Why we fell out of love with algorithms inspired by nature

June 3 2015, by Alexander Brownlee And John Woodward

Strictly for the birds? Credit: muratart

While computers are poor at creativity, they are adept at crunching through vast numbers of solutions to modern problems where there are numerous complex variables at play. Take the question of finding the best delivery plan for a distribution company – where best to begin? How many vehicles? Which stretches of road need to be avoided at which times? If you want to get close to a sensible answer, you need to ask a computer.

This is just one of millions of problems that are addressed by the field of metaheuristics, which is about developing algorithms that help you come up with the best possible answer in any situation where there are a large number of possible solutions.

It could be about devising job rosters that are as fair as possible. It could be about tuning the design of an engine or building to minimise energy or fuel usage. It could be about putting together the most economic flight schedule for an airport. For any discipline where a measure of quality can be provided, vast quantities of numbers have probably been crunched.

The solutions are far from perfect, however. Even for the fastest computers currently available, these problems are often challenging due to their sheer size. Exhaustively checking every possibility would typically take longer than the universe has existed.

Coming up with the perfect solutions to these kinds of problems is one of most high-profile conundrums for mathematics, known as the p=np question. While we wait for it to be solved, we have focused on developing algorithms that come up with solutions that are approximately the best instead.

Natural inspiration

Among the best known types of tools to come up with these approximate answers are called evolutionary algorithms (fully explained here). They take this name from the fact that they draw on the same narrative as Darwin's theory of evolution: that of a "population" of individuals competing, the fittest "parents" then producing "offspring" which form the next generation, so that the population becomes gradually "fitter" over time.

To see how this works in practice, take chemotherapy treatment. For every kind of cancer, the problem for oncologists is what dosage and frequency of each drug produces the best balance between eradicating the cancer and minimising the side-effects.

An evolutionary algorithm would start by randomly generating a few treatment regimes (the population). It would predict the resulting tumour size and side-effects after each treatment, then estimate the overall quality (fitness) of each regime. Pairs of regimes (parents) make a new regime (offspring) by choosing a proportion of the drug levels from each parent. This is repeated, the offspring replacing parents, until a good solution is found.

This is not about seeking to simulate biology as such. It is about taking how nature tackles problems as inspiration for how computers should solve problems. Having drawn on this since the days of Alan Turing, computer scientists have taken the ball and run with it. They have looked at how nature operates in specific situations, such as flocks of birds or ant colonies, and applied the same rules to their algorithms.

These have produced approaches to specific problems that have been remarkably effective, spanning engineering, medicine, economics, marketing, genetics, art, robotics, social sciences, physics and chemistry.

The problem

Over the past couple of decades, the research literature has filled up with endless new nature-based metaphors for algorithms. You can find algorithms based on the behaviour of cuckoos, bees, bats, cats, wolves, galaxy formation and black holes. Sometimes the metaphors even go beyond nature: musical composition, fireworks and even colonisation by imperial nations.

Much of this coincided with an old misconception that it was possible to develop one tool that can solve all these complex problems better than all the others. The way to get published in this field has been to show that your new algorithm solves a few test problems, making a case that yours could be the optimum tool that everyone has been looking for. But the reality is that while each new tool can be shown to perform well in specific cases, this Holy Grail doesn't exist.

All researchers have been doing is wasting time on developing new approaches that are probably little better than existing ones. And the language of each metaphor then invades the literature, distracting people from using the already sufficiently expressive terminology of mathematics and, above all, working together to find the best way forward.

Where next?

The backlash has begun: the Journal of Heuristics has revised its editorial policy to address this issue. Major figures in the field are calling for new approaches to be written in "metaphor-free language".

Yet this doesn't mean that nature-inspired algorithms are going to decline – not while arriving at approximate solutions to our complex modern problems is still the best that we can do. Instead the focus is shifting towards improving our understanding of how existing approaches work and improving their scientific value.

One theme is about devoting more time to looking at the relationships between the variables and solution quality in a given problem. In the past we have tended to know that they are connected but haven't tried to work out how. Remedying this should help us refine the tools that have already been developed so that they can search all the possible solutions to a problem in a more intelligent way.

Another theme has been about combining algorithms with classical mathematics to help reach solutions that we can be more confident are better than what we have had in the past. We are also looking at introducing rules from software engineering known as formal design patterns, which essentially set down prescribed ways of solving a given problem to stop people constantly trying to come up with radical alternatives.

All this work represents a move in the right direction. Perhaps a retreat from all the bats and the bees will make the research area harder to communicate to the public. But it can only be good for science that good old-fashioned computer science and mathematics are making a comeback. If it means that we build better houses, develop better cancer treatments, improve our airline scheduling and so forth, it will have been worth the effort.

This story is published courtesy of The Conversation *(under Creative Commons-Attribution/No derivatives).*

Source: The Conversation

Citation: Why we fell out of love with algorithms inspired by nature (2015, June 3) retrieved 27 April 2024 from https://phys.org/news/2015-06-fell-algorithms-nature.html