ELSEVIER

Robotics and Autonomous Systems 39 (2002) 1–18

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Embodied Evolution: Distributing an evolutionary algorithm in a population of robots

Richard A. Watson*, Sevan G. Ficici, Jordan B. Pollack

Dynamical and Evolutionary Machine Organization, Volen National Center for Complex Systems, Brandeis University, Waltham, MA 02454, USA

Received 3 June 2000; received in revised form 10 May 2001
Communicated by T.C. Henderson

Abstract

We introduce Embodied Evolution (EE) as a new methodology for evolutionary robotics (ER). EE uses a population of physical robots that autonomously reproduce with one another while situated in their task environment. This constitutes a fully distributed evolutionary algorithm embodied in physical robots. Several issues identified by researchers in the evolutionary robotics community as problematic for the development of ER are alleviated by the use of a large number of robots being evaluated in parallel. Particularly, EE avoids the pitfalls of the simulate-and-transfer method and allows the speed-up of evaluation time by utilizing parallelism. The more novel features of EE are that the evolutionary algorithm is entirely decentralized, which makes it inherently scalable to large numbers of robots, and that it uses many robots in a shared task environment, which makes it an interesting platform for future work in collective robotics and Artificial Life. We have built a population of eight robots and successfully implemented the first example of Embodied Evolution by designing a fully decentralized, asynchronous evolutionary algorithm. Controllers evolved by EE outperform a hand-designed controller in a simple application. We introduce our approach and its motivations, detail our implementation and initial results, and discuss the advantages and limitations of EE. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Evolutionary robotics; Artificial Life; Evolutionary algorithms; Distributed learning; Collective robotics

1. Introduction

1.1. Vision

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.

This vision, to our knowledge first described by Husbands et al. [1], aspires to an ideal where the robot population evolves in a completely hands-free and autonomous manner; in so doing, it offers intriguing possibilities for the future of evolutionary robotics

* Corresponding author.
E-mail address: richardw@cs.brandeis.edu (R.A. Watson).

PII: S0921-8890(02)00170-7

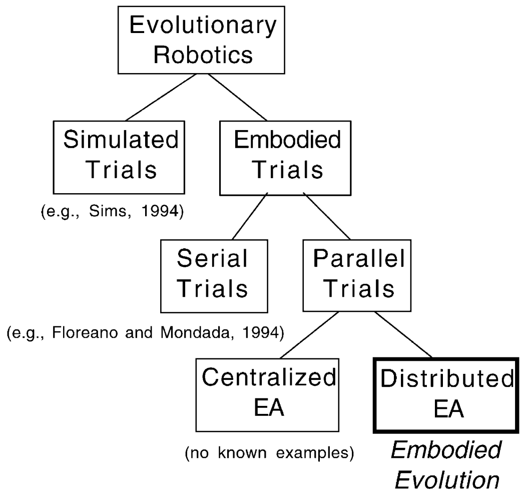

Fig. 1. Embodied Evolution is an evolutionary robotics methodology that embodies a distributed evolutionary algorithm within a population of real robots.

(ER). Nevertheless, many substantial technological demands are made by this conception of robot evolution, and considerable algorithmic detail must be added before it is implementable.

We have developed this vision into a working implementation and have termed our methodology *Embodied Evolution* (EE) [2,3]. By this we mean that the evolutionary process itself has no central locus of control, and is distributed in a population of robots—not merely that each robot has a physical body. Accordingly, we define EE as evolution taking place within a population of real robots where evaluation, selection, and reproduction are carried out by and between the robots in a distributed, asynchronous, and autonomous manner. Thus, EE follows the principles of organization outlined by Brooks [4]. As Fig. 1 shows, we distinguish Embodied Evolution from ER methods that use simulated trials, and also from methods that use embodied trials (the evaluation of candidate controllers on real robots) where either (a) only one or a small number of real robots are used [5–8], or (b) the evolutionary algorithm maintains and manipulates 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), using a distributed evolutionary algorithm.

1.2. Motivation

The following three areas each contribute to the motivation of our Embodied Evolution experiments.

1.2.1. An Artificial Life experiment

The vision of Embodied Evolution described above is largely inspired by experiments in Artificial Life. In natural evolution the adaptive mechanism is completely decentralized and distributed: evaluation is implicit and reproduction is carried out autonomously by the agents in the population. This contrasts with the common practice of artificial evolution, where reproduction is performed by some centralized authority. The Artificial Life literature provides several examples of simulated systems where agent behavior and reproductive activity are integrated together [9–12]. 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, ER experiments that use physical robots have not been able to integrate reproduction with other autonomous behaviors. Although some evolutionary robotics research 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 cannot 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 autonomy and distributed control not only in the task behavior of robots, but in their adaptive mechanism as well.

1.2.2. An evolutionary robotics methodology

Mataric and Cliff [13] identify several key problems with existing ER methods, such as simulator fidelity, or, when using real robots, evaluation time, and hardware robustness. Many efforts to deal with simulator fidelity exist, for example, the work of Jakobi [14,15]. In contrast, our intent here is to examine ways to use real robots that avoid incurring prohibitively long evaluation times and problems with hardware robustness. We argue below that one way to ameliorate the concerns of time and robustness, while avoiding the difficulty of simulation fidelity, is to use what we will term *embodied trials in parallel* (ETP)—that is, the parallel evaluation of a large number of evolving controllers on a large number of physical robots. Generally, the more robots available, the more evaluations can be done in parallel, and the better the speed-up. Additionally, such parallelization makes hardware robustness less critical than when using a single robot, and, of course, using real robots precludes the need for a simulator. However, the parallelization of the evaluation procedure converts costs in fidelity, time, and robustness to costs in hardware duplication and coordination. The economics involved in this balance determines whether ETP provides a net gain. The optimal number of robots to use in a particular case is affected by this balance, as well as a number of other factors, such as robot-to-robot interference (discussed below), and the normal issues of population sizing in evolutionary algorithms, such as diversity and convergence rates [16].

Embodied Evolution uses the ETP approach: EE presumes a large number of robots, each simultaneously evaluating a controller. But, additionally, EE employs a distributed evolutionary algorithm. Though embodied trial in parallel are, to some extent, feasible using a centralized evolutionary algorithm, scaling issues regarding coordination persist in principle; moreover, centralized ETP is not fundamentally distinct from conventional ER approaches. Specifically, centralized ETP is merely the marriage of existing ER work that uses a small number of real robots with existing EA work that farms the evaluation of individuals to parallel processors. In contrast, Embodied Evolution using parallelized embodied trials coupled with an evolutionary algorithm that is distributed throughout the population of robots—with no centralized component—is both inherently scalable and methodologically novel.

1.2.3. Opportunities for interactive and collective tasks

Our most intriguing, yet currently unexplored, motivation for EE arises from the fact that, since EE uses a distributed EA in a population of real robots, it may provide an effective way to investigate interactive and collective tasks. Pursuit and evasion, and collective foraging are but two examples of domains that would naturally be amenable to the use of EE. Also, EE provides interesting opportunities for experiments that investigate the interaction of reproductive behaviors and task behaviors. We develop these points in Section 4.

1.3. Paper outline

Our first motive, outlined above, is essentially an issue of curiosity, but the latter two motives warrant additional detail—this follows in Sections 2–4. Section 5 summarizes the suitability and utility of the EE approach. Section 6 describes our implementation of EE. Section 7 describes our experiments and presents results that show the first proof-of-concept for EE. Finally, Section 8 outlines some caveats regarding our approach, followed by conclusions in Section 9.

2. Embodied trials in parallel: a response to challenges in evolutionary robotics

As an alternative to the hand design of robotic controllers, evolutionary robotics [5,17,18] has generated a great deal of exuberance. While expected by many to emerge as an important technology [19], ER is currently a new research area in which a number of potentially serious problems are known to exist. Mataric and Cliff's [13] comprehensive summary of the challenges that face evolutionary robotics attracted considerable attention in the ER community. The issues they raise ultimately question whether ER techniques can produce a net savings of human effort when applied to complex robotic domains.

Our continued investment in developing ER techniques, and the EE methodology in particular, is done with the belief that problem domains of interest will soon be, if they are not already, too difficult for the hand design of solutions; indeed, the seminal work in evolutionary robotics was done with this realization in mind [18]. That evolutionary techniques have the potential to find unexpected, yet effective solutions is their real promise.

Nonetheless, for further progress to occur in ER, we must at least address the issues Mataric and Cliff discuss. Here we present these issues and argue that one way to alleviate them is to use Embodied trails in parallel.

2.1. Simulation

- "The difficulty of accurately simulating physical systems is well known in robotics." [13, p. 76]
- "As the complexity of robotic systems grows and the gap between the simulation and the real system widens, the question of the value of investing in a specialized simulation will become increasingly important." [13, p. 76]

The use of simulation in ER is a common and valuable approach. When appropriate simulators are available, they can offer fast evaluation, and they do not suffer from the problems of power delivery and physical robustness, which are often problematic when using real robots (as we discuss below). However, Mataric and Cliff, in agreement with Brooks [20], raise the concern that a lack of simulation fidelity can lead to problems of *transference*, where robotic controllers that evolve in simulation are unable to perform effectively when transferred to real robots because they exploit features of the simulator that are not found in the physical world. They argue that this problem is magnified as the system to be modeled becomes more complex. Realistic models of physical worlds require significant engineering effort [21–23]. Even virtual worlds that are not intended to faithfully model the real world can become complex and difficult to create [24,25].

More recently Jakobi has devised a method to provably eliminate transference risks from the design of a simulator, but only provided that environmental factors sufficient for enabling correct behavior are known a priori [14,15]. This allows the environmental factors that are not salient to the robot's behavior, or that cannot be modeled accurately, to be masked with noise, which minimizes the computational requirements of the simulation and prevents the evolution of behaviors that depend upon inaccurate features of the simulation. However, as environments continue to become more complex (for example, if they involve a multiplicity of robots, or robots with high-resolution sensory apparatus such as vision), the critical environmental factors will become more difficult to ascertain, more difficult to model, and more computationally expensive to simulate, arguably to the extent that simulation is prohibited.

One approach to alleviate the concerns of simulator fidelity is to not use a simulator—to conduct evaluations on a real robot [6,26]. Obviously, with respect to the properties of the robot itself, embodied trials have perfect fidelity and the problem of transference is side-stepped completely. With respect to the properties of the environment in which trials take place, the use of a physical environment removes the difficulties of accurate modeling and the computational expense of simulation. In short, "the world is its own best model" [4]. The physical environment used for embodied trials may be one and the same as that in which the robots are ultimately deployed. But, if this is not the case, then the use of embodied trials still requires care in identifying the critical environmental factors, such that transference from the physical test environment to the physical deployment environment is successful. Though this difficulty should not be underestimated, the problem of modeling these factors, once identified, is an additional burden for the simulation approach. In these respects, embodied trials offer clear advantages over the use of simulation. However, evaluations on real robots can be time consuming.

2.2. Time

- "Evolution on physical systems takes prohibitively long." [13, p. 76]
- "[Consider a problem of] ... five free parameters, each of which is tested with four different values, [and where] ... each trial takes 15 s [seconds] ... With a population of size 100 ... the 100th generation will finish in roughly five years." [13, p. 80]

The issue of time raised by Mataric and Cliff is in contrast to the (supposed) speed of simulation. Though simplifying assumptions and stochastic approximations minimize simulator complexity, they do not eliminate it; a sufficiently complex environment can still cause simulation to run slower than real time. Trials on real robots, although they may be slower than approximate models of simple domains, are never slower than real time and they have perfect fidelity.

Nevertheless, serial evaluation of candidate robot controllers on a single robot can take a very long time indeed. Hardware parallelization can alleviate this concern for both simulated and embodied approaches. Even in Mataric and Cliff's example scenario above, which they characterize as "exaggerated", a population of 100 physical robots reduces five years to 18 days (dividing five years by population size). However, where virtual robots perform interactive tasks in a simulated shared environment, parallelization is not trivial: for example, collision detection and mutual-sightedness can easily incur high computation costs. Parallelization in physical robots, however, accommodates shared environments and interactive tasks naturally.

Thus, concerns over simulation fidelity and evaluation time combine to direct us towards the use of embodied trials in parallel. As we point out above, the time savings gained by the ETP approach obviously come at the cost of significant hardware duplication. Yet, as large-scale multi-agent domains become increasingly important arenas of research, large numbers of robots will become commonplace; ETP is a straightforward way to exploit the ubiquity of hardware as it becomes available.

2.3. Power

- “The unavoidable need to recharge robot batteries further slows down the experimental procedure.” [13, p. 76]

The issue of power pervades robotics. Battery power is able to sustain a robot only for a period on the order of hours, often no more than two or three [20]. Longer periods of uninterrupted power can be achieved by either tethering a robot directly to a power source [27], or by providing battery recharge stations for the robot to visit periodically. Nevertheless, tethers easily tangle with only a few robots, and recharge stations cannot be made transparent with respect to the robotic task, as they force robots to interrupt their activity for non-trivial amounts of time.

However, though the issue of power continues to raise problems, we do not view it as a fundamental restriction to ER. There are, at least in some cases, technological solutions to this issue. For example, in our experiments, we use an electrified floor to provide power to our robots. While building our floor, we learned of two other research groups that have built floors of similar construction [28,29]. These parallel achievements attest to the viability and utility of this power supply approach. Other approaches [30], like earlier prototypes of our own, use a floor-and-ceiling ‘bumper-car’ style setup. Together these examples demonstrate acceptable solutions for laboratory conditions. Given these technologies, we see that electrical power need not be a restriction to the use of parallelized embodied trials. However, we concede that, except perhaps for some industrial settings, the use of such power technologies presumably implies that the physical environment in which trials occur is not identical to the actual deployment environment. While technologies such as the powered floor are important to the implementation of our experiments, we do not consider them to be an intrinsic part of the EE methodology.

2.4. Robustness

- “. . . a robotic system cannot survive the necessary continuous testing . . . ” [13, p. 76]

Robots used in research are rarely endowed with the robustness that is engineered into industrial robots, usually for reasons of economy of development time or expense. As a result, research robots demand almost constant care and attention to keep them in operational order. The robots we built for our experiments are no exception in this respect.

Nevertheless, the population of robots that ETP uses is a valuable source of redundancy, which allows the performance of the evolutionary system to degrade gracefully with the number of robot failures. While we concede that physical failures are inevitable for long running times, especially when many robots are involved in physical interaction, hardware parallelization reduces the amount of run-time per robot by a factor equal to the size of the robot population. Thus, the ETP approach appears to provide a feasible way forward for ER.

2.5. Other issues

We mention here, for completeness, that Mataric and Cliff raise several other issues that pertain to any method of machine learning, or even to hand-design methods. While our EE methodology is silent on these points, they are important to recognize. One such point in particular concerns the creation of an effective metric of agent success; all automated learning methods, including Embodied Evolution, require feedback to function. The question of how researchers are to construct good metrics of behavior for autonomous robots, especially as environments become more complex and interactive, is undeniably important and will continue to require special attention.

3. Embodied Evolution: the transition from ETP to scalable ETP

We have examined a number of core issues raised by Mataric and Cliff and have argued that embodied trials in parallel on a large number of robots may be one approach to alleviate their concerns. If we are to take the ETP approach seriously, however, we must address a new concern that does not arise in the methods considered by Mataric and Cliff, namely the *scalability* of the evolutionary algorithm. The scalability of ETP using a centralized evolutionary algorithm is limited, as is that of any centralized control system. For example, centralized coordination of evaluation and reproduction can potentially result in communication bottlenecks [28] as the number of robots increases. Hence, we aspire to decentralization in the adaptive mechanism in the same way that Brooks aspires to decentralization in the control of individual robots.

Accordingly, we arrive at the suggestion of an ER methodology that uses a large population of robots, performing evaluations in parallel (i.e., ETP), but that also implements a distributed evolutionary algorithm. This conforms to our definition of Embodied Evolution. The use of an evolutionary algorithm that has no centralized component—that is distributed throughout the population of robots—is inherently scalable, in principle enabling the use of hundreds or perhaps thousands of robots. As we state in the introduction, we wish to create a population of physical robots that evolve autonomously as well as perform their tasks autonomously. In this way, we adhere to the goals of autonomy and distributed control [4] now common in the field of robotics, not only in the task behavior of the robots, but in their adaptive mechanism as well.

4. Opportunities for interactive and collective tasks

The previous two sections assert the benefits of Embodied Evolution from the perspective of an evolutionary robotics methodology. In this section, we emphasize potential advantages of EE that are neither directly motivated by the issues that Mataric and Cliff raise, nor by the need for scalability. Specifically, since EE requires the use of a large number of robots in a shared task environment, it offers an appropriate substrate for experiments in which robot interaction is of interest (for example, pursuit and evasion tasks). Similarly, it offers a substrate for collective problem solving where robots must work together to solve some task (for example, collective foraging or box pushing).

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 may 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/or independent sub-systems, for example, flocking and foraging. Typically, agents in such experiments have used hand-built (non-learning) controller architectures [31–34]. Work that does involve learning typically occurs in simulation [35–38], or in relatively simple physical domains/environments [39–43].

The difficulty of accomplishing highly coordinated multi-robot behavior in complex interactive domains provides an additional area of motivation for the development of EE. To date, evolutionary robotics has not addressed collective tasks in real robots 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 and deployed in more complex environments these challenges become more acute.

Although the experiments we describe below do not involve an interactive task, nor collective behaviors, we suggest that EE may provide an appropriate substrate for future work in these areas. As an intrinsically population-based method where robots adapt in the task environment, EE appears potentially well-suited for the study of emergent group behavior and mechanisms that adaptively discover problem decomposition. This research direction returns to a theme from our ALife-inspired motivations, specifically, the integration of reproductive and task-oriented behaviors. Behaviors that modify reproduction can be used as a tool to enable speciation and heterogeneous teams/collective behaviors. This can be enabled by placing reproductive behaviors explicitly under the adaptive control of the robots, allowing certain environmental cues to trigger reproductive events. Alternatively, segregated reproduction may occur via some subtle interaction with task behaviors. This is not as esoteric as it may seem. The reproductive events in our implementation of EE (which we shall detail shortly), are subject to the co-locations of the robots, and thus an experimental setup not much more complicated than ours may enable reproductive 'preference' to be implemented simply by a behavioral disposition to move to one area of the task environment more often than another area.

Although much can be learned about these topics from studies that use simulation, the caveats regarding simulation fidelity, originally raised with respect to the task domain, apply to the 'reproductive domain', as well. For task behaviors, subtleties afforded by the physical substrate may be precluded by the use of a simplified or approximate simulation. Similarly, there exists the potential for subtle forms of sexual segregation to emerge in a physical substrate that may be precluded by simulation. Further, if the robots' choices for mating are expressed and determined by their behaviors, then a centralized reproductive algorithm that determines which robots reproduce with which other robots is excluded: reproductive and task-oriented behaviors are no longer categorically distinct, and a centralized EA would require an interpretive process to disentangle reproductive from non-reproductive behaviors, or else be reduced to a proxy for the robot's reproductive choices.

5. Suitability of Embodied Evolution

This section summarizes the strengths and suitability of Embodied Evolution according to the issues we have described above. We suggest that EE is suitable in any of the following circumstances. Taken together, the points below provide a strong motivation for a distributed, embodied approach:

- *Where a simulator for the task domain is impossible, unavailable, or insufficiently accurate*.

  The use of embodied trials, as in EE, allows the world to act as "its own best model" and avoids the need for a simulator.
- *Where a centralized, globally coordinated adaptive algorithm is not implementable or is unavailable, or where coordination of parallelized embodied trials is difficult*.

  The distributed architecture of EE's evolutionary algorithm is intrinsically scalable with respect to the number of robots.

In addition, the EE approach offers possibilities for future work in the following circumstances, the last two of which are applications we have not previously discussed.

- *Where we are interested in evolving interactive or collective behaviors*.

  The shared task environment utilized in EE provides a natural substrate for interactive and collective tasks. EE may be especially suitable where the interactions involve subtle complexities that are difficult to simulate.
- *Where we are interested in the interaction between task behaviors and reproductive behaviors*.

  Studies in Artificial Life that involve the interaction of task and reproductive behaviors, for example through implicit fitness metrics, may be implemented using the EE approach. Again, EE may be especially suitable where the interactions involve subtle complexities that are difficult to simulate.
- *Where the agents must learn 'in the field'*.

  One can easily imagine applications where learning must occur after the agents have been deployed in the actual task domain, for example, if the agents are to operate in a remote region (e.g. Mars or perhaps a Micro Electro-Mechanical System (MEMS) substrate). In such a case, a centralized coordinator of agent learning would be both difficult to design and perhaps precarious to use, as it would give the learning system a single point of failure. The distributed nature of EE provides a method for adaptation in the field that is robust in these respects.

- *Where agents have been evolved in simulation and must now adjust their control programs to cross the 'reality gap'.*

  Embodied Evolution can be applied as the final stage of a hybrid ER process, as described in [44]. Agents are first evolved in simulation, then downloaded into a population of physical robots where EE is used to fine-tune their control programs in the real world.

6. Implementing Embodied Evolution

This section describes the general components that implement Embodied Evolution. Section 7 gives implementation details that are specific to our experimental task.

6.1. A distributed evolutionary algorithm

The principal components of any evolutionary algorithm are *evaluation*, *selection*, and *reproduction*, and all of these are carried out autonomously by and between the robots in a distributed fashion according to our definition of Embodied Evolution.

Because the process of evaluation is carried out autonomously by each robot, some metric must be programmed into the robots with which they can measure their performance. This can be quite implicit, for example, where failing to maintain adequate power results in 'death' [27]. Or, it can be explicitly hard-coded, for example, where fitness is a function of objects collected and time. Whatever metric is used, performance against it must be monitored by the robot itself, as no external observer exists to measure a robot's ability explicitly [26].

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 we 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. Harvey's Microbial GA [45] 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. To achieve decentralized and asynchronous reproduction in EE, we have developed a probabilistic version of the Microbial GA.

6.2. Probabilistic Gene Transfer Algorithm

The Probabilistic Gene Transfer Algorithm (PGTA) is our probabilistic version of the Microbial GA. This algorithm requires minimal inter-agent communication, and eliminates the need to coordinate the communication of each reproduction event.

In the PGTA, reproduction is concurrent with task behavior—there is no 'reproduction mode' as such. Each robot maintains a virtual 'energy level', which reflects the robot's performance or fitness at the task,[1] 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 (i.e., one parameter from the robot's control specification). If another robot receives the broadcast, that robot may allow the received gene value to overwrite its own corresponding gene. The receiving robot will accept 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 do robots with lower energy. Nevertheless, 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 robots are left with a mixture of genes in proportion to their relative energy levels. This implements a fitness-proportionate recombinative evolutionary algorithm.

[1] To be clear, this energy level is not related to the electrical power level in the robot.

Using the PGTA, each gene transfer event requires only one-way communication—there is no need for robots to coordinate reproductive acts, for a robot to know the fitness or identity of another robot, or even to know that any robot received its broadcast. Thus, the PGTA naturally allows the complete decentralization of selection and reproduction. Though the PGTA provides an interesting mechanism for EAs in general, its robustness to genetic information ‘dropped’ in communication makes the PGTA particularly advantageous for implementation in a population of real robots. In a later section, we describe how the maintenance of energy levels for the PGTA is integrated into our experimental task.

6.3. 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 [46], which uses a PIC micro-controller. Shown in Fig. 2, each robot measures 12 cm in diameter, has two light sensor inputs, two motor outputs, as well as local-range omnidirectional infrared (IR) communication. The IR range is such that a robot may communicate with any other robot when the peripheries of their bodies are less than about 4 cm apart.

6.4. Continuous power technology

We have developed and refined a powered-floor technology that transparently provides continuous, untethered power to our robots, without the use of recharging stations. Our powered-floor is surfaced with strips of stainless steel tape, which are alternately connected to the positive and negative poles of a DC power source. Each robot has four contact points on the underside of its body, with which it draws power from the floor. The geometry of the contacts, shown in Fig. 3, guarantees that at least one point can make contact with each pole of the DC power supply, regardless of the rotation or translation of the robot on the floor (some examples are given in the figure). The strip a contact touches may be either polarity, according to the position of the robot on the floor, and so the power is rectified before being delivered to the robot’s controller and motors. It transpires that this method of power delivery was described by Claude Shannon in an internal memo during his tenure at the AT&T Bell Laboratories in 1950 [47, p. 678]. We add a rechargeable cell to cover intermittency in contact.

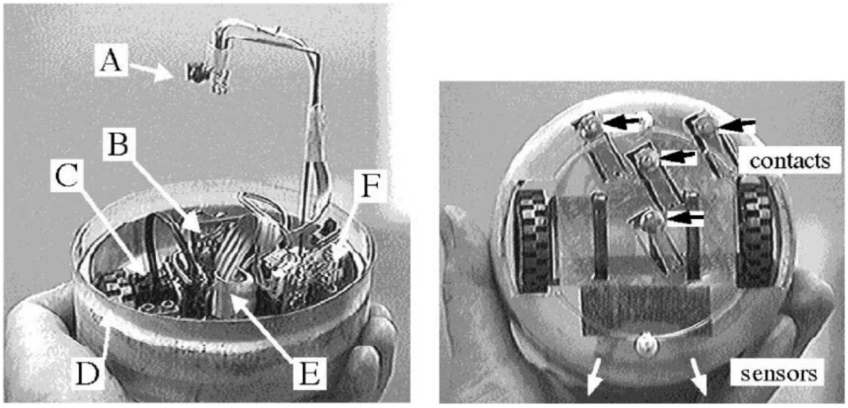

Fig. 2. (Left) The robot design used in our initial EE experiments. (A) Infrared transmit/receive. The directional infrared diodes are directed vertically downwards and use reflectance off the floor to achieve local omnidirectional communication. (B) PIC micro-controller. (C) Lego^{TM} motor. (D) Body from plastic food container. (E) Rechargeable cell. (F) Recharge circuit. (Right) Robot underside showing the two light sensors, and four contact points that collect power from the floor.

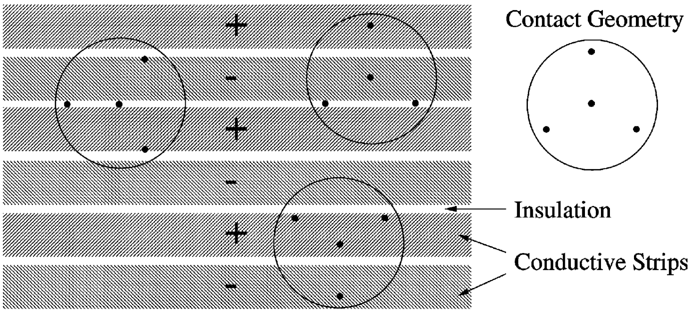

Fig. 3. Shannon's design for power delivery [47]. The floor is constructed of conductive strips (separated by a thin insulator) that are alternately connected to the positive and negative poles of a DC power supply. Power is picked up from the floor by four contacts arranged in an equilateral triangle with one in the center. This geometry guarantees contact with both poles of the power supply regardless of its rotation and translation on the floor. Since an individual contact point may be on either pole of the supply, power drawn through the contacts is rectified.

7. Experiments and results

7.1. A phototaxis task

Our Embodied Evolution experiments employ 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 by Braitenberg [48]. The task environment consists of a 130 cm × 200 cm pen with a lamp located in the middle, visible from all positions on the floor plane, as seen in Fig. 4. The robot task is to reach the light from any starting point in the pen. An IR beacon mounted above the light emits a signal that can be detected by a robot when it reaches the light source. This beacon signal 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. If a robot's sensor values do not change for some period of time, it is assumed that the robot is stuck against a wall (or another robot) and the robot invokes a second built-in behavior designed to free the robot by rotating it a random amount. Both of these 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 [49]; therefore, the task implicitly requires that each robot also successfully overcome this interference.

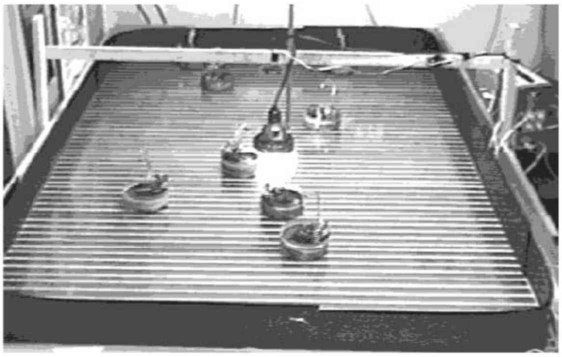

Fig. 4. The robot pen for the phototaxis experiments. Eight robots, the power floor, and the light in the center are shown. The unique ID of a robot is collected when it reaches the light (via infrared receivers on the overhead beam above the lamp). This data is time-stamped and stored for monitoring experiment progress.

7.2. Control architecture

Our initial experiments use a simple artificial neural-network control architecture to serve as the evolving substrate, depicted in Fig. 5. 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;

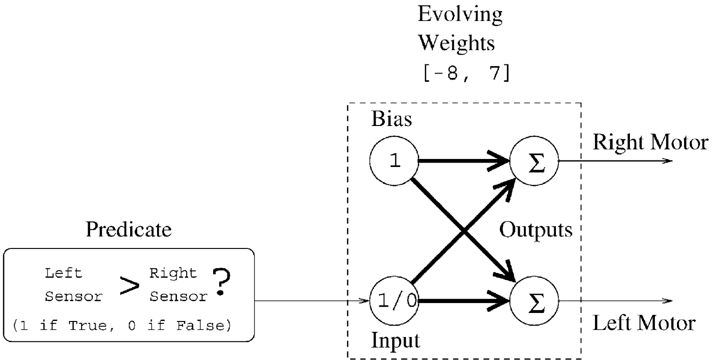

Fig. 5. Control architecture for phototaxis experiment. The one-bit input is 1 if left sensor is brighter than right sensor, 0 otherwise; the bias node has constant activation of 1.

no sigmoid function is used. This network is simple enough to be computable by the PIC micro-controller in real time, yet provides a non-trivial search space size 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 (omnidirectional signals that can be received by any robot within a short range) on the robots' IR communications channel. This limited communication range (just over one body-width in radius) combined with the freely mixed population of our shared environment essentially implements random selection of mates, as appropriate for the PGTA.

7.3. Maintaining reproductive energy levels

Energy levels regulate reproduction events and should reflect the robots' performance at the task. The virtual energy level maintained by a robot is updated as follows: whenever a robot reaches the light, its energy is increased by a fixed amount, up to a maximal energy value; whenever a robot transmits a gene for reproduction (regardless of whether another robot receives the transmission or accepts the gene), its energy is decreased by a small fixed amount, down to a minimal energy value. 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 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). Fig. 6 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.

Note that the energy level maintained by a robot is an odd representation of its performance compared to the usual meaning of 'fitness'. In our implementation, the energy level is not reset in a robot when it receives a new gene during a reproductive event—and so, the energy level is related to the performance of the various controllers that have been resident on that robot. In contrast, one would normally expect the fitness of previously resident controllers not to affect the current fitness of a robot. However, assuming that the offspring is similar to the parent, our method of using inherited energy potentially shortens the evaluation process that must occur before the fitness measurement of a new controller is reliable.

7.4. Results

Fig. 7 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 for the networks is that all weights have a value of zero (this configuration produces no output to the motors and provides a neutral starting point). Accordingly, to avoid an initial spatial configuration where robots are reproductively isolated, they are placed in a tight cluster at one end of the pen. The other two experiments are controls where the robots do not evolve. The

```
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
```

Fig. 6. Pseudo-code 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 functions and variables in the general form of the PGTA are defined as follows. 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? becomes *true* when a gene is received on infrared, and *false* once the gene value is read. indexof and valueof return the locus and value of the received gene, respectively. limit bounds the energy value between min_energy and max_energy. random returns an integer in the range of its argument. do_task_specific_behavior includes monitoring performance at the task and setting the values of reward and penalty. In the specific implementation of the PGTA for 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 one of $\{0, 1, -1\}$ with uniform probability. do_task_specific_behavior includes reading sensor values, updating network outputs, setting motor speeds/directions accordingly, monitoring sensor readings to perform a random turn if the 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.

first control experiment fixes the robots' weights to random values to provide an indication of the expected performance of random behaviors. In the second control experiment, we intend to provide an upper bound by hand-designing the neural-network weights for our light-seeking task. The optimal solution for the task depends on the particulars of our robotic hardware and task environment, and is therefore difficult to determine precisely. But, we presume a hand-designed behavior built upon the principles of Braitenberg's basic phototropic behavior [48] to be a plausible candidate. In this behavior, the robot's trajectory follows one of two curves, to the left or to the right, according to the input from the sensors (see Fig. 8, left). The exact network weights we used were the best we found from hand tuning.

As Fig. 7 shows, the two control experiments delimit 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—the Wilcoxon rank-sum test[2] indicates that EE outperforms the designed solution with probability $p = 0.935$. 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.

There are several additional points of interest in this result. First, though the robots learn to approach the light in a multi-robot environment, they are able to perform effectively in isolation, as well (not shown here). Second, despite its minimal structure, the artificial neural-network control architecture used in the robots allows a surprising variety of phototropic behaviors. Evolved behaviors include a 'looping' trajectory (see Fig. 8), where the robot continually turns in one direction, but changes the radius of the curve according to sensor input. Another evolved behavior performs a 'back-stitch' maneuver, where the robot reverses on a small arc to correct its heading. On some occasions, we have even seen the robots approach the light entirely in reverse.

In many of the runs reported here, the robots converge on the looping solution. There are many possible reasons why the looping solution may be favored. For example, the looping behavior may compose a larger proportion of the search space than other behaviors. Related to this is the possibility that the looping behavior is more robust to variation than other behaviors. Not least of all, we would expect looping to be favored by selection, since it appears to have a higher performance.

[2] The Wilcoxon test is preferred over the Student's *t*-test in this case because the variances of the designed and evolved runs are very different from each other.

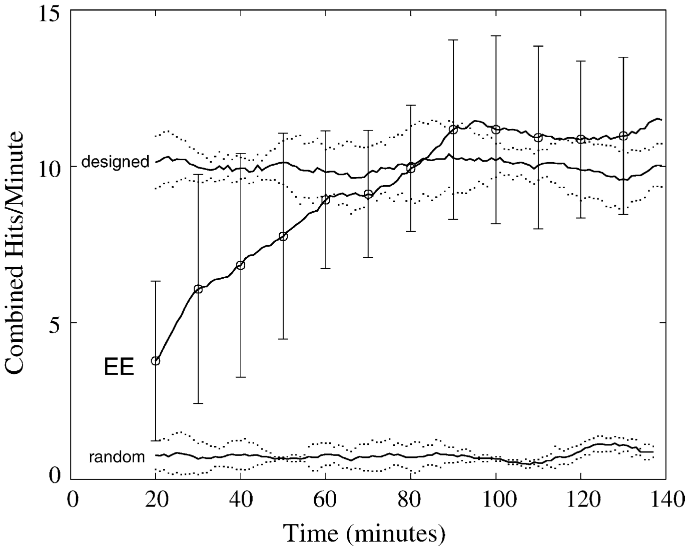

Fig. 7. Performance over time. Upon detecting the beacon's signal, a robot sends a reply signal to the beacon. These replies are then delivered to a desktop computer where they are time-stamped. Three solid curves show the performance of the robot population using hand-designed (non-evolved), evolved, and random (non-evolved) networks. The data from the hand-designed and evolved experiments are averaged over six runs, while the data from the random-solution 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 the population of robots reaches 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). Vertical bars on the evolved run, shown every 10 minutes, and the dotted lines on the control experiments, 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 exceed the hand-designed hit rate (the Wilcoxon rank-sum test gives $p = 0.935$).

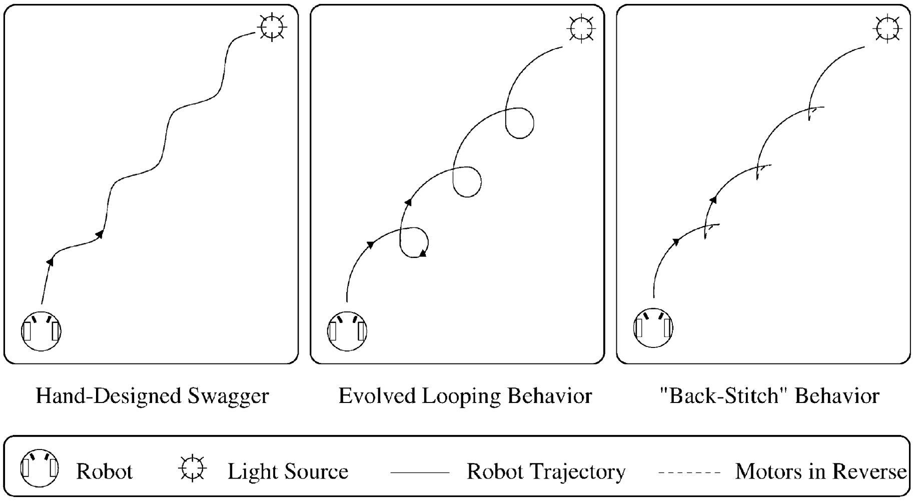

Fig. 8. Trajectories of light-seeking solutions.

That looping should outperform our hand-built swagger behavior may seem counter-intuitive, since the hand-built behavior gives a more direct path to the light. However, we must remember that the velocities on these trajectories may be different. Indeed, we had some difficulty tuning the swagger behavior because the light sensors are not sampled with high frequency; if the robot moves too fast, then it often travels too far before detecting that its orientation to the light has changed, and occasionally the robot misses the light completely. Consequently, the fastest motor speed at which we could obtain an effective swagger was less than the top speed of the motors. The looping behavior, however, has been observed to use faster motor speeds. Therefore, the looping behavior is hypothesized to provide more robust correction of robot heading, given a low-frequency sampling of the light sensors. Other reasons why looping may out-perform the hand-built swagger may be that the looping behavior overcomes the physical interference caused by the walls of the pen, or the other robots in the pen, more efficiently. 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, and dependent on our choice of algorithm parameters. These issues of light sensor sampling frequency and its interaction with motor speed, physical interference between robots, and the hardware variances between robots are exactly the sorts of reasons that make simulation and hand-design difficult. Regardless of the actual reasons for the improved performance, the point to be made is that Embodied Evolution was able to find effective solutions.

8. Caveats

In spite of our enthusiasm for the Embodied Evolution approach, we must recognize that it is still a developing methodology, and although EE offers solutions for some issues, it also introduces new difficulties.

- An environment that contains a multitude of robots also includes some amount of robot-to-robot interference [49]; e.g., our phototaxis environment implicitly requires that each robot also successfully overcome such interference. Hence, we suggest that EE is more suitable where interaction is native to the task domain.
- Our experiments to date have concerned only simple tasks for which many other learning approaches have also been effective. Though we suggest EE is suited to multi-agent systems, we are only just now designing experiments that involve explicitly interactive tasks.
- Our powered floor is constrained to research or industrial environments; other applications will require different power technologies.
- Because we eschew centralization, the jobs of monitoring our experiments and collecting data are made more awkward. With our current robotic hardware we are unable to monitor reproductive activity, for example.
- Because reproduction is based upon the principle of locality in our particular experimental setup, our implementation of EE is susceptible to failure if the agents become physically, and therefore reproductively, isolated. Moreover, the need to prevent reproductive isolation produces a selective pressure that may interfere with the objective of the task. And if, as we suggest, the reproductive mechanism is modified to allow speciation, we can imagine that reproductive behaviors could become quite elaborate, worsening this interference.
- Though the shared adaptive environment used in EE has interesting possibilities for team tasks, the precise manner in which EE should be applied to team evolution is unclear. Any requirement of locality (for reproduction) seems to suggest that an awkward overlapping of multiple teams is needed for reproduction to take place, and the mechanism for organizing games between teams is problematic.

Thus, we see that a great deal of research effort is still required to meet the long-term goals of EE, and although EE provides advantages in some domains it is not suitable for all applications.

9. Future work and conclusions

9.1. Future work

There exist a number of control experiments that will help us map the parameter space of the PGTA. We expect these controls to refine the PGTA, and provide better understanding of the algorithm's dynamics and the settings that provide the most robust operation. For example, a simulation built by Schneider-Fontán [50] 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 to investigate include the rates at which a robot's energy is increased and decreased as it reaches the light and attempts reproduction, respectively. More complex control architectures for our experiments may include a recurrent version of the neural network that will operate on raw sensor inputs. To more fully explore the merit of the EE approach, we will need to apply it to more complex domains that provide richer sensory data, and tasks that require multiple behaviors (such as navigation with obstacle avoidance).

Though the phototaxis task described in this report is simple and does not involve explicit robot interaction, the transparency of this domain allows us to investigate the strong *implicit* interactive forces that exist within the EE approach. For example, the reproductive process and physical robot-to-robot interference are two types of interaction that would be interesting to investigate before moving on to a more complex task. The hypotheses (discussed in Section 7.4) that concern why the looping behavior is selected by evolution also deserve investigation.

As we state above, a long-term goal of distributed robotics is a method for the automatic discovery of problem decomposition and the balancing of local autonomy with group coordination. By employing a large number of robots together in the task environment and allowing them to evolve collective 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 many critical issues to be addressed: credit assignment, the balance of cooperation and competition, homogeneity and heterogeneity, encapsulation and modularity.

Finally, future work may address advanced robotic platforms, including evolvable hardware [51] and evolvable morphology [52,53].

9.2. Conclusions

Embodied Evolution uses a population of physical 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.

In good part, our motivation for the development of EE was simply to see if the artificial evolution of a population of robots could be implemented in a distributed and autonomous manner. But, EE is also a new evolutionary robotics methodology that addresses several significant issues identified by ER researchers as problematic.

We must be careful to separate technological issues from fundamental ones when considering the long-term prospects for evolutionary robotics. Our implementation of Embodied Evolution reveals some concerns, such as power, to be more technical in nature. Other issues, such as combinatorics and the limits of simulation, are more fundamental and cannot be solved technologically. Nevertheless, some of these fundamental issues are ameliorated by the embodied parallelism that EE employs. Specifically, the use of embodied trails in parallel eliminates problems of simulation transference, alleviates the slow running time of numerous serial embodied trials, and provides hardware robustness. But, the more important contributions of the Embodied Evolution approach as an ER methodology are two-fold. First, it provides inherent scalability by decentralizing evaluation and reproduction. Second, it provides a substrate to explore the evolution of interactive and collective behaviors, which are particularly difficult to simulate.

Our experiments in EE have employed a population of eight robots that are supplied continuous power by an electrified floor. We have developed an evolutionary algorithm that operates through the probabilistic transfer of genetic information between robots via local-range communication. This Probabilistic Gene Transfer Algorithm is entirely distributed and is robust in ways that make it effective for implementation in a population of robots. 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 better than our best hand-designed solutions. In summary, we have provided the first proof-of-concept for Embodied Evolution—the first example of an evolutionary algorithm distributed in a population of physical robots.

Acknowledgements

The industrious contributions of several people were essential to the work reported in this paper. Miguel Schneider-Fontán built an EE simulator, which allowed us to subsequently conduct the experiments on real robots 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. Louis Lapat provided additional help with robot maintenance. Paaras Kumar designed and built an early prototype of the power backup circuit as well as the IR communication circuit to be used for future work. Greg Hornby, Hod Lipson, and other members of DEMO at Brandeis University[3] challenged us with many insightful questions. We are also grateful for careful review and suggestions from Dave Cliff, and to Fred Martin of the MIT Media Laboratory, who supplied the Cricket micro-controllers and considerable technical expertise. We would also like to acknowledge helpful reviewer comments.

References

[1] P. Husbands, I. Harvey, D. Cliff, Central issues in evolutionary robotics, unpublished manuscript; presented at ALife III, Santa Fe, NM, 1992.

[2] R.A. Watson, S.G. Ficici, J.B. Pollack, Embodied Evolution: Embodying an evolutionary algorithm in a population of robots, in: P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, A. Zalzala (Eds.), Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Press, New York, 1999, pp. 335–342.

[3] S.G. Ficici, R.A. Watson, J.B. Pollack, Embodied Evolution: A response to challenges in evolutionary robotics, in: J.L. Wyatt, J. Demiris (Eds.), Proceedings of the Eighth European Workshop on Learning Robots, Lausanne, Switzerland, 1999, pp. 14–22.

[4] R.A. Brooks, Intelligence without reason, in: B. Grosz, J. Mylopoulos, R. Reiter (Eds.), Proceedings of the 12th International Joint Conference on AI, (IJCAI-91), Sydney, Australia, Morgan Kaufmann, Los Altos, CA, 1991, pp. 569–595.

[5] I. Harvey, P. Husbands, D. Cliff, Issues in evolutionary robotics, in: J.-A. Meyer, H.L. Roitblat, S.W. Wilson (Eds.), From Animals to Animats 2, MIT Press, Cambridge, MA, 1993, pp. 364–373.

[6] D. Floreano, F. Mondada, Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot, in: D. Cliff, P. Husbands, J.-A. Meyer, S. Wilson (Eds.), From Animals to Animats 3, MIT Press, Cambridge, MA, 1994, pp. 421–430.

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

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

[9] G.M. Werner, M.G. Dyer, Evolution of communication in artificial organisms, in: C. Langton, C. Taylor, J. Farmer, S. Rasmussen (Eds.), Artificial Life II, Addison-Wesley, Reading, MA, 1991, pp. 659–687.

[10] W. Fontana, Algorithmic chemistry, in: C. Langton, C. Taylor, J. Farmer, S. Rasmussen (Eds.), Artificial Life II, Addison-Wesley, Reading, MA, 1991, pp. 159–209.

[11] T.S. Ray. An approach to the synthesis of life, in: C. Langton, C. Taylor, J. Farmer, S. Rasmussen (Eds.), Artificial Life II, Addison-Wesley, Reading, MA, 1991, pp. 371–408.

[12] J. Ventrella, Attractiveness vs efficiency (how mate preference affects locomotion in the evolution of artificial swimming organisms), in: C. Adami, R.K. Belew, H. Kitano, C. Taylor (Eds.), Artificial Life VI, MIT Press, Cambridge, MA, 1998, pp. 178–186.

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

[14] N. Jakobi, Evolutionary robotics and the radical envelope of noise hypothesis, Adaptive Behavior 6 (2) (1997) 325–368.

[15] N. Jakobi, Half-baked, ad hoc, and noisy: Minimal simulations for evolutionary robotics, in: P. Husbands, I. Harvey (Eds.), Proceedings of the Fourth European Conference on Artificial Life, MIT Press, Cambridge, MA, 1997, pp. 348–357.

[16] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[17] D. Cliff, I. Harvey, P. Husbands, Explorations in evolutionary robotics, Adaptive Behavior 2 (1) (1993) 73–110.

[18] P. Husbands, I. Harvey, Evolution versus design: Controlling autonomous robots, in: Proceedings of the Third Annual Conference on Artificial Intelligence, Simulation and Planning, IEEE Press, New York, 1992, pp. 139–146.

[19] J.-A. Meyer, P. Husbands, I. Harvey, Evolutionary robotics: A survey of applications and problems, in: P. Husbands, J.-A. Meyer (Eds.), Proceedings of the First European Workshop on Evolutionary Robotics (EvoRobot'98), Springer, Berlin, 1998, pp. 1–21.

[20] R.A. Brooks, Artificial life and real robots, in: F. Varela, P. Bourgine (Eds.), Proceedings of the First European

[3] http://www.demo.cs.brandeis.edu

Conference on Artificial Life, MIT Press, Cambridge, MA, 1992, pp. 3–10.

[21] H. Lipson, J.B. Pollack, Evolving physical creatures, in: M. Bedau, J. McCaskill, N. Packard, S. Rasmussen (Eds.), Artificial Life VII, MIT Press, Cambridge, MA, 2000, pp. 282–287.

[22] Mathengine, http://www.mathengine.com

[23] B. Mirtich, Impulse-based dynamic simulation of rigid body systems, Ph.D. Thesis, University of California, Berkeley, CA, 1996.

[24] K. Sims, Evolving 3D morphology and behavior by competition, in: R.A. Brooks, P. Maes (Eds.), Artificial Life IV, MIT Press, Cambridge, 1994, MA, pp. 28–39.

[25] M. Komosiński, S. Ulatowski, Framsticks: Towards a simulation of a nature-like world, creatures and evolution, in: D. Floreano, J.-D. Nicoud, F. Mondada (Eds.), Proceedings of the Fifth European Conference on Artificial Life, Springer, Berlin, 1999, pp. 261–265.

[26] S. Nolfi, D. Floreano, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, MIT Press, Cambridge, MA, 2000.

[27] F. Mondada, D. Floreano, Evolution and mobile autonomous robotics, in: E. Sanchez, M. Tommasini (Eds.), Towards Evolvable Hardware, Springer, Berlin, 1996, pp. 221–249.

[28] A. Martinoli, E. Franzi, O. Matthey, Towards a reliable set-up for bio-inspired collective experiments with real robots, in: A. Casals, A.T. de Almeida (Eds.), Proceedings of the Fifth Symposium on Experimental Robotics (ISER-97), Springer, Berlin, 1997, pp. 597–608.

[29] D. Keating, Personal communication, 1998.

[30] Applied AI Systems Inc., Continuous Power Supply for Khepera, Product Literature, Kanata, Ont., 1998.

[31] R. Beckers, O.E. Holland, J.L. Deneubourg, From local actions to global tasks: Stigmergy and collective robotics, in: R.A. Brooks, P. Maes (Eds.), Artificial Life IV, MIT Press, Cambridge, MA, 1994, pp. 181–189.

[32] T. Balch, R.C. Arkin, Motor schema-based formation control for multiagent robot teams, in: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), AAAI Press, Menlo Park, CA, 1995, pp. 10–16.

[33] D. Rus, B. Donald, J. Jennings, Moving furniture with teams of autonomous robots, in: K. Ikeuchi, P. Khosla, P. Dario, O. Khatib, K. Kosuge (Eds.), Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95), IEEE Press, New York, 1995, pp. 235–242.

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

[35] M. Tan, Multi-agent reinforcement learning: Independent vs. cooperative agents, in: Proceedings of the 10th International Conference on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993, pp. 330–337.

[36] M. Littman, Markov games as a framework for multi-agent reinforcement learning, in: W.W. Cohen, H. Hirsch (Eds.), Proceedings of the 11th International Conference on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1994 pp. 157–163.

[37] G.M. Saunders, J.B. Pollack, The evolution of communication schemes of continuous channels, in: P. Maes, M. Mataric, J. Pollack, S. Wilson (Eds.), From Animals to Animats 4, MIT Press, Cambridge, MA, 1996, pp. 580–589.

[38] T. Balch, Learning roles: Behavioral diversity in robot teams, in: Proceedings of the 1997 AAAI Workshop on Multiagent Learning (AAAI Technical Report WS-97-03), AAAI Press, Menlo Park, CA, 1997, pp. 7–12.

[39] S. Mahadevan, H. Connell, Automatic programming of behavior-based robots using reinforcement learning, in: T. Dean, K. McKeown (Eds.), Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91), AAAI Press, Menlo Park, CA, 1991, pp. 768–773.

[40] M.J. Mataric, Learning to behave socially, in: D. Cliff, P. Husbands, J.-A. Meyer, S. Wilson (Eds.), From Animals to Animats 3, MIT Press, Cambridge, MA, 1994, pp. 453–462.

[41] M.J. Mataric, Reward functions for accelerated learning, in: W.W. Cohen, H. Hirsch (Eds.), Proceedings of the 11th International Conference on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1994, pp. 181–189.

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

[43] E. Uchibe, M. Asada, K. Hosoda, Cooperative behavior acquisition in multimobile robots environment by reinforcement learning based on state vector estimation, in: G. Giralt, P. Dario (Eds.), Proceedings of the International Conference on Robotics and Automation, IEEE Press, New York, 1998, pp. 1558–1563.

[44] J.B. Pollack, H. Lipson, S.G. Ficici, P. Funes, G. Hornby, R.A. Watson, Evolutionary techniques in physical robotics, in: J. Miller (Ed.), Proceedings of the Third International Conference on Evolvable Systems (ICES 2000), Springer, Berlin, 2000, pp. 175–186.

[45] I. Harvey, The microbial genetic algorithm, Unpublished manuscript, 1996.

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

[47] N. Sloane, J. Alexander, A.D. Wyner (Eds.), Claude Elwood Shannon: Collected Papers, IEEE Press, New York, 1993.

[48] V. Braitenberg, Vehicles: Experiments in Synthetic Psychology, MIT Press, Cambridge, MA, 1984.

[49] M. Schneider-Fontán, M.J. Mataric, A study of territoriality: The role of critical mass in adaptive task division, in: P. Maes, M. Mataric, J. Pollack, S. Wilson (Eds.), From Animals to Animats 4, MIT Press, Cambridge, MA, 1996, pp. 553–561.

[50] M. Schneider-Fontán, Personal communication, Brandeis University, Waltham, MA, 1998.

[51] A. Thompson, Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution, Springer, Berlin, 1998.

[52] H. Lipson, J.B. Pollack, Automatic design and manufacture of robotic lifeforms, Nature 406 (August 2000) 974–978.

[53] P. Funes, J. Pollack, Evolutionary body building: Adaptive physical designs for robots, Artificial Life 4 (1998) 337–357.

Richard A. Watson received his B.A. in Computing with Artificial Intelligence, 1990, and M.Sc. in Knowledge-Based Systems, 1996, from The University of Sussex, UK. He is currently completing his Ph.D. in Computer Science at Brandeis University, USA. His publications include works on evolutionary robotics, genetic algorithms, evolutionary theory, and, in particular, coevolution and cooperative interaction.

Sevan G. Ficici has a B.M., 1988, from the Cleveland Institute of Music, USA, and an M.A. in Music Theory, 1990, from the Eastman School of Music, USA. He is currently completing his Ph.D. in Computer Science at Brandeis University, USA. He has published research in computer analysis of music, evolutionary robotics, and adaptive behavior. His more recent publications concern coevolution and evolutionary game theory.

Jordan B. Pollack is Associate Professor of Computer Science and Complex Systems at Brandeis University, with interests and refereed publications across artificial intelligence, Artificial Life, neural and evolutionary computing, robotics, complex and dynamic systems, educational technology, and intellectual property law. His laboratory's work on automatically designed robots made front page news in all media, and he was named one of MIT's Technology Review "TR 10" in January 2001. In his spare time he acts as founder and CTO of Thinmail, which makes software to liberate email from the desktop.