Origami:
Complexity in Creases (Again)

by Robert J. Lang

In this origami composition, "Hummingbird and Trumpet Vine," Lang folded the bird, each blossom, and each leaflet from a single, uncut square of paper.

Over a decade ago, I wrote an article for *Engineering & Science* magazine about origami, the Japanese art of paper folding, and its appeal to scientists and mathematicians. Toward the end of the article, in a fit of wild speculation, I asked:

"Could a computer someday design a model deemed superior to that designed by man?"

Little did this would-be futurist know what the following decade would bring. The past 10 years have seen an astonishing cross-fertilization of ideas between origami, math, and computer science. We have origami solutions to ancient problems, such as how to double a cube or trisect an angle, and origami solutions to new ones, including how to fold airbags to fit into steering columns, or telescope mirrors to fit into spacecraft. And certain origami crease patterns have been found to encode some of the hardest problems known to computer science. But most remarkably, yes, there is indeed a computer program that can, in 30 seconds or so, design origami models more complex than anything conceived over the previous thousand years. When I wrote that *E&S* article in 1989, the field of origami mathematics was almost nonexistent, but over the past 10 years, researchers from many fields have developed the principles that led to that program and to the application of origami to real-world engineering problems.

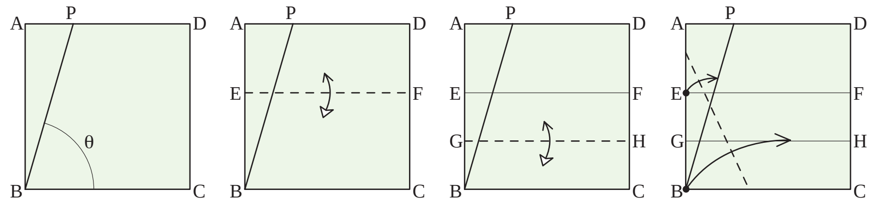
Paper folding did not start out as an engineering discipline; it started as a craft. Origami is the art of folding uncut sheets of paper, usually squares, into decorative shapes. The name is Japanese and the Japanese form of the art is the most well known, although other countries (notably Spain) have their own independent tradition of paper folding as entertainment. There are two kinds of origami in Japan: abstract, ceremonial shapes, such as the good-luck pattern known as *noshi*, and representational origami—origami that looks like something. Historically, the usual subjects for representational origami were birds, fish, flowers, and the like. It was a woman's art: simple figures passed down from mother to daughter, valuable primarily for teaching or entertaining the young. The ceremonial figures were imbued with great symbolism, but for the most part, representational origami was viewed with the same respect that we give cootie catchers and paper airplanes—which is to say, not very much.

That began to change in the early part of the twentieth century, when a Japanese factory worker named Akira Yoshizawa began creating artistic new designs. He also promoted origami in books and exhibitions, initially in Japan, and eventually around the world. Origami as an art form caught on in the West in the 1950s and 1960s. Some people seem to have a peculiar susceptibility to the charms of origami—the simplicity of folding a pedestrian sheet of paper into unexpected and beautiful shapes. Through the 1960s and 1970s, the number of people infected by this particular bug grew at an exponential rate.

Somewhere along the way, the ranks of the infected were joined by mathematicians and scientists, who began asking questions like: What is possible in origami? How can I fold any given object? Can one quantify the difficulty of an origami design? Of course, scientists don't just ask questions—they set out to answer them.

One of the first areas to be explored was the problem of geometric constructions. You probably recall from high-school geometry that you can draw an equilateral triangle or bisect a given angle using nothing but a compass and a straightedge. But some constructions, the most famous being the trisection of an angle, are impossible with just those tools. It comes as a surprise to many people that it is possible to trisect any angle using origami—it came as a surprise to the editors of the *American Mathematical Monthly*, which printed an article in 1996 "proving" the impossibility of origami angle trisection, and then printed a correction six months later noting that an origami solution for angle trisection was over 20 years old.

Tsune Abe's trisection method works for any angle less than 90°. There are other methods that work for larger angles.

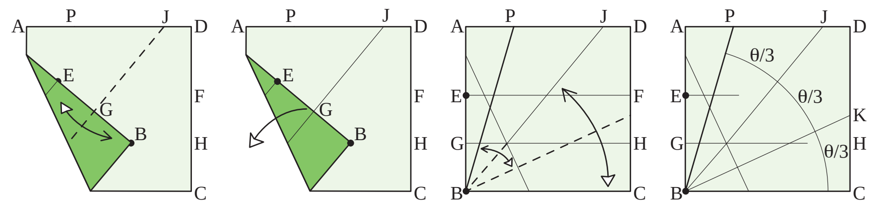


1. Draw the desired angle, PBC, so that point B is in the corner of a square of paper as shown.

2. Make a horizontal fold anywhere across the square, defining line EF.

3. Fold line BC up to line EF and unfold, creating line GH.

4. Fold the bottom left corner up so that point E touches line BP and point B touches line GH.



5. With the corner still up, fold both layers to continue the crease that ends at point G all the way to J, then unfold.

6. Unfold corner B.

7. Fold along the crease that runs to point J, extending it to point B. Fold the bottom edge BC up to line BJ and unfold.

8. The two creases BJ and BK divide the original angle PBC into thirds.

There are always a few adventurous high-schoolers who, when told of the impossibility of angle trisection, seek to find a method on their own. However, it's been mathematically proven that a compass and unmarked straightedge don't allow angle trisection—at least, not without cheating. The way this was proven was to show that all the different operations you could make with compass and straightedge—striking arcs, drawing straight lines through points, and so on—were only enough to solve quadratic equations, while trisecting an arbitrary angle requires the solution of a cubic equation. One of the compass-and-straightedge cheats involves holding your compass against the ruler and manipulating the two as a single object, thus effectively letting you do things with a *marked* straightedge. This simple change adds another new operation to compass-and-straightedge that allows the solution of cubic equations, and thus, angle trisections. In the origami angle trisection, the action in step four—folding two different points to lie on two different lines—fills the role of the marked straightedge. This maneuver, or one like it, is at the heart of several origami solutions to problems that bested Euclid. One of the most elegant is "doubling the cube," that is, constructing two line segments in the ratio of 1: $\sqrt[3]{2}$. An approach devised by Peter Messer is shown on page 12.

It came as a surprise to the editors of the *American Mathematical Monthly*, which printed an article in 1996 "proving" the impossibility of origami angle trisection, and then printed a correction six months later noting that an origami solution for angle trisection was over 20 years old.

Origami geometric constructions are part of a family of pure mathematical problems in which the object is to fold an arbitrary geometric shape or a pure number represented as a distance proportional to the edge of the paper. While the origami construction of a 13-gon has a certain allure, for many folders (myself included), origami's appeal has always been that you folded a specific subject: a bird, fish, or cuckoo clock. My own interest has been more practical: given the subject, how can I use mathematics to figure out how to fold it? This has been dubbed mathematical origami design.

This field owes a great debt to computational geometry, itself only about 30 years old. One of the first formal results was proven in 1994, when Marshall Bern and Barry Hayes, computer scientists at Xerox PARC in Palo Alto, California, showed that the problem of origami crease assignment—given a pattern of creases on a square, how to decide whether each crease should be a mountain fold (making a peak) or valley fold (making a trough)—could be computationally intractable for relatively small problems.

In lay terms, Bern and Hayes proved that "origami is hard"—a point most people don't need to be convinced of. But in fact, they proved its difficulty in a significant way. They showed that crease assignment was one of a broad class of problems known as "*NP*-complete" that contains some of the most challenging problems known to computer science (see sidebar). These problems share two characteristics: if you find a quick way to solve one of them, you can use the same approach to quickly solve all the others; and no one has ever found a quick way to solve any of

NP-Completeness and Origami

NP-complete problems are defined by their computational complexity, which measures how the work involved in solving a problem relates to the size of the problem itself. For example, when you add two n-digit numbers, you start at the right and add each pair of digits (plus any carries), record the result, and go on to the next pair. You do this n times and thus the problem's complexity is said to be of order n, abbreviated as $O(n)$.

For simple addition, complexity increases linearly, but often it grows much faster. For example, you "convolve" two lists of numbers by multiplying every number in one list by every number in the other list and then adding them up in groups. (Convolution is what Adobe Photoshop does when it blurs or sharpens an image.) For two lists of n numbers, there are n^2 multiplications and n^2 additions, so the problem is said to be $O(n^2)$. Now, if you double the problem's size, you quadruple the program's running time.

Sometimes there are faster approaches. While multiply-and-add is $O(n^2)$, the Fast Fourier Transform allows you to do a convolution in $O(n \log n)$, meaning that the number of steps is proportional to the product of n and its natural logarithm. Of course, $n \log n$ still grows, but much more slowly than n^2. A fast algorithm can make the difference between minutes and days of computing.

Addition and convolution are called class *P* problems, where *P* stands for "polynomial time," because the time needed to solve them is bounded by some finite polynomial in n (meaning n raised to a finite power; thus, $n \log n$ is bounded by n^2).

But a host of nasty problems appear to scale as an exponential of their size and quickly become intractable as n increases. Running an exponential-time algorithm might easily take longer than the age of the universe even for fairly small values of n.

One famous example is the traveling salesman problem: given the locations of n cities, what is the shortest route that visits each city? A related form of the problem asks if there is a route shorter than a specified distance. Although people have figured out relatively fast ways of finding pretty good answers—routes that are among the shortest—the only known way to guarantee you've really found the shortest one is to compare all possible routes, or at least a fairly large subset of them. The traveling salesman is in a class of problems, called *NP* for "nondeterministic polynomial time," which may or may not be solvable in polynomial time, but whose solutions, once found, can be *checked* in polynomial time. For example, it's easy to see whether a route is under 100 miles long.

The traveling salesman problem and several others are in a special corner of *NP*, called *NP*-complete, which means that they are hard in a particular way. As in the case of convolution, problems can sometimes be converted, or "reduced," to other problems. *NP*-complete problems have the property that *every* problem in *NP* can be converted into *any NP*-complete problem, which means that if you could knock off one of these incorrigibles in polynomial time, you could use the same approach to solve all *NP* problems and make millions of dollars along the way. The frustrating thing is that although almost everyone believes that there are no polynomial-time algorithms for *NP*-complete problems, no one has been able to prove it.

Which brings us to an origami problem: given a pattern of creases, how can you assign valley and mountain folds to the creases so that the result can be folded flat? It's pretty easy to analyze a single vertex where creases intersect. For example, if four creases come together, they will only fold flat if there are three mountain folds and one valley fold or vice versa, *and* the sums of opposite angles are equal. (To see this, fold a square in half and then in half again to make a new square one-fourth the original size.)

The complexity arises when edges and layers start to collide in a large pattern. You can't pass the paper through itself, and a crease that runs all the way across the paper can make widely separated regions interfere with one another. Such long-range interconnectedness is a hallmark of the traveling salesman problem and its ilk.

Bern and Hayes showed that assigning mountain and valley folds is equivalent to the so-called "not-all-equal three-variable satisfiability" problem, which is known to be *NP*-complete: given a collection of clauses, each containing exactly three true-false logic variables, determine whether you can make each clause have either one or two, but not zero or three, "trues." A simple example is shown in the margin. Bern and Hayes converted the clauses into small crease patterns connected by long, skinny pleats. A noninterfering set of mountains and valleys corresponded to a valid set of trues and falses. So a pleat that went mountain-valley might mean "Pat is the husband, Kim is the wife," whereas valley-mountain would mean "Pat is the wife, Kim is the husband." Thankfully, only one particular class of crease-assignment problems is *NP*-complete, or this article would not have been written.

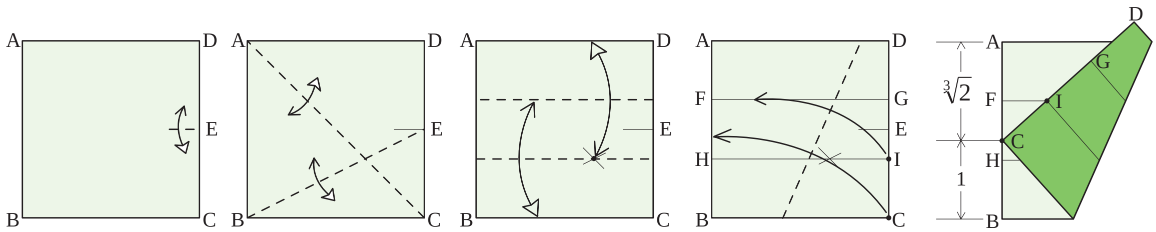
As noted earlier, if you could solve one *NP*-complete problem efficiently, you've solved them all, but proving that a problem is *NP*-complete does not prove that no efficient algorithm for solving it exists. It just means that while I can't find one, neither can all the famous folk.

This brainteaser uses genders instead of "true" and "false." Four couples attend a party: Pat and Kim, Renay and Leslie, Lynn and Lee, and Sydney and Chris. Each couple consists of a husband and wife, though not necessarily in that order. Each of the following trios includes two members of one sex and one of the other:

1. Leslie, Lynn, and Sydney;
2. Pat, Leslie, and Chris;
3. Kim, Renay, and Lynn;
4. Kim, Leslie, and Lee.

If all the husbands get together in one room and all the wives in another, who is in the room with Renay? **For the answer, see page 44.**

Peter Messer's construction of the cube root of 2.



1. Make a small fold halfway up the right side of the paper.

2. Make a crease connecting points A and C and another connecting B and E. Only make them sharp where they cross each other.

3. Fold the top edge down horizontally to touch the crease intersection and unfold. Then fold the bottom edge up to touch this new crease and unfold.

4. Fold corner C to lie on line AB while point I lies on line FG.

5. Point C divides edge AB into two segments whose proportions are 1 and the cube root of 2.

them. Absent that magic bullet, cracking such problems basically boils down to trying out an appreciable fraction of all the possible solutions and seeing which one works. Bern and Hayes showed that some origami crease patterns can be used to encode *NP*-complete logic problems; solving the crease pattern would be equivalent to solving the logic problem.

The difficulty of assigning mountain and valley folds to an existing crease pattern grows quickly

Creating a rigorous definition of a "flap" was fundamental to our solution. A not-so-rigorous definition is "a loose bit of paper that gets turned into an appendage."

with the number of creases; therefore it is possible to construct patterns for which mountain-fold and valley-fold assignments would stump even the most powerful computer. Small problems are amenable to trial and error—just try every possible combination of mountain and valley folds—but this quickly becomes impractical as the size of the pattern grows.

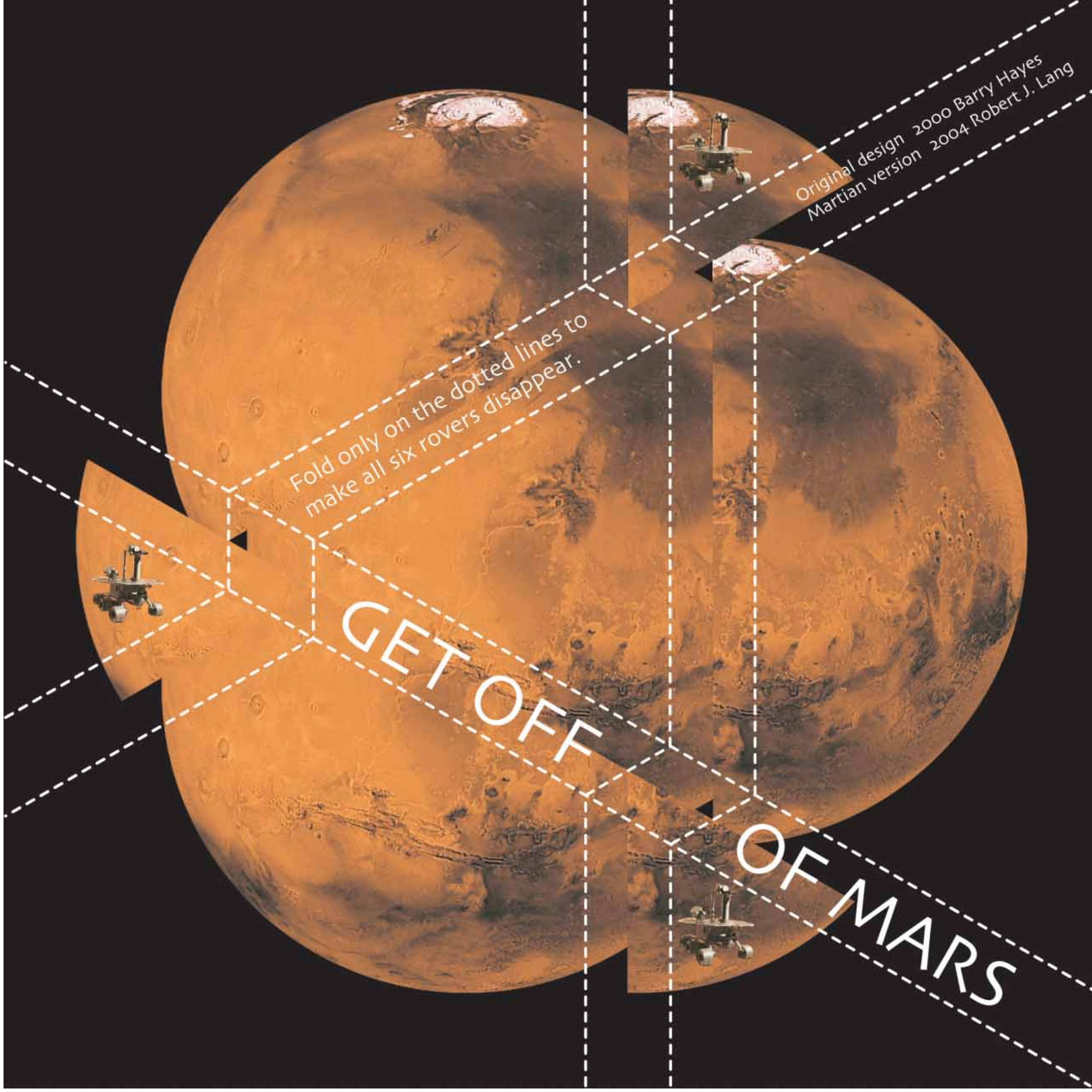
Not all crease problems are intractable; in fact, some of them are at just the right level of difficulty to make good puzzles. On the opposite page is a crease pattern designed by Hayes, who called it "Get Off the Moon!" In honor of JPL's current successes on the red planet, I've created a modified version titled "Get Off of Mars!" Cut out the black square and fold it, making creases only on the dotted lines, to conceal all six rovers—three on each side of the paper.

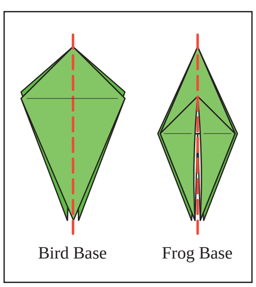
Bern and Hayes's proof would seem to rule out developing a computer algorithm for origami design; after all, how can you hope to design an unknown crease pattern if you can't even assign mountain-valley status to a known crease pattern? Fortunately, their result only applies to patterns that *may* encode *NP*-complete logic problems. If you can lay out the creases to avoid such logical challenges, then the problem might be quite tractable.

During the 1990s, Japanese biochemist Toshiyuki Meguro and I independently developed a set of techniques for expressing the structure of a large class of folded shapes in a way that could be transformed into creases on a sheet. Just as importantly, the mountain-valley status of most of the creases was predetermined by the shape itself, and the remaining creases could easily be assigned using simple, polynomial-time rules.

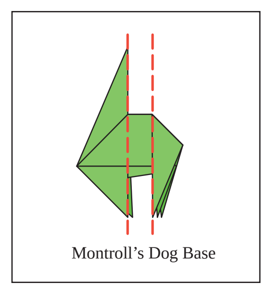
Creating a rigorous definition of a "flap" was fundamental to our solution. A not-so-rigorous definition is "a loose bit of paper that gets turned into an appendage." Flaps become wings, legs, arms, feet, ears, horns—basically, anything that sticks out from the rest of the model. In origami, a shape with a bunch of flaps is called a "base." In general, a base resembles the subject to be folded by having the same number and length of flaps as the subject has appendages. For example, a base for a bird might have four flaps, corresponding to a head, tail, and two wings. A slightly more complicated subject such as a lizard would require a base with six flaps for the head, four legs, and a tail. And an extremely complicated subject such as a flying horned beetle might have six legs, four wings, three horns, two antennae, and an abdomen, requiring a base with 16 flaps.

The number of flaps required depends on the level of anatomical accuracy desired by the paper folder. Historically, much origami design was performed by trial and error—manipulating a piece of paper until it began to resemble something recognizable. For a complex subject, this is rather inefficient, since one is unlikely to stumble upon a 16-pointed base with flaps of the right size in the right places purely by luck. A more directed approach was clearly needed.



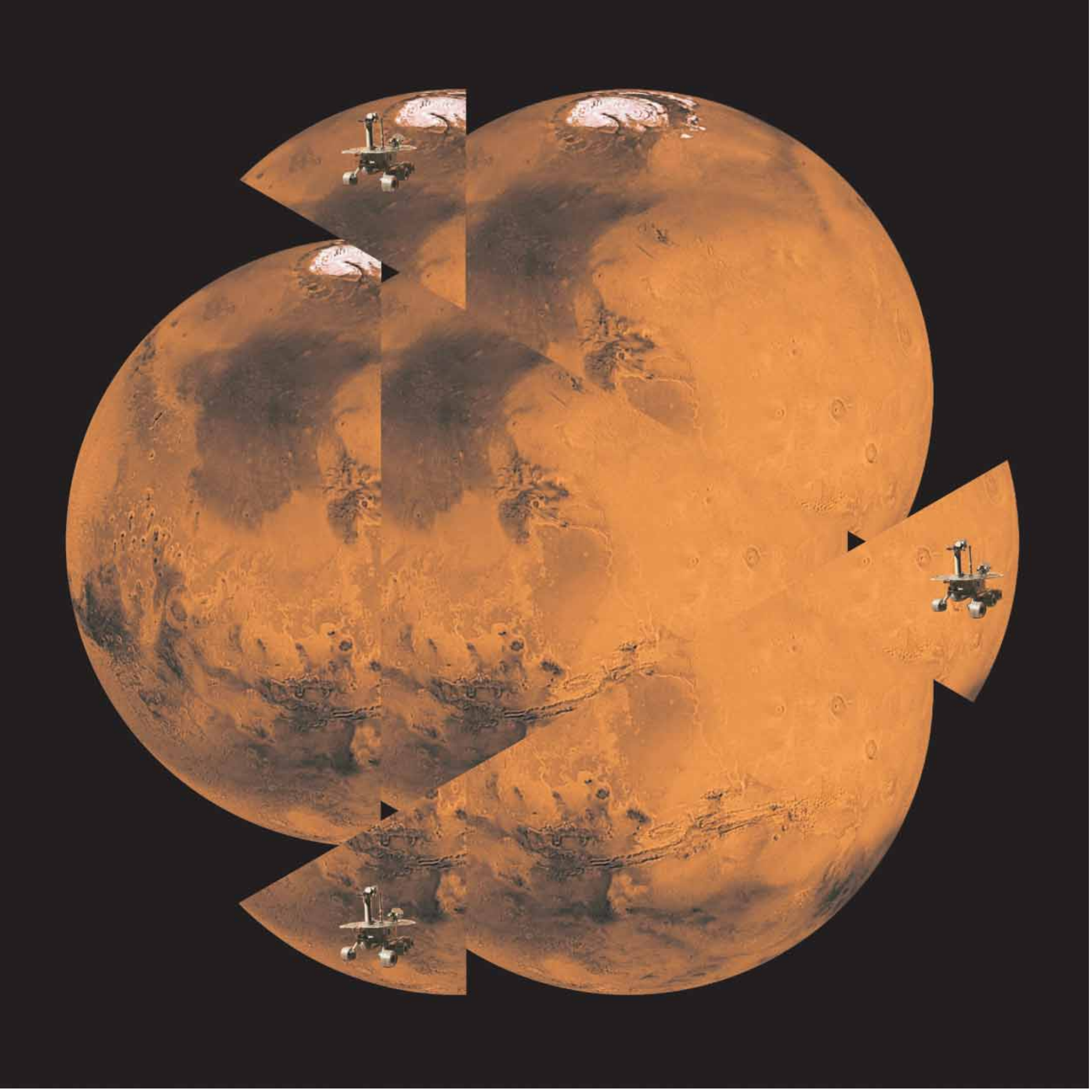


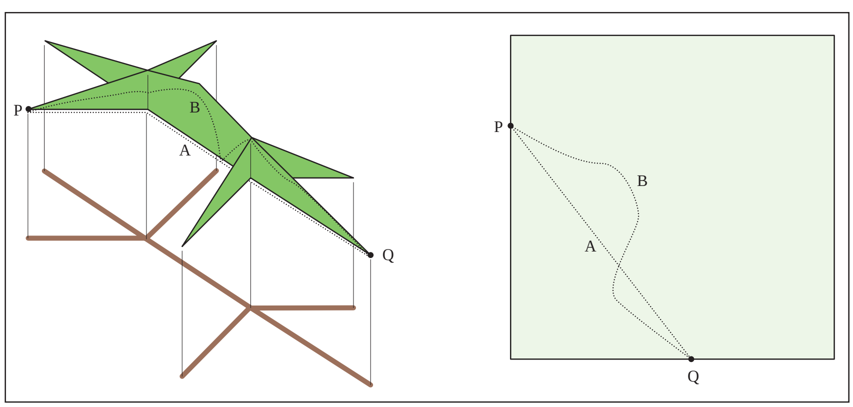
Left: The traditional Bird Base and Frog Base are uniaxial bases. Right: John Montroll's famous Dog Base is not, having two distinct axes.



I focused my attention on a class of bases that can be oriented so that all of the layers run up and down and all of the flaps have their tips and at least one edge in a horizontal plane. If you take either base shown at left and rotate it 90 degrees around the red axis, you'll see what I mean. This class, which I named the "uniaxial" base, takes in all of the traditional origami bases, including the Kite, Fish, Bird, and Frog [see *E&S*, Winter 1989] and many (though not all) modern bases as well.

If you illuminate a uniaxial base from directly above, its shadow will consist solely of lines, as you can see on the next page. It turns out that the most important properties of a uniaxial base—indeed, much of its structure—can be determined solely from the properties of its shadow. In mathematical terms, this shadow forms a "tree graph," which is a fancy term for a "stick figure." The tree graph consists of "edges," or line segments, and "nodes," which are points where edges either come together or terminate. The flaps have a one-to-one correspondence with the graph's edges; similarly,





A folded uniaxial base (right) casts a tree-graph shadow. We can take many paths between leaf vertices P and Q, the shortest of which (path A) is the same length as the shadow. When the paper is unfolded (far right), this path becomes the crease connecting P and Q.

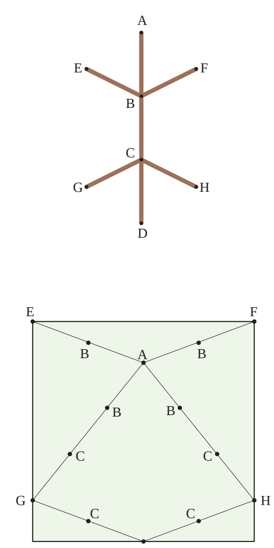
Origami subjects with relatively large, rounded bodies are not so well suited to tree theory. This figure, "Night Hunter," uses a mixture of tree theory and intuition in its design.

the flaps' tips match up with the "leaf nodes," which are the nodes that have exactly one edge connected to them. While graph theory generally doesn't care about the lengths of the graph's line segments, we do; we assign the length to each edge that we desire in the corresponding flap of the base. With this, we are ready to start figuring out the creases.

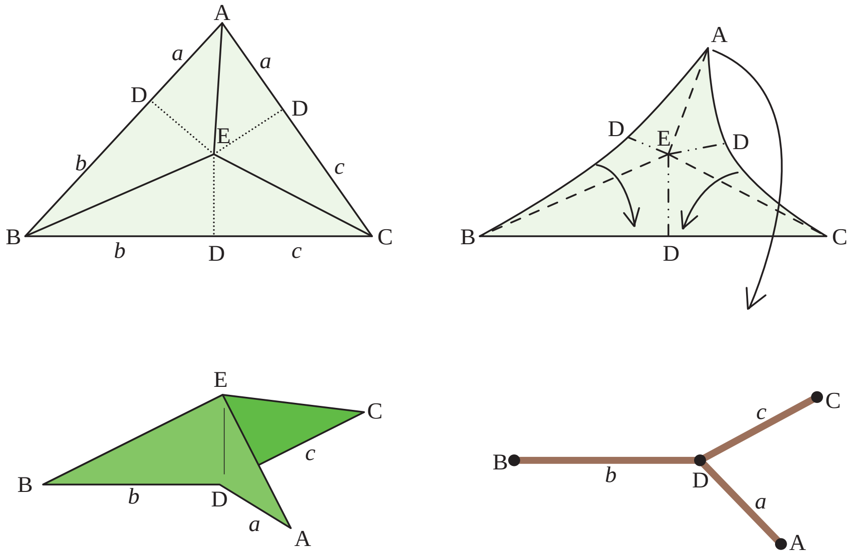
Let's consider the hypothetical base at left, and the relationship between paths on the unfolded paper, the same paths in the folded base, and the shadow. Suppose you drew a line, not necessarily straight, on the paper. What would that line look like in the folded base, and how long would its shadow be?

A point on the paper whose shadow is a leaf node is called a "leaf vertex." Each vertex corresponds to the tip of a flap, so that any path between two leaf vertices will, in the folded base, run from the tip of one flap to the tip of another—say between points P and Q. This path might travel in a horizontal plane in the base, as path A does, or it might go uphill and downhill within the folds of paper, like path B. How does the length of the path compare to the length of its shadow? If the path is purely horizontal, like path A, then the two lengths are equal. Any other path, including path B, is longer than its shadow. Thus the distance between any two leaf vertices on the unfolded crease pattern must be greater than or equal to the distances between the corresponding leaf nodes on the tree graph. I named this mathematical expression the "path condition" for the two vertices, and there is a path condition for every possible pair of leaf nodes.

This seems pretty obvious, but in the early 1990s I was able to show something not so obvious: that the inequalities embodied in the path conditions were not only *necessary* for a valid crease pattern, but they were *sufficient*, as well—a much more useful result. In other words, if you found a set of points on a piece of paper that satisfied all possible path conditions, then those points were the leaf vertices of a pattern that would fold into a base whose shadow was the graph. Furthermore, whenever the length of a path between two vertices exactly equaled the distance between the corresponding nodes, a fold line ran between those vertices, and that fold was almost always a valley. For example, path A is perfectly horizontal and runs along the bottom of the base. Any path that descends to this line, like path B, has to change direction in the folded base or leave the paper; consequently, there must be a crease there. Constructing all such valley folds produces the creases that serve as a framework for the base.



Top: The tree graph for a six-flap base. Bottom: The skeleton of the crease pattern that will fold into the corresponding base. Now all we have to do is construct the secondary folds that will, for example, bring together all the points labeled B.



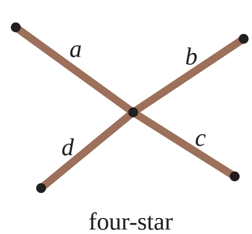
Top left: The crease pattern for the triangle molecule. Top right: The molecule is formed by folding along the angle bisectors and bringing the three points marked D together as you flip the paper over. (Dashed lines are valley folds, dot-dashed lines are mountain folds.) Bottom left: The folded molecule. If you now folded tips B and C forward to meet tip A, all of the original triangle's edges would now lie on a common line. Bottom right: The molecule's tree graph.

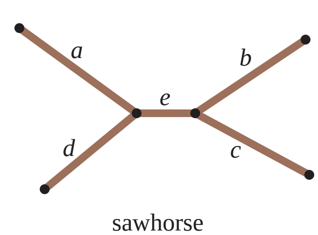
These first folds aren't the entire crease pattern, of course, but they establish its overall structure by dividing the paper into polygons that correspond to various pieces of the tree graph. Now we need to fill in these polygons with creases in such a way that each polygon folds flat with all its edges along a common line. Several crease patterns that do this—dubbed *bun-shi* (molecules) by biochemist Meguro—were found for triangles and some special quadrilaterals by Koji Husimi, Jun Maekawa, Fumiaki Kawahata, Toshikazu Kawasaki, and me during the 1980s and early 1990s. The creases that fill in a triangle are very simple to construct; they bisect the three corners.

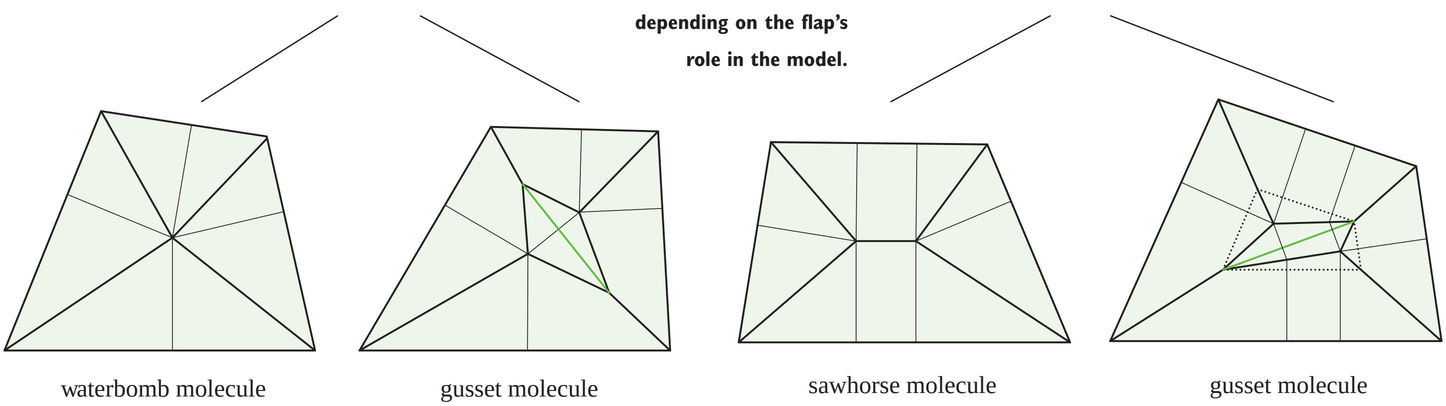
There's a close relationship between a polygon and its tree graph: the sides of the polygon, when folded, become the edges of the graph. For a triangle, it's a one-to-one relationship; there is exactly one triangle for a given three-leaf-node tree graph and vice versa. Quadrilaterals are a bit more complicated, first, because there are two possible tree graphs with four leaf nodes, and second, because there can be many different quadrilaterals for the same tree graph. The two tree graphs are called the "four-star" and the "sawhorse," and are illustrated below, along with two molecules for each. As the number of sides goes up, the number of graphs and molecules grows rapidly.

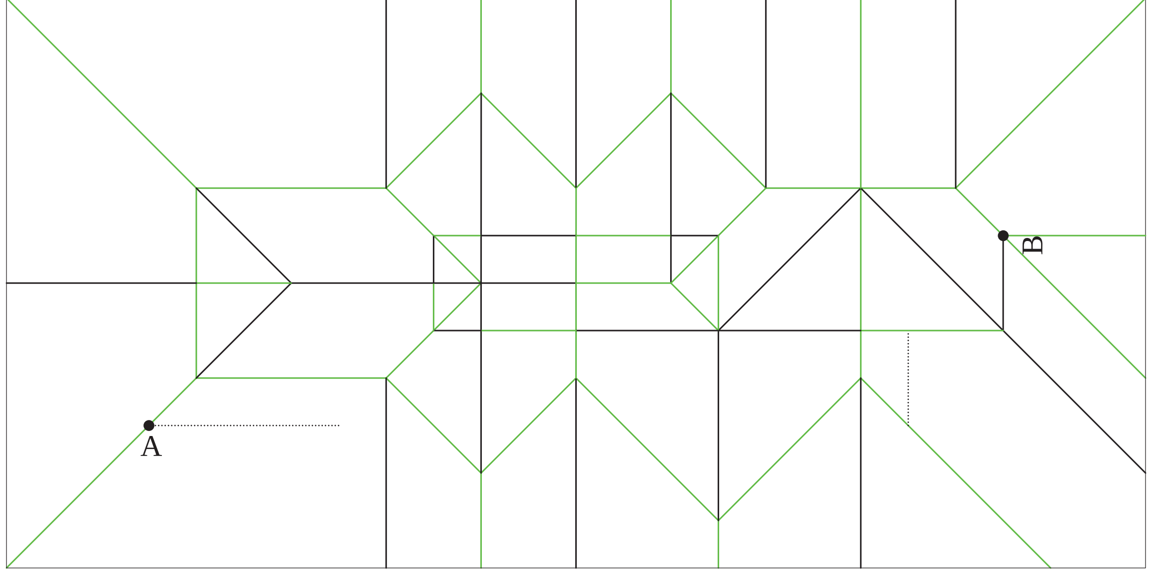
One type of molecule, called the gusset, is particularly versatile; one version of it can be folded into a four-star, and another into a sawhorse. In 1995, I discovered a generalization of it that worked for any convex polygon, no matter how many sides it had. I dubbed the algorithm that creates these solutions the "universal" molecule. Any tree graph can be decomposed into one or more polygons, each of which can be folded into a universal molecule, giving a full crease pattern for any uniaxial base.

Below: The two four-node graphs, and some molecules that fold into them. Heavy black lines are valley folds, heavy green lines are mountain folds, and the light black lines are "hinge creases," which may be mountains, valleys, or unfolded, depending on the flap's role in the model.









A one-cut puzzle. Cut out this rectangle and fold it only on the solid lines: mountain folds are green and valley folds are black. Then cut through all layers along a line that runs from point A to point B (from dot to dot) and carefully unfold all the pieces.

The universal molecule has an interesting property: it enables you to make any convex polygon from a folded sheet of paper with a single straight cut. The "one-cut" problem was independently solved for all polygons, including concave and multiple ones, by University of Waterloo grad student Erik Demaine, now an assistant professor at MIT, whose research revolves around folding of all kinds. Demaine's cutting algorithm bears a

In much of science and engineering, the most productive way to deal with a problem is to turn it into one that somebody else has either already solved or proven impossible. Or, put another way, the key to productivity is letting dead guys do your work for you.

surprisingly close relationship to several issues in pure uncut origami design. The creases' precise locations within a universal molecule depend on the polygon's size and shape and on the lengths of the edges of the tree graph. If you freeze the polygon but shrink the graph, the universal molecule evolves toward, and eventually becomes, the solution to the one-cut problem.

I applied Demaine's algorithm for multiple concave polygons to create the one-cut puzzle above. First, fold the figure—which is, in itself, something of a challenge. Then cut along the dotted line that runs from A to B and unfold the paper. If you've done it correctly, you should obtain the initials of a well-known institution of higher learning.

Let's turn now to the concept of efficiency, around which many computational-geometry problems revolve. For example, the usual goal of the traveling-salesman problem is to find the shortest route among the salesman's cities. In origami design, the most efficient crease pattern is the one that gives the largest possible base for a given tree graph and a given-sized sheet of paper.

We measure a base's efficiency by m, which we call the "scale"; it quantifies how large the finished base is relative to the size of the unfolded square, whose sides we define to be 1 unit long. If m is very small, then all the distances specified by the tree graph are short. The leaf vertices are close together and you can always find a set of them that satisfies all possible path inequalities—in fact, there will be many possible arrangements. But these bases will be very small and, because all that paper must be tucked into them somewhere, they will also be thick, and difficult to fold. On the other hand, if m is made too large, no arrangement of points will work. If we have two flaps, each 1 unit long, then the separation between their leaf vertices must be at least 2 units, and you can't fit two points that far apart into a 1-unit square. Somewhere between the possible and the impossible lies the most efficient base—a crease pattern whose leaf-vertex arrangement satisfies all possible path inequalities for the largest possible value of m.

In much of science and engineering, the most productive way to deal with a problem is to turn it into one that somebody else has either already solved or proven impossible. Or, put another way, the key to productivity is letting dead guys do your work for you. In this case, the problem can be posed in a form known as a "nonlinear constrained optimization," namely: "find a set of variables (the scale m, and the coordinates of the leaf nodes in the crease pattern) that maximizes the value of m subject to a set of inequalities (the path conditions and inequalities that constrain all points to the square of paper)." Thankfully, nonlinear constrained optimization problems have been thoroughly studied by computer scientists. Finding the provably best possible solution is often computationally intractable, but fast, efficient algorithms for near-optimal solutions are known. "Good enough for government work" is also usually good enough for origami design.

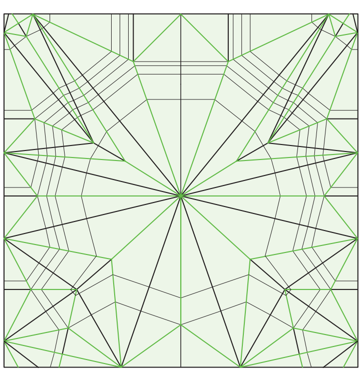
Origami tree theory works best for subjects that can be approximated by a stick figure. It is especially suited to insects like this cicada, in which each of the legs, wings, antennae, and body can be represented by an edge of the tree graph.

Over the 1990s I developed a computer model of tree graphs, bases, and crease patterns and combined it with CFSQP, a nonlinear constrained optimization code created by Andre Tits and his research group at the University of Maryland. The resulting program, TreeMaker, does exactly what I speculated about in 1989; the user enters a tree graph and the software performs the optimization and finds the base's full crease pattern, which may then be transformed into a finished model—like the scorpion below—using common origami shaping techniques. Using TreeMaker, I've been able to design many figures whose complexity is considerably beyond what I (or others) could do by hand.

It turns out that these algorithms are good for more than just making cool animals. The theory of foldable paper also describes what's possible for other materials: cloth, metal, plastic, and so on.

One of the most direct applications has been in modeling airbags for cars. An airbag must inflate in a few milliseconds and be firm enough to stop a rapidly accelerating body, yet provide cushioning. Hitting a rigidly inflated, brick-hard airbag could do as much damage as no airbag at all, and an airbag must work for small children and large adults over a wide range of collision speeds and impact angles. So airbag design involves a lot of computer simulation—if your client is Mercedes-Benz, you don't want to crash more cars than you absolutely have to. The simulations start with the airbag folded up into a small packet and tucked into the (simulated) steering wheel or dashboard.

Calculating the crease pattern for this scorpion base required solving a complex optimization problem using CFSQP.



And that's where the problem arises. While flattening an airbag in real life is fairly easy—you just squash all the air out of it—simulating the process is quite a challenge. You need to treat the airbag as a rigid object, as if it were made out of cardboard; find creases that flatten it; and then fold it up into a small packet. Several years ago, I was contacted by an airbag-design firm, and the universal molecule proved to be just the solution. Thus, origami can not only make beautiful art; it can save lives.

Origami can also expand our view of the heavens. Eyeglass, a brainchild of the Lawrence Livermore National Laboratory, is a space telescope with a projected 100-meter aperture that would be able to examine Earth-like planets around nearby stars.

Eyeglass is a radical new design, but also a very old one. Most high-performance telescopes, like the Hubble Space Telescope, are "reflective." Their main optical element is a curved mirror, which lets the telescope be fairly short—just a few times the diameter of the lens. The Hubble, with its 2.4-meter-diameter mirror, is just 13 meters long. But "transmissive" telescopes, like Galileo and the pirates of the Caribbean used, are tubes with lenses at each end. Transmissive telescopes are by their nature much longer than the lens diameter, and one with a 100-meter lens would need to be thousands of meters long. This does not seem, on first consideration, like a good thing.

But there's a lot of space in space. When the nearest interfering object is 40,000 kilometers away, a kilometer or two doesn't matter much. Even better, you don't actually need to build a tube between the two lenses—simply put your main lens into one orbit, and the other lens plus the camera and associated electronics into another orbit a few kilometers away.

Lang in front of the five-meter Eyeglass prototype at Lawrence Livermore National Laboratory.

PICTURE CREDITS: 8, 10, 12-18, – Robert Lang; 19 – Rod Hyde, LLNL

Eyeglass will use a diffractive lens—a large sheet of glass or plastic with precise grooves machined into its surface, like the lens on an overhead-transparency projector. A thin plastic lens wouldn't be very stiff or strong, of course, but in orbit, this doesn't matter.

But how do you get it up there in good shape? That 100-meter sheet of plastic is going to have to get crumpled, folded, or otherwise stuffed into a tube about four meters in diameter and 10 meters long, like a sleeping bag going into a stuff sack. Although diffractive lenses have looser tolerances than mirrors, one thing they can't tolerate is being crumpled up. The only way such a surface could go into a rocket would be if it were collapsed along a precise set of creases carefully laid out so as not to degrade the optical performance.

Roderick Hyde and his colleagues at Livermore's diffractive optics group discovered that a folding, origami-based solar panel designed by Koryo Miura had powered Japan's Space Flyer Unit back in 1995. Further research led Hyde to my own work, and a phone call revealed the happy coincidence that I lived just five miles from his lab. Over the next few months, I met with the Eyeglass team several times and adapted several origami structures for their consideration. They needed something that was radially symmetric, so that it could be spin-stabilized; would collapse on a finite number of creases; and would then fit into a cylinder. They chose the "Umbrella," which, when furled, looks like a collapsible umbrella. This design can easily be scaled up, has mass-producible parts, and folds from a large flat disk down to a much smaller flanged cylinder. A five-meter prototype has been built, and when unfolded and hung in a test rig, it successfully focused an expanded laser beam fired at it from 100 meters away.

As often happens, once you solve one problem, four or five or ten new ones pop up. Over the past two years, MIT's Demaine and I have collaboratively extended origami tree theory to address crease patterns with regular angles, and the construction of two-dimensional patterns and three-dimensional polyhedra, among other forms. (Regular angles means forcing all crease angles to be some multiple of an integral division of 360°, for example, multiples of 45°.) Computer-aided design solved one set of problems but introduced another: once you've computed the locations of hundreds of creases, how do you figure out how to fold them? Forcing the crease angles to be regular makes it possible to develop tractable step-by-step folding sequences.

So origami has finally hit the big time, it appears—at least in the world of computational geometry. But mathematical origami is also affecting the ancient Japanese craft. There has been a dramatic shift in the art of origami as geometric techniques—what one might call "algorithmic" origami design—have become more widespread. For years, we concentrated on getting the right number and lengths of appendages, to the near-exclusion of considerations such as line, form, and character. With algorithmic origami design, point count comes automatically. Origami art and origami science have sometimes been at odds in the past, but now the origami designer can focus on the art of folding, secure in the knowledge that the science will take care of itself. □

Robert J. Lang (BS '82, PhD '86) was a Member of the Technical Staff at JPL when he wrote "Complexity Increasing" in the Winter 1989 issue of Engineering & Science*. He is now a full-time origami artist, the author of eight books, and a consultant specializing in the application of origami to engineering problems. His most recent book,* Origami Design Secrets *(2003, A K Peters, Ltd.), describes the theory of origami design.*