



© 1997–2009, Millennium Mathematics Project, University of Cambridge.

Permission is granted to print and copy this page on paper for non-commercial use. For other uses, including electronic redistribution, please contact us.

January 2001

Features

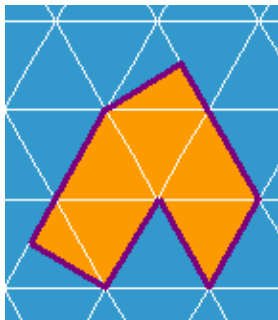


Prize specimens

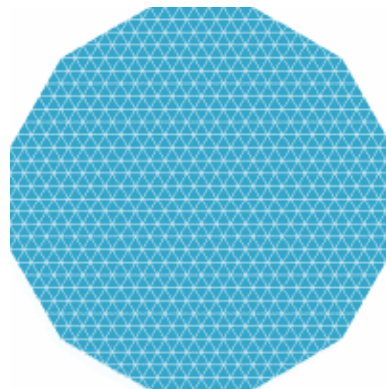
by Mark Wainwright



Alex Selby and Oliver Riordan, two mathematicians, with the help of a couple of computers, have shared a £1m prize by solving the "Eternity" puzzle. The puzzle was like an enormously difficult jigsaw. There were 209 pieces, all different, but all made from equilateral triangles and half-triangles, as in the example on the left. (All the pieces had the same total area as 6 triangles.) The puzzle was to fit them together into the almost-regular 12-sided figure aligned to a triangular grid, shown on the right.



An example piece



The board

The prize was announced when the puzzle hit the market in June 1999. It would go to the first correct solution submitted. However, no solutions would be opened until September 2000. Until then, no-one would know for sure whether a winning solution had already been sent in.

Prize specimens

In July 1999, Alex was given a copy for his birthday by his brother. "I didn't really think about it until November," he says. At that point, two friends came round, sneakily typed the pieces into his computer while he was out, and got him interested in it. Oliver was then in the US, but when he returned in January, Alex recruited him to help with the puzzle.

It seemed that, putting pieces in by hand, one was likely to get stuck after about 150 pieces. They found an internet discussion group on Eternity, where some people had already found ways of placing 206 or 207 of the 209 pieces. "That was interesting, because it suggested the puzzle might be both doable and not yet done," says Alex. Of course, there was no guarantee that someone else hadn't already solved it without posting to the group. By 1 February, someone on the group announced a 208-piece partial solution. "Not only that – it had a piece-shaped hole, though not the right piece, of course. If he could repeat that 1000 times, he would probably get a solution."

By now it seemed that other solvers were getting close to a solution. In fact, it turned out much later that one puzzler, Guenter Stertenbrink, had had a 208-piece partial solution by September 1999! Alex and Oliver submitted their tiling on May 15th, but had to wait until September before they knew that it was the first. Guenter is the only other puzzler known to have found a solution.

The puzzle's inventor, Christopher Monckton, had created the puzzle by starting with a much larger set of pieces, and fitting them into the grid until it was full. "Of course, this made it easy for him to find a tiling, since he had so many extra pieces to choose from," says Alex. "Effectively he's planting a solution. When he had filled up the shape, he threw away the pieces he hadn't used, and the ones that were left formed the puzzle."

Monckton had thought the money safe for a long time: "He thought the sheer size of Eternity would make it intractable." Indeed, exhaustively searching all possibilities would take so long that the fastest computer would have taken many millions of years to find a solution this way. Yet Alex and Oliver took less than seven months to solve it, using mathematical analysis and computer searching. So how did they do it?

Unexpected bonus

They calculated that the large number of pieces would, supprisingly, make the puzzle easier to solve rather than harder. Monckton had run computer searches on much smaller versions of the puzzle, which convinced him that it would get harder as it got bigger. For the cases he examined this was true, but Alex and Oliver realised that as the puzzle grew, it would get harder up to some critical size, after which it would start to get easier again.

Why does the critical size arise? Let's see first why one should expect a large enough puzzle to have solutions at all. Suppose there are N pieces in all. We can imagine dividing the board into N rough areas and trying to put one piece in each, but not worrying if they overlap a bit. We know the number of ways of choosing which piece to put in which area: it's $(1 \times 2 \times 3 \times \dots \times N)$, usually written $N!$

For each piece, there is a number of triangles round its edge where it might overlap with another piece. Indeed, the probability of it not overlapping might be some very small number, say p . And we have a complete solution only if none of the tiles overlap, whose probability is a tiny p^N . On average, therefore, the number of solutions will be $N!(p^N)$.

When N is fairly small, $N!(p^N)$ will be small – far less than 1 – and there are probably no solutions, other than the one planted by the inventor. But $N!$ is a function that grows faster than k^N for any fixed k (in fact, nearly as fast as N^N). This means that however small p is, as N grows large enough $N!(p^N)$ will grow above 1 and more

Prize specimens

and more solutions will appear. The critical size happens when the expected (average) number of solutions is 1.

This explanation by reference to probabilities may seem a little unsatisfactory at first. Surely there either is a solution or there isn't? But the same comment applies in any situation we think of as probabilistic (either the coin will come down heads or it won't). Probability is just a measure of our lack of knowledge. There might be some systematic reason why some tilings will be impossible, but if there isn't, the probabilistic estimate of how many solutions exist will (probably!) be a reasonable one.

Bigger means easier

They calculated the critical size was about 70 pieces. An Eternity puzzle this size would be essentially impossible to solve, so it is not surprising that Monckton's searches did not get this far. The actual puzzle, with 209 pieces, was far beyond this critical size. Its level of difficulty would be critically determined by how well-behaved the shapes of the pieces turned out to be. "It seemed it would be just possible to solve, with a lot of work," says Oliver. "We thought the pieces must have been carefully designed to make it very, very difficult, without being impossible. But it turns out the inventor was just lucky – or perhaps unlucky."

The pair estimate that there are probably about 10^{95} solutions, more than the number of subatomic particles in the universe. "That might sound like a lot of solutions, but there are far, far more non-solutions," says Oliver. In other words, the puzzle is much too large to solve by an exhaustive search. The fastest computer would take millions of years to find even one solution by this method. This would still be true even if there were many more solutions, as there may well be. "Our estimate was on the low side compared with some others, which were more like 10^{120} ."

Why, then, does the problem get easier as the size grows? Alex explains: "For a region of the critical size, c say, we expect one solution. That's an average: some regions will have more solutions and be easier to tile, others will have none. If the number of pieces N is bigger than c , we are free to put in the first $(N-c)$ pieces in many different ways. Some of the resulting tiling problems will be much easier than average. If we can identify the easy ones, we only have to look at those."

Why did Monckton miss this idea? "He wasn't looking at it in a mathematical way at all. In fact, he said he didn't expect a mathematician to solve it."

Unhelpful hints

At the same time as Eternity, Monckton released various smaller puzzles. Solving any of these allowed one to send off for one of various "hints", showing where on the Eternity grid particular pieces were in Monckton's solution. Alex points out that these would not have been any help at all. "Suppose you're going fishing, and you know that in the Atlantic there is one fish per cubic mile, and in the Pacific there are two fish per cubic mile. Then you'll fish in the Pacific rather than the Atlantic. If someone throws one extra fish into the Atlantic, it won't make any difference. With total fish stocks of 10^{95} or more, one fish more or less is neither here nor there."

In fact, if solvers had been required to use the hints, the puzzle would have become much harder. If you are building up a solution by carefully slotting in pieces one by one, small holes in the middle of the region to be tiled will enormously increase the number of constraints. "Having to fit in with pieces in the middle of the board would be crippling."

Good and bad pieces

Knowing there are lots of solutions doesn't give us much idea where to look for them, but we did get one clue: the approach must be to try to identify what kinds of positions are easier to tile when only a few pieces are left. Then we will try to find a way to generate such positions, and whenever we get one, we will do an exhaustive search from it for a complete solution.



You can compare trying to fit the pieces into the grid with loading packages into the boot of a car. Some of the packages are awkwardly shaped, and you usually put those ones in first, keeping the nicer shapes till later when you have less room for manoeuvre. In the same way, some Eternity pieces will be better than others when you've placed most of the pieces and want to fill up the space that's left.

Unlike the car-packing case, though, it's not easy to tell which pieces are good just by looking at them. One might expect that the pieces with the most convex, least knobby shape will be best, and indeed it turns out that this is mainly true. But to get a handle on such a vast problem, Alex and Oliver needed much more accurate information about just how valuable different pieces were. Their plan was to collect this information by running exhaustive searches to tile small regions of the board with subsets of the Eternity pieces. Using the knowledge gained from this, they would try to build up a tiling piece by piece, always trying to leave themselves with as good a collection of pieces in hand as possible.

Going back to the car-boot analogy: as you put things in, you're not only trying to keep the nice packages till last; you're also trying to ensure that, at each stage, the space that's left is as sensible a shape as you can arrange. Similarly, some shapes of Eternity region will be easier to tile than others, but again, it's not always obvious which ones. At the same time as identifying good and bad pieces, Alex and Oliver intended to try to identify good and bad regions. Of course, they could not examine all possible region shapes, but they collected data about what features made a region good or bad. For example, they looked at how many corners a region's boundary had, and tried to work out how much harder each corner made it to tile.

Exhausting work

Since they were going to have to run many exhaustive searches of small regions, the first challenge was to write a fast and efficient solver program that would find all tilings of a given region of a moderate size with a given set of pieces.

Prize specimens

One useful idea in writing the solver was the idea of always branching on the most constrained site first. This can be explained by yet another reference to packing a car boot. If, at a particular point, you have an awkward-shaped corner to fill up, you will probably look first for a package that will make efficient use of that corner. The solver used a similar principle: at each stage, it scanned the whole region to find which site could be tiled in the fewest ways. Then it would try all the possibilities for that site.

Another important feature of a good solver is the ability to discard as many unfillable positions as possible at an early stage. Of course the search will find that the position is impossible, but that takes time. If a large number of positions can be eliminated early on, it will make the solver much faster. Their innovation here was to look at the shape of the boundary. Oliver expands: "We looked at all possible shapes of path in the triangular grid of length 11, and for each one checked if it could occur as part of a tiling – roughly, if it could be tiled on both sides. We stored that information in a table, and when we examined a position, the first thing we did was to check every subpath of its boundary of length 11 against our table."

Suppose you start tiling by fitting pieces together in the grid, until all but, say, 24 of the pieces have been used. The position that remains consists of two things: the region that has yet to be tiled, whose size is that of 24 pieces – note that all the pieces have the same area – and the particular 24 pieces still available. We'll refer to these together as an endgame position. The resulting puzzle is small enough to search exhaustively, but unfortunately is extremely unlikely to have any solutions. (If it does, you've solved the puzzle and you can go home!) A different, random selection of 24 pieces is no more likely to be able to tile the region.

Oliver and Alex generated a large number of such size-24 endgame positions, using the computer to put tiles down using their solver's boundary checker and a small amount of look-ahead to make sure it didn't get stuck too early. They kept the regions from these positions, but ignored the pieces. Instead of the actual 24 pieces left in each case, they asked the solver to tile the region using a random subset of 90 of the original 209 Eternity pieces. In each case, they then did an exhaustive search to find all tilings of the region using any 24 of the 90 pieces. "By giving yourself extra pieces, you make the problem easier, so that you can begin to find some solutions", says Oliver. "The expected number of solutions is multiplied by 90 choose 24, which is a huge factor, for a much smaller increase in the amount of work."

Getting the parameters

From the results of these searches, they planned to find just how useful each piece was, and how much a tiling problem was affected by features of the boundary such as corners. To do this they had to make a priori assumptions about what kinds of effects they were looking for. For example, they assumed that the probability of being able to tile a region with a particular set of pieces varied independently with each piece in the set. Each piece would have an associated probability, and to get the overall probability you simply multiply together all the individual probabilities. Similarly, the probability would depend exponentially on the number of corners in the region's boundary (and on certain other properties of the boundary).

The problem now was to estimate values for the probabilities (for the pieces), and the exponential factor for boundary corners (and similarly for other boundary features), using the data from the exhaustive searches of the size-24 regions. They proceeded as follows:

1. Obviously, if we knew the probabilities for the pieces and boundary features, then given a particular set of pieces, we could calculate the probability of tiling a given region. What we need at the moment, though, is the opposite: for a fairly large data set we know how often we succeeded in tiling a number of regions; from this we want the piece probabilities. A well-known theorem in probability theory, Bayes's Theorem, gives a formula for the probability-of-A-given-B in terms of the probability-of-B-given-A. If we assume that, in the absence of any data, all models (i.e. guesses

Prize specimens

about the piece probabilities) are, in some sense, "equally likely", then Bayes's Theorem allows us to find how any model's probability is affected by the data.

2. The formula given by Bayes's Theorem might not be easy to apply in practice. The pair had to adjust their probabilistic model to make the calculations more tractable. Of course, the trick is to choose adjustments which make only a negligibly small difference to the result, but give a big payoff in computation time.

This part of the procedure – finding a likelihood that could actually be calculated – was the most challenging and interesting, because in principle, calculating each likelihood depended on adding a huge and impracticable number of terms. They circumvented this problem by arranging for this sum to be calculable by a far more manageable short-cut.

3. Finally, when we can calculate how likely any set of piece probabilities is, we have to maximise this likelihood over all possible sets of probabilities! This may seem even more implausible: when one could invent an unbounded number of trial sets of probabilities, how do we find the likeliest one? In general this would indeed be a sticky (though not necessarily impossible) problem. However, for a certain class of functions, called convex functions, this maximisation problem can be solved efficiently. The likelihood function of the piece probabilities that Alex and Oliver obtained was thoroughly nice and convex, so they were able to use standard methods to find the set of piece probabilities that was likeliest to be correct.

If they had done their calculation with too small a data set, the results would have been fairly random. How would they know when they had looked at enough positions that the resulting parameters could be trusted? They took a simple approach to this: they divided their searches into two sets, and did separate calculations for each. Only when the two calculations gave similar answers for most or all of the 209 pieces were they satisfied that the sample was big enough.

Putting the data to work

They now had a set of probabilities expressing how useful each piece was in tiling a region the size of 24 pieces. Unfortunately, using exactly the same values on larger regions was not likely to work, so they needed similar values for tiling larger regions. (They could not use the same method as for size 24, since the exhaustive searches would take too long.) They modified the size-24 values for tiling progressively larger and larger regions. For example, for a size-25 region, they tried to arrange that the parameters gave the same predicted chance of a tiling as placing a piece and calculating the probability using the parameters they already had for size-24 regions. "Our procedure for this step wasn't very good," admits Alex. "That was one of the things we were trying to improve when we were so rudely interrupted by the computer announcing that it had found a solution."

Having decided on the model parameters for sizes up to 209, they had a score for every position. They used these scores to do a version of a standard "breadth-first search" which remembered good positions and investigated them further. "To start with, you find the most constrained site, and find all the ways to tile it. You keep the best, say, 1000 positions at time 1, and throw the rest away. Then you look at all the children of those positions: that's to say, in each one, you find the most constrained site and look at all the ways to tile that site. Of the resulting positions at time 2, you again keep the best 1000 and throw the rest away." As well as calculating the score for the position, they used the length-11 boundary information that they had calculated for the solver to avoid keeping provably untileable positions, and used a little lookahead. "This is a really nice search method," says Alex. "It's a standard method in computer science, called beam search, but all the same I was surprised how well it worked for this problem." The performance is improved by increasing the number of positions kept (the "width") from 1000 to a much larger number. "We were using a width of about 10,000 on the run that found the solution, although we have since tried widths up to 1,000,000 and they

seem to be better."

A lucky break

By this point they were confident of finding a solution; it was just a matter of when. When the computer announced that it had found one, though, Alex at first suspected a mistake. "From our estimate of the number of solutions, we didn't think the program had much chance as it stood. On average, it would have needed to run for years before it found a solution. Of course, there might be more solutions than we thought, but even so, extensive runs since for longer periods haven't found any more solutions. It looks as if we were lucky." Oliver was away for the weekend when the solution came up: "Alex came round and told me it had found a solution while I was away. I didn't believe him – I thought he was having me on."

Without such a piece of luck, how much longer would it have taken them to find a solution? Alex says it is difficult to estimate: "The algorithm was constantly improving. For example, we were working on improving our scoring method for regions larger than 24 pieces. And the main speed-up would have come from using more computers: we were near the point of enlisting more computers when the solution appeared. If I had to make a guess, I'd say it would have been about two months." Would this have been in time to claim the prize? "Nope. Guenter Stertenbrink sent in his solution just six weeks after ours."

Finally, how do the pair intend to use their new-found wealth? Oliver will use his share to buy a house, "but I don't know where or when". At present he is a research fellow in Cambridge while his girlfriend lives in Germany, "which is a long commute". Alex has no definite plans. "But I'm sure the money will come in handy."

Monckton is already planning a successor puzzle, Eternity 2, with another headline-grabbing prize for the first solution, but Alex and Oliver have no plans to compete this time. "A second time round, it wouldn't be interesting enough to spend another six months on," Alex says. That sounds like good news for other would-be solvers.

About the author

Mark Wainwright is an assistant editor of Plus.



Plus is part of the family of activities in the Millennium Mathematics Project, which also includes the NRICH and MOTIVATE sites.