<table>
<thead>
<tr>
<th>項目</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>タイトル</td>
<td>じゃばら折りの一般化とその複雑さの研究</td>
</tr>
<tr>
<td>著者</td>
<td>上原 隆平</td>
</tr>
<tr>
<td>資料</td>
<td>2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録</td>
</tr>
<tr>
<td>インデックス</td>
<td></td>
</tr>
<tr>
<td>注</td>
<td>ERATOセミナ No.10. 2010年8月4日</td>
</tr>
</tbody>
</table>

---

HOKKAIDO UNIVERSITY
ERATO セミナ 2010 - No. 10
じゃばら折りの一般化とその複雑さの研究

上原隆平
北陸先端科学技術大学院大学 准教授
2010/8/4

概要

日本での折紙の研究は、数学的な観点からの研究が先行してきている。しかし Lang, Domain, O’Rourke といった研究者の活躍により、最近 computational origami という言葉が市民権を得つつある。しかし理論計算機科学としての枠組みは、まだ未整備である。折紙を計算機科学の対象としたときの単純なモデルとは何だろうか。ここではごく単純な折紙モデルとして、次のものを考える。入力として与えられるのは、長さ n+1 の紙と、長さ n の M, V 上の文字列である。ここで M は山折り、V は谷折りを意味している。つまり長さ n+1 の紙の上に、等間隔に山/谷が指定されている。この文字列にしたがって紙を折る問題を考える。まず時間計算量に対しては、折りの回数が自然に対応すると考えることができる。この観点から、じゃばら折りとその一般化に対して、講演者らは、ほぼ最適な折りアルゴリズムを設計した。では領域計算量に対応する概念は何だろうか。これに対してはみんなが合意する概念は存在しない。講演者は、折り目に挟まる紙の枚数を基準として提案し、これに関する最適化問題を研究している。部分的な解は得られているものの、まだ残された研究課題の方が多いのが実状である。
Ryuhei Uehara
JAIST
uehara@jaist.ac.jp
http://www.jaist.ac.jp/~uehara
or, ask Google with "uehara origami"

1/33

Background story...
- At 4th Japan Meeting on Origami in Science, Mathematics, and Education, 2008/6/22,
  - Toshikazu Kawasaki, a mathematician and designer of Kawasaki rose, said that
    "For a mathematician, it is OK if solution exists."
  - A computer scientist, or I, cannot agree;
    "For a mathematician, it is OK if solution exists."
  - how to find the solution
  - the cost to find/construct the solution
  - Goodness…
  - Hardness…
- Is there any good problem just about computational cost??

2/33

Complexity of folding(?)
From the viewpoint of Computer Science
- Two resources of an origami model;
  1. time: the number of steps of operations
  2. space: the number of memory cells required to compute

3/33

http://www.jaist.ac.jp/~uehara/etc/origami/

5/33

http://www.jaist.ac.jp/~uehara/etc/origami/

7/33

Complexity of folding
From the viewpoint of Computer Science
- Two resources of an origami model?
  1. time…the number of foldings (operations)
  2. space…minimization of "stretch" that is
    - the number of papers between two hinged papers
Least stretch folding problem
- The number of folding ways for a random pattern
  - \( \Theta(1.65^n) \) by experiments
  - \( \Omega(1.53^n) \) and \( \Omega(2^n) \) by theoretical lower/upper bounds
- So a naive program runs veeeeerrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr slow.

(Note) These results are based on enumeration & rough counting, described hereafter...
Least stretch folding problem

- Upper bound $f(n) = \Omega(4^n)$ comes from the Catalan number.

[Proof] If the paper of length $n+1$ is folded, the crease points should be nested.

$$\begin{align*}
&\text{(00)(00)} \\
\text{Next} &\quad \text{(00)(00)} \\
\text{Next} &\quad \text{(00)(00)} \\
\text{Combination of } n^2 \text{ pairs of } (i =) \text{ Catalan Number } C_n.
\end{align*}$$

\[\text{[Proof]} \quad \text{We consider “folding of the last } k+1 \text{ unit papers”;} \]

\[f(n) \geq (g(k))^k = (g(k)^{k+1})^k \]

Thus, we have the lower bound.

Least stretch folding problem

[Proof] We consider “folding of the last $k+1$ unit papers”;

- $g(k)$: the number of folding ways of length $k+1$ s.t. the leftmost endpoint is not covered
- "the number of ways a semi-infinite directed curve can cross a straight line $k$ times",

A000682 in "The On-Line Encyclopedia of Integer Sequences".
From that site, we have $g(43) = 8307625906531894760$. Thus, by

$$f(n) \geq (g(k))^k = (g(k)^{k+1})^k$$

we have the lower bound.

Open problem and Future work

- Extension to...
  - non-unit length (operation-restricted linkage)
  - 2D (a kind of map folding)
- What "space complexity" of Origami is?
- area for folding?

Universal theorem:

- Universal theorem of unit length folding in a simple folding model;
  - Input: a folded state of a paper strip of length $n+1$ to unit length
  - Question: Is it foldable on a suitable set of operations?
  - Simple folding model

1. (from flat state)
2. pick up a point
3. valley fold most inner layers
4. to flat state

http://www.jaist.ac.jp/~uehara/etc/origami/
Let's turn to the folding complexity of pleat folding

- Repeating of mountain and valley foldings
- Basic operation in some origami
- Many applications

Pleat folding (in 1D)
- Naive algorithm: $n$ time folding is a trivial solution
- We have to fold at least $\log n$ times to make $n$ creases
- More efficient ways…?
- General Mountain/Valley pattern?

Repeating "folding in half" is the best way to make many creases

- proposed at Open Problem Session on CCCG 2008 by R. Uehara.
- T. Ito, M. Kiyomi, S. Imahori, and R. Uehara: "Complexity of pleats folding", EuroCG 2009...
• Complexity of Pleat Folding

[Main Motivation] Do we have to make $n$ foldings to make a pleat folding with $n$ creases??

1. The answer is "No"!
   - **Any pattern can be made by** $\left\lceil \frac{n}{2} \right\rceil \times \left\lceil \log_2 n \right\rceil$ foldings
2. Can we make a pleat folding in $o(n)$ foldings?
   - Yes! It can be folded in $O(\log^2 n)$ foldings.
3. Lower bound; $\log n$
   - $\Omega(\log \log n)$ lower bound for pleat folding!!

• Upper bound of **Unit FP** (1)

[Next Motivation] What about general pleat folding problem for a given M/V pattern of length $n$?

- **Any pattern can be made by** $\left\lceil \frac{n}{2} \right\rceil \times \left\lceil \log_2 n \right\rceil$ foldings
- **Upper bound**:
  - Any M/V pattern can be folded by $(4 + \epsilon) \frac{n}{\log n} \left( \frac{n}{3 \log_2 n} \right)$ foldings
- **Lower bound**:
  - Almost all mountain/valley patterns require $\frac{n}{3 \log_2 n}$ foldings

[Note] Ordinary pleat folding is exceptionally easy pattern!

• Decor

Difficulty/Interest come from two kinds of Parities:
- "Face/back" determined by layers
- Stackable points having the same parity

Input: Paper of length $n+1$ and a string $s$ in $\{M, F\}^*$

Output: Well-creased paper according to $s$ at regular intervals.

**Basic operations**
1. Flat {mountain/valley} fold (all/some) papers at an integer point (= simple folding)
2. Unfold (all/rewind/any) crease points (= reverse of simple foldings)

**Rules**
1. Each crease point remembers the last folded direction
2. Paper is rigid except those crease points

**Goal**: Minimize the number of folding operations

**Note**: We ignore the cost of unfoldings

Upper bound of **Pleat Folding** (1)

[Observation]
If $f(n)$ foldings achieve $n$ mountain foldings, $n$ pleat foldings can be achieved by $2f(n/2)$ foldings.

The following strategy works;
- Make $f(n/2)$ mountain foldings at odd points;
- Reverse the paper;
- Make $f(n/2)$ mountain foldings at even points.

We will consider the "mountain folding problem"
• Lower bound of Unit FP

[Thm] Almost all patterns except \(2^n\) exceptions require \(\Omega(n/\log n)\) foldings.

[Proof] A simple counting argument:

- # patterns with \(n\) creases > \(2^n/4 > 2^{n-2}\)
- # patterns after \(k\) foldings < \((2 \times n) \times (n+1) \times (2 \times n) \times (n+1) \times \ldots \times (n+1) \times (2 \times n)\)

We cannot fold most patterns after at most \(k\) foldings if

\[\sum_{i=0}^{k} (2n(n+1))^{i} \leq (2n(n+1)+1)^{i} < 2^{k+2}\]

Letting \(n \geq 2, k = O\left(\frac{n}{\log n}\right)\) we have \((2n(n+1)+1)^{k} = o(2^n)\)

Any pattern can be folded in \(cn/\log n\) foldings

Prelim.
- Split into chunks of size \(b\);
  1. Each chunk is small and easy to fold
  2. # kinds of different \(b\) are not so big

Main alg.
- For each possible \(b\)
  1. pile the chunks of pattern \(b\) and mountain fold them
  2. fix the reverse chunks
  3. fix the boundaries

Repeat for all chunks

Open Problems
- Pleat foldings
  - Make upper bound \(O(\log^2 n)\) and lower bound \(\Omega(\log n/\log\log n)\) closer
- “Almost all patterns are difficult”, but…
  - No explicit M/V pattern that requires \(cn/\log n\) foldings
- When “unfolding cost” is counted in…
  - Minimize #foldings + #unfoldings

Analysis is omitted