The CCCG 2001 Logo

by Erik D. Demaine*, Martin L. Demaine*, and Anna Lubiw*

1 Introduction

The logo for CCCG this year is in fact an origami crease pattern, with the property that if you fold it flat, all the bold lines outlining the letters line up, so making one complete straight cut along the bold line and unfolding the pieces results in a piece of paper with holes spelling the name of the conference, as well as 3 copies of 'C' and 1 copy of 'G'. This note briefly describes how to fold the example, how it was created, and what problem it illustrates.

2 Previous Work

There is some history of CCCG logos based on applying geometric constructions to the letters C-C-C-G; see [BV01]. The CCCG 1994 logo is a triangulation with certain edges emboldened to show the letters. The CCCG 1995 logo is the generalized Voronoi diagram of the exterior face of a polygonal model of the letters.

3 How to Fold

The notation of the crease pattern is as follows. The bold edges are just outlines of the target shapes; do not fold along them. The remaining solid edges are *mountains*, where the paper folds "away" from the folder. The dashed edges are *valleys*, where the paper folds "toward" the folder. (Of course, you can reverse all the creases without any effect.)

We suggest using an enlarged photocopy of the crease pattern. In practice, we precrease all of the edges as if they were mountains, and then work from one end to the other, folding all the creases in the correct direction. This is difficult without practice.

Good luck folding!

*Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, email: {eddemaine, mldemaine, alubiw}@uwaterloo.ca. Supported by NSERC.

4 Fold-and-Cut Problem

This example illustrates the following problem.

> Given a planar graph, drawn with straight edges on a polygonal piece of paper, is there a flat folding of that piece of paper that maps all vertices and edges of the graph to a common line, and maps nothing else to that line? Thus, cutting along the line would achieve precisely the desired pattern of cuts.

There are two approaches for solving versions of this problem, one based on the straight skeleton [DDL98, DDL99], and the other based on disk packing [BDEH98]. While the disk-packing method has better bounds on the number of creases required (linear in the number of disks required to suitably pack the graph), the straight-skeleton method results in more practical foldings. In this example, we use the straight-skeleton method.

5 Construction

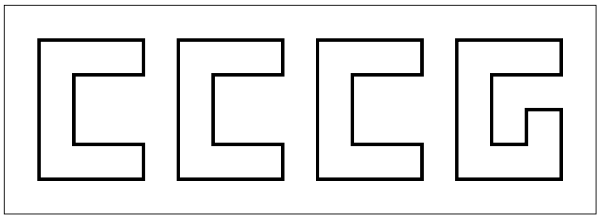
The graph is designed on the square lattice; see Figure 1(a). We first construct the *straight skeleton* as shown in Figure 1(b). One way to define this structure is by continuously shrinking each face of the graph, moving each edge perpendicularly inwards at the same speed, and tracing the loci of the vertices of the graph. Every resulting crease is an angular *bisector* of two edges of the graph, so folding along that crease brings together the two edges of the graph. Intuitively, folding along all of the straight skeleton brings together all of the edges to a common line.

Unfortunately, the straight skeleton by itself cannot be folded flat. Thus we introduce additional *perpendicular* creases, each of which is perpendicular to an edge of the graph. Intuitively, such a perpendicular crease does not affect the "lining up" property of the straight skeleton. A complete diagram of all perpendicular creases is shown in Figure 1(c). Notice how the perpendicular creases "reflect" at edges of the straight skeleton, before extending to infinity.

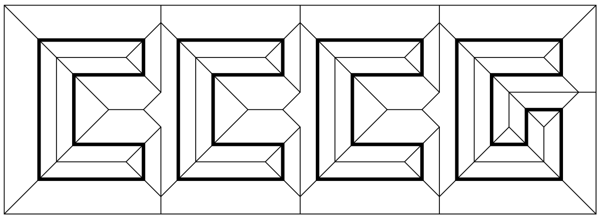
The result is that a perpendicular crease is perpendicular to the unique edge of the graph contained in the same face of the straight skeleton.

We do not need to use all of the perpendicular creases. A sufficient set is shown in Figure 1(d). How this subset is produced algorithmically is beyond the scope of this note; see [DDL98].

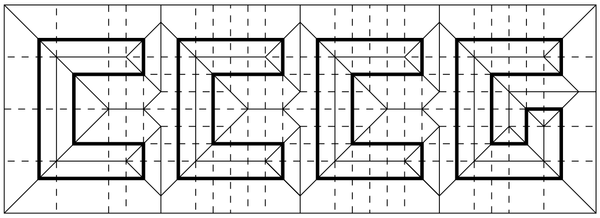
The final step is to assign an appropriate direction (mountain or valley) to each crease, as in Figure 1(d). Again, refer to [DDL98] for details, but we can point out here one essential rule. An edge of the straight skeleton, interior to a polygon, is a mountain if it bisects a convex angle (0 included), and valley if it bisects a reflex angle (360° included), and edges of the straight skeleton exterior to any of the polygons follow the opposite rule.



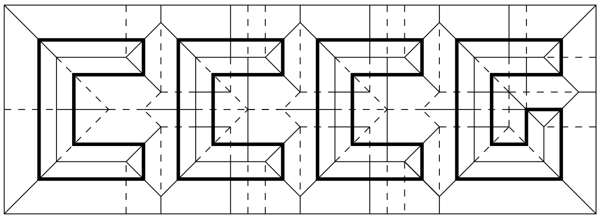
(a) Input graph (bold).



(b) Straight skeleton (thin).



(c) Perpendicular creases (dashed).



(d) Final crease pattern. Mountains are solid, and valleys are dashed.

Figure 1: Constructing the CCCG logo.

6 Future Work

We leave the main open problem, constructing a CCCG 2002 logo, to the next conference organizers.

Acknowledgments

We thank Therese Biedl for suggesting this project.

References

[BDEH98] Marshall Bern, Erik Demaine, David Eppstein, and Barry Hayes. A disk-packing algorithm for an origami magic trick. In *Proceedings of the International Conference on Fun with Algorithms*, pages 32–42, Isola d'Elba, Italy, June 1998.

[BV01] Broňa Brejová and Tomáš Vinař. CCCG 2001: Past CCCG conferences. 2001. `http://compgeo.math.uwaterloo.ca/~cccg01/history.php3`.

[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. Folding and one straight cut suffice. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 891–892, Baltimore, MD, January 1999.