How the Mathematical Conundrum Called the ‘Knapsack Problem’ Is All Around Us

A litany of issues in business, finance, container ship loading and aircraft loading derive from this one simple dilemma

The "knapsack problem" is a widespread computing challenge—and no, it doesn't have to do just with literal backpacks. (golubovy / iStock)

By Elizabeth Landau
smithsonianmag.com
March 9, 2020

360 2 4 152

Imagine you’re a thief robbing a museum exhibit of tantalizing jewelry, geodes and rare gems. You're new at this, so you only brought a single backpack. Your goal should be to get away with the most valuable objects without overloading your bag until it breaks or becomes too heavy to carry. How do you choose among the objects to maximize your loot? You could list all the artifacts and their weights to work out the answer by hand. But the more objects there are, the more taxing this calculation becomes for a person—or a computer.

This fictional dilemma, the “knapsack problem,” belongs to a class of mathematical problems famous for pushing the limits of computing. And the knapsack problem is more than a thought experiment. “A lot of problems we face in life, be it business, finance, including logistics, container ship loading, aircraft loading — these are all knapsack problems,”

says Carsten Murawski, professor at the University of Melbourne in Australia. “From a practical perspective, the knapsack problem is ubiquitous in everyday life.”

Researchers once took advantage of the problem’s complexity to create computer security systems, but these can now be cracked since the problem has been so well studied. Today, as technology capable of shattering the locks on our digital communications loom on the horizon, the knapsack problem may inspire new ways to prepare for that revolution.

All or Nothing

The knapsack problem belongs to a class of “NP” problems, which stands for “nondeterministic polynomial time.” The name references how these problems force a computer to go through many steps to arrive at a solution, and the number increases dramatically based on the size of the inputs—for example, the inventory of items to choose from when stuffing a particular knapsack. By definition, NP problems also have solutions that are easy to verify (it would be trivial to check that a particular list of items does, in fact, fit in a backpack).

“The problem the theoreticians started to look at was how *efficiently* a particular task can be carried out on a computer,” writes Keith Devlin in the book *The Millennium Problems*. For example: Given a list of 1 million museum artifacts with their weights and monetary values, and a backpack limited to 25 pounds, a computer would have to run through every possible combination to generate the single one with the most lucrative haul. Given an indefinite amount of time, a computer could use brute force to optimize large cases like this, but not on timescales that would be practical.

“We think you could cover the entire Earth with processors and run them until the heat death of the universe and still fail to solve relatively small instances of appropriate versions of these problems,” says Noah Stephens-Davidowitz, a Microsoft Research Fellow at the Simons Institute in Berkeley, California.

Some NP problems like the knapsack example have a special property: In the early 1970s, Stephen Cook and Richard Karp showed that a variety of NP problems could be converted into a single problem of formal logic. Therefore, if one could be solved and verified efficiently with an algorithm, they all could. This property is known as “NP completeness.”

One of the most stubborn questions in computer science and mathematics is whether these “NP” problems, including the knapsack problem, are truly different from “P” problems, those that can be solved in what is called polynomial time. If P=NP, then it’s possible to solve every problem whose solutions are easy to verify, says Stephens-Davidowitz. So, if this inequality persists, the general knapsack problem will always be hard.

Keeping Things Secret

Cryptography researchers love problems that are difficult for computers to solve because they’re useful in encrypting digital messages. Knapsack-problem-like security codes are not useful for this, as they're too easily cracked, but more complicated methods inspired by this problem are being developed, and may one day play a role in outwitting the next generation of computing.

In an early knapsack-style encryption method, one person’s private key would be a list of numbers in which each is larger than the sum of its predecessors. Exchanges involving that person would use a public key that looks random but is made up of numbers from the first list with specific transformations applied. For example, if the public key is [2, 3, 4, 5], the transmitted message “1, 0, 0, 1” would be encoded as 2+0+0+5 = 7 (because 2*1=2, 3*0=0, 4*0=0, and 5*1=5). Secret numbers involved in the conversions between keys allow the original message to be unveiled.

For this to work, a computer must also figure out whether any given number can be written as the sum of a subset of numbers in the private key, which becomes an easy knapsack problem. It’s akin to filling a backpack with a batch of such differently sized items — like a ring, a painting, a car and a house — and knowing you can’t stuff in anything else after you’ve checked that the ring and the painting fit. Cryptographers Ralph Merkle and Martin Hellman described this idea in 1978, but others figured out how to crack it by the early 1980s.

Private information exchanges on today’s internet often use keys involving large prime numbers, and while factoring big numbers is difficult, it’s not thought to belong to the same “NP complete” class as the knapsack problem. However, computer scientists are already gearing up for a future in which quantum computers can quickly unlock *these* keys.

Quantum computers rely on the principles of quantum mechanics, which says a particle is not located in a single position but has a probability of being in many different places unless it is pinned down and measured. While normal computers encode information in 0s and 1s, each “qubit” in a quantum computer would have a wide range of possible

states related to the properties of particles. Quantum computers wouldn't be useful for browsing the internet or writing a screenplay in a coffee shop, but they would unleash never-before-seen power on a few types of math problems. Unfortunately, those math problems make up the foundations of modern cybersecurity.

"In some sense, we got really unlucky," Stephens-Davidowitz says. "We managed to rest the security of the internet on the hardness of some of the very few problems that seem to be hard for classical computers but easy for quantum computers."

While quantum computing is in its infancy, some researchers say we're behind in preparing for it. In 2016, the National Institute of Standards and Technology (NIST) called for new quantum-resistant encryption methods, announcing 26 semi-finalists last year. One such type of algorithm being developed is called lattice-based cryptography. Instead of using numbers, it uses keys that exist in multiple dimensions and involve the formation of a lattice structure made of equally-spaced points in space. The question is where those points are, and how close a given random point is to the coordinates of a lattice. At its heart, this is a knapsack problem in more than one dimension.

"My current obsession is trying to figure out how secure these lattice-based things are, ideally before we use them to run the internet," Stephens-Davidowitz says.

It remains unclear how far we really are from game-changing quantum computing. Still, many cryptography researchers see an urgent threat. Hackers could be intercepting encrypted private communications and saving the for the day quantum computers are available.

"This means that we need quantum-resistant cryptography much earlier than we expect quantum computer[s] to reach their full potential," said Leo Ducas, researcher at the Centrum Wiskunde & Informatica in the Netherlands.

Routing and Rerouting

Beyond cryptography research, the knapsack problem and its NP complete cousins are everywhere in real life. For example, you may have heard of the "traveling salesman" problem, which is also NP complete. The challenge here is to find the shortest route for a salesman to travel between a given number of cities before returning to the starting point. Closely related is the vehicle routing problem, which considers multiple vehicles making deliveries.

Luciana Buriol, associate professor at the Universidade Federal do Rio Grande do Sul in Brazil, has attacked this problem to try to find new approaches for the health care sector. She worked with a home care service where physicians and nurses visit patients in their homes and helped optimize their routes, given a limited number of cars available for transportation.

"Given 300 patients and 15 cars, you cannot find the solution in a reasonable time," she said. "If you have days for running the algorithm you will find — but you have to find [it] in less than 2 hours, otherwise you will never use [it] in practice."

No single one-size-fits-all algorithm can solve these problems. Instead, Buriol finds quick ways to arrive at useful approximations so they can be put into action.

Knapsacks All Around Us

For those of us who are not computer scientists and face these kinds of problems in real life, how good are we? Murawski's group finds preliminary results that when you give humans knapsack-like problems, we also struggle mightily. In small experiments in which participants were asked to fill a backpack on a computer screen with items carrying stated values and weights, people tended to have a harder time optimizing the backpack's contents as the number of item options increased—the same problem computers have. The researchers say this finding may be related to "choice overload": the way we freeze up when given too many choices, even in simple situations like buying jam at a grocery store.

Yet, in the real world, we get by. Paying attention is also a knapsack problem. When driving, we face a cornucopia of possible distractions such as birds, clouds, the radio, and surrounding buildings. We must put only the most pertinent stimuli in our mental knapsacks—and generally, we do.

The question remains: Given that NP complete problems are more difficult for computers than other kinds of conundrums, are they also harder for people? The limited initial results suggest they could be, which surprised

Murawski.

“If this turns out to be the case, it would suggest that hardness of such problems is a feature of the problems—a property of nature—and not in the eye of the beholder,” Murawski says.

About Elizabeth Landau

Elizabeth Landau is a science writer and editor who splits her time between Pasadena, California, and Washington, D.C. She holds degrees from Princeton University and the Columbia University Graduate School of Journalism.

|

360 2 4 152