

Analysis shows 'Super Mario Brothers' is harder than NP-hard

May 31 2016

Completing a game of "Super Mario Brothers" can be hard—very, very hard. That's the conclusion of a new paper from researchers at MIT, the University of Ottawa, and Bard College at Simon's Rock. They show that the problem of solving a level in "Super Mario Brothers" is as hard as the hardest problems in the "complexity class" PSPACE, meaning that it's even more complex than the traveling-salesman problem, or the problem of factoring large numbers, or any of the other hard problems belonging to the better-known complexity class NP.

In a standard "Super Mario Brothers" game, Mario runs across terrain that unspools from the right side of the screen. While battling monsters, he must complete various tasks, which can involve navigating brick structures that may rise from the ground plane of the game but may also hang in the air unsupported. The completion of a level is marked by Mario's reaching a flagpole.

The new paper doesn't attempt to establish that any of the levels in commercial versions of "Super Mario Brothers" are PSPACE-hard, only that it's possible to construct PSPACE-hard levels from the raw materials of the "Super Mario" world.

The work follows on a paper from two years ago, with two of the same coauthors, which showed that "Super Mario Brothers" is at least as hard as the hardest problems in NP. But at the time, the researchers couldn't determine whether it was any harder. "PSPACE is its final home," says Erik Demaine, an MIT professor of electrical engineering and [computer](#)

[science](#) and a co-author on both papers.

Demaine and his colleagues—Giovanni Viglietta, a postdoc in electrical engineering and computer science at the University of Ottawa and a coauthor of the earlier paper; and Aaron Williams, a professor of computer science at Bard College at Simon's Rock—will present their new paper at the International Conference on Fun with Algorithms next week.

Questions of proportion

Theoretical computer scientists categorize algorithms according to their execution times, which they measure in terms of the number of data items the algorithms manipulate. An algorithm for finding the largest number in a list of N numbers, for instance, has a running time proportional to N . An algorithm that, say, calculates the flying distances between N airports on a map has a running time proportional to N^2 , because for every airport, it has to calculate the distance to each of the others.

Algorithms whose running times are proportional to N raised to a power are called "polynomial." A polynomial algorithm whose running time is proportional to, say, N^3 is slower than one whose running time is proportional to N . But those differences pale in comparison to the running times of exponential algorithms, whose running time is proportional to 2^N .

If an algorithm whose execution time is proportional to N takes a second to perform a computation involving 100 elements, an algorithm whose execution time is proportional to N^3 takes almost three hours. But an algorithm whose execution time is proportional to 2^N takes 300 quintillion years.

The complexity class NP is a set of problems whose solutions can be verified in polynomial time, even if finding those solutions takes—as far as anyone knows—exponential time. To use the most familiar example, factoring a 1,000-digit number is probably beyond the capacity of all the computers in the world in the lifetime of the universe, but verifying a solution—multiplying the factors together—is something a smartphone could do.

Like NP, PSPACE contains problems that appear to require exponential time to solve. But the hardest problems in PSPACE—the PSPACE-hard problems—also take exponential time to verify. In some sense, that makes PSPACE a natural place for a video game to reside. Figuring out how to complete a fiendishly difficult level of "Super Mario Brothers" could take a long time, but so could navigating that level, even with the solution in hand.

Fundamental components

In their earlier paper, Demaine, Viglietta, and colleagues described a generic video-game structure that they call a locked door. The structure must have a path through it that can be either safe to traverse or not, and there must be a way for the player to switch the state of the path.

Because the locked door has two possible states, it can represent a bit of computer memory, and because it has a path through it that can be opened or closed, it can serve as an element of a computational circuit. The researchers were able to show that any computational problem could be described by locked doors strung together in the right configuration. If the problem is exponentially hard, then figuring out how to complete the level is exponentially hard, too.

In the earlier paper, Demaine, Viglietta, and their colleagues demonstrated how to build locked doors in several versions of the game

"Donkey Kong Country," but they couldn't figure out how to build one in "Super Mario Brothers." "We thought it was impossible," Demaine says.

But it's not. The locked door described in the new paper uses a monster from the "Mario Brothers" world called a "spiny," which will move back and forth continuously between two barriers but will never spontaneously jump either of them. As the spiny approaches a barrier, however, Mario can bump the floor beneath it and send it over. In the researchers' new locked door, if the spiny is on one side of a barrier, the path through the structure is untraversable; if it's on the other, the path is open. And separate paths through the structure allow Mario to bump the spiny from one side to the other.

Fun and games

The result could have implications beyond the design of ever-more-baffling games of "Super Mario Brothers." Mathematically, video games are not very different from computational models of real-world physical systems, and the tools used to prove complexity results in one could be adapted to the other.

"I'm really excited about these kinds of hardness proofs, and I've been pushing them a lot in the last couple years," Demaine says. "I even taught an entire course about them. I'm pretty good at them, just through practice, and I wanted to somehow distill that into a form that other people could learn. So the class was a first attempt to do that. But it's already a really useful reference. I go and look at these lecture notes all the [time](#) to see, 'Is that variation of this problem hard?'"

"My hope is through this class and these kinds of papers to encourage more people to do this, because it really does build up a lot of expertise that makes it easier to conquer problems," he continues. "The more practice we get as a collective, the better we are at solving these types of

problems. And it's important to know the limitations of algorithms."

Provided by Massachusetts Institute of Technology

Citation: Analysis shows 'Super Mario Brothers' is harder than NP-hard (2016, May 31) retrieved 26 April 2024 from <https://phys.org/news/2016-05-analysis-super-mario-brothers-harder.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.