

# じゃばらを高速に折る --- folding complexity ---

上原隆平(うえはらりゅうへい)

北陸先端科学技術大学院大学 情報科学研究科

Japan Advanced Institute of Science and Technology (JAIST)

uehara@jaist.ac.jp

<http://www.jaist.ac.jp/~uehara>

Google に “上原 折り紙” でもOK!

# じゃばら折り = 1次元折り紙



- 山折りと谷折りが等間隔に交互に続く
- 折り紙の基本ツール
- 応用も...?
- 山/谷を一般化したときの複雑さとは?

「(M/Vによる)2進数文字列上の操作の問題」と考えると、コンピュータサイエンスと相性がよい



# 折り紙の複雑さ/効率(?)

- 計算機科学の視点で考えよう...
- 例えばチューリング機械モデルにおける2種類の資源とは
  1. 時間: 基本演算の適用回数
  2. 領域: 計算に必要なメモリ量

# 折り紙の複雑さ/効率(?)

- 計算機科学の視点で考えよう...
- 折り紙モデルにおける2種類の資源とは？

## 1. 時間...折り(基本演算)の回数

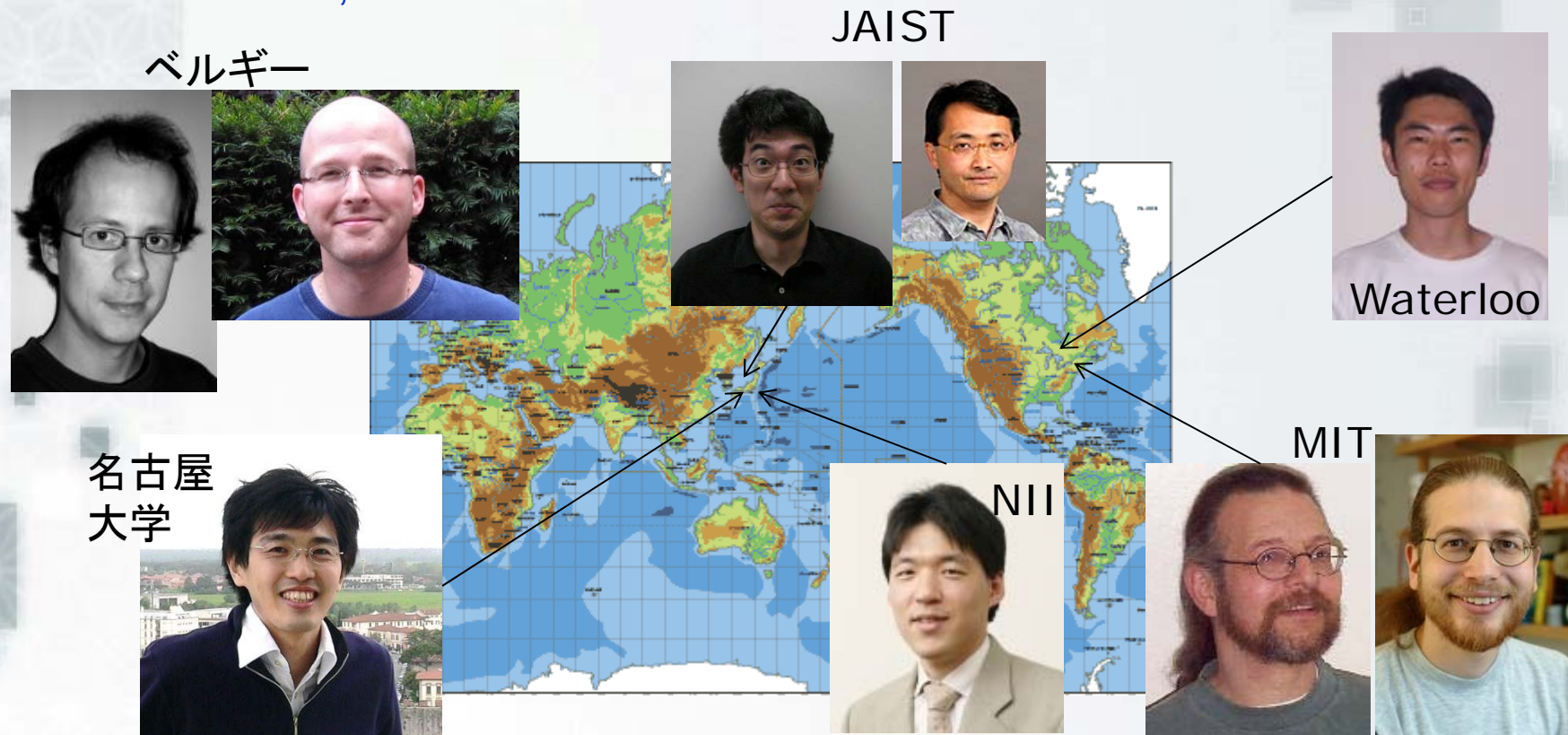
- ◆ J. Cardinal, E. D. Demaine, M. L. Demaine, S. Imahori, T. Ito, M. Kiyomi, S. Langerman, R. Uehara, and T. Uno: Algorithmic Folding Complexity, *Graphs and Combinatorics*, Vol. 27, pp. 341-351, 2011.

## 2. 領域...???

- R. Uehara: Stretch Minimization Problem of a Strip Paper, *5th International Conference on Origami in Science, Mathematics and Education*, 2010/7/13-17.
- R. Uehara: On Stretch Minimization Problem on Unit Strip Paper, *22nd Canadian Conference on Computational Geometry*, pp. 223-226, 2010/8/9-11.

## 1. 時間...折り(基本演算)の回数

J. Cardinal, E. D. Demaine, M. L. Demaine, S. Imahori, T. Ito, M. Kiyomi, S. Langerman, R. Uehara, and T. Uno: Algorithmic Folding Complexity, *Graphs and Combinatorics*, Vol. 27, pp. 341-351, 2011.



# じゃばら折り

「中央で半分に折る」ことが  
もっとも多くの折り目をつ  
けることができる

## → じゃばら折り(1次元折り紙)



- 単純な方法:  $n$  回折ればよい
- $n$  個の折り目をつけるには  $\log n$  回は折らないといけない
- もっと効率よく折り目をつけることはできるのか...?
- 一般のM/Vパターンに対してはどうか?



- CCCG 2008 の “Open Problem Session” にて上原が提案 (Ron Graham が絶賛してくれた!!)
- 何度かの発表と結果の改善を経て多くの共著者を  
得て論文に至る

# じゃばら折りの複雑さ

モデル: 紙の厚みは0

**[最大の疑問]**  $n$  個の折り目からなるじゃばら折りは  
 $n$  回折らなければ作れないのか？

1. 答えは“No”!

- 任意のM/Vパターンは高々  $\lfloor n/2 \rfloor + \lceil \log n \rceil$  回折れば作れる!
- では  $o(n)$  回ではどうか?
- Yes!! ...わずか  $O(\log^2 n)$  回の折りで作る方法がある

2. 下界;  $\log n$

- じゃばら折りは  $\Omega(\log^2 n / \log \log n)$  回未満の折りでは作れない!!

# 一般化じゃばら折りの複雑さ

**[次の疑問]** 与えられた長さ  $n$  の任意の M/V パターンを折る問題の複雑さはどうか？

□ **任意のM/Vパターン** 高々  $\lfloor n/2 \rfloor + \lceil \log n \rceil$  回折れば作れる!!

1. 上界:

どんなM/Vパターンも  $(4+\varepsilon)\frac{n}{\log n} + o\left(\frac{n}{\log n}\right)$  回の折りで作れる

2. 下界:

ほぼすべてのM/Vパターンは  $\frac{n}{3+\log n}$  回以上折らないと作れない

**[おまけ]** じゃばら折りは例外的に簡単なパターンであった！



## ● 定義

- 2種類の**パリティ**があるところが{難しい|面白い}
- “表/裏”は層のパリティで決まる
  - 同じパリティでないと重ねて折れない

入力: 長さ  $n+1$  の紙と  $\{M, V\}^n$  の文字列  $s$

出力:  $s$  にしたがって等間隔に折り目のついた紙

### 基本操作

1. 整数点で{一部/全部}の紙を{山/谷}折りにして平坦にする  
(=単純折りモデル)
2. {すべて/任意/直前}の折り目で開く (=単純折りの逆操作)

### 規則(仮定)

1. それぞれの折り目は**最後に折られた方向を記録している**
2. 紙は折り目を除いて**剛体**

目標: 折り操作の回数の最小化

**注意:** 紙を開く操作のコストは気にしない

# • 単位長折りの上界(1)

- どんなパターンも  $\lfloor n/2 \rfloor + \lceil \log n \rceil$  回で折れる
  1. 指定にしたがって紙を中心に半分に折る
  2. 折られた紙の中心を見て、 $M$ と $V$ の多数決をとる  
(裏返った紙に注意)
  3. 多数決に従って中心で半分に折る
  4. 2と3を繰り返す
  5. 全部広げる
  6. 間違った折り目を逐一直す

ステップ1~4の折りは  $\log n$  回でステップ6は高々  $n/2$  回の折り



# じゃばら折りの上界(1)

## [観測]

もし  $f(n)$  回の折りで「 $n$  個の山折り」が作れるなら、 $n$  個のじゃばら折りは  $2f(n/2)$  回の折りで作れる

## 以下の方法でよい:

- 偶数点だけに着目して  $f(n/2)$  回折って全部山折りにする;
- 紙を裏返す;
- 奇数点だけに着目して  $f(n/2)$  回折って全部山折りにする.

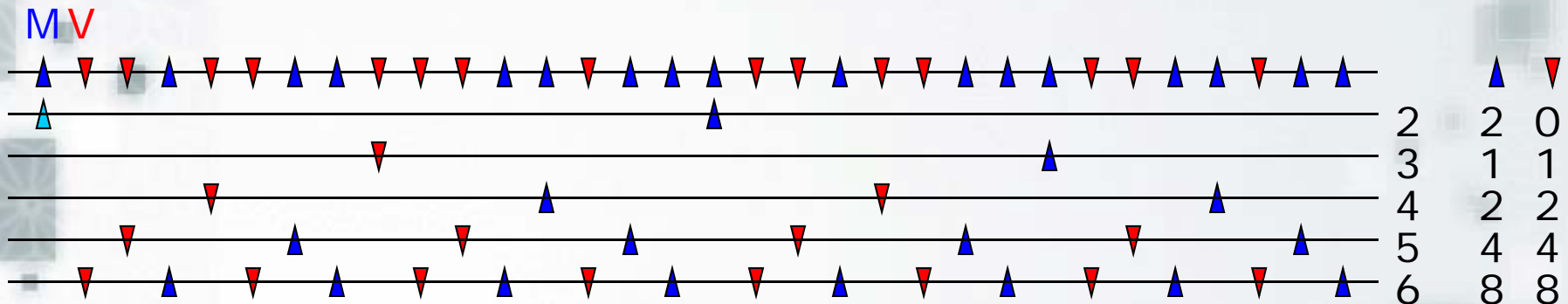
よって以下では「山折り問題」を考える

## 参考:「山折り問題」の $O(n^{0.69})$ アルゴリズム

### [定理] 山折り問題を解く $O(n^{0.69})$ アルゴリズム

[証明]  $n=2^k$  として以下のアルゴリズムを使う

1. 左端を山折りにして、長さを  $2^{k-1}$  にする
2. 中央で山折りにする
3. 長さ1になるまで2を繰り返す
4. 全部開いて...

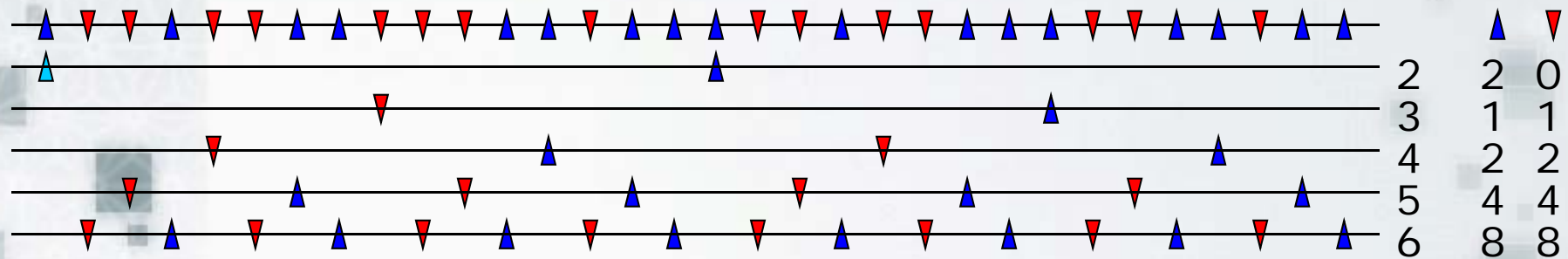


$k+1$ 回折って  $2^{k-1}+1$  個の山と  $2^{k-1}-1$  個の谷ができる

# 参考:「山折り問題」の $O(n^{0.69})$ アルゴリズム

## [定理] 山折り問題を解く $O(n^{0.69})$ アルゴリズム

[証明]



$2^{k-1}-1$  個の谷は独立で等間隔な  $k-1$  個の層に分けられる!!

$$\therefore f(2^k) = 1 + k + f(2^{k-2}) + f(2^{k-3}) + \dots + f(4) + f(2) + f(1)$$

$$f(2^{k-1}) = k + f(2^{k-3}) + \dots + f(4) + f(2) + f(1)$$

$$f(2^k) - f(2^{k-1}) = f(2^{k-2}) + 1$$

$$(f(2^k) + 1) = (f(2^{k-1}) + 1) + (f(2^{k-2}) + 1)$$

$k$  に関するフィボナッチ数列!



## 参考: 「山折り問題」の $O(n^{0.69})$ アルゴリズム

### [定理] 山折り問題を解く $O(n^{0.69})$ アルゴリズム



[証明]

$k$  に関するフィボナッチ数列!

$$(f(2^k) + 1) = (f(2^{k-1}) + 1) + (f(2^{k-2}) + 1)$$

初期条件:

$$f(2^0) = 1, f(2^1) = 2, f(2^2) = 4$$

よって  $f(2^k) + 1 = F_{k+3}$

[フィボナッチ数列]

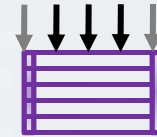
$$F_0 = 0, F_1 = 1, F_i = F_{i-1} + F_{i-2} \quad (i > 1)$$

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

$$F_i = \frac{1}{\sqrt{5}} \left( \left( \frac{1+\sqrt{5}}{2} \right)^i - \left( \frac{1-\sqrt{5}}{2} \right)^i \right)$$

$$\begin{aligned} \therefore f(n) = f(2^k) &= \frac{1}{\sqrt{5}} \left( \left( \frac{1+\sqrt{5}}{2} \right)^{k+3} - \left( \frac{1-\sqrt{5}}{2} \right)^{k+3} \right) - 1 \square \frac{1}{\sqrt{5}} \left( \frac{1+\sqrt{5}}{2} \right)^{k+3} \\ &= O \left( \left( \frac{1+\sqrt{5}}{2} \right)^{\log n} \right) = O \left( n^{\log \frac{1+\sqrt{5}}{2}} \right) = O(n^{0.694242}) \quad \square \end{aligned}$$

# 山折り問題を $\log^2 n$ 回の折りで解く アルゴリズム



## ステップ1;

1. 「半分に折る」を繰り返して以下をえる ( $\log n - 2$  回の折り): [vvv]
2. 3回山折りして以下を得る: [MMM]
3. 開く; vMMMvvvvvMMMvvvvvMMMvvvvvMMMvvvvv...

## ステップ2;

1. すべての“vvvvv”が重なるように半分折りを繰り返す ( $\log n - 3$  回折る)
2. 5回山折りして以下を得る: [MMMMMMM]
3. 開く; vMvMMMMMMMvMvvvvvMvMMMMMMMvMvvvvvMvM

[MvvvvvM]

ステップ3; “vvvvv” が一つ残るまでステップ 2 を繰り返して以下を得る:

vMvMMMvMMMvMMMMMMMvMMMvMMMvMvvvvvMvM

ステップ4; すべてのとびとびの v を一つずつ折る.

- ステップ2~3の繰り返し回数;  $\log n$
- ステップ4での v の個数;  $\log n$

折りの回数のトータル  $\sim (\log n)^2$

# • 単位長折り問題の下界

[定理]  $o(2^n)$ 個の例外を除くほぼすべてのパターンは  
 $\Omega(n/\log n)$  回折らないと作れない

{表/裏} × {前/後}

[証明] 単純な数え上げ法による:

- $n$  個の折り目のパターンの数  $> 2^n/4 = 2^{n-2}$
- $k$  回折ったことで作れるパターンの数  $<$

$$(2 \times n) \times (n+1) \times (2 \times n) \times (n+1) \times \dots \times (n+1) \times (2 \times n)$$

山/谷

可能な展開

$$< (2n(n+1))^k$$

位置

よって以下が成立する場合、高々  $k$  回の折りではすべてのパターンは作れない:  $\sum_{i=0}^k (2n(n+1))^i \leq (2n(n+1)+1)^k < 2^{n-2}$

ここで  $n \geq 2, k = O\left(\frac{n}{\log n}\right)$  とおくと  $(2n(n+1)+1)^k = o(2^n)$  をえる。

□



# 任意のパターンを $cn/\log n$ 回の折りで 作るアルゴリズム(概要)

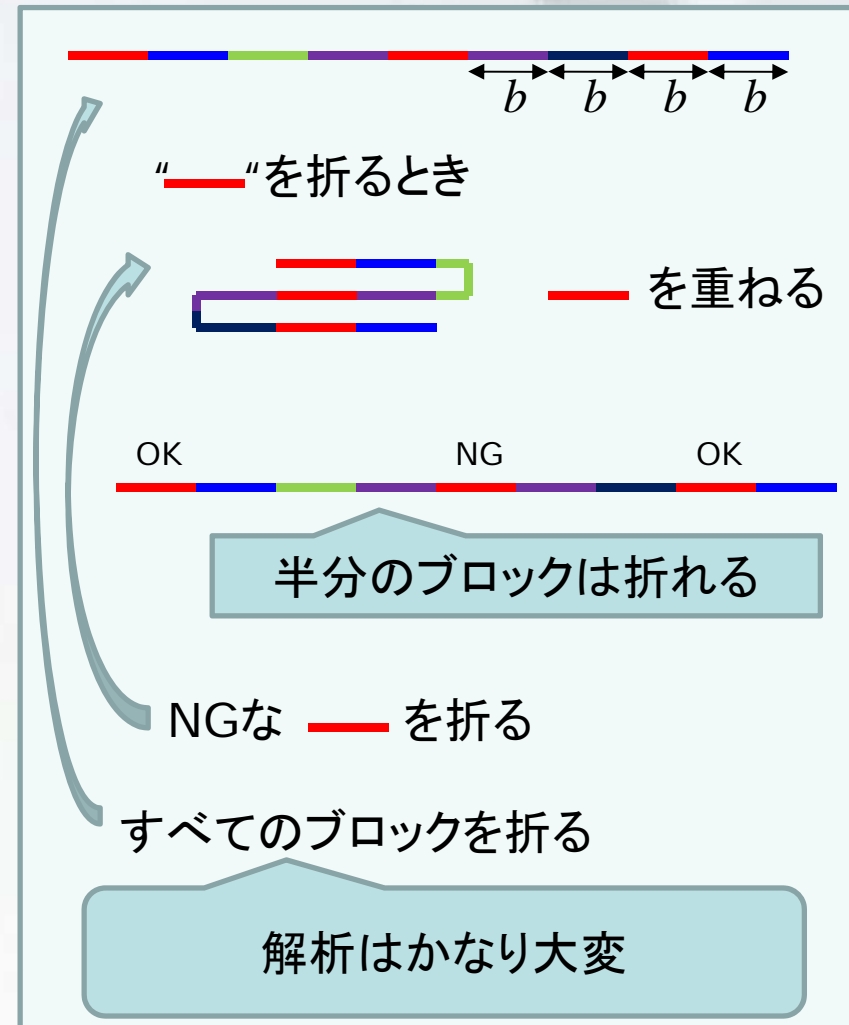
## 準備:

$n$ に応じて適切に  $b$  を選ぶ.

- パターンを大きさ  $b$  のブロックに分割;
  1. 各ブロックは  $O(n/\log n)$  回で折れる
  2.  $2^b$  はそれほど大きくない

## アルゴリズム:

- 可能なそれぞれのブロックパターンに対して
  1. 同じパターンをもつブロックが重なるように半分折りする
  2. 裏返っているブロックを直す
  3. 重ねるために折った所を直す



# 未解決問題

## ➤ じゃばら折り

- 上界  $O(\log^2 n)$  と下界  $\Omega(\log^2 n / \log \log n)$  を近づける

## ➤ 「**ほぼすべての**パターンは難しい」と言うけれど...

- $(cn/\log n)$  回の折りが本当に必要な具体的な M/V パターンの構成方法はわかっていない

## ➤ 紙を開くコストは無視しているけど...

- 「折る回数」+「開く回数」を最小化するとよいかも
- (たかだか折った回数しか開けないけど...)

- もちろん2次元への拡張も...