

1 of 2 DOCUMENTS

NewScientist

New Scientist

August 11, 2012

The world maker

BYLINE: Richard Elwes**SECTION:** FEATURES; No. 1014**LENGTH:** 3767 words**HIGHLIGHT:** Is time running out for the clever piece of maths that runs modern life, asks Richard ElwesThe **algorithm** that **runs the world**

Its services are called upon thousands of times a second to ensure the world's business runs smoothly -; but is its mathematics sound?

YOU MIGHT not have heard of the **algorithm** that **runs the world**. Few people have, though it can determine much that goes on in our day-to-day lives: the food we have to eat, our schedule at work, when the train will come to take us there. Somewhere, in some server basement right now, it is probably working on some aspect of your life tomorrow, next week, in a year's time.

Perhaps ignorance of the algorithm's workings is bliss. The door to Plato's Academy in ancient Athens is said to have borne the legend "let no one ignorant of geometry enter". That was easy enough to say back then, when geometry was firmly grounded in the three dimensions of space our brains were built to cope with. But the algorithm operates in altogether higher planes. Four, five, thousands or even many millions of dimensions: these are the unimaginable spaces the algorithm's series of mathematical instructions was devised to probe.

Perhaps, though, we should try a little harder to get our heads round it. Because powerful though it undoubtedly is, the algorithm is running into a spot of bother. Its mathematical underpinnings, though not yet structurally unsound, are beginning to crumble at the edges. With so much resting on it, the algorithm may not be quite as dependable as it once seemed.

To understand what all this is about, we must first delve into the deep and surprising ways in which the abstractions of geometry describe the world around us. Ideas about such connections stretch at least as far back as Plato, who picked out five 3D geometric shapes, or polyhedra, whose perfect regularity he thought represented the essence of the cosmos. The tetrahedron, cube, octahedron and 20-sided icosahedron embodied the "elements" of fire, earth, air and water, and the 12-faced dodecahedron the shape of the universe itself.

Things have moved on a little since then. Theories of physics today regularly invoke strangely warped geometries unknown to Plato, or propose the existence of spatial dimensions beyond the immediately obvious three. Mathematicians, too, have reached for ever higher dimensions, extending ideas about polyhedra to mind-blowing "polytopes" with four, five or any number of dimensions.

A case in point is a law of polyhedra proposed in 1957 by the US mathematician Warren Hirsch. It stated that the maximum number of edges you have to traverse to get between two corners on any polyhedron is never greater than the number of its faces minus the number of dimensions in the problem, in this case three. The two opposite corners on a six-sided cube, for example, are separated by exactly three edges, and no pair of corners is four or more apart.

Hirsch's rule holds true for all 3D polyhedra. But it has never been proved generally for shapes in higher dimensions. The expectation that it should translate has come largely through analogy with other geometrical rules that have proved similarly elastic (see "Edges, corners and faces", page 35). When it comes to guaranteeing short routes between points on the surface of a 4D, 5D or 1000D shape, Hirsch's rule has remained one of those niggling unsolved problems of mathematics -; a mere conjecture.

How is this relevant? Because, for today's mathematicians, dimensions are not just about space. True, the concept arose because we have three coordinates of location that can vary independently: up-down, left-right and forwards-backwards. Throw in time, and you have a fourth "dimension" that works very similarly, apart from the inexplicable fact that we can move through it in only one direction.

But beyond motion, we often encounter real-world situations where we can vary many more than four things independently. Suppose, for instance, you are making a sandwich for lunch. Your fridge contains 10 ingredients that can be used in varying quantities: cheese, chutney, tuna, tomatoes, eggs, butter, mustard, mayonnaise, lettuce, hummus. These ingredients are nothing other than the dimensions of a sandwich-making problem. This can be treated geometrically: combine your choice of ingredients in any particular way, and your completed snack is represented by a single point in a 10-dimensional space.

Brutish problems

In this multidimensional space, we are unlikely to have unlimited freedom of movement. There might be only two mouldering hunks of cheese in the fridge, for instance, or the merest of scrapings at the bottom of the mayonnaise jar. Our personal preferences might supply other, more subtle constraints to our sandwich-making problem: an eye on the calories, perhaps, or a desire not to mix tuna and hummus. Each of these constraints represents a boundary to our multidimensional space beyond which we cannot move. Our resources and preferences in effect construct a multidimensional polytope through which we must navigate towards our perfect sandwich.

In reality, the decision-making processes in our sandwich-making are liable to be a little haphazard; with just a few variables to consider, and mere gastric satisfaction riding on the outcome, that's not such a problem. But in business, government and science, similar optimisation problems crop up everywhere and quickly morph into brutes with many thousands or even millions of variables and constraints. A fruit importer might have a 1000-dimensional problem to deal with, for instance, shipping bananas from five distribution centres storing varying numbers of fruit to 200 shops with different numbers in demand. How many items of fruit should be sent from which centres to which shops while minimising total transport costs?

A fund manager might similarly want to arrange a portfolio optimally to balance risk and expected return over a range of stocks; a railway timetabler to decide how best to roster staff and trains; or a factory or hospital manager to work out how to juggle finite machine resources or ward space. Each such problem can be depicted as a geometrical shape whose number of dimensions is the number of variables in the problem, and whose boundaries are delineated by whatever constraints there are (see diagram, left). In each case, we need to box our way through this polytope towards its optimal point.

This is the job of the algorithm.

Its full name is the simplex algorithm, and it emerged in the late 1940s from the work of the US mathematician George Dantzig, who had spent the second world war investigating ways to increase the logistical efficiency of the US air force. Dantzig was a pioneer in the field of what he called linear programming, which uses the mathematics of multidimensional polytopes to solve optimisation problems. One of the first insights he arrived at was that the optimum value of the "target function" -; the thing we want to maximise or minimise, be that profit, travelling time or whatever -; is guaranteed to lie at one of the corners of the polytope. This instantly makes things much more tractable: there are infinitely many points within any polytope, but only ever a finite number of corners.

If we have just a few dimensions and constraints to play with, this fact is all we need. We can feel our way along the edges of the polytope, testing the value of the target function at every corner until we find its sweet spot. But things rapidly escalate. Even just a 10-dimensional problem with 50 constraints -; perhaps trying to assign a schedule of work to 10 people with different expertise and time constraints -; may already land us with several billion corners to try out.

The simplex algorithm finds a quicker way through. Rather than randomly wandering along a polytope's edges, it implements a "pivot rule" at each corner. Subtly different variations of this pivot rule exist in different implementations of the algorithm, but often it involves picking the edge along which the target function descends most steeply, thus ensuring each step takes us nearer the optimal value. When a corner is found where no further descent is possible, we know we have arrived at the optimal point.

Practical experience shows that the simplex method is generally a very slick problem-solver indeed, typically reaching an optimum solution after a number of pivots comparable to the number of dimensions in the problem. That means a likely maximum of a few hundred steps to solve a 50-dimensional problem, rather than billions with a suck-it-and-see approach. Such a running time is said to be "polynomial" or simply "P", the benchmark for practical algorithms that have to run on finite processors in the real world.

Dantzig's algorithm saw its first commercial application in 1952, when Abraham Charnes and William Cooper at what is now Carnegie Mellon University in Pittsburgh, Pennsylvania, teamed up with Robert Mellon at the Gulf Oil Company to discover how best to blend available stocks of four different petroleum products into an aviation fuel with an optimal octane level.

Since then the simplex algorithm has steadily conquered the world, embedded both in commercial optimisation packages and bespoke software products. Wherever anyone is trying to solve a large-scale optimisation problem, the chances are that some computer chip is humming away to its tune. "Probably tens or hundreds of thousands of calls of the simplex method are made every minute," says Jacek Gondzio, an optimisation specialist at the University of Edinburgh, UK.

Yet even as its popularity grew in the 1950s and 1960s, the algorithm's underpinnings were beginning to show signs of strain. To start with, its running time is polynomial only on average. In 1972, US mathematicians Victor Klee and George Minty reinforced this point by running the algorithm around some ingeniously deformed multidimensional "hypercubes". Just as a square has four corners, and a cube eight, a hypercube in n dimensions has 2^n corners. The wonky way Klee and Minty put their hypercubes together meant that the simplex algorithm had to run through all of these corners before landing on the optimal one. In just 41 dimensions, that leaves the algorithm with over a trillion edges to traverse.

A similar story holds for every variation of the algorithm's pivot rule tried since Dantzig's original design: however well it does in general, it always seems possible to concoct some awkward optimisation problems in which it performs poorly. The good news is that these pathological cases tend not to show up in practical applications -; though exactly why this should be so remains unclear. "This behaviour eludes any rigorous mathematical explanation, but it certainly pleases practitioners," says Gondzio.

Flashy pretenders

The fault was still enough to spur on researchers to find an alternative to the simplex method. The principal pretender came along in the 1970s and 1980s with the discovery of "interior point methods", flashy algorithms which rather than feeling their way around a polytope's surface drill a path through its core. They came with a genuine mathematical seal of approval -; a guarantee always to run in polynomial time -; and typically took fewer steps to reach the optimum point than the simplex method, rarely needing over 100 moves regardless of how many dimensions the problem had.

The discovery generated a lot of excitement, and for a while it seemed that the demise of Dantzig's algorithm was on the cards. Yet it survived and even prospered. The trouble with interior point methods is that each step entails far more computation than a simplex pivot: instead of comparing a target function along a small number of edges, you must analyse all the possible directions within the polytope's interior, a gigantic undertaking. For some huge industrial problems, this trade-off is worth it, but for by no means all. Gondzio estimates that between 80 and 90 per cent of today's linear optimisation problems are still solved by some variant of the simplex algorithm. The same goes for a good few of the even more complex non-linear problems (see "Straight down the line", right). "As a devoted interior-point researcher I have a huge respect for the simplex method," says Gondzio. "I'm doing my best trying to compete."

We would still dearly love to find something better: some new variant of the simplex algorithm that preserves all its advantages, but also invariably runs in polynomial time. For US mathematician and Fields medallist Steve Smale, writing in 1998, discovering such a "strongly polynomial" algorithm was one of 18 outstanding mathematical questions to be dealt with in the 21st century.

Yet finding such an algorithm may not now even be possible.

That is because the existence of such an improved, infallible algorithm depends on a more fundamental geometrical assumption -; that a short enough path around the surface of a polytope between two corners actually exists. Yes, you've got it: the Hirsch conjecture.

The fates of the conjecture and the algorithm have always been intertwined. Hirsch was himself a pioneer in operational research and an early collaborator of Dantzig's, and it was in a letter to Dantzig in 1957 musing about the efficiency of the algorithm that Hirsch first formulated his conjecture.

Until recently, little had happened to cast doubt on it. Klee proved it true for all 3D polyhedra in 1966, but had a hunch the same did not hold for higher-dimensional polytopes. In his later years, he made a habit of suggesting it as a problem to every freshly scrubbed researcher he ran across. In 2001 one of them, a young Spaniard called Francisco Santos, now at the University of Cantabria in Santander, took on the challenge.

As is the way of such puzzles, it took time. After almost a decade working on the problem, Santos was ready to announce his findings at a conference in Seattle in 2010. Last month, the resulting paper was published in the *Annals of Mathematics* (vol 176, p 383). In it, Santos describes a 43-dimensional polytope with 86 faces. According to Hirsch's conjecture, the longest path across this shape would have $(86e_{Snnb} - e_{Snnb}43)$ steps, that is, 43 steps. But Santos was able to establish conclusively that it contains a pair of corners at least 44 steps apart.

If only for a single special case, Hirsch's conjecture had been proved false. "It settled a problem that we did not know how to approach for many decades," says Gil Kalai of the Hebrew University of Jerusalem. "The entire proof is deep, complicated and very elegant. It is a great result."

A great result, true, but decidedly bad news for the simplex algorithm. Since Santos's first disproof, further Hirsch-defying polytopes have been found in dimensions as low as 20. The only known limit on the shortest distance between two points on a polytope's surface is now contained in a mathematical expression derived by Kalai and Daniel Kleitman of the Massachusetts Institute of Technology in 1992. This bound is much larger than the one the Hirsch conjecture would have provided, had it proved to be true. It is far too big, in fact, to guarantee a reasonable running time

for the simplex method, whatever fancy new pivot rule we might dream up. If this is the best we can do, it may be that Smale's goal of an idealised algorithm will remain forever out of reach -; with potentially serious consequences for the future of optimisation.

All is not lost, however. A highly efficient variant of the simplex algorithm may still be possible if the so-called polynomial Hirsch conjecture is true. This would considerably tighten Kalai and Kleitman's bound, guaranteeing that no polytopes have paths disproportionately long compared with their dimension and number of faces. A topic of interest before the plain-vanilla Hirsch conjecture melted away, the polynomial version has been attracting intense attention since Santos's announcement, both as a deep geometrical conundrum and a promising place to sniff around for an optimally efficient optimisation procedure.

As yet, there is no conclusive sign that the polynomial conjecture can be proved either. "I am not confident at all," says Kalai. Not that this puts him off. "What is exciting about this problem is that we do not know the answer."

A lot could be riding on that answer. As the algorithm continues to hum away in those basements it is still, for the most part, telling us what we want to know in the time we want to know it. But its own fate is now more than ever in the hands of the mathematicians. n

Richard Elwes is a visiting fellow at the University of Leeds, UK, and the author of *Maths 1001: Absolutely everything that matters in mathematics* (Quercus, 2010)

The simplex algorithm directs wares to their destinations the world over

"Probably tens or hundreds of thousands of calls are made of the simplex algorithm every minute"

Edges, corners and faces

Since Plato laid down his stylus, a lot of work has gone into understanding the properties of 3D shapes, or polyhedra. Perhaps the most celebrated result came from the 18th-century mathematician Leonhard Euler. He noted that every polyhedron has a number of edges that is two fewer than the total of its faces and corners. The cube, for example, has six faces and eight corners, a total of 14, while its edges number 12. The truncated icosahedron, meanwhile, is the familiar pattern of a standard soccer ball. It has 32 faces (12 pentagonal and 20 hexagonal), 60 corners -; and 90 edges.

The French mathematician Adrien-Marie Legendre proved this rule in 1794 for every 3D shape that contains no holes and does not cut through itself in any strange way. As geometry started to grow more sophisticated and extend into higher dimensions in the 19th century, it became clear that Euler's relationship didn't stop there: a simple extension to the rule applies to shapes, or polytopes, in any number of dimensions. For a 4D "hypercube", for example, a variant of the formula guarantees that the total number of corners (16) and faces (24) will be equal to number of edges (32) added to the number of 3D "facets" the shape possesses (8).

The rule derived by Warren Hirsch in 1957 about the maximum distance between two corners of a polyhedron was thought to be similarly cast-iron. Whether it truly is turns out to have surprising relevance to the smooth workings of the modern world (see main story).

2000 YEARS OF ALGORITHMS

George Dantzig's simplex algorithm has a claim to be the world's most significant (see main story). But algorithms go back far further.

c. 300 BC The Euclidean algorithm

From Euclid's mathematical primer *Elements*, this is the granddaddy of all algorithms, showing how, given two numbers, you can find the largest number that divides into both. It has still not been bettered.

820 The quadratic algorithm

The word algorithm is derived from the name of the Persian mathematician Al-Khwarizmi. Experienced practitioners today perform his algorithm for solving quadratic equations (those containing an x^2 term) in their heads. For everyone else, modern algebra provides the formula familiar from school.

1936 The universal Turing machine

The British mathematician Alan Turing equated algorithms with mechanical processes -; and found one to mimic all the others, the theoretical template for the programmable computer.

1946 The Monte Carlo method

When your problem is just too hard to solve directly, enter the casino of chance. John von Neumann, Stanislaw Ulam and Nicholas Metropolis's Monte Carlo algorithm taught us how to play and win.

1957 The Fortran compiler

Programming was a fiddly, laborious job until an IBM team led by John Backus invented the first high-level programming language, Fortran. At the centre is the compiler: the algorithm which converts the programmer's instructions into machine code.

1962 Quicksort

Extracting a word from the right place in a dictionary is an easy task; putting all the words in the right order in the first place is not. The British mathematician Tony Hoare provided the recipe, now an essential tool in managing databases of all kinds.

1965 The fast Fourier transform

Much digital technology depends on breaking down irregular signals into their pure sine-wave components -; making James Cooley and John Tukey's algorithm one of the world's most widely used.

1994 Shor's algorithm

Bell Labs's Peter Shor found a new, fast algorithm for splitting a whole number into its constituent primes -; but it could only be performed by a quantum computer. If ever implemented on a large scale, it would nullify almost all modern internet security.

1998 PageRank

The internet's vast repository of information would be of little use without a way to search it. Stanford University's Sergey Brin and Larry Page found a way to assign a rank to every web page -; and the founders of Google have been living off it ever since.

STRAIGHT DOWN THE LINE

When a young and nervous George Dantzig spoke about his new simplex algorithm at a conference of eminent economists and statisticians in Wisconsin in 1948, a rather large hand was raised in objection at the back of the room. It was that of the renowned mathematician Harold Hotelling. "But we all know the world is non-linear," he said.

It was a devastating put-down. The simplex algorithm's success in solving optimisation problems (see main story) depends on assuming that variables vary in response to other variables along nice straight lines. A cutlery company

increasing its expenditure on metal, for example, will produce proportionately more finished knives, forks and profit the next month.

In fact, as Hotelling pointed out, the real world is jam-packed with non-linearity. As the cutlery company expands, economies of scale may mean the marginal cost of each knife or fork drops, making for a non-linear profit boost. In geometrical terms, such problems are represented by multidimensional shapes just as linear problems are, but ones bounded by curved faces that the simplex algorithm should have difficulty crawling round.

Surprisingly, though, linear approximations to non-linear processes turn out to be good enough for most practical purposes. "I would guess that 90 or 95 per cent of all optimisation problems solved in the world are linear programs," says Jacek Gondzio of the University of Edinburgh, UK. For those few remaining problems that do not submit to linear wiles, there is a related field of non-linear programming -; and here too, specially adapted versions of the simplex algorithm have come to play an important part.

"Cases where the algorithm fails have tended not to show up in practice -; a pleasing behaviour that eludes explanation"

LOAD-DATE: August 17, 2012

LANGUAGE: ENGLISH

PUBLICATION-TYPE: Magazine

Copyright 2012 Reed Business Information, UK, a division of Reed Elsevier Inc.
All Rights Reserved