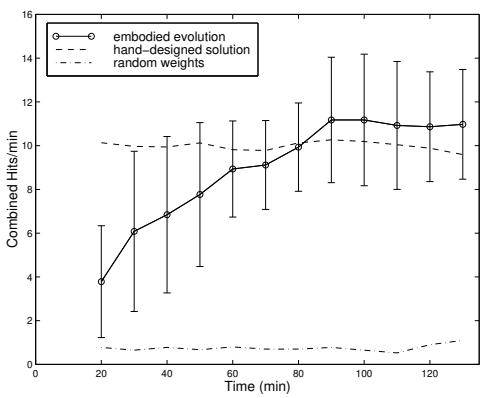

Figure 5: Average hit rates over time. Three curves show performance of the robot population using hand-designed (non-evolved), evolved, and random (non-evolved) network weights. The data from the hand-designed and evolved experiments are averaged over six runs, while the data from the random-networks experiment are averaged over two runs. Each run lasts 140 minutes and uses eight robots. The vertical axis represents the average rate (in hits per minute) at which robots reach the light. A time window of 20 minutes is used to compute the instantaneous hit rate for each data point on the graph (hence the first data points appear at Time = 20 minutes). Error bars on the evolved run, shown every 10 minutes, show +/- one standard deviation. Though the evolved solutions begin with network weights of zero, we see that the robots achieve an average performance of four hits per minute within the first twenty minutes of the experiment and eventually meet the hand-built hit rate.

Despite its minimal structure, the artificial neural-network control architecture used in the robots allows a surprising variety of solutions to be discovered by the evolutionary process. Interestingly, the best evolved solutions exhibit behaviors that are qualitatively different from our hand-designed solution; evolution appears to favor a "looping" solution, whereas, with our hand-designed solution, the robot "swaggers" to the light, as shown in Figure 6. The reasons for this are not known and we intend to address this in future work.

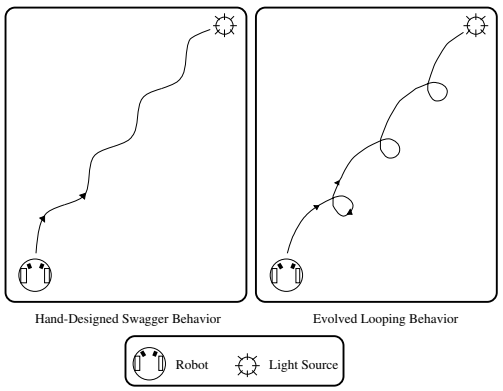

Figure 6: Trajectories of light-seeking solutions.

5 Future Work and Conclusions

5.1 Future Work

There exist a number of control experiments that will help us map the parameter space of the PGTA. Through these controls, we expect to refine the PGTA and understand more precisely the dynamics of the algorithm and the settings that provide the most robust operation. For example, simulation suggests that good solutions are not stable in the population if we remove the robots' ability to resist the reproductive attempts of others. The resistance model we use, while effective, is not known to be optimal. Other parameters we will investigate include the rates at which a robot's energy is increased and decreased as it reaches the light and attempts reproduction, respectively. We are in the process of developing more complex task environments and control architectures for our experiments, beginning with a recurrent version of the network architecture that will operate on raw sensor inputs. Though the phototaxis task described in this report is simple and does not to involve explicit robot interaction, the transparency of this domain allows us to investigate the strong implicit interactive forces within the EE approach. For example, the reproductive process and physical robot-to-robot interference are two types of interaction that we are currently investigating before moving to a more complex task.

Although the performance of the looping behavior (discovered by evolution) appears slightly more effective than the (hand-built) swagger behavior, this result is not statistically significant with the data collected to-date. If this result should prove reliable, one question we hope to answer is why the looping, which seems less efficient, is more effective than the swagger. One hypothesis is that the looping behavior over-

comes the physical interference caused by the other robots in the pen more efficiently than does our hand-designed solution. Another hypothesis is that looping is more robust to the inevitable hardware variances that exist between the robots. Or, perhaps, we will find the cause is more mundane.

As stated, a long-term goal of distributed robotics is a method for the automatic discovery of problem decomposition and balancing local autonomy with group coordination. By employing a large number of robots together in the task environment and allowing them to evolve interactive behaviors, we avoid introducing preconceptions about how a problem should be decomposed, how many robots should be assigned to each task/sub-task, or how many groups/sub-groups will be needed. Potentially, we allow the robots to discover appropriate working groups and interactive behaviors that reflect the nature and structure of the task at hand. Achieving this will require that we address many critical issues: credit assignment, the balance of cooperation and competition, homogeneity and heterogeneity, encapsulation and modularity.

5.2 Conclusions

Embodied evolution is a new methodology for evolutionary robotics. EE uses a population of robots that evolve together while situated in the task environment. The adaptive mechanism is distributed in the population using robot-to-robot reproduction that is carried out autonomously by the robots. Evolutionary adaptation is seamlessly integrated with the robot's task behavior. Our experiments in EE have employed a population of eight robots that are supplied continuous power via an electrified floor. We have developed an evolutionary algorithm that operates via the probabilistic transfer of genetic information between robots on local-range communication. This PGTA is entirely distributed and is robust in ways that make it effective for implementation in a population of robots.

EE provides a number of opportunities. Firstly, EE enables the study of the effects of integrating reproduction with other autonomous behaviors into real robots in a manner that has previously only been possible in simulated ALife experiments. Secondly, EE offers advantages over other ER methods: specifically, speed-up in time by parallelizing evaluations, and the elimination of transference problems, since all evaluations are carried out on real robots. Thirdly, EE provides a substrate for future research to investigate collective robotics behaviors. However, EE also introduces some complications from which established ER methods do not suffer; for example, because we do not use a centralized mechanism, the collection of experimental data is made more difficult. Also, because reproduction in EE is based upon the principle of locality, EE is susceptible to failure if the robots become physically, and therefore reproductively, isolated. Finally, though embodied evolution appears particularly suited to team tasks, the precise manner in which EE should be applied to team evolution is unclear—reproduction may interfere with task behavior.

Our experiments provide the first proof-of-concept for embodied evolution. We have successfully applied EE to a simple phototaxis task. The neural-network control architecture, though minimal, has a non-trivial search space and provides surprisingly novel solutions for phototaxis. Results show solutions evolved with EE to perform comparably to our best hand-designed solutions. Future experiments will provide greater clarity on the advantages and difficulties of the EE method.

Acknowledgments

The industrious contributions of several people were essential to this paper. Miguel Schneider-Fontán built the EE simulator, which allowed us to conduct the embodied experiments with confidence. Giovanni Motta designed and built the power backup and battery recharge circuit for our robots. Prem Melville administered many of the experiments and maintained the robots. Paaras Kumar designed and built an early prototype of the power backup circuit as well as the IR communication circuit for future work. Greg Hornby, Hod Lipson, and other members of DEMO challenged us with many insightful questions. We also thank Fred Martin of the MIT Media Lab, who supplied the Cricket micro-controllers.

Bibliography

[AAIS, 1998] AAIS (1998). *Continuous Power Supply for Khepera*. Applied AI Systems, Inc., Kanata, Ontario, Canada. Product literature.

[Balch, 1997] Balch, T. (1997). Learning roles: Behavioral diversity in robot teams. In *1997 AAAI Workshop on Multiagent Learning*. AAAI.

[Balch and Arkin, 1995] Balch, T. and Arkin, R. (1995). Motor schema-based formation control for multiagent robot teams. In *Proceedings of the First International Conference on Multi-Agent Systems ICMAS-95*, pages 10–16. AAAI Press.

[Beckers et al., 1994] Beckers, R., Holland, O., and Deneubourg, J. (1994). From local actions to global tasks: Stigmergy and collective robotics. In Brooks, R. and Maes, P., editors, *Artificial Life IV*, pages 181–189. MIT Press.

[Braitenberg, 1984] Braitenberg, V. (1984). *Vehicles: experiments in synthetic psychology*. MIT Press.

[Brooks, 1991] Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence Journal*, 47:139–159.

[Brooks, 1992] Brooks, R. (1992). Artificial life and real robots. In Varela, F. and Bourgine, P., editors, *Proceedings of the First European Conference on Artificial Life*, pages 3–10. MIT Press.

[Cliff et al., 1993] Cliff, D., Harvey, I., and P., H. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73–110.

[Donald et al., 1997] Donald, B., Jennings, J., and Rus, D. (1997). Minimalism + distribution = supermodularity. *Journal on Experimental and Theoretical Artificial Intelligence*, 9(2–3):293–321.

[Floreano and Mondada, 1994] Floreano, D. and Mondada, F. (1994). Automatic creation of an autonomous agent: Genetic

evolution of a neural-network driven robot. In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson, S., editors, *From Animals to Animats 3*, pages 421–430. MIT Press.

[Floreano and Mondada, 1996] Floreano, D. and Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 26(3):396–407.

[Fontana, 1991] Fontana, W. (1991). Algorithmic chemistry. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II*, pages 159–209. Addison-Wesley.

[Harvey, 1995] Harvey, I. (1995). Personal communication. University of Sussex, U.K.

[Harvey, 1996] Harvey, I. (1996). The microbial genetic algorithm. Submitted.

[Harvey et al., 1993] Harvey, I., Husbands, P., and Cliff, D. (1993). Issues in evolutionary robotics. In Meyer, J.-A., Roitblat, H., and Wilson, S., editors, *From Animals to Animats 2*, pages 364–373. MIT Press.

[Husbands and Harvey, 1992] Husbands, P. and Harvey, I. (1992). Evolution versus design: Controlling autonomous robots. In *Proceedings of the Third Annual Conference on Artificial Intelligence, Simulation and Planning*, pages 139–146. IEEE Press.

[Jakobi, 1997a] Jakobi, N. (1997a). Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6(1):131–174.

[Jakobi, 1997b] Jakobi, N. (1997b). Half-baked, ad hoc, and noisy: minimal simulations for evolutionary robotics. In Husbands, P. and Harvey, I., editors, *Fourth European Conference on Artificial Life*, pages 348–357. MIT Press.

[Keating, 1998] Keating, D. (1998). Personal communication.

[Littman, 1994] Littman, M. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the International Machine Learning Conference*, pages 157–163.

[Mahadevan and Connell, 1991] Mahadevan, S. and Connell, H. (1991). Automatic programming of behavior-based robots using reinforcement learning. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI '91)*, pages 8–14.

[Martinoli et al., 1997] Martinoli, A., Franzi, E., and Matthey, O. (1997). Towards a reliable set-up for bio-inspired collective experiments with real robots. In Casals, A. and de Almeida, A., editors, *Proceedings of the Fifth Symposium on Experimental Robotics ISER-97*, pages 597–608. Springer Verlag.

[Mataric, 1994a] Mataric, M. (1994a). Learning to behave socially. In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson, S., editors, *From Animals to Animats 3*, pages 453–462. MIT Press.

[Mataric, 1994b] Mataric, M. (1994b). Reward functions for accelerated learning. In Cohen, W. and Hirsh, H., editors, *Proceedings of the Eleventh International Conference on Machine Learning*, pages 181–189. Morgan Kaufman.

[Mataric and Cliff, 1996] Mataric, M. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems, Special Issue on Evolutional Robotics*, 19(1):67–83.

[Mondada and Floreano, 1996] Mondada, F. and Floreano, D. (1996). Evolution and mobile autonomous robotics. In Sanchez, E. and Tommasini, M., editors, *Towards Evolvable Hardware*, pages 221–249. Springer Verlag, Berlin.

[Nolfi, 1997] Nolfi, S. (1997). Evolving non-trivial behaviors on real robots: A garbage collecting robot. *Robotics and Autonomous Systems*, 22:187–198.

[Parker, 1997] Parker, L. (1997). Task-oriented multi-robot learning in behavior-based systems. *Advanced Robotics, Special Issue on Selected Papers from IROS '96*, 11(4):305–322.

[Ray, 1991] Ray, T. (1991). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II*, pages 371–408. Addison-Wesley.

[Resnick et al., 1997] Resnick, M., Berg, R., Eisenberg, M., and Turkle, S. (1997). Beyond black boxes: Bringing transparency and aesthetics back to scientific instruments. MIT project funded by the National Science Foundation (1997-1999).

[Rus et al., 1995] Rus, D., Donald, B., and Jennings, J. (1995). Moving furniture with teams of autonomous robots. In *Proceedings of IEEE/RSJ IROS'95*, pages 235–242.

[Saunders and Pollack, 1996] Saunders, G. and Pollack, J. (1996). The evolution of communication schemes of continuous channels. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S., editors, *From Animals to Animats IV*, pages 580–589. MIT Press.

[Schneider-Fontán and Mataric, 1996] Schneider-Fontán, M. and Mataric, M. (1996). A study of territoriality: The role of critical mass in adaptive task division. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S., editors, *From Animals to Animats IV*, pages 553–561. MIT Press.

[Tan, 1993] Tan, M. (1993). Multi-agent reinforcement learning: independent vs. cooperative agents. In *Proceedings of the Tenth International Machine Learning Conference*, pages 330–337.

[Uchibe et al., 1998] Uchibe, E., Asada, M., and Hosoda, K. (1998). Cooperative behavior acquisition in multi mobile robots environment by reinforcement learning based on state vector estimation. In *Proceedings of International Conference on Robotics and Automation*, pages 1558–1563.

[Ventrella, 1998] Ventrella, J. (1998). Attractiveness vs efficiency (how mate preference affects locomotion in the evolution of artificial swimming organisms). In Adami, C., Belew, R., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 178–186. MIT Press.

[Werner and Dyer, 1991] Werner, G. M. and Dyer, M. G. (1991). Evolution of communication in artificial organisms. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II*, pages 659–687. Addison-Wesley.