

review articles

DOI:10.1145/2493883

Taking a biologically inspired approach to the design of autonomous, adaptive machines.

BY JOSH C. BONGARD

Evolutionary Robotics

THE AUTOMATED DESIGN, construction, and deployment of autonomous and adaptive machines is an open problem. Industrial robots are an example of autonomous yet nonadaptive machines: they execute the same sequence of actions repeatedly. Conversely, unmanned drones are an example of adaptive yet non-autonomous machines: they exhibit the adaptive capabilities of their remote human operators. To date, the only force known to be capable of producing fully autonomous as well as adaptive machines is biological evolution. In the field of evolutionary robotics,⁹ one class of population-based metaheuristics—evolutionary algorithms—are used to optimize some or all aspects of an autonomous robot. The use of metaheuristics sets this subfield of robotics apart from the mainstream of robotics research, in which machine learning algorithms are used to optimize the control policy^a of a robot. As in other branches of computer science the use of a metaheuristic algorithm has a cost and a benefit. The cost is that it is not possible to guarantee if (or when) an optimal control policy will be found for a given robot. The benefit is few assumptions must be made

about the problem: evolutionary algorithms can improve both the parameters and the architecture of the robot's control policy, and even the shape of the robot itself.

Because the trial-and-error nature of evolutionary algorithms requires a large number of evaluations during optimization, in many evolutionary robotics experiments optimization is first carried out in simulation. Typically an evolutionary algorithm generates populations of virtual robots that behave within a physics-based simulation.^b Each robot is then assigned a fitness value based on the quality of its behavior. Robots with low fitness are deleted while the robots that remain are copied and slightly modified in some random manner. The new robots are evaluated in the simulator and assigned a fitness, and this cycle is repeated until some predetermined time period has elapsed. The most-fit robot may then be manufactured as a physical machine and deployed to perform its evolved behavior.

To illustrate the distinction between mainstream and evolutionary robotics, consider two experiments drawn from the two fields. Legged locomotion—

^b Interested readers may download and perform their own evolutionary robotics experiments at <http://www.uvm.edu/~ludobots>.

» key insights

- **Manual design of a mobile robot that is autonomous and adaptive is extremely difficult.**
- **As an alternative, computers can ‘evolve’ populations of robots in a simulator to exhibit useful behavior and then manufacture physical versions of the best ones, very much like how farmers breed crops for high yield. This approach is known as evolutionary robotics.**
- **This evolutionary approach changes the way we view robotics: rather than machine-learning techniques improving behaviors for a hand-designed robot, focus shifts to creating an evolutionary system that continuously designs and manufactures different robots with increasing abilities.**

^a A control policy is some function that transforms a robot's sensor signals into commands sent to its motors.



optimizing a control policy that allows a two, four, or six-legged robot to move over rugged terrain—is a popular area of study in robotics. In mainstream robotics, machine-learning algorithms can now optimize walking behavior for a physical two-legged robot in a matter of minutes.⁷ Alternatively, a recent investigation in simulation has shown if robots are evolved to move over rough terrain, robots will eventually evolve from amorphous shapes into robots exhibiting the rudiments of appendages (Figure 1b).¹

The former experiment can enable walking behaviors for a certain kind of robot; the latter experiment can continuously produce different robots adapted to different environments. Put differently, mainstream robotics aims to continuously generate better behavior for a given robot, while the long-term goal of evolutionary robotics is to create general, robot-generating algorithms.

History

The goal of artificial intelligence, since its beginnings, has been to reproduce aspects of human intelligence (such as natural language processing or deductive reasoning) in computers. In contrast, most roboticists aim to generate noncognitive yet adaptive behavior in robots such as walking or object manipulation. Once these simpler behaviors are realized successfully in robots, it is hoped the behavior-generating algorithm will scale to generate ever more complex behavior until the adaptive behavior exhibited by a given robot might be characterized by an observer as intelligent behavior. This operational definition of intelligence bears a resemblance to the Turing Test: if a robot looks as if it is acting intelligently, then it is intelligent.

Note the emphasis in robotics on “behavior:” the action of a robot generates new sensory stimulation, which in turn affects its future actions. This differs from non-embodied AI algorithms, which have no body with which to affect, or be affected by the environment. In non-embodied AI, intelligence is something that arises out of introspection; in robotics, the belief is that intelligence will arise out of ever more complex interactions between the machine and its environment. This

idea that intelligence is not just something contained within the brain of the animal or control policy of a robot but rather is something that emerges from the interaction between brain, body, and environment, is known as embodied cognition.²⁷

The very first experiments in evolutionary robotics⁹ began to shed light on embodied cognition. In one set of experiments a robot equipped with a camera had to move toward certain shapes and away from others. Based on the way the robot evolved to move, the control policy of the robot often only made use of two small pixel patches rather than the entire video stream. In other words, the robot evolved the ability to recognize objects through a combination of motion and sensation. This approach is non-intuitive to a human designer, who might implement object-recognition algorithms that draw on all of the pixels in the video stream.

Applications

Evolutionary algorithms have been applied in several branches of robotics and thus evolutionary robotics is not strictly a subfield of robotics. When applied well, an evolutionary approach can free the investigator from having to make decisions about every detail of the robot’s design. In many cases the evolutionary algorithm discovers solutions the researcher might not have thought of, especially for robots that are non-intuitive for a human to control or design. For example it is often difficult to see how best to control a soft robot (Figure 1j) using traditional machine learning techniques, let alone determine the best combination of soft and rigid materials for such a robot.

Moreover, ideas can flow not just from biology to robotics but back again: evolved robots that exhibit traits observed in nature—such as a robot swarm that evolves cooperative rather than competitive tendencies—often provide new ways of thinking about how and why that trait evolved in biological populations. In this way evolutionary robotics can give back to biology (“Why did this trait evolve?”) or more cognitively oriented fields such as evolutionary psychology (“Why did this cognitive ability evolve?”).

Evolutionary biorobotics. In biorobotics, investigators implement anatomical details from a specific animal in hardware and then use the resulting robot as a physical model of the animal under study. Although much work in this area has been dedicated to non-human animals (see supplemental material available in the ACM Digital Library; <http://dl.acm.org>), many roboticists choose to model the human animal: a humanoid robot is more likely to be able to reach a doorknob, climb steps, or drive a vehicle than a wheeled robot or one measuring only a few inches in length. The humanoid form, however, requires mastery of bipedal locomotion, a notoriously difficult task. As an example, Reil et al.³⁰ evolved a bipedal robot in simulation that first mastered walking and then evolved the ability to walk toward a sound source.

In short, bioroboticists attempt to model, in robot form, the products of evolution: individual organisms. Evolutionary roboticists in contrast attempt to re-create the process of evolution, which generates robots that may or may not resemble existing animals.

Evolutionary biorobotics is a blend of these two approaches: investigators build robots that resemble a particular animal, and then evolve one aspect of the robot’s anatomy to investigate how the corresponding aspect in the animal might have evolved. For example Long and his colleagues¹⁹ have evolved the stiffness of artificial tails attached to swimming robots: robots with tails of differing stiffness have differing abilities to swim fast or turn well. This provides a unique experimental tool for investigating how backbones originally evolved in early vertebrates.

Developmental robotics. The field of developmental robotics²² shares much in common with evolutionary robotics. Practitioners of developmental robotics draw inspiration from developmental psychology and developmental neuroscience: how do infants gradually mature into increasingly complex and capable adults? Like evolutionary robotics, work in developmental robotics tends to have either a scientific or an engineering aim. Developing robots can be used as scientific tools: they can serve as physical models for investigat-

ing biological development. Alternatively, engineers can draw on insights from biological development to build better robots.

Evo-devo-robo. Developmental robotics tends to focus on post-natal change to a robot's "body" and "brain" as the robot learns to master a particular skill. Evolutionary robotics experiments on the other hand generate robots that become more complex from generation to generation, but typically each individual robot maintains a fixed form while it behaves.

Biological systems however exhibit change over multiple time scales: individual organisms grow from infants into adults, and the developmental program that guides this change is in turn altered over evolutionary time. This process is known as the evolution of development, or *evo-devo*. This biological phenomenon has recently been exploited in evolutionary robotics:³ At the outset of evolution, robots change from a crawling worm into a legged walking machine over their lifetime. As evolution proceeds, this infant form is gradually lost until, at the end of evolution, legged robots exhibit the ability to walk successfully without the need to crawl first. It was found this approach could evolve walking machines faster than a similar approach that does not lead robots through a crawling stage.

In the initial experiments of evo-devo-robo,³⁴ the genetic instructions were encoded as a specific class of formal grammars known as Lindenmayer systems, or L-systems.^c L-systems were initially devised to model plant growth: their recursive nature can produce fractal or otherwise symmetric forms. Hornby¹² demonstrated that robots evolved using such grammars do indeed produce repeated forms (Figure 1a). He also showed this repetition can make it easier for evolutionary algorithms to improve such robots, compared to robots lacking in genetically determined self-similarity.

The evolution of robot bodies and brains differs markedly from all other approaches to robotics in that it does not presuppose the existence

^c Sims' work had a large impact on the computer graphics community and L-systems remain a popular technique within that field.

The evolution of robot bodies and brains differs markedly from all other approaches to robotics in that it does not presuppose the existence of a physical robot.

of a physical robot. Rather, the user provides as input a metric for measuring robot performance along with a simulation of the robot's task environment, and the algorithm produces as output the body plan and control policy for a robot capable of performing the task. This can then be used to manufacture a physical version of the evolved robot. Such an algorithm could, in principle, continually receive new desired behaviors and task environments and continuously generate novel robots.

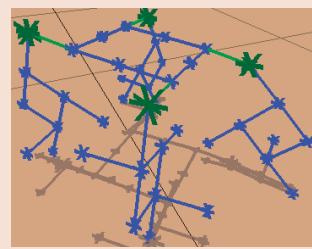
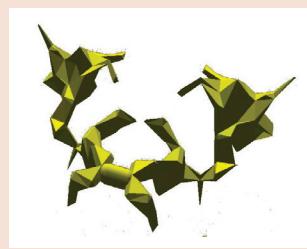
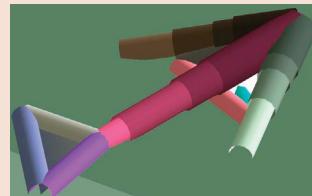
In this way, the roboticist can make fewer assumptions about the final form of the robot and have greater confidence the final evolved robot is better adapted to the environment in which it must operate. For example, there is often a debate about whether a wheeled or legged robot is more appropriate for moving over a given surface. Although not yet demonstrated, an evolutionary robotics algorithm should generate wheeled robots if supplied with a simulation of flat terrain and legged robots if supplied with a simulation of rugged terrain. Recent work in mainstream robotics has demonstrated the possible advantage of combining wheels and legs in the same robot: an evolutionary system should rediscover this manually devised solution if it is indeed superior to either wheels or legs alone.

Another advantage of this approach over mainstream robotics is its potential for better scalability: by genetically encoding assembly instructions rather than the blueprint of a robot, more complex machines can be evolved with little or no increase in the amount of information encoded in the genome. For example, consider an approach in which robots are specified by a formal grammar such that the invocation of a rewrite rule replaces one part of the robot with two or more parts. Thus the more times a given set of rewrite rules are invoked, the more complex the resulting robot becomes. If evolution increases the number of rewrite rule invocations, then simple robots can evolve into more complex robots with no increase in the information content of the underlying genomes describing those robots.

Despite the promise of this ap-

Figure 1. A sampling of representative work in evolutionary robotics.

Evolutionary robotics often involves optimizing not only the controller of a robot but also its body plan. Formal grammars (a¹²) and algorithms that simulate development (b¹) have been used to optimize robots in simulation. Additive manufacturing has been employed to build physical versions of evolved simulated robots semiautomatically (c,d¹⁸). Once deployed as physical machines, evolutionary algorithms have been used to allow damaged robots to recover from injury (e⁴) as well as ease the transferral of newly evolved controllers from simulation to the physical robot (f¹⁸). In addition to locomotion, researchers have evolved more cognitively demanding behaviors such as discriminating between differently shaped objects by manipulating them (g³⁵) or physically demanding tasks like aerial swarming (h¹¹). Behaviors have also been evolved for robots with non-traditional body plans such as tensegrity robots (i²⁶), robot built by S. Fivat), soft robots (j³²), modular robots (k³⁹) and robot swarms (l³³).

(a) Hornby and Pollack¹²(b) Auberbach and Bongard¹(c) The GOLEM Project Lipson and Pollack¹⁸

(d)

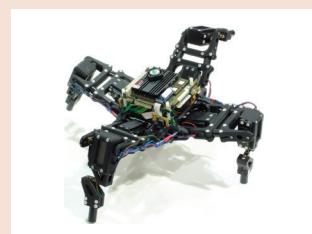
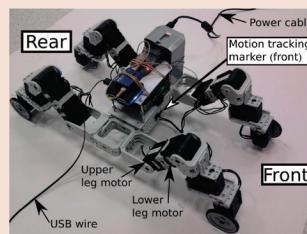
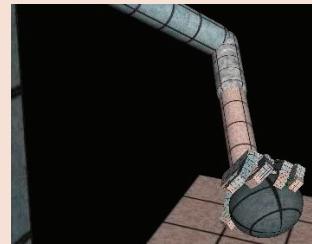
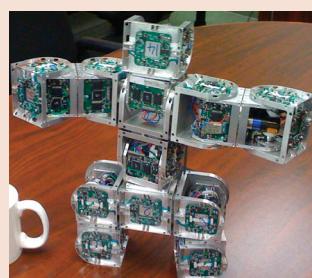
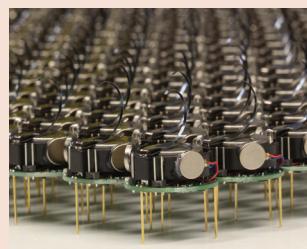
(e) Bongard et al.⁴(f) Loos et al.¹⁸(g) Tuci et al.³⁵(h) Hauert et al.¹¹(i) Paul et al.²(j) Rieffel et al.³²(k) Yim et al.³⁹
(USC Polymorphic Robotic Lab)(l) Rubenstein et al.³³

Figure 1c,d,f,g,j,k,l reprinted courtesy of IEEE.
Figure 1e reprinted courtesy of AAAS.
Figure 1h reprinted courtesy of Springer.

proach, only a handful of such algorithms have yet been developed. There are five main reasons for this. First, implementing such an algorithm is extremely difficult, as it requires a robust physics-based simulator that can accurately simulate complex mechanical constructs of arbitrary topology. Second, even with today's available computing power it can be computationally prohibitive to evaluate the thousands or millions of candidate robots required to generate one of sufficient quality. Third, an evolutionary algorithm must be devised that is expressive enough to encode diverse robot forms and evolvable in the sense that successive slight mutations lead to successively more complex and capable robots. Fourth, building a physical copy of the often complex virtual robots produced by such systems can be prohibitive. And finally, such systems have yet to automatically generate a robot that is more complex and capable than those designed and built manually. Overcoming these challenges remains a strong focus in the field.

Swarm robotics. One of the major challenges in swarm robotics is devising a control policy that, when executed by all members of the swarm, gives rise to some desired global behavior (Figure 1l). For example, if one wishes to program a group of robots to move collectively in a way similar to biological herds, flocks, or schools of fish, it has been shown³¹ that each robot must balance attraction toward its local neighbors with repulsion away from neighbors that are too close.^d However, if attraction is weighted too heavily, the swarm can contract into a traffic jam; if repulsion is weighted too strongly the group disperses.

This approach to controlling groups of robots is based on the principle of self-organization observed at many levels of biological systems: biological elements such as cells or organisms often form into cohesive patterns without a central control signal. This approach is desirable in robotics, in which communication limitations or

^d This basic algorithm has since become the cornerstone of computer graphics algorithms which simulate the movement of animal or human groups.

the danger of failure may make designating one robot as the leader a difficult or risky proposition.

Evolutionary approaches have been used to optimize individual behaviors within a robot swarm. In the first such work,²⁹ control policies for homogeneous robots evolved that allowed them to move in concert, despite the lack of a leader. Repeatedly evaluating large numbers of candidate controllers on groups of physical robots places severe demands on the underlying hardware, so much work in this area has relied on simulations of robot swarms. This has enabled researchers to investigate more complex group behaviors, such as group hunting.²⁰ Such experiments require an understanding of co-evolution: one group's ability to overcome a second group makes it likely the second group will evolve to defend against the original group. This in turn exerts pressure for the original group to evolve a new strategy, and so on.

Co-evolution requires competition between groups, but also cooperation between individual group members. Evolutionary robotics has been used to investigate the conditions under which cooperation will arise, and how communication may evolve to support it. In an early study communication evolved in groups of "male" and "female" simulated robots so that female robots could call out to and attract males for mating.³⁶ It was observed that different dialects would evolve and compete with one another. More recent work with populations of simulated robots has demonstrated how distinct communication strategies can arise and that there are evolutionary advantages to more complex strategies.³⁸ These and other studies may provide unique tools for studying the evolution of biological communication strategies in general, and human language in particular. Such work could also provide a physical substrate on which to test hypotheses from game theory that involve deception, cooperation, and competition.

Modular robotics. Advancing technology has now made modular robotics feasible: Individual robots, or modules, may dynamically attach and detach from one another to create a robot with a constantly changing form

(Figure 1k). It has been shown that evolutionary algorithms can be used to optimize behaviors for a modular robot in a fixed form (for example, see Zahadat⁴⁰). More recently evolutionary methods have been used to enable modular robots to self-assemble from their constituent parts or reconfigure into different functional forms (for example, see Meng²³). Continuously evolving novel forms and associated behaviors appropriate for a newly encountered environment remains an open problem in this area.

Soft robotics. With the exception of wheeled vehicles, robots are typically constructed from jointed collections of rigid parts, mirroring the skeletal linkages of higher animals and humans. Advances in materials science, however, have made non-traditional robot body plans possible. As one example, the evolution of behaviors for tensegrity robots was reported in Paul²⁶ (Figure 1i). Tensegrity structures are collections of rigid and elastic links attached in a particular way that provide several advantages over traditional robots, such as the ability to automatically revert to their default form if perturbed.

Soft robots are emerging as a new class of machine that combines discrete rigid parts with continuous, soft materials (Figure 1j). Such machines could "squeeze through holes, climb up walls, and flow around obstacles."³² Controlling such devices is non-trivial, as motion at one location of the robot can propagate in unanticipated ways to other parts of the body. Despite this, Rieffel et al.³² successfully evolved locomotion for a soft robot such that it exploited rather than fought against the synergies within its body. Evolving the architectures of such discrete and continuous devices demands new kinds of optimization methods. Coupled with the sudden recent interest in this field¹³ there are many contributions that computer scientists interested in optimization could make in this area.

Formalisms

Evolutionary robotics is a mostly empirical endeavor, although three formalisms—the nature of computation, dynamical systems theory, and information theory—are beginning to

provide a theoretical foundation for the field.

Morphological computation. As noted earlier, evolutionary robotics builds on the concept of embodied cognition, which holds that intelligent behavior arises out of interactions between brain, body, and environment.²⁷ An important corollary of embodied cognition is that, given the right body plan, a robot (or animal) can achieve a given task with less control complexity than another robot with an inappropriate body plan. For example, a soft robot hand can grip a complex object simply by enclosing it: the inner surface of the hand passively conforms to the object. A robot hand composed of hard material must carefully compute how to grasp the object. It has been argued that the physical aspect of a robot—its morphology—can actually perform computations that would otherwise have to be performed by the robot's control policy if situated in an unsuitable body plan. This phenomenon of morphological computation²⁵ cannot be completely abstracted away from the physical substrate that gives rise to it in the way a Turing Machine can. Practitioners in this area would greatly benefit from the aid of theoretical computer scientists to formalize this concept.

Dynamical systems theory. Dynamical systems theory is increasingly a useful tool for creating controllers for autonomous robots.² Often these controllers take the form of artificial neural networks that have their own intrinsic dynamics: they exhibit complex temporal patterns spontaneously. Evolutionary algorithms can then be used to shape the parameters of these networks such that they can be pushed by incoming sensor stimuli to fall into desired attractor states. For example, a neural network that falls into a periodic attractor may generate a rhythmic gait in a legged robot. However, it has been demonstrated that a one-to-one mapping between a basin of attraction in a neural network and a distinct robot behavior may be overly simplistic,¹⁴ indicating there is much work to be done at the interface of dynamical systems theory and evolutionary robotics.

Information theory. Typically in an evolutionary robotics experiment, the “fitness” of a robot is measured based on its ability to perform a given behavior, such as how far it can walk or how well it can grasp an object. Surprisingly, it has been found that maximizing certain information-theoretic measures within the neural network of evolving robots can lead to useful behavior.²⁸ Why information maximization produces desired behaviors rather than useless, random, or uninteresting behavior remains mostly unresolved, although some progress has been made in this direction.⁸

In addition to helping with the synthesis of behavior, information theory can also be used to analyze evolved behaviors. Williams et al. have recently shown³⁷ that information flow—the transfer of information from one variable to another—can be employed to measure how behaving robots “offload” computed information to their body and/or their environment. This technique therefore holds promise for formalizing the concept of morphological computation.²⁵

Challenges

There are a number of challenges currently facing the field, including transferring evolved robots from simulation to physical machines; scalability issues; and the difficulty of defining appropriate fitness functions for automatically measuring behavior.

The Reality Gap Problem. Both biological and artificial evolution are notorious for exploiting the potential relationship between the animal (or robot) and its environment to produce new behaviors. For instance the lightweight property of feathers, which are thought to have originally evolved for heat regulation, was later exploited for flight.^c

As an example of the exploitative tendencies of evolutionary algorithms applied to robots, a robot was initially designed to brachiate along a suspended beam.¹⁰ The robot was composed of a main body slung under two arms, and a heavy battery pack attached to the main body. Gradually, the evolutionary algorithm discovered

control policies for the robot that exploited, rather than fought against the weight of the batteries. These control policies would cause the robot to move such that the battery pack swung forward under the robot's body before it changed hand holds. This would cause the robot's center of mass to move forward, thus requiring much less force to release contact with the beam and grasp it further forward. This mimics the way primates exploit the weight of their bodies like a pendulum to bring them into reach of a new tree limb. It is also reminiscent of the energy-saving passive dynamics of bipedal locomotion (see more details in the supplemental material).

However, if robots are optimized in a simulator, artificial evolution may exploit simplifications or inaccuracies in how physics is simulated. Such evolved control policies may then fail to reproduce the desired behavior when transferred from simulated to physical robots. For example, if there is no noise in the simulator, a control policy may evolve to generate behavior based on a very narrow range of sensor values. If this control policy is then transferred to a physical robot with a sensor that registers a wider range of values due to limitations in its electronics or mechanics, the physical robot may not behave as intended. This failure of evolved solutions to “cross the gap” from simulation to reality is known as the “reality gap” problem¹⁵ and is one of the major challenges facing the field. However, a number of solutions have been proposed and significant progress is being made in this area.

In early work, sampling of the physical sensors was conducted and used to simulate the robot's sensors during evolution.²⁴ Alternatively, noise can be added to different aspects of the robot and its interaction with the environment: noise can be added to the sensors, to the effects of the motors, or the position of the robot itself.¹⁵ This keeps evolution from exploiting artifacts of the simulation.

However, neither of these approaches scale well. If the robot must interact with increasingly complex and asymmetric objects, more samples must be taken from the sensors that detect the object: the sensor must be polled

e This tendency of evolution to repurpose traits is known as “exaptation.”

at many more distances and positions relative to the object because there are more unique views of the object from the standpoint of the sensor. If noise is added to the simulation, each control policy must be evaluated several times such that controllers evolve to be robust to the noise in the simulation. For more complex robots, noise must be added to greater numbers and types of sensors and actuators. This requires even more evaluations to evolve robustness against this larger number of noise sources.

In more recent work the typically unidirectional approach of transferring evolved control policies from simulated to physical machines has been replaced with bidirectional approaches in which optimization alternates between simulation and reality.^{4,16} For example, in Bongard et al.⁴ three different evolutionary algorithms were employed. The first optimized a population of physical simulators to better reflect reality: The fitness of a simulator was defined as its ability to predict the behavior of the physical robot (Figure 1e).

The second evolutionary algorithm optimized exploratory behaviors for the physical machine to perform. These behaviors were assigned a high fitness if, when executed by the physical machine, they extracted the most new information about the way in which the robot could interact with its environment. This new information then became new training data for the first evolutionary algorithm. Gradually, after several alternations between these two optimization methods, a physical simulation would automatically emerge that was adapted to the details of the quadrupedal physical robot that was used in the experiment. The third evolutionary algorithm then uses this highly fit simulator to evolve control policies for the physical robot, and it was found that many such evolved behaviors transferred successfully from simulation to reality.

This approach turned out to have an added advantage over previous attempts to cross the reality gap: the robot could recover from physical damage such as the mechanical separation of one of its four legs. If the robot experienced such damage

It was identified early on that the time required to evaluate a single robot might grow exponentially with the number of parameters used to describe its task environment.

while behaving, the robot could not directly sense the damage but there would be an inevitable change in the incoming sensor values. This change would be automatically incorporated by the first evolutionary algorithm into new simulations: simulations of a three-legged robot would gradually replace simulations of a four-legged robot. These new simulations would then be used to evolve new control policies for the damaged robot that would allow it to automatically compensate for its injury.

Future work in this area would benefit from collaborations with developers of physical simulation such that evolution could alter the physical constants of the simulation itself, such as those used to model friction, collision, as well as aero- and hydrodynamics.

Koos et al.¹⁶ recently proposed a different approach to the reality gap problem. Control policies evolved in simulation are transferred to a physical machine (Figure 1f), and the disparity between the behavior observed in simulation and reality is measured. This is done for several controllers, and the resulting disparity measures are used to create a model that predicts the disparity of control policies that have yet to be validated on the physical machine. A multi-objective optimization is then employed to maximize the desired behavior in simulation and to minimize predicted disparity: control policies are sought that generate the desired behavior in the simulated robot and are likely to reproduce that behavior in the physical robot.

This work attempted to address a seeming trade-off between behavioral efficiency and transferability: the more efficient the robot is at exhibiting a desired behavior the less likely it is to transfer to the physical machine. For example if fast-legged locomotion is selected for in simulation, running is more desirable than walking. However, running requires the robot's control policy to carefully manage its center of mass to avoid falling. If the mass distributions of the simulated and physical robot are slightly different, running behaviors may fail when transferred. This failure is less likely for walking behaviors in which the robot's mass distribution is less important.

Most of the work on the reality gap problem has assumed that only the control policy of robots will be transferred. Lipson and Pollack,¹⁸ however, integrated an evolutionary robotics simulation with rapid prototyping technology to automate robot manufacture as well as robot design (Figure 1c, 1d). They first evolved the body plans and control policies for robots composed of linked assemblages of linear actuators. Then, the 3D architectures of the best of these evolved robots were printed out of plastic; motors, circuitry, and batteries were then added by hand. Many of these automatically designed and manufactured robots were able to successfully reproduce the locomotion patterns originally evolved in the simulator.

Combinatorics of evaluation. It was identified early on that the time required to evaluate a single robot might grow exponentially with the number of parameters used to describe its task environment.²² For example, consider a robot that must grasp m different objects under n different lighting conditions. Each robot must be evaluated for how well it grabs each object under each lighting condition, requiring mn evaluations per robot. If there are p parameters describing the task environment and each parameter has s different settings, then each robot must be evaluated s^p times.

This is a serious challenge in the field that has yet to be resolved. However, one possible solution to this challenge may be addressed using co-evolution. Consider a population of robots and a second population of task environments competing against one another. The robots evolve to succeed when exposed to environments drawn from the pool of evolving environments, and environments evolve to foil the abilities of the evolving robots. This is not unlike prey evolving to elude predators, while the predators evolve to catch prey. This approach could, in the future, be used to evolve robots that successfully generalize against a subset of task environments they might encounter when manufactured and deployed.

Evolvability. Evolving all aspects of a complex machine such as a robot is a daunting, high-dimensional optimization problem. Biological evolution

One goal in evolutionary robotics in particular, and the field of evolutionary computation in general, is to create increasingly evolvable algorithms.

faces the same challenge yet seems to have addressed it by a process known as the evolution of evolvability. A species with high evolvability is defined as one that can more rapidly adapt to changes in its environment than a similar species with lower evolvability.

One goal in evolutionary robotics in particular, and the field of evolutionary computation in general, is to create increasingly evolvable algorithms. Rather than independently optimizing individual parameters of a candidate solution, such algorithms should rapidly discover useful aggregate patterns in candidate solutions and subsequently elaborate them. It has been shown, for example, that genomes that encode formal grammars produce robots with regular structure, and that such genomes are more evolvable than genomes that do not produce regular structures.¹²

Similarly, when an evolutionary algorithm biased toward producing regular patterns was used to evolve artificial neural networks for robots it was found, again, that such networks more rapidly discover desired behavior compared to other evolutionary methods that do not generate such regularity.⁶ Auerbach and Bongard¹ have expanded the reach of this evolutionary algorithm to shape robot body plans as well.

Despite these recent advances, little is known about how to design evolutionary algorithms that reorganize genetic representations to maximize evolvability and thus automatically generate adaptive complex machines in a reasonable amount of time.

Fitness Function Design. The original and continued goal of evolutionary robotics is to make as few assumptions about the final form of the robot or the kind of behavior that should be generated. However, designing a fitness function that rapidly discovers desirable solutions without biasing it toward particular solutions is notoriously difficult. For this reason there have been efforts in the field to eliminate the usage of a fitness function altogether. One recent example is novelty search, which begins with simple candidate solutions and gradually creates more complex solutions as optimization proceeds.¹⁷ The fitness of any given solution is simply how much

Copyright of Communications of the ACM is the property of Association for Computing Machinery and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.