TO BECOME A PROFESSIONAL ANTENNA DESIGNER, you can follow one of two paths: you can enroll in college- and graduate-level courses on electromagnetism, immerse yourself in the empirical study of antenna shapes, and apprentice yourself to an established technician willing to impart the closely guarded secrets of the discipline.

Or you can do what Jason Lohn did: let evolution do the work.

Physicists know a lot about Maxwell's equations and the other principles governing wireless communications. But antenna design is still pretty much a dark art, says Lohn, a computer scientist working at NASA Ames Research Center outside Mountain View, CA. "The field is so squirrelly. All your learning is through trial and error, the school of hard knocks."

So why not automate trial and error? Antenna design, Lohn believes, is one of many engineering problems that could best be solved by evolutionary algorithms, an emerging class of software that produces lots of different designs, rejecting the less fit in order to select the most functional. The resulting designs often seem a little inhuman—inelegant and uncanny.

Evolutionary algorithms, also known as genetic algorithms or GAs, take their cue from biological evolution, which can turn a crawling reptile into a soaring bird without any kind of forward-looking blueprint. In sexual reproduction, the shuffling of each parent's genes—combined with random genetic mutation—creates organisms with new characteristics, and the less fit organisms tend not to pass on their genes to succeeding generations. Evolutionary algorithms work much the same way, but inside a computer.

Machines using genetic algorithms are better than humans at designing other machines

UNNATURAL SELECTION

By Sam Williams ILLUSTRATION BY JOEL LARDNER

When Lohn creates a new antenna, for example, he starts off with a population of randomly generated designs and grades their relative performance. Designs that come close to preset goals win the right to intermingle their properties with those of other successful candidates. Designs that disappoint go the way of the archaeopteryx: oblivion.

Breeding antennas takes time, of course. Most designs are downright awful, and it takes a large number of computing cycles to find decent performers. Still, when you've got a computer that can generate and test 1,000 generations an hour, interesting ideas do emerge*. Lohn, a PhD who hasn't taken a course on electromagnetism since his undergraduate years, expects to have at least one of his team's antenna designs go into space this year as part of NASA's Space Technology 5 mission, which will test a trio of miniature satellites. His favorite computer-designed antenna: a corkscrew contraption small enough to fit in a wine glass, yet able to send a wide-beam radio wave from space to Earth. It resembles nothing any sane radio engineer would build on her own.

"Evolutionary algorithms are a great tool for exploring the dark corners of design space," Lohn says. "You show [your designs] to people with 25 years' experience in the industry and they say, 'Wow, does that really work?'" The slightly spooky answer is that yes, they really do, as Lohn established after months of testing. "If we're lucky, we could have as many as six antenna designs going into space" in 2005, Lohn says.

Not every problem will succumb to the evolutionary approach. But those that will share a common characteristic: they all sit beyond what mathematician John von Neumann dubbed the "complexity barrier," the dividing line between problems that can be solved using traditional, reductionist methods and those that require a more intuitive, throw-it-up-and-see-what-sticks approach. Until recently, crossing this barrier was an expensive proposition. But today's computers are fast enough to sift through millions of offbeat designs in hope of finding one that works. Couple that with modern designers' growing skill in applying evolutionary algorithms, says David Goldberg, director of the Illinois Genetic Algorithms Laboratory at the University of Illinois at Urbana-Champaign, and you get what engineers lovingly call "scalability": the ability to tackle both miniature and massive design challenges.

NASA's corkscrew antenna resembles nothing any sane radio engineer would build on her own.

"Just as the steam engine created mechanical leverage to do larger tasks, genetic algorithms are starting to give individuals a kind of intellectual leverage that will reshape work," Goldberg says. "By automating some of the heavy lifting of thought, we free ourselves to operate at a higher, more creative level." Such freedom comes at a price, of course. It requires that engineers recognize the impossibility of peering into each and every "dark corner" and put their trust in yet another layer of mechanical assistance. But more and more of them are taking that leap.

* WWW.TECHNOLOGYREVIEW.COM We got our hands on a few clickable simulations that show how evolvable algorithms work. Keyword evolution.

From Toys to Tools

Reproducing in microseconds on a computer a process that takes millions of years in nature is an idea that long predates the ability to realize it. John H. Holland, a 76-year-old computer science professor at the University of Michigan, says he first came up with the notion while browsing through the Michigan math library's open stacks in the early 1950s.

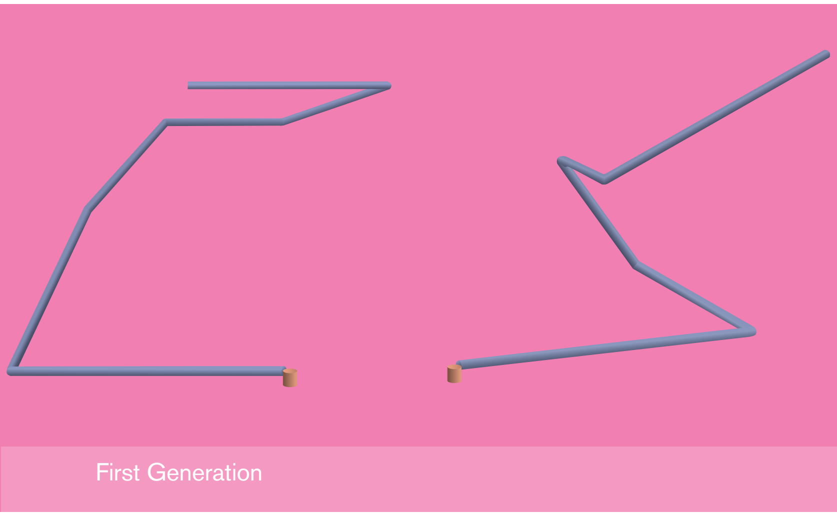
"Every once in a while I'd pick up a book that looked interesting and just read it," he says. That habit led him to *The Genetical Theory of Natural Selection*, a 1930 book by British mathematician-turned-biologist Ronald Fisher. Inspired by the pea plant experiments of 19th-century Austrian monk Gregor Mendel, Fisher worked out mathematical descriptions of natural selection at the level of individual genes. While researchers wouldn't crack the biochemistry behind that process until the 1950s, Fisher's work nevertheless jibed with what farmers and shepherds had known for centuries: sexual reproduction ensures variation and novelty.

"That's really where the genetic algorithms came from," says Holland. "I began to wonder if you could breed programs the way people would, say, breed good horses and breed good corn."

Holland wrote his first paper on "adaptive algorithms" in 1962. But it

Bent Is Better

For an experimental trio of miniature satellites, NASA needed antennas with both a wide beam and a wide bandwidth. The Evolvable Systems Group at NASA Ames Research Center created evolutionary software that generated random antenna designs, simulated the antennas' behavior, and recombined features of the best performers. The result, after many generations: the bizarrely twisted design on the far right.



wasn't until the late 1970s that he and his graduate students had amassed the computational resources to put the idea into play. Holland credits one of his students, Edward Codd, with convincing his former employer, IBM, to sell the Michigan research group a low-cost mainframe. (Codd would go on to win the A. M. Turing Award, computer science's equivalent of the Nobel Prize, for designing the first relational databases.) Even then, however, the computer's paltry 32 kilobytes of memory limited the size and scope of the researchers' initial experiments.

One of the first scientists to give evolutionary algorithms a serious test drive was Goldberg, who worked under Holland as a PhD student in the early 1980s. Goldberg resurrected a problem that he had faced during his days in the natural-gas industry: minimize the power consumption of a long-distance pipeline, given variations in regional demand. His evolutionary algorithms yielded solutions as efficient as those produced by the existing fluid mechanics software used by pipeline designers. But as Goldberg fed his algorithms bigger and more complicated problems, they began to stumble: they got stuck exploring evolutionary dead ends or spitting out hopelessly wild solutions. "I understood the problems I was solving better than the tools I was using to solve them, and that bothered me," Goldberg says.

Goldberg focused his dissertation and then another half-decade of work on making genetic algorithms more predictable. He found that adjusting the parameters of each new algorithm—the starting population size or the rate of mutation, for example—smoothed out a few wrinkles. But for the most part, his research left him with a sobering realization: evolutionary algorithms were often more complex than the problems they tried to solve. Eventually, Goldberg learned to steer clear of what he calls "needle in the haystack" problems, which demand a single, best solution; these tended to cause evolutionary algorithms to spin out of control. Instead, he aimed at friendlier problems that had a range of viable solutions, depending on how you approached them. "If there are dozens of needles scattered around in such a way that the [evolutionary algorithm] can break the haystack down into smaller haystacks, you at least guarantee yourself a shot at a better outcome," Goldberg says.

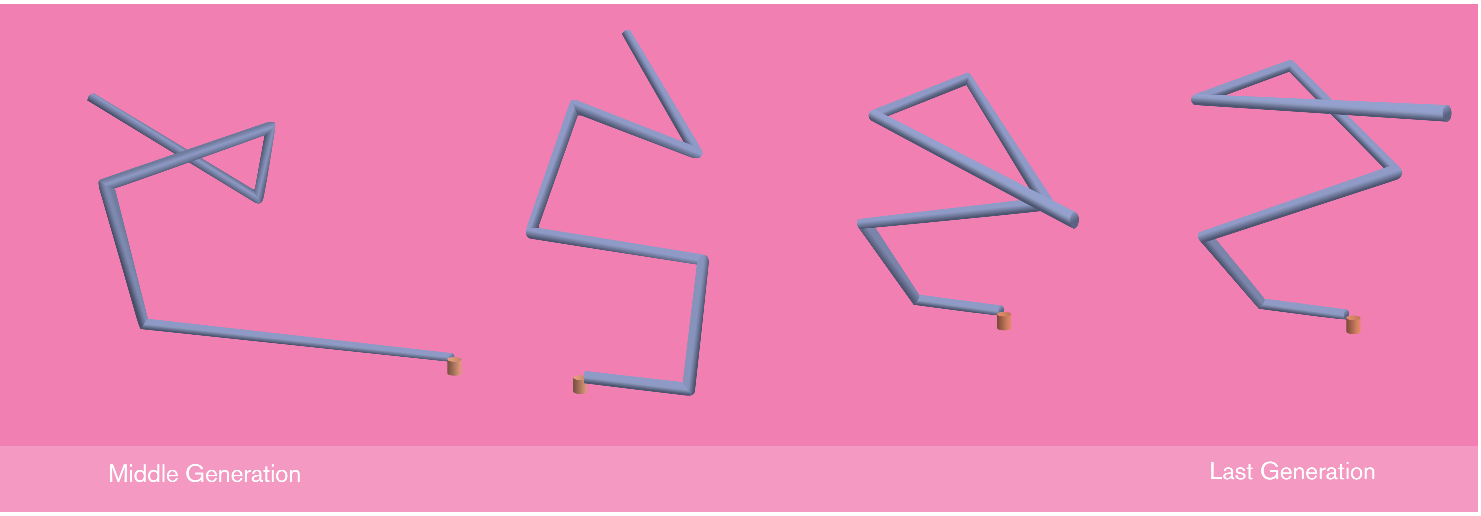
Goldberg documented his work in a 1989 textbook, a volume that would inspire other computer-savvy engineers to begin their own tinkering. By the mid-1990s, engineers at General Electric Research Center in Niskayuna, NY, had built evolutionary methods into an in-house design tool called EnGENEous, which was used to find the most efficient shape for the fan blades in the GE90 jet engines used on Boeing's 777 aircraft. EnGENEous allowed the GE90 team to eliminate one stage of the engine's compressor, which meant a reduction in engine weight and manufacturing costs without any sacrifice in aerodynamic performance. "After this initial success, the floodgates opened to use these types of tools in many different applications across all of GE's businesses," says Pete Finnigan, laboratory manager for advanced mechanical-design applications at the research center. Engineers at Rolls Royce, Honda, and Pratt and Whitney have followed suit, incorporating genetic algorithms into their own design processes.

Flagging Fraud

But while computers have grown powerful enough to apply evolutionary principles to all sorts of problems, the "haystacks" have been multiplying at an even more dramatic rate. Consider consumer fraud. Credit card companies estimate that $.07 per $100 charged to credit cards is lost to fraud, costing the industry more than $1 billion per year in the United States alone. Yet writing traditional software to identify fraudulent charges remains phenomenally difficult. Why? Because the people perpetrating the fraud are experts at modifying their behavior to evade detection. It's simply not possible to write a program that anticipates every possible scam.

But evolutionary algorithms can at least make computerized fraud detection more likely to succeed, argue the artificial-intelligence researchers who founded New York City–based Searchspace. The company sells a variety of programs that split up the haystack by looking for aberrant activity within precisely defined slices of existing account data, says Michael Recce, Searchspace's chief scientist. The software uses tools dubbed "sentinels," programmed with fraud detection rules. Multiple charges to the same debit card at a single store on a single day, for example, might automatically raise a red flag.

But the person racking up these purchases may simply be a forgetful Christmas shopper, not a thief. So the sentinels



JASON LOHN, NASA AMES RESEARCH CENTER

weigh in a variety of factors, such as a person's prior activity at that store, in order to avoid "false positives" and flag only accounts that human experts would agree are suspect. Says Recce, "You can set the fitness criteria in a way that delivers both minimal fraud loss and minimal good-customer loss."

Searchspace routinely hosts "pilots," essentially software bake-offs that pit its algorithms against potential clients' existing fraud detection systems. Participants bring in blind samples of historical data to see if Searchspace's sentinels plant red flags in all the right places. Invariably, says Recce, the sentinels turn up not only the preflagged accounts but also a few more miscreants lurking in the background noise. "I don't think there's been one of those presentations where we haven't had to pause things for a moment so that an executive could go step out to make a quick phone call," says Recce, smiling.

Helping Businesses Evolve

Ascent Technologies, Cambridge, MA
Ascent developed the SmartAirport Operations Center, which uses evolutionary algorithms to coördinate airport operations such as gate and ground traffic, baggage routing, and security staff scheduling.

Deere and Company, Moline, IL
The famous maker of John Deere tractors and lawnmowers uses evolutionary algorithms to breed assembly line schedules that best meet projects' cost, time, and safety requirements.

First Quadrant, Pasadena, CA
Evolutionary algorithms developed by Santa Monica, CA, think tank Rand have helped this hedge fund time stock trades and other investment decisions. First Quadrant has used evolutionary models to manage up to $6 billion in capital.

Nutech Solutions, Charlotte, NC
Nutech's evolutionary algorithms help General Motors make more money on automobile resales by suggesting the best times, locations, and prices at which to sell vehicles, given regional variations in auction prices, transportation costs, and the like.

Schlumberger, New York, NY
Software from the giant oil-field-services company uses evolutionary algorithms to help oil drillers decide when to drill new wells, based on the production of existing wells.

Patently Original

Now that evolutionary algorithms are outwitting humans, some researchers want to raise the bar even higher. At Stanford University, for example, professor of biomedical informatics John Koza—yet another Holland protégé—is exploring a closely related field called genetic programming. Evolutionary algorithms have fixed sets of instructions and merely vary the data they manipulate. Genetic programs are more like sexual organisms, capable of improving over time by shuffling bits of code among themselves. The "discoveries" made so far by Koza's programs range from novel computerized methods for sorting proteins to cutting-edge designs for electronic circuits.

Evolved designs can be unintelligible. But if they actually work, does it matter if we don't understand how?

The circuit designs emerged from Koza's work with Matthew Streeter of Carnegie Mellon University and Martin Keane of Econometrics, a marketing strategy consultancy based in Chicago. Together, the researchers built a program that draws schematic circuit diagrams. Their first challenge was to see whether the genetic approach could derive from scratch circuit designs already patented by past engineers. The program had little trouble generating simple designs that matched those patented in the 1930s and 1940s. Indeed, Koza began referring to the program as an "invention machine" and created a Web page that tracks the latest discoveries by "human competitive" software.

By the time Koza's group tested the fourth or fifth versions of their program, however, something even more surprising began to happen: the program kicked out circuit designs unpublished anywhere in the patent literature. Two of these designs—a pair of controller circuits that regulate feedback—were so original that Koza and his colleagues have taken out patents on them.

As proud as he is of his software, Koza isn't about to assign responsibility for the new designs to the program itself. The patents credit Keane, Koza, and Streeter, in that order. But there are a few new pseudophilosophical conundrums lurking here: If something is invented with no human near, is it really an invention? Who is the inventor? And if the invention actually works, does it matter if we don't understand how?

On that last point, says NASA's Lohn, "There are two schools of thought. One says I just need something that does X, Y, and Z, and if evolution gives me X, Y, and Z, that's all I care about. The other school wants to know what's in there and how it works. We can't really help those people, because we frequently see evolved designs that are completely unintelligible."

There's no need yet for humans to feel jealous of "human competitive" software, says Koza, since the ultimate goal is simply to hand over engineering's hardest drudge work to computers. He does foresee a time in the near future—perhaps 20 years from now—when genetic algorithms running on ultrafast computers will take over basic design tasks in fields as diverse as electronics and optics. But even then, Koza believes, human and machine intelligence will work in partnership. "We've never reached the place where computers have replaced people," Koza says. "In particular narrow areas, yes—but historically, people have moved on to work on harder problems. I think that will continue to be the case." ■

Sam Williams is a freelance technology writer based in Staten Island, NY. He is a frequent contributor to Salon.