

HOW ENGINEERS USE EVOLUTION TO INVENT THINGS

Sean P. Walton^{1*}, Ben Evans², Ben Smith² and Jakub Vincalek¹

¹Computational Foundry, Department of Computer Science, Faculty of Science and Engineering, Swansea University, Swansea, United Kingdom

²Aerospace Engineering, Faculty of Science and Engineering, Swansea University, Swansea, United Kingdom

YOUNG REVIEWERS:



CHRIS
AGE: 15



LORNA
AGE: 13

EVOLUTION

The process by which plants and animals change over a long period of time.

You may have heard of evolution in terms of plants and animals, but did you know that this natural process can also be used by engineers to invent things? Animals and plants have evolved in amazing ways to survive in their environments. Biologists have been investigating how evolution works for a long time. Mathematicians and computer scientists have worked alongside biologists to create computer programs that can evolve designs, to help engineers invent things. These are called evolutionary optimization algorithms, and they can be used to evolve faster airplanes, stronger bridges, or even better video games. In this article, we will explain how these algorithms work and what their strong and weak points are.

UNDERSTANDING EVOLUTION

Evolution is a word we use to describe how plants and animals change over long periods of time. Most animals in a single species may look

INHERITANCE

The process by which traits, such as hair color, are passed on from parents to their children.

MUTATION

A random change to genes that can cause the change of a trait.

NATURAL SELECTION

The process by which only creatures with useful traits live long enough to have children, and pass on those traits.

EVOLUTIONARY OPTIMIZATION ALGORITHM

An algorithm which uses the concepts of evolution to optimize something, often using computer simulations.

OPTIMIZATION

The process of making something the best it can be.

Figure 1

Evolutionary optimization algorithms are used for lots of different things, and it is impossible to list them all. This diagram shows some examples of where evolutionary optimization algorithms have been used successfully. Where else do you think they could be used?

similar, but they are all slightly different—which means I can pick my dog out from a crowd of dogs! When two parents have a child, the child will be a bit like its parents. Perhaps the child has the same hair color as its mother and is tall like its father. The passing on such similarities between generations is called **inheritance**. There will also be slight differences between the child and its parents, maybe a bigger nose or better eyesight. These differences are called **mutations**.

In nature, small mutations can mean the difference between life and death for animals and plants. For example, if two zebras are running away from a lion that is looking for lunch, the fastest runner will be the one that escapes and survives. The surviving zebra will then be able to have babies, which might inherit their parent's running speed. Fast zebra babies are also more likely to survive and have babies themselves, so over time, the population of zebras will become fast runners. This process is called **natural selection**.

Scientists have observed that, over a long period of time, species evolve to survive in their environments. This is due to a combination of inheritance, mutation, and natural selection. Creatures have evolved in amazing ways, to do incredible things—from moths that can change color to hide from predators to lizards that can drink water through their skin. The question is, can we use these ideas from nature to create amazing *inventions* that can do incredible things?

EVOLUTIONARY OPTIMIZATION ALGORITHMS

Evolutionary optimization algorithms use ideas from evolution to help engineers invent things. Several computer scientists came up with the idea of evolutionary **optimization** algorithms at around the same time, between 1950 and 1960 [1]. Evolutionary optimization algorithms can be used for all sorts of things, like designing airplanes [2], building levels in video games [3], and even creating art [4]. Figure 1 gives

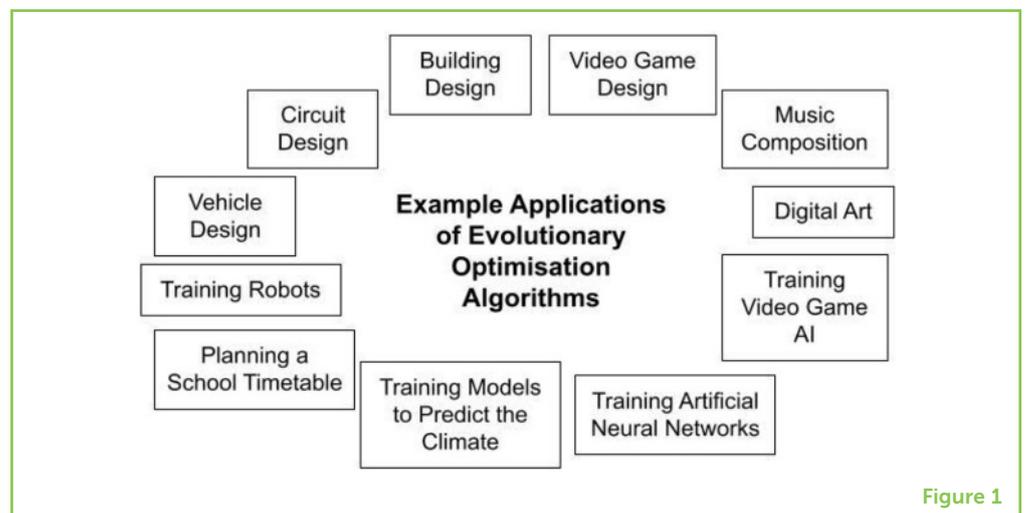


Figure 1

more examples of how evolutionary optimization algorithms have been used. To explain how evolutionary optimization algorithms work, let us imagine we are trying to design a bridge.

The first thing we must do is decide what makes a bridge design good and what makes it bad. To keep things simple, we will say that we want our bridge to hold as much weight as possible—this is our purpose, or objective. Once we have decided on an objective, we can compare two bridge designs and decide which is better—the one that holds the most weight.

The next step is to create some starting bridge designs. This could be done randomly using a computer, or we could spend some time designing them ourselves. Either way, we will need a lot of designs! We call this set of designs our population. By running simulations on a computer, we can predict how much weight each of our bridge designs can hold. Using this information, we can put our designs in order based on how good they are (Figure 2).

Figure 2

Once a population of designs has been created by a computer program or by engineers, computer simulations are run to see how well each design meets the objective. In this example, we have created four possible bridge designs. For each design, we have measured how much weight the bridge can carry, measured in **Newtons**, before it breaks. The best bridges in the population—those that have the highest weight capacity—are then used as “parents” to create “children” for the next generation.

NEWTONS

We measure forces in Newtons. For example, weight is the force at which gravity pulls an object down toward the Earth – measured in Newtons.

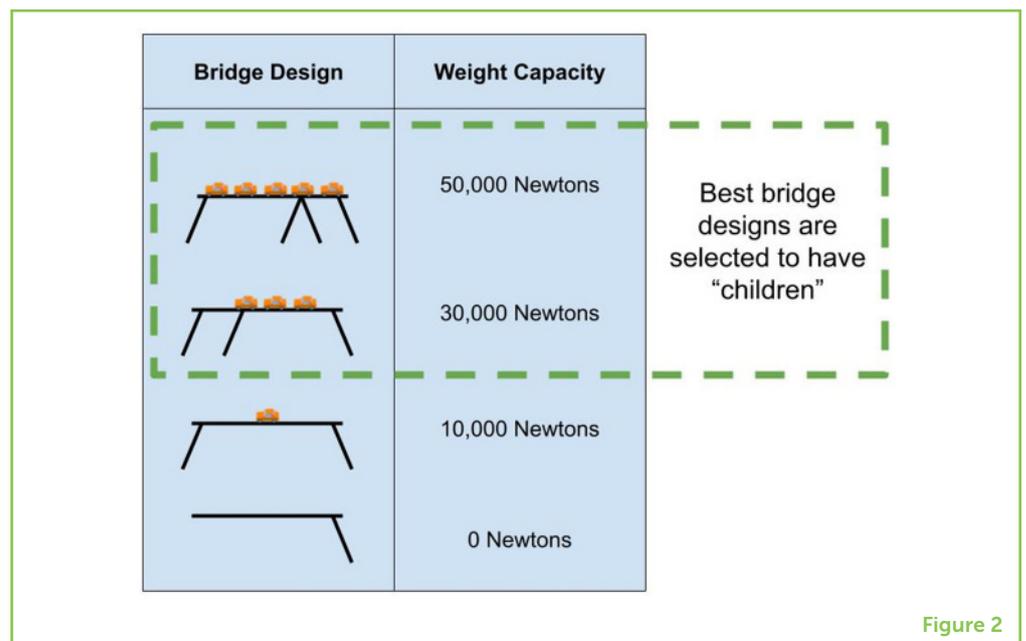
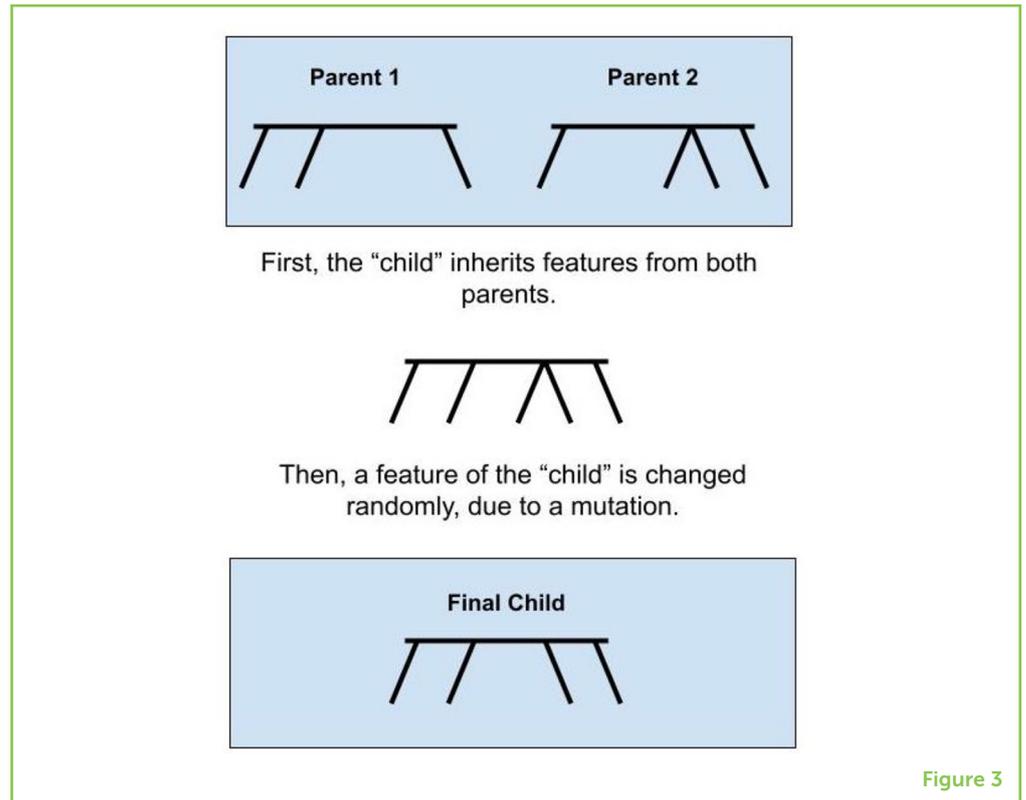


Figure 2

Once we know which of the designs in our population can hold the most weight, we can select them to be “parents”—similar to the situation in nature, in which only the fastest zebras survive to have babies. Of course, bridges cannot actually have “children,” but scientists can write computer programs which are able to combine two good designs to produce a new “child” design that shares features of both “parents.” Perhaps the child bridge has the same height as one of its parents, and the same number of legs as the other. Then, we apply a mutation by randomly changing something in the new design. Maybe we add an extra leg or double the height. An example of two parent designs creating a child design is shown in Figure 3.

Figure 3

Once the best designs have been selected from the population, those designs are combined by the algorithm, to produce “children.” First, we create a new design that has features from both parents. In this example the child design inherits the legs from both of its parent designs. Then we apply a mutation (random change) to the child. Here, the mutation caused the child to lose one of the legs inherited from Parent 2.



We keep generating new designs using inheritance and mutation until we have a new population of designs. Then we can run simulations on these new designs, decide which ones are best, and repeat the process. All of this is done automatically for us, by a computer. Over a long period of time, which could be weeks or months, we will find better and better bridge designs. Eventually, if we are really lucky, we might find a bridge design no one has ever thought of before!

WHAT ARE EVOLUTIONARY OPTIMIZATION ALGORITHMS GOOD AND BAD AT?

Evolutionary optimization algorithms can be really useful when we need to design something new and have no idea where to start. These computer programs are very good at surprising engineers with weird and sometimes wonderful designs. For example, NASA has used evolutionary optimization algorithms to design tiny antennae parts for satellites [5]. The evolutionary optimization algorithm created strange designs that surprised the engineers. When they tested the evolved designs, they found that those designs worked better than antennas designed by humans.

Evolutionary optimization algorithms use randomness in lots of different ways. While some scientists think this is a good thing, many engineers and designers disagree. If you use an evolutionary optimization algorithm to do the same thing twice, it might give you

two different answers—which means you can never quite be sure that the algorithm is giving you the best design possible. This randomness makes some engineers distrust evolutionary optimization algorithms [6] because it is impossible to understand why the algorithm picks one design instead of another. Would you fly in an airplane designed by an evolutionary optimization algorithm? Scientists are currently trying to design algorithms that can explain the decisions they make, to help engineers trust them. The randomness also means that designers will want to run an evolutionary optimization algorithm many times, to be sure they are getting the best designs. Repeatedly running evolutionary optimization algorithms can take a long time, even more time than it would take experienced engineers to sit down and design something themselves.

There are some situations in which evolutionary optimization algorithms struggle. Sometimes there is more than one objective to think about. When designing bridges, imagine if we also cared about how long or high the bridge was. How would we decide which bridges were the best when considering both weight capacity *and* height/length? Adding more objectives means it will take the evolutionary optimization algorithm longer to find a solution. Another example would be that of a car-design engineer who needs to think about how good the car looks. It is difficult to get a computer to predict if a design looks good or bad. The evolutionary optimization algorithm might design a car that no one would want to be seen driving! For these sorts of things, we need a human to decide.

Another problem caused by relying on an algorithm to make decisions is that the **algorithm** can only make decisions based on **computer simulations**, which do not always get things right. It is not unusual for an evolutionary optimization algorithm to take hours to suggest a design that the simulation thinks will be good, only for an engineer to take seconds to realize the simulation is wrong. When this happens scientists and engineers need to fix the errors in the simulation. So, while evolutionary optimization algorithms are useful tools, they will never replace human designers.

THE FUTURE OF EVOLUTIONARY OPTIMIZATION ALGORITHMS

As we have seen, evolutionary optimization algorithms take processes we see in nature and use them to solve problems we cannot solve ourselves. One of the biggest problems we face right now is climate change and the damage we are causing to nature through pollution. Researching new evolutionary optimization algorithms and new ways to use them will help us design better, less polluting, vehicles, and aircraft. But even though we always need humans to design and use these algorithms, in a way, nature has come to rescue itself. What other

ALGORITHM

A process or set of rules that must be followed to solve a problem.

COMPUTER SIMULATION

A computer program which predicts what will happen in real life. For example, computer simulations are used for predictions of weather forecasts. Simulations will never be perfect and are always just our best guess on what might happen.

problems do you think evolutionary optimization algorithms can help us solve?

TRY IT YOURSELF!

If you would like to have a go designing a car using an evolutionary algorithm, then have a play with a game we created! You can download or play the game in a browser at this link (<https://pillbuginteractive.itch.io/genetic-car-designer>) to get a feel of what it is like working with an evolutionary algorithm. Be sure to share your best designs with your friends!

ACKNOWLEDGMENTS

David Chena, Ben Vincenta, and Kafyad Wasti worked as summer research interns to help structure and draft this article. BS's is part funded by an EPSRC Doctoral Training Partnership at Swansea University. JV is funded by the EPSRC Centre for Doctoral Training in Enhancing Human Interactions and Collaborations with Data and Intelligence Driven Systems (EP/S021892/1).

REFERENCES

1. Mitchell, M., and Taylor, C. E. 1999. Evolutionary computation: an overview. *Annu. Rev. Ecol. Syst.* 30:593–616.
2. Naumann, D. S., Evans, B., Walton, S., and Hassan, O. 2016. A novel implementation of computational aerodynamic shape optimisation using Modified Cuckoo Search. *Appl. Math. Model.* 40:4543–59. doi: 10.1016/j.apm.2015.11.023
3. Walton, S. P., Rahat, A. A. M., and Stovold, J. 2021. Evaluating mixed-initiative procedural level design tools using a triple-blind mixed-method user study. *IEEE Trans. Comput. Intell. AI Games* doi: 10.1109/TG.2021.3086215
4. den Heijer, E., and Eiben, A. E. Investigating aesthetic measures for unsupervised evolutionary art. *Swarm Evol. Comput.* (2014) 16:52–68. doi: 10.1016/j.swevo.2014.01.002
5. Hornby, G., Globus, A., Linden, D., and Lohn, J. 2006. "Automated antenna design with evolutionary algorithms," in *Space 2006* (Reston, VA: American Institute of Aeronautics and Astronautics). doi: 10.2514/6.2006-7242
6. Vincalek, J., Walton, S., and Evans, B. 2021. "It's the Journey Not the Destination: Building Genetic Algorithms Practitioners Can Trust," in *2021 Genetic and Evolutionary Computation Conference Companion (GECCO'21 Companion)* (New York, NY: ACM). doi: 10.1145/3449726.3459483

SUBMITTED: 14 September 2021; **ACCEPTED:** 28 July 2022;

PUBLISHED ONLINE: 16 August 2022.

EDITOR: Marco Aldi, Virginia Commonwealth University, United States

SCIENCE MENTORS: Kostas Karpouzis and Carol Jagger

CITATION: Walton SP, Evans B, Smith B and Vincalek J (2022) How Engineers Use Evolution to Invent Things. *Front. Young Minds* 10:776918. doi: 10.3389/frym.2022.776918

CONFLICT OF INTEREST: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

COPYRIGHT © 2022 Walton, Evans, Smith and Vincalek. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

YOUNG REVIEWERS

CHRIS, AGE: 15

I like playing games on my laptop or mobile, so I started studying advanced math and computer science as an excuse: I just had to try math concepts in computer code and then put them to use in games, e.g., when shooting, running or performing a free fall! I am a reserve member of the National Team for the math Olympiad and currently working toward my first game.



LORNA, AGE: 13

I am in High School and love Maths and Science—I also love playing sports like water polo and volleyball. I have 2 siblings—an older brother and a younger sister, and two amazing parents. I take great inspiration from my mentor, a very accomplished mathematician and my friend.



AUTHORS

SEAN P. WALTON

Sean P. Walton is a computer scientist at the Computational Foundry at Swansea University. His research interests are in developing computer tools that can help people be creative and trying to understand how those tools make people feel. The tools he creates all use evolutionary optimization algorithms. Sean is also an independent video game designer. His company, Pill Bug Interactive, has created three games on PC and Nintendo Switch. *s.p.walton@swansea.ac.uk



BEN EVANS

Dr. Ben Evans is an associate professor in aerospace engineering at Swansea University and is a member of the design team for the Bloodhound Land Speed Record project, with overall responsibility for aerodynamic modeling. Dr. Evans' research interests involve developing computer simulations to help design



better aircraft. He also has interests in engineering public engagement and engineering education.



BEN SMITH

Ben Smith is a Ph.D. researcher in aerospace engineering at Swansea University, funded jointly by EPSRC and an industry sponsor. Ben's background is in mathematics and computer science, and he worked in the defense sector prior to starting in academia. He currently researches human-in-the-loop design optimization techniques.



JAKUB VINCALEK

Jakub Vincalek is a Ph.D. researcher and is a candidate on the EPSRC CDT in Enhancing Human Interactions and Collaborations with Data and Intelligence Driven Systems. The EPSRC CDT is a training scheme for future computer science researchers. Jakub has master's degrees in aerospace engineering and computer science, and he has previously worked in the space and defense industry.