Puzzles, Art, and Magic with Algorithms*

Erik D. Demaine† Martin L. Demaine*

Abstract

Solving and designing puzzles, creating sculpture and architecture, and inventing magic tricks all lead to fun and interesting algorithmic problems. This paper describes some of our explorations into these areas.

1 Puzzles

Solving a puzzle is like solving a research problem. Both require the right cleverness to see the problem from the right angle, and then the careful execution of that idea until you find a solution. (The main difference is that the puzzle poser usually guarantees that the puzzle is solvable.) Puzzles also lead to research problems which form a sort of metapuzzle: to analyze a family of puzzles and their solutions algorithmically. Solutions to these metapuzzles in turn influence puzzle design, often leading to interesting puzzles that can be appreciated without mathematics.

1.1 Sliding Coins

A sliding-coin puzzle consists of two arrangements of coins on a common grid, as in Figure 1. The goal is to reconfigure one arrangement into the other via a sequence of moves. In each move, the player can move any coin to any grid position that is adjacent (along the grid) to at least two other coins. The coins may be labeled to distinguish which coins should go where, while other groups of coins may be considered identical.

Unlike many puzzles which are NP-hard or worse [Dem01], the majority of sliding-coin puzzles on the square and triangular grids can be solved (or determined unsolvable) in polynomial time [DDV02]. In particular, these puzzles have polynomial-length solutions. What seems to make these puzzles nonetheless challenging for humans to solve is that the polynomials can be large—$\Theta(n^3)$ for n coins on the square grid [DDV02].

This algorithmic understanding gives us a lot of insight into sliding-coin puzzle design, which was our original motivation for this work. In particular,

*A preliminary version of this paper appeared in *Proceedings of the 3rd International Conference on Fun with Algorithms*, Isola d'Elba, Italy, May 2004, pages 7–15.

†MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, {edemaine,mdemaine}@mit.edu

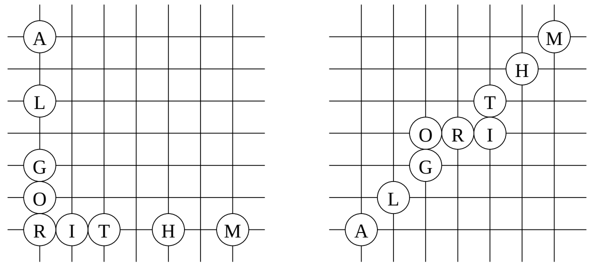

Figure 1: A difficult sliding-coin puzzle with coins that spell ALGORITHM: can you reconfigure the coins from the left arrangement to the right arrangement, at each step moving a coin to a grid position adjacent to at least two other coins?

there are simple conditions on what puzzles are solvable, enabling the puzzle designer to be sure that a puzzle is solvable without explicitly having to try it. What is more, the study of the asymptotic number of moves required for a few types of puzzles gives the designer a rough understanding of what puzzles are hard. In particular, we know the asymptotically "hardest" puzzle, in the sense of requiring the most moves. For example, the type of puzzle in Figure 1 should be of an intermediate difficulty—the number of moves grows as $\Theta(n^2)$ instead of $\Theta(n^3)$—though at the figure's scale of n the solution is probably quite long.

1.2 Map Folding

Map folding is a problem frequently encountered during road trips. One mathematical formulation of this problem is that you are given a rectangular paper map divided into a grid of squares, where each interior edge is a crease marked either mountain or valley, and the goal is to fold the map adhering to all of the crease directions. (In particular, all of the creases must be folded.) In most real maps, the folding is achieved by a sequence of simple folds, each of which folds along a single line. In this case, the map-folding puzzle can be solved in polynomial time, even linear time [ABD$^+$04]. However, if the map has diagonal creases, the problem becomes NP-hard [ABD$^+$04].

A different kind of map-folding puzzle arises when we allow general origami foldings, which fold along multiple creases at once, instead of just simple folds. Mathematically, we can model origami foldings as simply specifying a valid ordering of all the squares in the grid, valid in the sense that it leads to a non-self-intersecting folded state of the map. Although in principle every such folded state can be reached by a continuous folding motion [DDMO04], physically executing such a folding can be a challenging puzzle. For example, try folding the map in Figure 2 so that the squares are stacked in order to spell ALGORITHM.

Even more challenging puzzles arise when the puzzler is told only partial information about the desired origami folding. For example, it may be NP-hard to decide whether there is a folding consistent with a specified mountain-valley

O	T	H
R	I	G
A	M	L

Figure 2: A challenging map-folding puzzle: can you fold the 3×3 map so that reading the squares in their folded stacking order spells the word ALGORITHM?

assignment. This problem was posed by Jack Edmonds.[1]

Unfortunately, the previous two types of map-folding puzzles suffer from the practical difficulty that it is difficult to verify whether you have actually solved the puzzle. Reading the stacking order of the squares in the grid in an already folded map requires careful untucking of flaps of paper, and it is easy to miss a square. (This problem can be fixed by using transparent paper and arranging the labels to not overlap, but transparent paper is usually hard to fold.) Checking for correct mountains and valleys is difficult even when the creases are colored accordingly, because it is even easier to miss a deeply hidden crease.

This difficulty in verification led us to a different family of map-folding puzzles, where verification simplify involves looking at the top and bottom sides of the folding. For such a puzzle to be interesting, we must allow holes cut out of portions of the grid squares that show through to the square behind. We implemented a computer program that considers all 1,368 valid folded state orderings of a 3×3 map, and displays their appearance on either side when the squares have specified labels and holes. This tool allowed us to design a puzzle based on constraining the two visible sides, while still guaranteeing that the solution folding is unique. See Figure 3. A graphical variation of this puzzle is available on the web.[2]

2 Art

Elegant algorithms are beautiful. A special treat is when that beauty translates visually. Sometimes this is by design, when you develop an algorithm to compose artwork within a particular family. Other times the visual beauty of an algorithm just appears, without anticipation.

2.1 Hyparhedra

If you crease a square of paper along several concentric squares, alternating mountain and valley, and along the diagonals, the paper relaxes into a pleated

[1] Personal communication, 1997.
[2] http://theory.csail.mit.edu/~edemaine/puzzles/LCS2003/

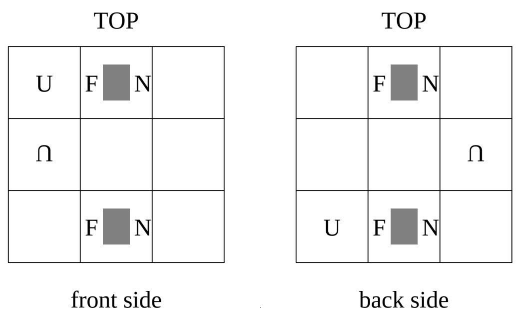

Figure 3: A challenging map-folding puzzle: can you fold the 3×3 map so that both sides read FUN? Label both sides of the map as shown, and cut out the shaded rectangles.

form as shown in Figure 4. We call this pleated folding a *hypar*, short for "hyperbolic paraboloid", the mathematical surface that it approximates.

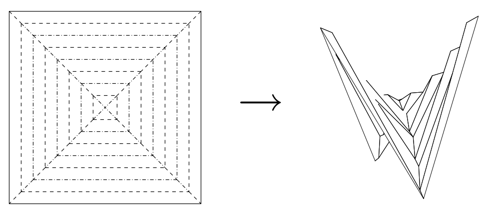

Figure 4: Folding a square of paper into a hypar. This folding was likely originally discovered by John Emmet in England [Jac89, p. 138].

We began experimenting with gluing multiple hypars along their edges. There are many possible ways to glue together complex arrangements of hypars. One particularly interesting family of gluings comes from an algorithm that converts any polyhedron into a hypar gluing [DDL99]. The resulting *hyparhedra* are attractive paper sculptures. Figure 5 shows one example, resulting from the cube.

2.2 Voronoi Architecture

One of our collaborations with MIT's Department of Architecture [ACD+03] explored the use of Voronoi diagrams in architectural design. The particular setting we considered was a museum for the work of Nam Jun Paik, a pioneer of video art. Figure 6 shows one view of our design.

We can think of the Voronoi diagram as an algorithm whose input is a set of points in 2D/3D and whose output is a decomposition of 2D/3D into polygonal/polyhedral cells, one for each input point. The cell corresponding to each input point is the region of 2D/3D points for which that input point is the nearest among all input points. The Voronoi diagram can also be thought

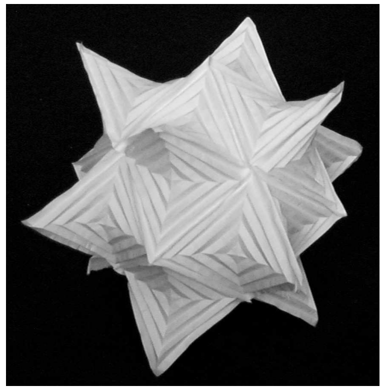

Figure 5: Hyparhedron sculpture resulting from the algorithm in [DDL99] applied to a cube.

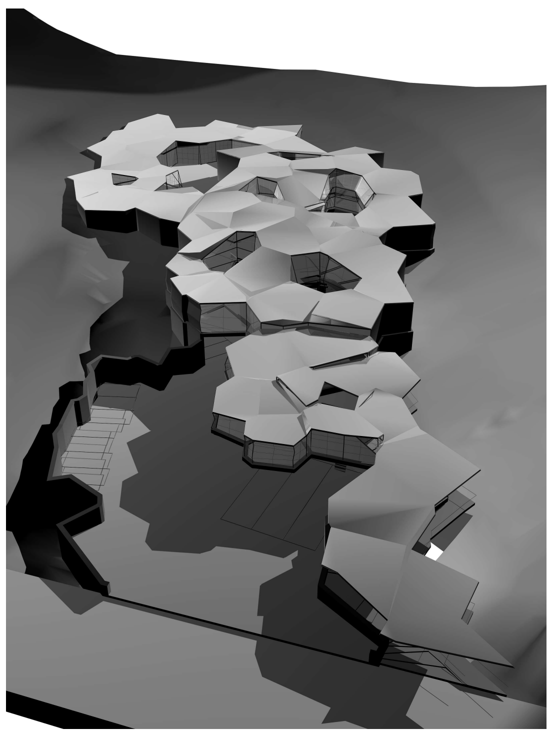

Figure 6: Overhead view of Voronoi architecture from [ACD$^+$03].

of as the result of cellular growth from each of the input points, and the cell decompositions consequently have an organic sense. In fact, Voronoi diagrams arise in surprisingly many contexts in nature, such as the spots on a leopard's skin, honeycombs of bees, and the arrangement of galaxies in the universe. These connections to nature made the Voronoi diagram an attractive source of inspiration for architecture.

Our main challenge in the collaboration was how to give the architects enough control over the Voronoi diagram for the cell decomposition to be meaningful as a structural element of the building. The difficulty is that manipulation of a Voronoi diagram is indirect: you can only change the input points. In 2D, the mapping from input points to cells is sufficiently natural and well-behaved that control comes relatively easily. In 3D, however, the process becomes substantially more complicated. For example, a building requires roughly horizontal surfaces for the floor and roof. How should we choose a set of points to guarantee such surfaces? One effective approach we found for generating one such surface is to hand-pick a set of points in 3D that roughly following a desired shape (e.g., a plane), and then automatically duplicate each point with a random but small vertical offset and optionally with a small random horizontal offset. Together, these points tend to produce Voronoi diagrams with only infinite cells, and the facets in between forming a kind of shell surface.

2.3 Hinged Blocks

A *hinged dissection* is a collection of 2D/3D shapes hinged together at vertices or edges in such a way that the linkage can be folded to form two or more solid 2D/3D shapes. While many hinged dissections between various pairs of shapes have been designed [Fre02], it remains open whether every pair of equal-area 2D polygons have a hinged dissection. The broadest family of 2D hinged dissections is based on polyforms [DDE$^+$05]; in particular, this family includes a hinged dissection, for each $n \geq 1$, that folds into all connected edge-to-edge joinings of n unit squares (polyominoes of size n). This result was recently generalized to 3D shapes [DDLS05], in particular establishing an edge-hinged dissection, for each $n \geq 1$, that folds into all connected face-to-face joinings of n unit cubes (polycubes of size n). The pieces in this dissection can even be constrained to be cubes themselves, in which case the number of pieces is $8n = 2^3 n$.

This last result was recently applied in a collaboration with artist Laurie Palmer from the Art Institute of Chicago. Figure 7 shows the resulting sculpture, called “The Helium Stockpile” [Pal04]. Her sculpture consists of about 1000 identical blocks hinged together in groups of eight. Each group can fold into a $2 \times 2 \times 2$ “macroblock” or various other shapes. If the groups were connected together in one chain, the assembly could fold into any polymacroblock of size around 125. Visitors are encouraged to manipulate, experiment, and interact with the sculpture.

Figure 7: Laurie Palmer's "The Helium Stockpile".

3 Magic

Mathematics is the basis for many magic tricks, particularly "self-working" tricks. One of the key people at the intersection of mathematics and magic is Martin Gardner, whose work has inspired much of what we write about in this paper. Algorithmically, our goal is to design magic tricks within a particular family automatically.

3.1 One-Cut Hell

A classic paper-folding magic trick goes something like this. Two people, Good and Evil, die and arrive at the gates of heaven. Only Good has a ticket to enter heaven. Evil begs Good for help, so Good folds his ticket as shown in Figure 8, rips along a line, and hands Evil the smaller pieces. Unsure of what to do with the pieces, Evil hands them to St. Peter, who re-arranges them to spell H-E-L-L, to which Evil is appropriately directed. Good hands the remaining piece to St. Peter, who is pleased to unfold a cross.

This trick is a special case of a general algorithmic result: any desired collection of line segments can be simultaneously cut by folding flat and making one complete straight cut [DDL98]. Applying this algorithm, you can fold a square of paper flat and make one cut to produce the silhouette of a swan, an angelfish, or a butterfly; or produce your initials; or in principle produce any desired collection of polygonal shapes. In particular, we have applied this result to make a more precise form of the Good-and-Evil magic trick described above [DD04], where each letter is a single piece. See Figure 9. The folding is more complicated, and the pieces are not in perfect proportion, but there is a certain elegance to having exactly the five desired pieces. In particular, this

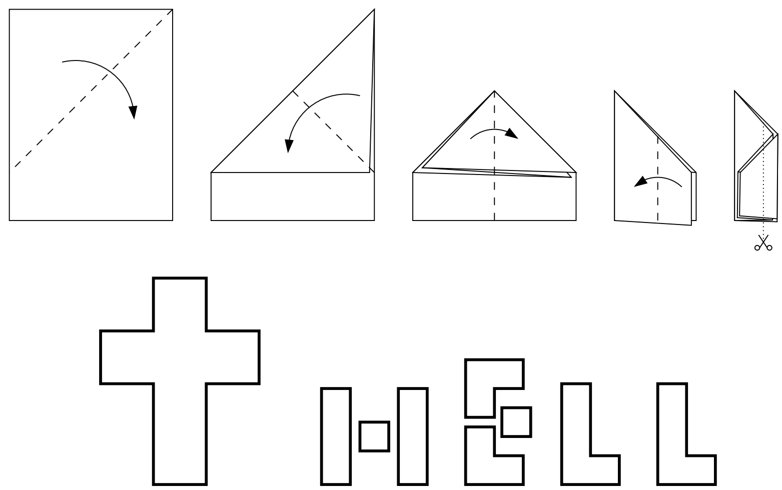

Figure 8: Classic method for producing cross and multipiece H-E-L-L.

version on the trick is easier to perform “standing up”, without a surface to arrange the pieces.

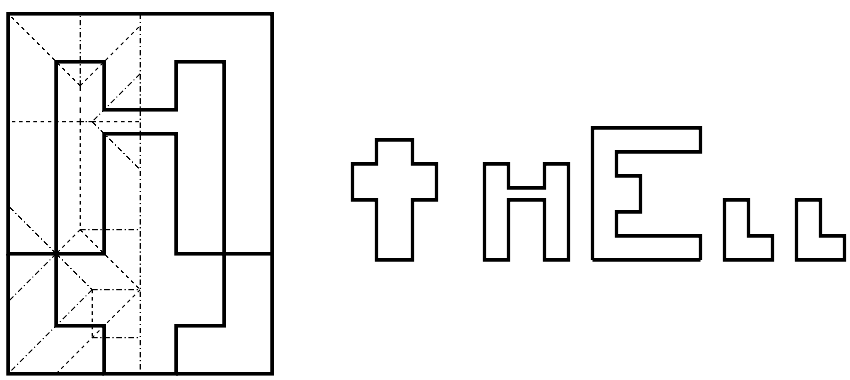

Figure 9: Five-piece design for cross and H-E-L-L from [DD04]. Fold in half first, then use the specified creases.

3.2 Picture Hanging

The magician can hang a picture on two nails in such a way that, no matter which nail you choose to remove, the picture falls. This topological curiosity has circulated the puzzle community in the past few years. Looking at the problem in the right way, the two-nail picture-hanging problem is precisely the Borromean rings, three interlocked loops no two of which are interlocked. Recently, this mathematical trick has been generalized to arbitrary feats [DDM$^+$04]. For example, a picture can be hung on n nails such that removing any k nails causes the picture to fall, but removing fewer nails leaves it hanging, for any $1 \leq k \leq n$. Figure 10 shows such a picture hanging with $k = 1$ and $n = 3$. Even more, a picture can be hung on n red nails and n blue nails such that removing k nails of each color causes the picture to fall, but removing fewer of either color leaves it hanging. In general, you can specify any family of subsets of labeled nails that should cause the picture to fall, and hang a picture so that it falls precisely when removing these subsets or (necessarily) their supersets. These results connect the otherwise disparate fields of puzzles and magic, algebra and topology, and monotone function theory.

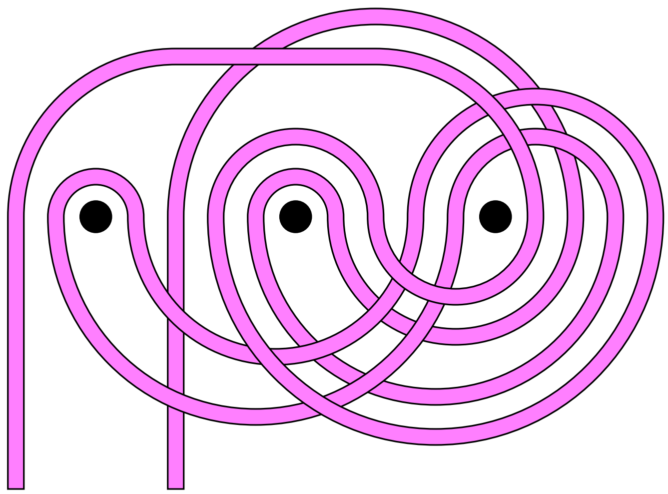

Figure 10: Hanging a picture on three nails such that removing any one nail causes the picture (hung on the two strands in the lower left) to fall.

References

[ABD+04] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S. B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? *Computational Geometry: Theory and Applications*, 29(1):23–46, September 2004.

[ACD+03] Saeed Arida, Eddie Can, Erik D. Demaine, Martin L. Demaine, Talia Dorsey, John Ochsendorf, and Xu Wang. Poetic synecdoches. Architectural design for Nam Jun Paik Museum International Competition, August 2003.

[DD04] Erik D. Demaine and Martin L. Demaine. Fold-and-cut magic. In *Tribute to a Mathemagician*, pages 23–30. A K Peters, 2004.

[DDE+05] Erik D. Demaine, Martin L. Demaine, David Eppstein, Greg N. Frederickson, and Erich Friedman. Hinged dissection of polyominoes and polyforms.

Computational Geometry: Theory and Applications, 31(3):237–262, June 2005.

[DDL98] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and cutting paper. In J. Akiyama, M. Kano, and M. Urabe, editors, *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 104–117, Tokyo, Japan, December 1998.

[DDL99] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Polyhedral sculptures with hyperbolic paraboloids. In *Proceedings of the 2nd Annual Conference of BRIDGES: Mathematical Connections in Art, Music, and Science*, pages 91–100, Winfield, Kansas, 1999.

[DDLS05] Erik D. Demaine, Martin L. Demaine, Jeffrey F. Lindy, and Diane L. Souvaine. Hinged dissection of polypolyhedra. In *Proceedings of the 9th Workshop on Algorithms and Data Structures*, volume 3608 of *Lecture Notes in Computer Science*, pages 205–217, Waterloo, Canada, August 2005.

[DDM+04] Erik D. Demaine, Martin L. Demaine, Yair Minsky, Joseph S. B. Mitchell, and Ronald Rivest. Picture-hanging puzzles. Manuscript in preparation, 2004.

[DDMO04] Erik D. Demaine, Satyan L. Devadoss, Joseph S. B. Mitchell, and Joseph O'Rourke. Continuous foldability of polygonal paper. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 64–67, Montréal, Canada, August 2004.

[DDV02] Erik D. Demaine, Martin L. Demaine, and Helena Verrill. Coin-moving puzzles. In R. J. Nowakowski, editor, *More Games of No Chance*, pages 405–431. Cambridge University Press, 2002. Collection of papers from the MSRI Combinatorial Game Theory Research Workshop, Berkeley, California, July 2000. http://www.arXiv.org/abs/cs.DM/0204002.

[Dem01] Erik D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In *Proceedings of the 26th Symposium on Mathematical Foundations in Computer Science*, volume 2136 of *Lecture Notes in Computer Science*, pages 18–32, Marianske Lazne, Czech Republic, August 2001. Full paper available at http://www.arXiv.org/abs/cs.CC/0106019.

[Fre02] Greg N. Frederickson. *Hinged Dissections: Swinging & Twisting*. Cambridge University Press, August 2002.

[Jac89] Paul Jackson. *Origami: A Complete Step-by-Step Guide*. Hamlyn Publishing Group Ltd., London, 1989.

[Pal04] Laurie Palmer. The helium stockpile: Under shifting conditions of heat and pressure. Installation, Radcliffe College, Cambridge, Massachusetts, April 2004.