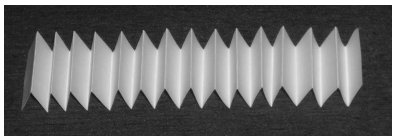
Algorithmic Folding Complexity

Jean Cardinal* Erik D. Demaine† Martin L. Demaine** Shinji Imahori‡
Stefan Langerman*§ Ryuhei Uehara¶

Abstract. How do we most quickly fold a paper strip (modeled as a line) to obtain a desired mountain-valley pattern of equidistant creases (viewed as a binary string)? Define the *folding complexity* of a mountain-valley string as the minimum number of simple folds required to construct it. We show that the folding complexity of a length-n *uniform* string (all mountains or all valleys), and hence of a length-n *pleat* (alternating mountain/valley), is polylogarithmic in n. We also show that the maximum possible folding complexity of any string of length n is $O(n/\lg n)$, meeting a previously known lower bound.

1. Introduction. What is the best way to fold an origami model? Origamists around the world struggle with this problem daily, searching for clever, more accurate, or faster folding sequences and techniques. Many advanced origami models require substantial *precreasing* of a prescribed mountain–valley pattern (getting each crease folded slightly in the correct direction), and then folding all the creases at once. For example, in his instructional video for folding the MIT seal *Mens et Manus* in "three easy steps" [3], Brian Chan spends about three hours precreasing, then three hours folding those creases, and then four hours of artistic folding. The precreasing component is particularly tedious, leading us to a natural algorithmic problem of *optimal precreasing*: what is the fastest way to precrease a given mountain–valley pattern? Although the standard method of "fold one crease, unfold, repeat" is usually the most accurate, it might be possible to fold the paper along some of the desired creases to bring several other desired creases into alignment, and thereby precrease them all at once.



(a) How fast can we fold this?

(b) An origami angel with many pleats, folded by Takashi Hojyo (reproduced with his kind permission).

Figure 1: Pleats.

We focus here on a simple kind of one-dimensional precreasing, where the piece of paper is a long rectangular strip, which can be abstracted into a line segment, and the creases uniformly subdivide the strip. A mountain-valley pattern is then simply a binary string over the alphabet $\{M, V\}$ (M for mountain, V for valley), which we call a *mountain-valley string*. Of particular interest in origami is the *pleat*, which alternates $MVMVMV\cdots$; see Figure 1.

2. Our Results. In this paper, we develop surprisingly efficient algorithms for precreasing a

*Département d'Informatique, Université Libre de Bruxelles (ULB), CP 212, B-1050 Brussels, Belgium, {jcardin,stefan.langerman}@ulb.ac.be.

†Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. {edemaine,mdemaine}@mit.edu

‡Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan. imahori@mist.i.u-tokyo.ac.jp

§Maître de recherches du F.R.S.-FNRS

¶School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa 923-1292, Japan. uehara@jaist.ac.jp

mountain-valley string, especially the pleat.

First, we show how to fold a uniform mountain–valley string $MMM\cdots$ of n mountain creases using just $O(\lg^{1+\sqrt{2}} n)$ simple fold operations. These operations fold only along desired creases, and the last direction that each crease gets folded is mountain. By combining two executions of this algorithm, we obtain the same bound for pleats. This folding is exponentially faster than both the standard folding and the best known folding of $O(n^\varepsilon)$ folds [7]. From a complexity-theoretic perspective, this is the first polynomial-time algorithm for pleat folding, because the only input is the number n.

Second, we show how to fold an arbitrary mountain–valley string of n creases using just $O(n/\lg n)$ folds. This algorithm is the first to beat the straightforward $n-1$ upper bound by more than a constant factor, and is asymptotically optimal [7]. We effectively exploit that every string has some redundancy in it, similar to how Lempel-Ziv can compress any string into $O(n/\lg n)$ block pointers.

Unfortunately, our algorithms are not about to revolutionize pleat folding or other practical paper precreasing, because they assume ideal zero-thickness paper. In reality, folding more than a few layers of paper leads to some inaccuracy in the creases, called *creep* in origami circles, and our algorithms require folding through $\Theta(n)$ layers. Nonetheless, our results lead the way for the development of practical algorithms that limit the number k of layers that can be folded through simultaneously, with speed increasing as k grows.

From an information-theory perspective, paper folding offers an intriguing new definition of the algorithmic complexity of a binary string. The *folding complexity* [7] of a mountain–valley string is the minimum number of folds needed to construct it. Similar to how Kolmogorov complexity compresses a string down to instructions for a Turing machine, folding complexity compresses a string down to instructions for an origamist. Unlike Kolmogorov complexity, however, folding complexity is computable, though its exact computational complexity (between P and EXPTIME) remains open. We lack a specific (deterministic) string whose folding complexity is asymptotically the maximum possible. (The pleat was an early candidate, now known to be far from the worst case.) Nonetheless, our results shed some light on the structure of this new measure.

3. Related Work. Uehara [6] posed the problem we tackle here in August 2008. In March 2009, Ito, Kiyomi, Imahori, and Uehara [7] formalized the problem and made some partial progress. On the positive side, they showed how to fold any mountain–valley string using $\lfloor n/2 \rfloor + \lceil \lg(n+1) \rceil$ folds, a bound we improve on by a logarithmic factor; and they showed how to fold the uniform string and hence a pleat using $O(n^\varepsilon)$ folds, for any $\varepsilon > 0$.[1] On the negative side, they showed that almost every mountain–valley string requires $\Omega(n/\lg n)$ folds using an information-theoretic argument. We tighten this lower bound to prove that a lead constant factor of 1 suffices, reasonably close to our asymptotically matching upper bound which has a lead constant factor of $4+\varepsilon$, for any $\varepsilon > 0$.

About n different mountain–valley strings of length n can be folded using the absolute minimum number of folds, $\lceil \lg(n+1) \rceil$. These strings are called *paper folding sequences* and have been studied much earlier [8, 4, 1].

Acknowledgments. This work was initiated at the WAFOL'09 workshop in Brussels. We thank all the other participants, as well as Guy Louchard, for useful discussions.

References.

[1] Jean-Paul Allouche. Sur la complexité des suites infinies. *Bull. Belg. Math. Soc.*, 1:133–143, 1994.

[2] 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? *Comput. Geom. Theory Appl.*, 29(1):23–46, 2004.

[3] Brian Chan. The making of Mens et Manus (in origami), vol. 1. `http://techtv.mit.edu/collections/chosetec/videos/`, March 2007.

[4] Michel Dekking, Michel Mendès France, Alf van der Poorten. Folds! *Math. Intell.*, 4:130–138, 173–181, 190–195, 1982.

[5] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms*. Cambridge University Press, 2007.

[6] Erik D. Demaine and Joseph O'Rourke. Open problems from CCCG 2008. In *Proc. 21st Canadian Conference on Computational Geometry (CCCG'09)*, to appear, 2009.

[7] Tsuyoshi Ito, Masashi Kiyomi, Shinji Imahori, and Ryuhei Uehara. Complexity of pleat folding. In *Proc. 25th Workshop on Computational Geometry (EuroCG'09)*, 53–56, 2009.

[8] M. Mendès France and A. J. van der Poorten. Arithmetic and analytic properties of paper folding sequences. *Bull. Austr. Math. Soc.*, 24:123–131, 1981.

[1] A somewhat more careful analysis shows that the same algorithm uses $2^{O(\sqrt{\lg n \lg\lg n})}$ folds.