Biologically Inspired Computing

Josh Bongard, *University of Vermont*

Evolutionary algorithms and robotics hold great promise as integrated design and modeling tools.

Editor's note: A column on intelligent systems and AI research appears passé, when there are exclusive journals and other outlets devoted to their study. But there is an opportunity to present ideas and developments in a way that appeals to experts and nonpractitioners alike.

This column is intended to fulfill that need by serving as a "what's happening in the AI disciplines" commentary. Every other month, it will feature articles written by prominent AI researchers on the latest work in their areas. We cast a broad definition of AI that embraces classical topics, interfaces with other disciplines, and applications in novel domains.

Send suggestions for articles and feedback to Naren Ramakrishnan, Department of Computer Science, Virginia Tech, at naren@cs.vt.edu.

Despite the relentless, breathtaking advances in computing and related technologies, we continue to be humbled by the variety, adaptability, and sophistication of the natural world around us. From the beginning, computing has been inspired by nature: Alan Turing asked whether computers could think like us, while John von Neumann, armed only with pencil and paper, sketched out an automaton that could self-replicate.

Since then, a divide has grown between computational scientists on whether to continue creating faster, more efficient algorithms and hardware that exhibit centralized control or to place less emphasis on speed and efficiency than on robustness, adaptability, and emergent organization from the interaction of many loosely coupled processes. These latter approaches have come to be known as *biologically inspired computing*, which is not so much a field as a philosophy that links various disciplines such as artificial intelligence, evolutionary computation, biorobotics, artificial life, and agent-based systems.

One argument against bio-inspired computing is that large-scale systems cannot emerge from blind, bottom-up processes. However, one study found that Wikipedia, with little centralized editorial control, is nearly as accurate as Encyclopedia Britannica (G. Giles, "Internet Encyclopedias Go Head to Head," *Nature*, 15 Dec. 2005, pp. 900-901). Social-networking platforms, peer-to-peer networks, and user-generated content sites—although not strictly bio-inspired—have likewise demonstrated that such systems can grow, organize, and improve themselves with little direction from above.

Bio-inspired computing has passed in and out of vogue during the past few decades, mostly due to the claim that it does not embody true computer science in the sense of delivering guaranteed performance in clearly defined domains. But bio-inspired algorithms can exhibit strength through flexibility, or strength in numbers: They often work well even when the desired task is poorly defined, adapt to unforeseen changes in the task environment, or achieve global behavior through interaction among many, simply programmed agents.

BODY AND BRAIN

AI in particular has suffered numerous "winters" in which hype and overly optimistic promises have led to unfulfilled expectations and collapses in funding. It has gradually been recovering since its last ice age in the early 1990s for many reasons, one of which was Rodney Brooks' argument that intelligent systems must have access to a body to interact with and thus learn from the environment ("Elephants Don't Play Chess," *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, P. Maes, ed., MIT Press, 1991, pp. 3-15).

This flew in the face of much AI work, which emphasized building ever more complex and elegant algorithms that, although inspired by what psychologists and

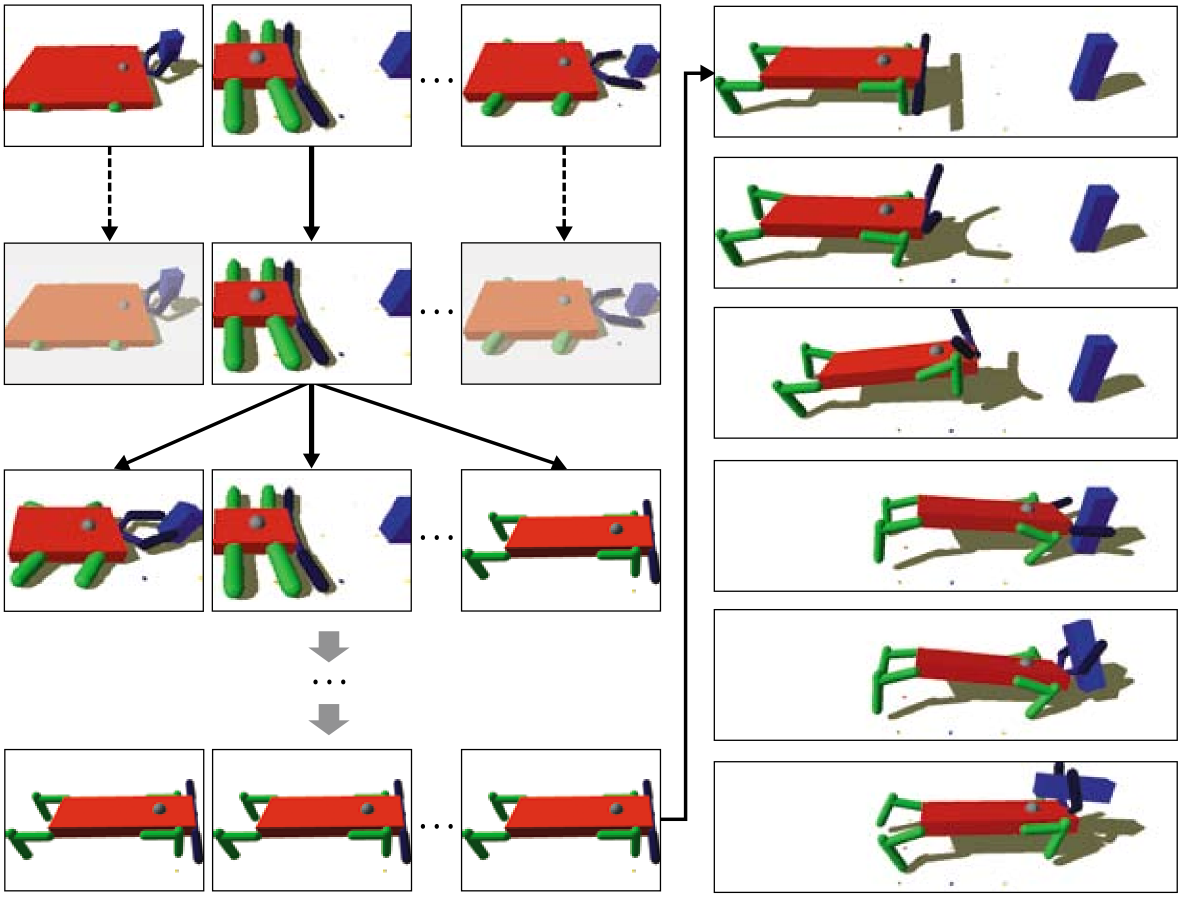

Figure 1. Evolutionary robotics. ER designers use software to breed virtual robots so that they evolve to carry out the desired task—in this case, picking up a block.

neuroscientists believed at the time was occurring in the human mind, were rarely exposed directly to the messiness of the physical world.

Brooks and, later, many others contended that what initially seemed to be difficult problems, such as playing a masterful game of chess, are actually relatively straightforward to solve with a powerful-enough algorithm that performs the same basic operation trillions of times. However, seemingly simple tasks such as walking over uneven ground while keeping one's balance require orchestration. A walking robot needs not only real-time control but also flexible ankles, sensors to detect tipping, and arms to swing to keep tipping from becoming falling.

EVOLUTIONARY ROBOTICS

Brooks' revolution has raised many interesting questions in robotics and AI that remain to be answered. One of the most engaging is that if both body and brain affect a robot's chances to exhibit useful behavior, how do we go about designing not only the robot's control algorithm but its body as well? After all, Mother Nature does not optimize the body plan for a given species and then optimize its behavior, but rather both body and brain gradually adapt to suit the demands of a species' ecological niche.

Enter the field of *evolutionary robotics*. ER combines evolutionary computation and autonomous robotics in an attempt to automate the process of robot design. Evolutionary computation refers to a collection of machine learning techniques that draw inspiration from the blind process of natural selection:

- generate a population of random solutions to a given problem;
- apply each solution to the problem;
- delete poorly performing solutions from the population;
- make copies of those solutions that survive;
- introduce small, random changes into the copies;
- apply these new solutions to the problem;
- repeat the process until you find a satisfactory solution.

In evolutionary robotics, the "problem" is the task that the robot must carry out, such as finding objects in its environment and carrying them to a central location. "Solutions" are either controllers for an existing robot or blueprints that describe both a robot body plan and a controller to go along with it.

As Figure 1 shows, ER designers use software to breed the bodies and brains of virtual robots so that they evolve to carry out the desired task. As rapid prototyping technologies mature, the hope is that manufacturing one of these automatically created robot designs will become relatively straightforward.

This method adheres to the bottom-up philosophy of bio-inspired computing in that there is no central designer; rather, a collection of solutions compete against one another in an attempt to solve the given problem, and through the process of natural selection, good solutions tend to bubble to the top—at least some of the time.

Industry is just beginning to adopt evolutionary computation and related algorithms. For example, the major Swiss supermarket chain Migros uses bio-inspired algorithms to optimize its goods distribution.

However, much work remains to be done to improve the automated design aspects of evolutionary computation and ER. A major challenge is that evolutionary algorithm designers often spend so much time studying the problem and determining a formula for measuring the quality of any given solution that they can then usually come up with a better one on their own, rather than evolving one (A.L. Nelson, G.J. Barlow, and L. Doitsidis, "Fitness Functions in Evolutionary Robotics: A Survey and Analysis," *Robotics and Autonomous Systems*, doi:10.1016/j.robot.2008.09.009).

RESILIENT MACHINES

ER promises to be useful not just for designing robots but also for enabling them to adapt, an ability that machines lack yet animals possess in abundance. For example, Figure 2 shows a robot that is resilient: It can recover from unanticipated events for which it was not preprogrammed, one important aspect of adaptation (J. Bongard, V. Zykov and H. Lipson, "Resilient Machines through Continuous Self-Modeling," *Science*, 17 Nov. 2006, pp. 1118-1121).

The robot uses three separate evolutionary algorithms to achieve resiliency. The first builds several models that describe the robot's body (Figure 2a), which may change as a result of damage. The second algorithm finds a new action for the robot to perform (Figure 2b). This action is designed to help the robot discover which among its current set of models are incorrect. The robot then replaces these inaccurate models with modified copies of the more accurate models. Finally, once the robot can no longer find a better model, it selects the most accurate of them and uses it to evolve a walking strategy (Figures 2c and 2d).

If the robot suffers some injury that reduces it from four legs to three, the models re-evolve to reflect this. The robot then uses one of these new models to find a walking strategy

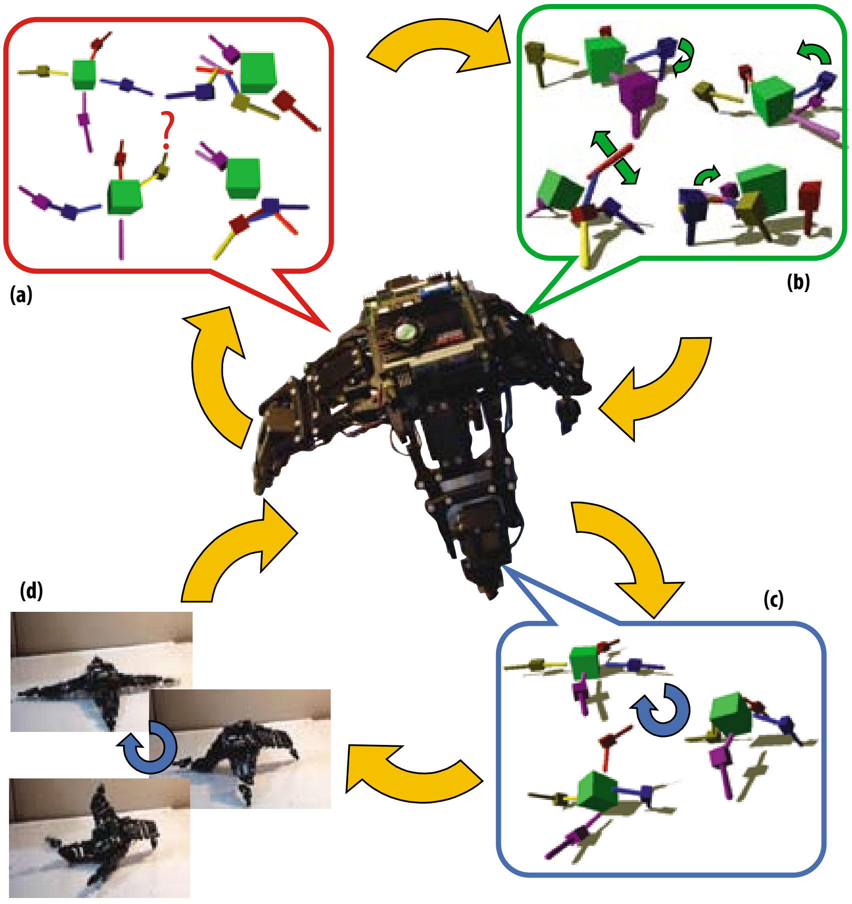

Figure 2. Using evolutionary algorithms to achieve resilience in a robot. The robot uses (a) one algorithm to build several models that describe its body and (b) another algorithm to find a new action to perform to discover which among its current set of models are incorrect. The robot replaces these inaccurate models with modified copies of the more accurate models (c and d) and uses the most accurate of them to evolve a walking strategy.

that compensates for its unexpected injury.

This work is part of a growing collection of next-generation adaptive machines, many of which rely on bio-inspired algorithms to adjust operation on the fly. Some can adapt in real time to novel situations, which lets them regain their footing after slipping on ice, for instance. Others, known as *social robots*, can enter into fluid interactions with people: shaking hands, maintaining eye contact, even raising their eyebrows when they see something surprising. Yet another class of machines known as *modular robots* are made up of collections of relatively independent modules, much like biological systems are made up of cells. If one module fails, the machine can reconfigure to continue operation.

INTEGRATED DESIGN AND MODELING

Much credit for these new machines is due to advances in motor and sensing technologies, but also to the unique advantages that evolutionary computation offers. More traditional machine learning algorithms tend to assume that the general form of the solution is known, but the parameters of that form must be optimized in some way.

Evolutionary computation really shines when confronted with an integrated design or modeling problem in which both the underlying form and parameters of the system are unknown. For example, the resilient machine shown in Figure 2 learns not only the parameters describing the different parts of its body but also the number of parts and how they interact. When designing a modular robot, we might ask how many modules should comprise the robot, how those modules should fit together, or how the robot should be able to reconfigure itself.

This ability to conduct integrated design and modeling is useful in other domains besides robotics. Complex systems have become a hot topic across the sciences, in which the connectivity between many heterogeneous units gives rise to behaviors of interest.

Often, we wish to both describe the units in a complex network as well as guess the connections between them. For example, evolutionary computation can be used to model genetic networks, in which both the behavior of genes and their influences on one another can be discovered by observing how they switch on and off over time (J. Bongard and H. Lipson, "Automated Reverse Engineering Nonlinear Dynamical Systems," *Proc. National Academy of Sciences*, June 2007, pp. 9943-9948).

Evolutionary algorithms and robotics are just in their infancy, with many obstacles yet to overcome including making them more automated and coupling design with manufacture. There are also exciting attempts to hybridize bio-inspired computing with more formal methods to combine the advantages of both into a single system.

Regardless, these techniques hold great promise as integrated design and modeling tools. Although they have only been applied to simple problems so far, there is hope that they may be scalable to larger problems.

It is becoming increasingly clear that the challenges we face in the 21st century are forbidding not just on account of their size but also because of their connectedness. Designing a better battery will not be sufficient for large-scale adoption of electric cars; it will also require designing new infrastructure to supply power to them. Combating global warming will require jointly designing economic, social, and technical solutions.

Creating machines or computer programs that grow, adapt, and multiply worry some. But it is worthwhile to keep in mind that self-replication and natural selection created the organisms that first filled our atmosphere with oxygen and others that continue to scrub CO_2 from it. It also produced intelligent agents—us—who are able to appreciate what we see around us and work to create new kinds of technologies that may yet help us to preserve it.

***Josh Bongard** is an assistant professor in the Department of Computer Science at the University of Vermont. Contact him at josh.bongard@uvm.edu.*