Introduction to Computational Origami

Ryuhei Uehara

Japan Advanced Institute of Science and Technology (JAIST)

School of Information Science

uehara@jaist.ac.jp

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

Origami?

- In Japanese, “Ori”=folding and “kami/gami”=paper.
  - It was born in 1500? with inventing paper, in some Asia, maybe. Of course, we have no record on paper!
  - Now, “ORIGAMI” is an English word, and there are some shelves in bookstores in North America and Europe.
  - Origami-like things…

There are some “Origami”s which are not folded, and not paper any more now a day!! Maybe by an NSF big fund?

Origami as paper folding

- Normal Origami
- Difficult Origami
- Impossible Origami (for most human!)

Kawasaki Rose

Maekawa Devil

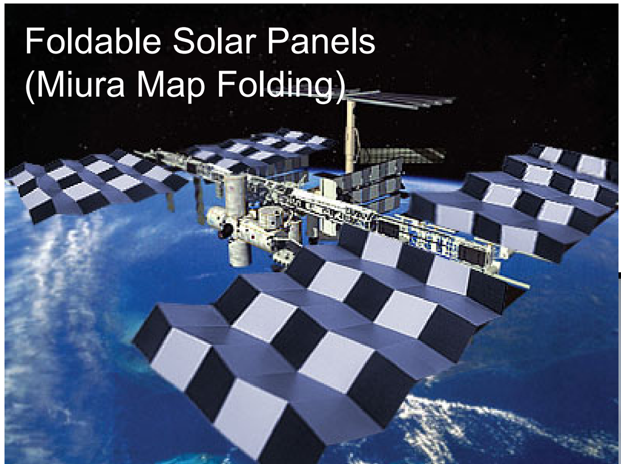
By Tetsushi Kamiya (Origami Champion)

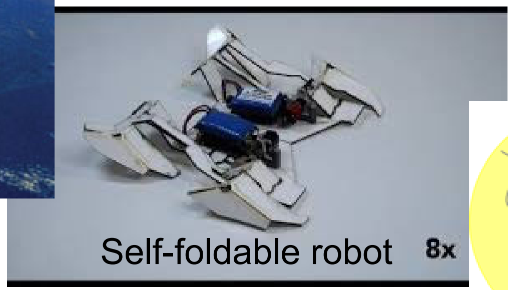
Application of Origami

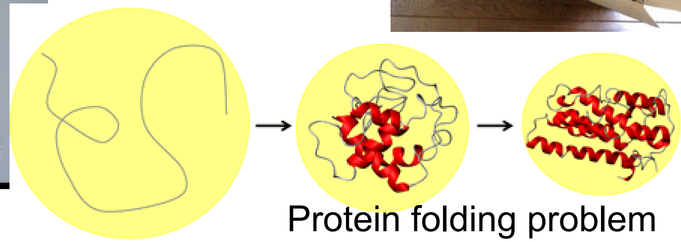
- There are many applications of "Folding"→ Computational Origami

Science based on the basic operations of "folding"

There are many applications and open problems of "folding"







Protein folding problem

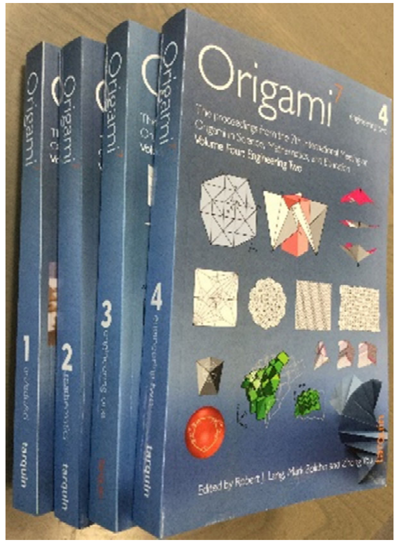
International Conference on Origami Science

1. 1989@Italy
   - International meeting of Origami Science and Technology
2. 1994@Japan
   - International meeting of Origami Science and Art
3. 2001@USA
   - 3OSME(International meeting of Origami Science, Mathematics, and Education)
4. 2006@USA
   - 4OSME
5. 2010@Singapore
   - 5OSME
6. 2014@Japan
   - 6OSME
7. 2018@UK
   - 7OSME

Proceedings is on market

Proceedings become 2 volumes

Proceedings become 4 volumes



Computational ORIGAMI=

Geometry＋Algorithm＋Computation

- Mathematics
- Theoretical Computer Science
- Real High Performance Computing

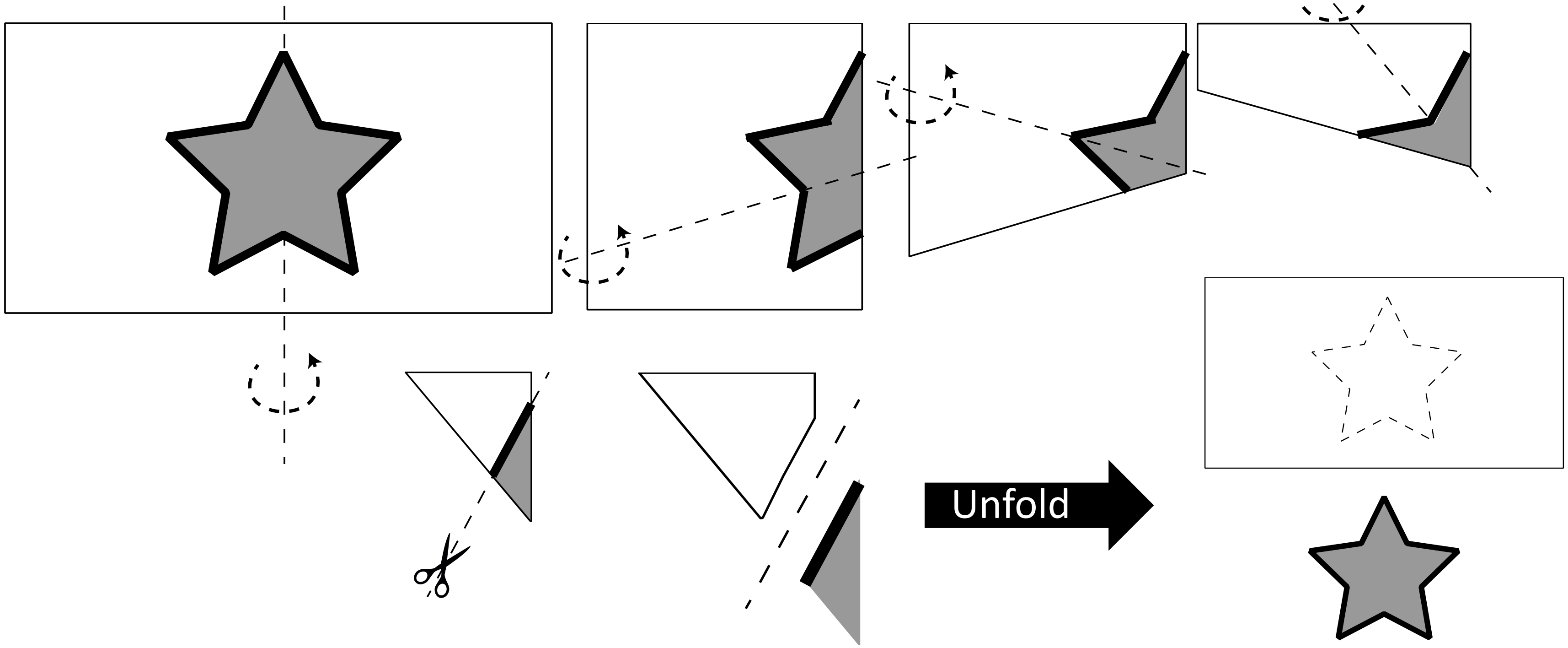
- Many Applications from micro-size to universe-size
  - Bioinfomatics (e.g., DNA folding),
  - Robotics, packaging,
  - Architecture

- Many young researchers;
  - even undergrad students, highschool students!

Today’s Topic: Fold-and-cut problem

Fold and Cut Problem:

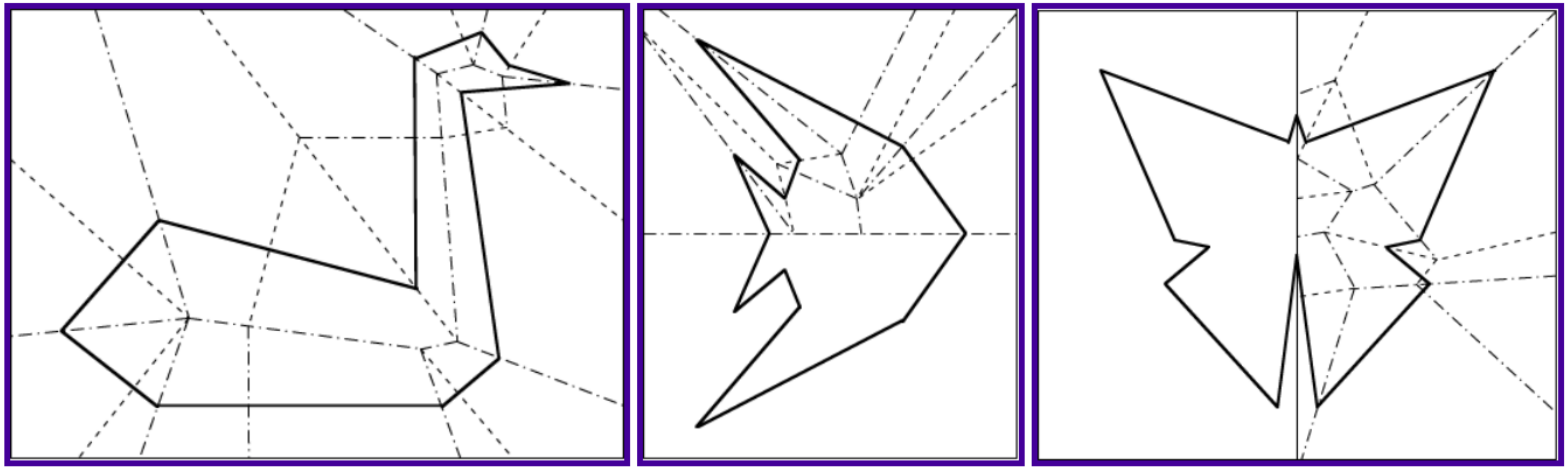
Take a sheet of paper, fold it flat however you like and make one complete straight cut, what shapes can you get from the unfolded pieces?



Universal theorem

Universal theorem [Demaine et al. 1998-2001]:

Every plane graph can be made by folding and one complete cut.



See http://erikdemaine.org/foldcut or

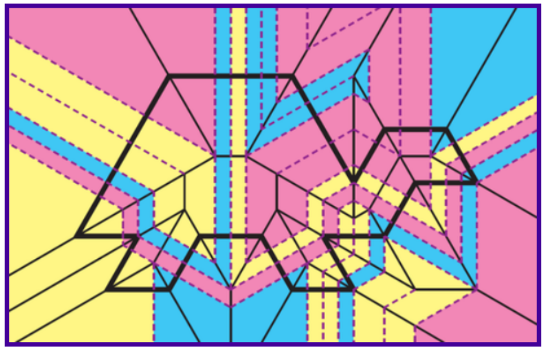
Erik Demaine and Joseph O'Rourke. Geometric Folding Algorithms: Linkages, Origami, Polyhedra. Cambridge University Press, 2007

Universal theorem

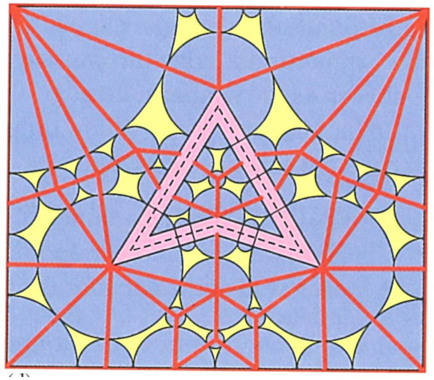
Universal theorem [Demaine et al. 1998-2001]:

Every plane graph can be made by folding and one complete cut.

[Proof] Two major approaches;



Straight-Skeleton method



Disk Packing method

Both are "complicated" and "hard" by, e.g., folding robots…

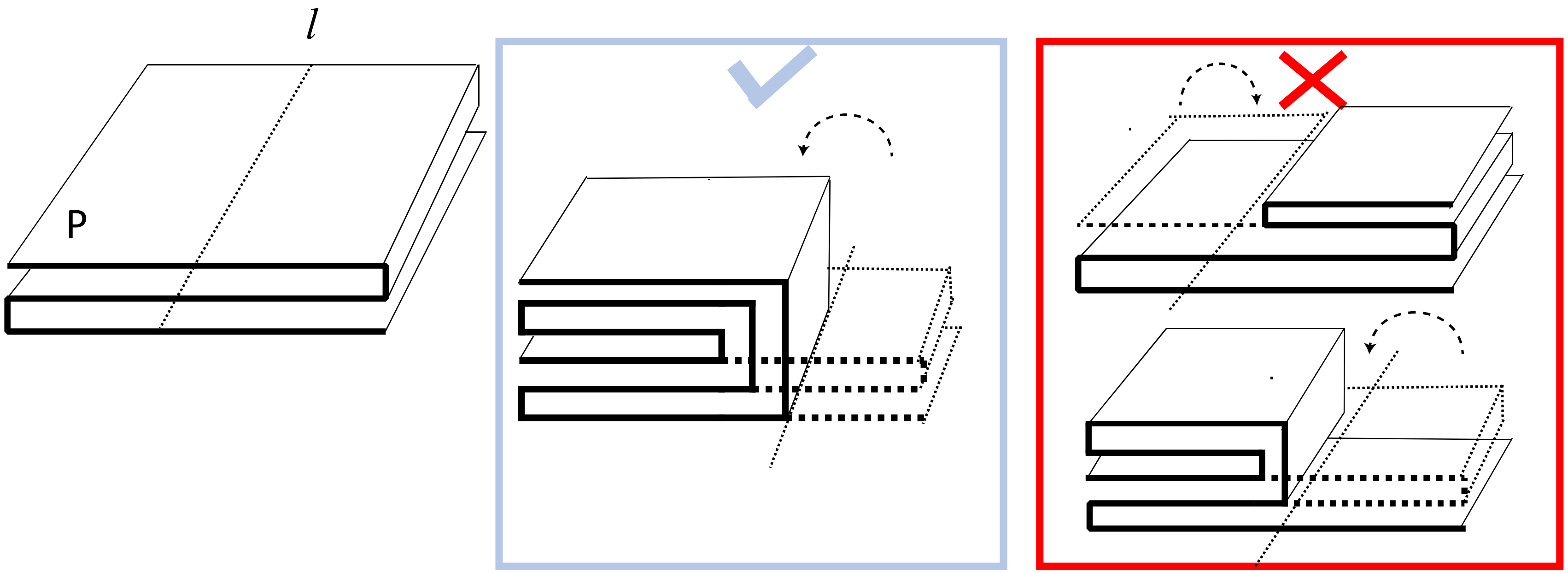
What happens if we use only "simple folding"?

See http://erikdemaine.org/foldcut or

Erik Demaine and Joseph O'Rourke. Geometric Folding Algorithms: Linkages, Origami, Polyhedra. Cambridge University Press, 2007

Simple folding

All-layer simple fold will fold through **all layers of the sheet of paper** by $\pm 180°$.



Single polygon with Simple Folding

Theorem

There is a strongly polynomial-time algorithm for determining whether a given (not necessarily convex) simple polygon P is simple-fold-and-cuttable, starting from a piece of paper.

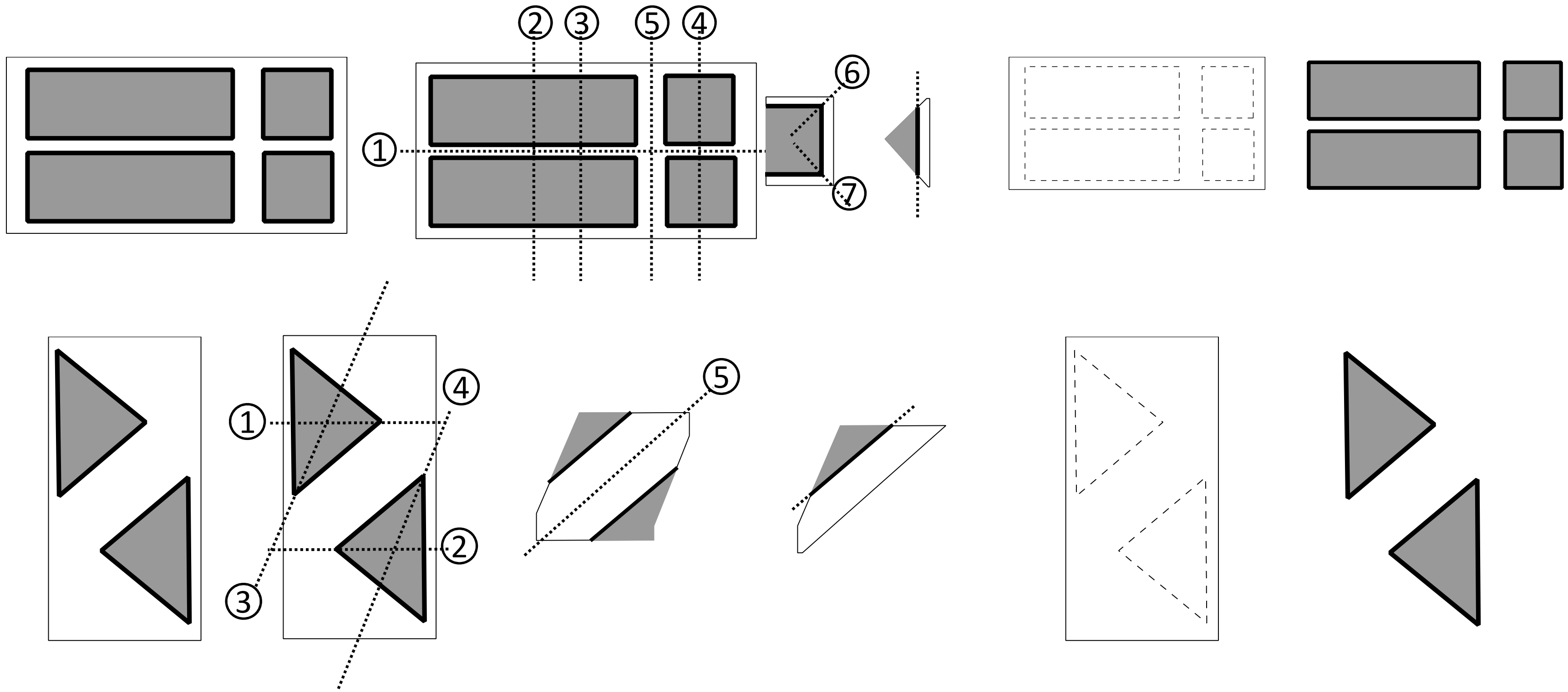
[Note] The first step is very special; P should be line symmetry.

How can we extend this problem to "many simple polygons"?

[Reference]

E. D. Demaine, M. L. Demaine, A. Hawksley, H. Ito, P. R. Loh, S. Manber, and O. Stephens. Making polygons by simple folds and one straight cut. In *Computational Geometry, Graphs and Applications,* LNCS Vol. 7033, pp. 27-43, 2011.

Many polygons: too difficult so far!

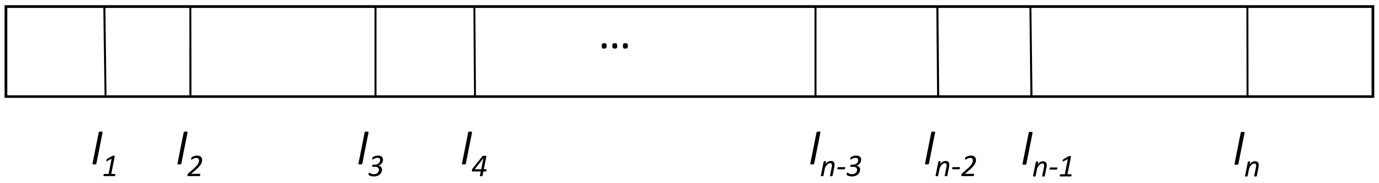


It is not easy at all! Especially, when they are close…

So we consider simpler problems…

Simpler Problems we posed

1. Line segments in parallel



- **For a given set of line segments in parallel, our goal is finding the way of overlapping all line segments (and cut).**

2. Just foldability of line segments (without cut)

- **For a given set of line segments, our goal is folding all line segments by a (shortest) sequence of simple foldings.**

[References]

•G. Hu, S.-I. Nakano, R. Uehara and T. Uno: Simple Fold and Cut Problem for Line Segments, *CCCG 2019*, pp. 158-163, 2019/08/08-10, Edmonton, Canada.

•F. Klute, I. Parada, T. Horiyama, M. Korman, R. Uehara and K. Yamanaka: Efficient Segment Folding is Hard, *CCCG 2019*, pp. 182-188, 2019/08/08-10, Edmonton, Canada.

Very rough summary!

1. **Line segments in parallel**
   - **For a given set of line segments in parallel, our goal is finding the way of overlapping all line segments (and cut).**
2. **Just foldability of line segments (without cut)**
   - **For a given set of line segments, our goal is folding all line segments by a shortest sequence of simple foldings.**

We can find the optimal way of folding in O(n^3) time by dynamic programming

It is NP-hard (intractable) for finding *n* lines in the optimal way

… and we have many unsolved problems on this topic

[References]

•G. Hu, S.-I. Nakano, R. Uehara and T. Uno: Simple Fold and Cut Problem for Line Segments, *CCCG 2019*, pp. 158-163, 2019/08/08-10, Edmonton, Canada.

•F. Klute, I. Parada, T. Horiyama, M. Korman, R. Uehara and K. Yamanaka: Efficient Segment Folding is Hard, *CCCG 2019*, pp. 182-188, 2019/08/08-10, Edmonton, Canada.

1. Simple-fold-and-cut problem for parallel line segments

We investigated the simple-fold-and-cut problem for line segments

Theorem 1: There is an algorithm for solving the simple-fold-and-cut problem for line segment when the distances are almost the same in $O(n^2)$ time and $O(n^2)$ space.

Theorem 2: There is an algorithm for solving the simple-fold-and-cut problem for line segment in general distances in $O(n^3)$ time and $O(n^2)$ space.

[Reference]

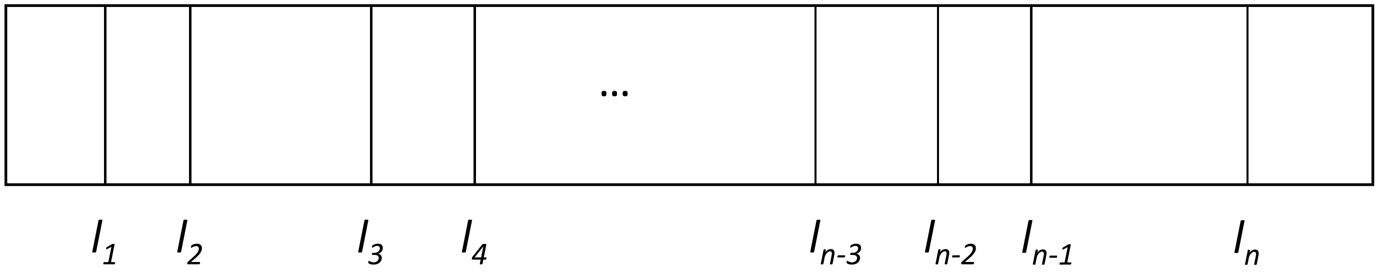
•G. Hu, S.-I. Nakano, R. Uehara and T. Uno: Simple Fold and Cut Problem for Line Segments, *CCCG 2019*, pp. 158-163, 2019/08/08-10, Edmonton, Canada.

1. Simple-fold-and-cut problem for parallel line segments

We are given a long paper strip P and a set of n parallel line segments.

All the given line segments are perpendicular to the two long edges of P

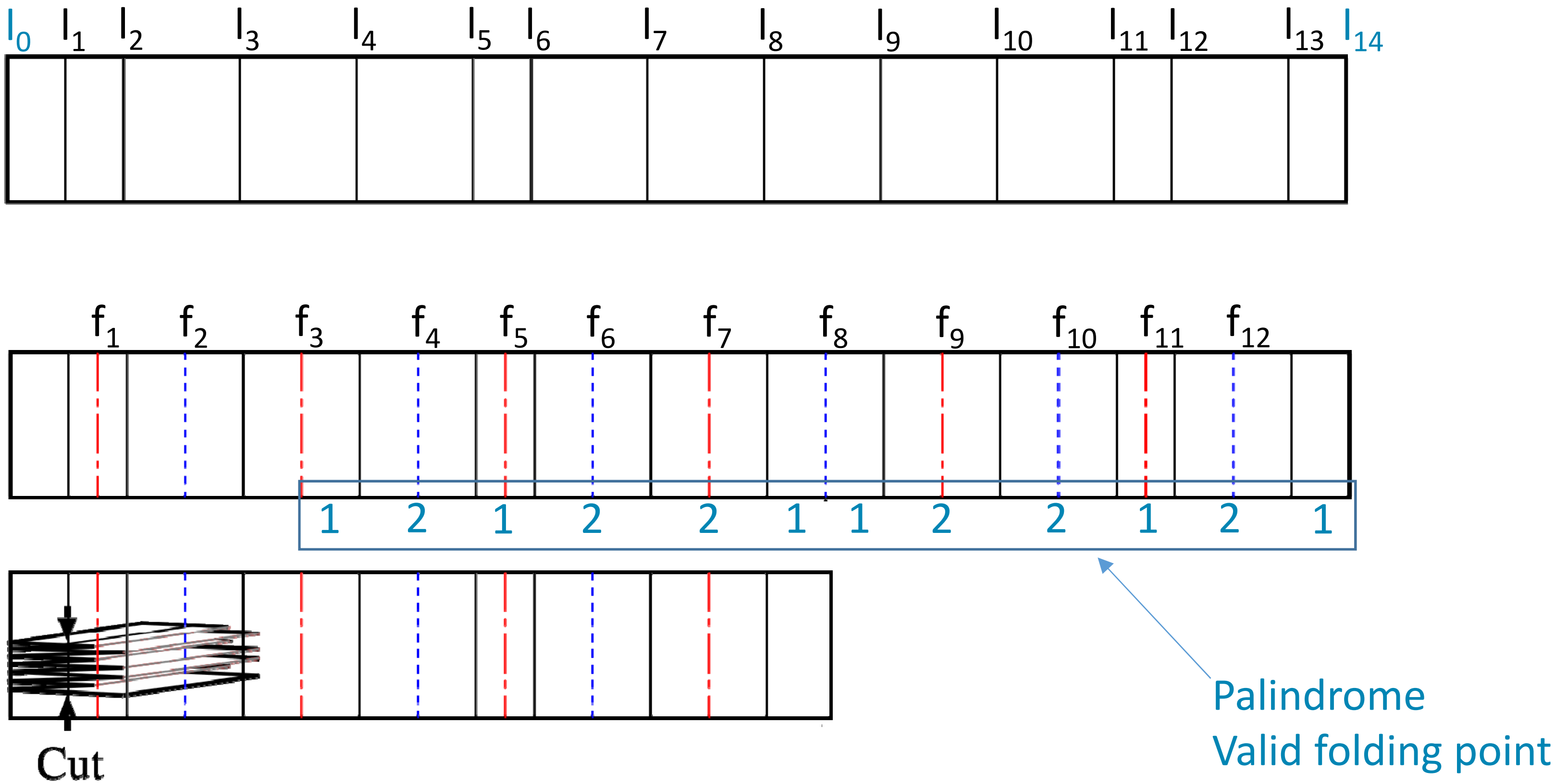
The distances between line segments are not equal



Our goal is to find the shortest sequence of all-layers simple fold to overlap all the line segments

1. Almost the same case
   $d_{min} \geqq d_{max}/2$
2. General case
   No limits for the distances

Example: Line segments



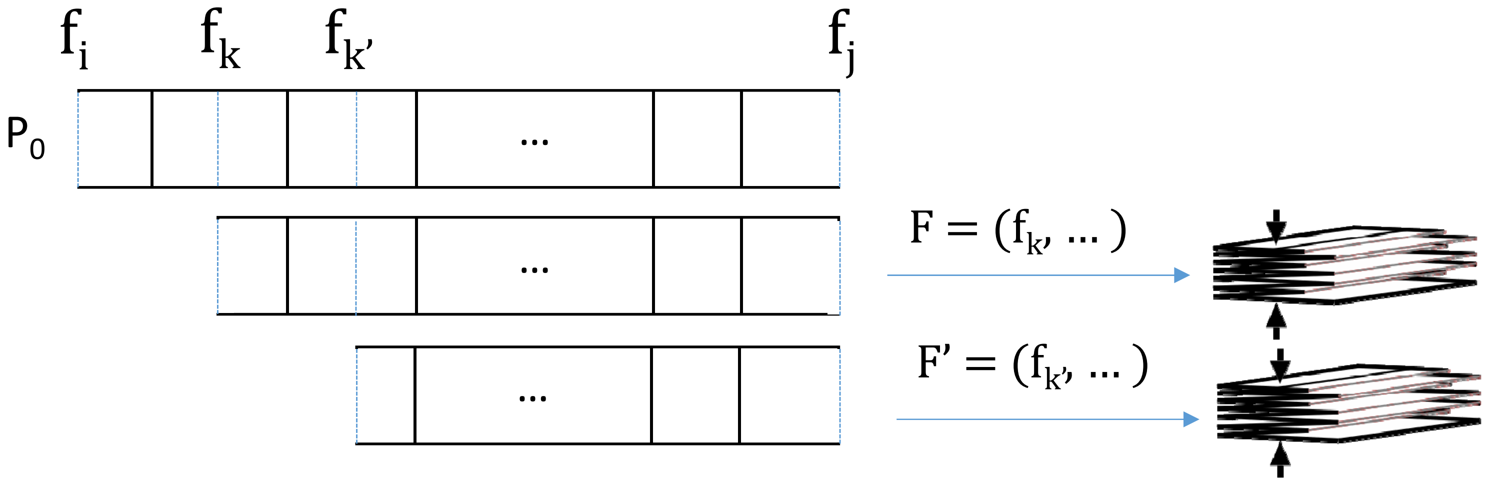
"Close to center" is better

Let P_0 be a folded state of P placed on $[f_i, f_j]$.

We assume that two simple foldings at f_k and $f_{k'}$ are both valid for some k and k' with $i < k < k' < (i+j) / 2$.

Then for any valid simple folding sequence $F = (f_k, \dots)$,

we have another valid simple folding sequence $F' = (f_{k'}, \dots)$ that is as short as F

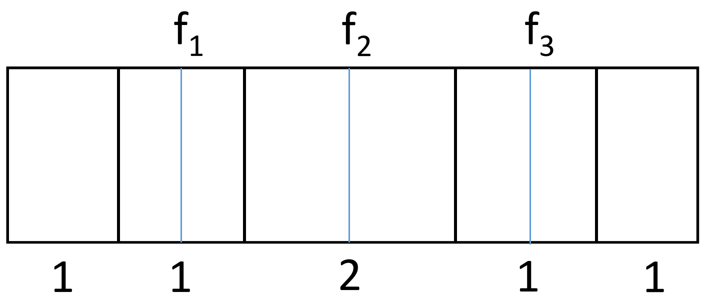


We can only check the leftmost in right half and rightmost in left half valid folding points to find the shortest folding sequence by using this lemma.

Data Structures for Our Algorithm

Array **Palindromes:**

To check valid simple folding, we use the sequence $(p_1, p_1, \ldots, p_n)$, where p_i is the length of the maximal palindrome centered at f_i for each $i = 1, \ldots, n$



P_1	P_2	p_3
1	3	1

We can use Manacher's algorithm to compute array Palindromes in $O(n)$ time and $O(n)$ space

Data Structures for Our Algorithm

Table LLINE:

For positive integers i and l, LLine$[i][l]$ indicates whether the paper segment $[l_i, l_{i+l}]$ can be folded to right along the line f_{i+l} or not.

LLine$[1][0] = 1$, from l_1 to l_{1+0} f_{1+0}

LLine$[1][1] = 1$ from l_1 to l_{1+1} f_{1+1}

LLine$[1][2] = 0$ from l_1 to l_{1+2} f_{1+2}

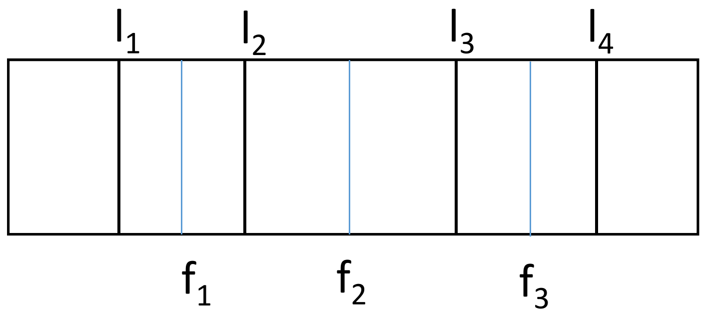


Table RLINE:

For positive integers i and l, RLine$[i][l]$ indicates whether the paper segment $[l_{i-l}, l_i]$ can be folded to left along the line f_{i-l} or not.

We can compute LLINE and RLINE in $O(n^2)$ time and $O(n^2)$ space

Data Structures for Our Algorithm

Table Fl:
For m = $\lfloor i + j \rfloor / 2$, each element Fl[i][j] gives the maximum index m in [i,m] such that LLine[i][m - i] = 1.

$\mathrm{Fl}[1][1] = 1, \mathrm{m} = 1$ $\quad$ $\mathrm{LLine}[1][0] = 1$

$\mathrm{Fl}[1][2] = 1, \mathrm{m} = 1$ $\quad$ $\mathrm{LLine}[1][1] = 1$

$\mathrm{Fl}[1][3] = 2, \mathrm{m} = 2$ $\quad$ $\mathrm{LLine}[1][2] = 0$

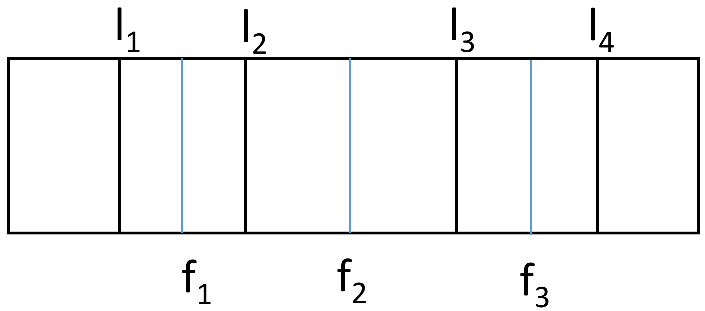


Table Fr:
For m = $\lfloor i + j \rfloor / 2$, each element Fr[i][j] gives the minimum index m in [m,j] such that LLine[j][j - m] = 1.

We can compute table Fl and table Fr in $O(n^2)$ time and $O(n^2)$ space

Data Structures for Our Algorithm

Table sf:

Each element sf[i][j] be the minimum number of simple foldings of the folded state P_0 placed on an interval $[f_i , f_i]$ of the paper strip

$$
\begin{aligned}
&\mathsf{sf}[i][j] = \infty \ \ if\ i > j,\\
&\mathsf{sf}[i][j] = 0 \ \ if\ j = i,\\
&\mathsf{sf}[i][j] = \min\{\\
&\qquad \mathsf{sf}[f_l(i,j)+1][j], \mathsf{sf}[i][f_r(i,j)-1]\\
&\qquad \} + 1 \ \ otherwise,
\end{aligned}
$$

We can compute table sf in $O(n^2)$ time and $O(n^2)$ space

Algorithm and complexity

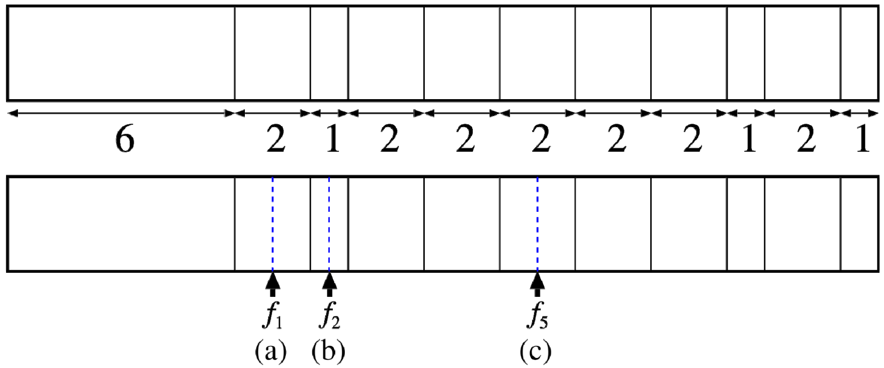
Input: $(d_0, d_1, \ldots, d_n)$
Output: Number of foldings to overlap all the line segments by using all-layer simple folding.

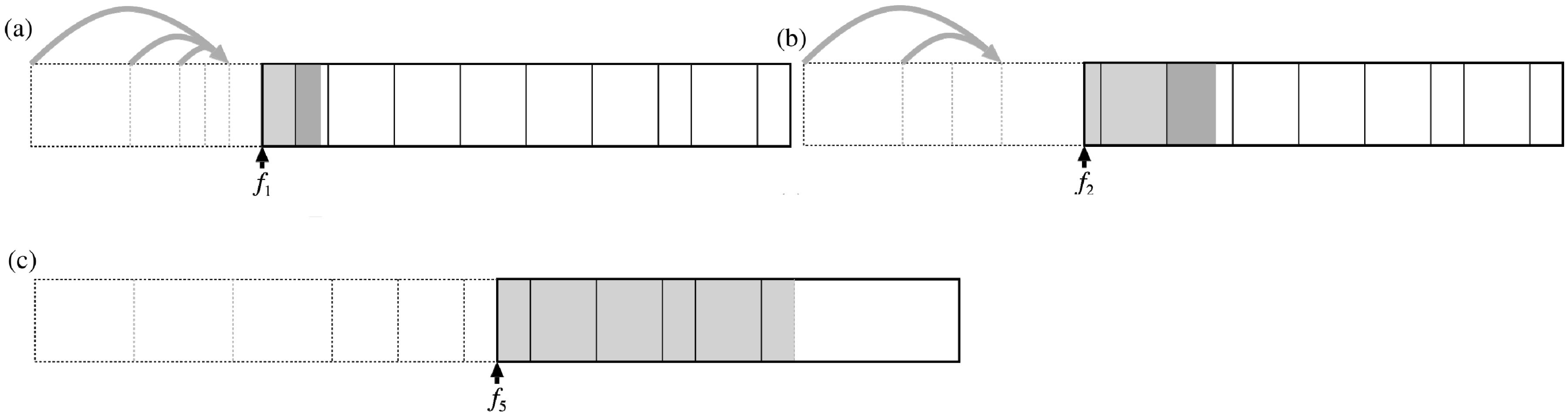
1. Compute array palindromes — O(n) time and O(n) space
2. Compute table LLINE and table RLINE — $O(n^2)$ time and $O(n^2)$ space
3. Compute table Fr and table Fl — $O(n^2)$ time and $O(n^2)$ space
4. Compute table sf — $O(n^2)$ time and $O(n^2)$ space

Totally, our algorithm for almost the same case runs in $O(n^2)$ time and $O(n^2)$ space

General case

In the general case,
we have to take care of the case that the leftmost paper segment or the rightmost paper segment covers some lines to be cut after folding





General case

For an interval $[f_i, f_j]$ of the paper strip, we define a new function $ef(d_i, dj)$

$$ef(d_i, dj) = 0 \qquad if\ di \text{ is short enough}$$

$$ef(d_i, dj) = \left\lceil \log_2\left(\frac{d_i}{2}\right) - \log_2(dj) \right\rceil \qquad 0 < i < n\ and\ \left(\frac{d_i}{2}\right) > dj$$

$$ef(d_i, dj) = \lceil \log_2(d_i) - \log_2(dj) \rceil \qquad i = 0\ or\ i = n\ (with\ j \neq 0\ and\ j \neq n)$$

For general case

“close to center” is not good any more…

$$sf[i][j] = \min\{sf[i'][j] + ef(d_i, d_{2_{i'}-i}), sf[i][j'] + ef(d_j, d_{2_{j'}-j})\} + 1$$

Our algorithm for general case runs in $O(n^3)$ time and $O(n^2)$ space

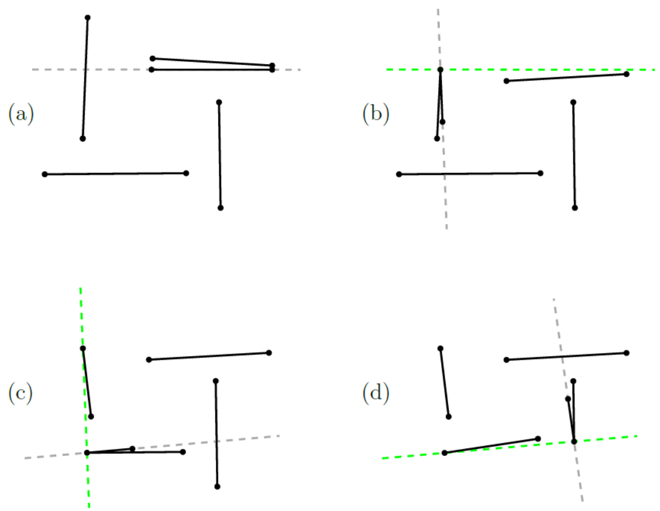
2. Foldability of line segments (without cut)

Input: Set of line segments on a sheet of paper

Output: Find a sequence of simple foldings along one of them to fold them all.

It is still quite difficult in general…

Why?



We *might* have to fold infinitely many to fold finite line segments. (Though this problem is not yet settled…)

Hardness result

So we modify the problem as follows to handle it:

Input: Set of *n* line segments on a sheet of paper

Question: Determine if there is a sequence of *n* simple foldings

along one of them to fold them all.

That is, determine if there is an ordering of line segments s.t. any folding does not cross other not-yet-folded lines.

Theorem: The problem is NP-complete.

Main result and proof

Theorem: The line segment folding problem is NP-complete.

(Key Issue) When you fold along a line, never go through other.

[Proof] Reduction from 3-SAT.

(Clause gadget)

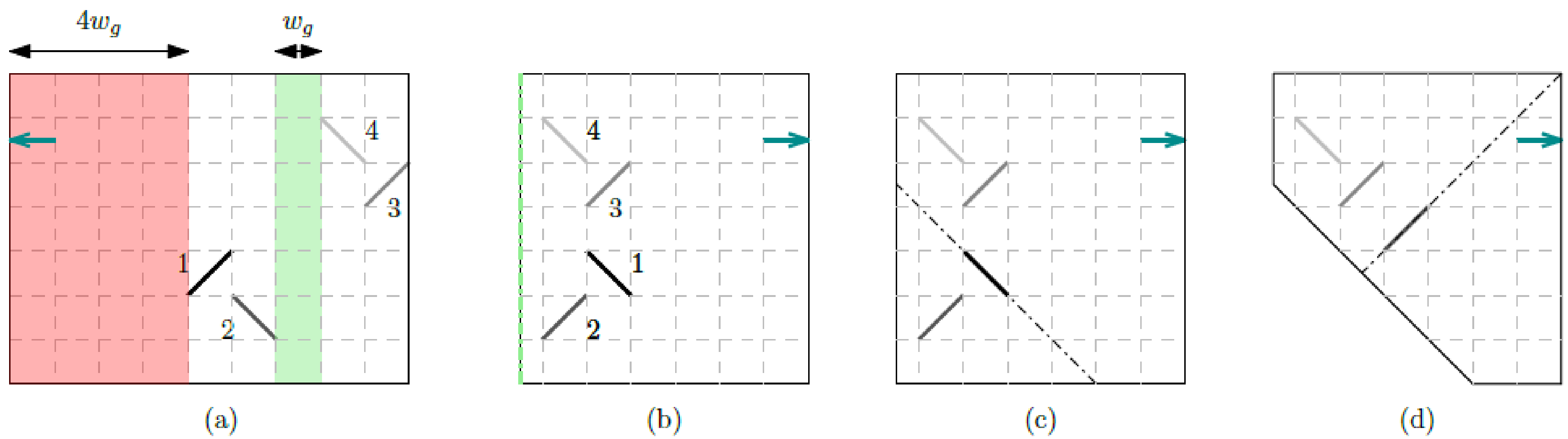


Figure 4: Clause gadget.

Main result and proof

Theorem: The line segment fol

(Key Issue) When you fold alor

[Proof] Reduction from 3-SAT.

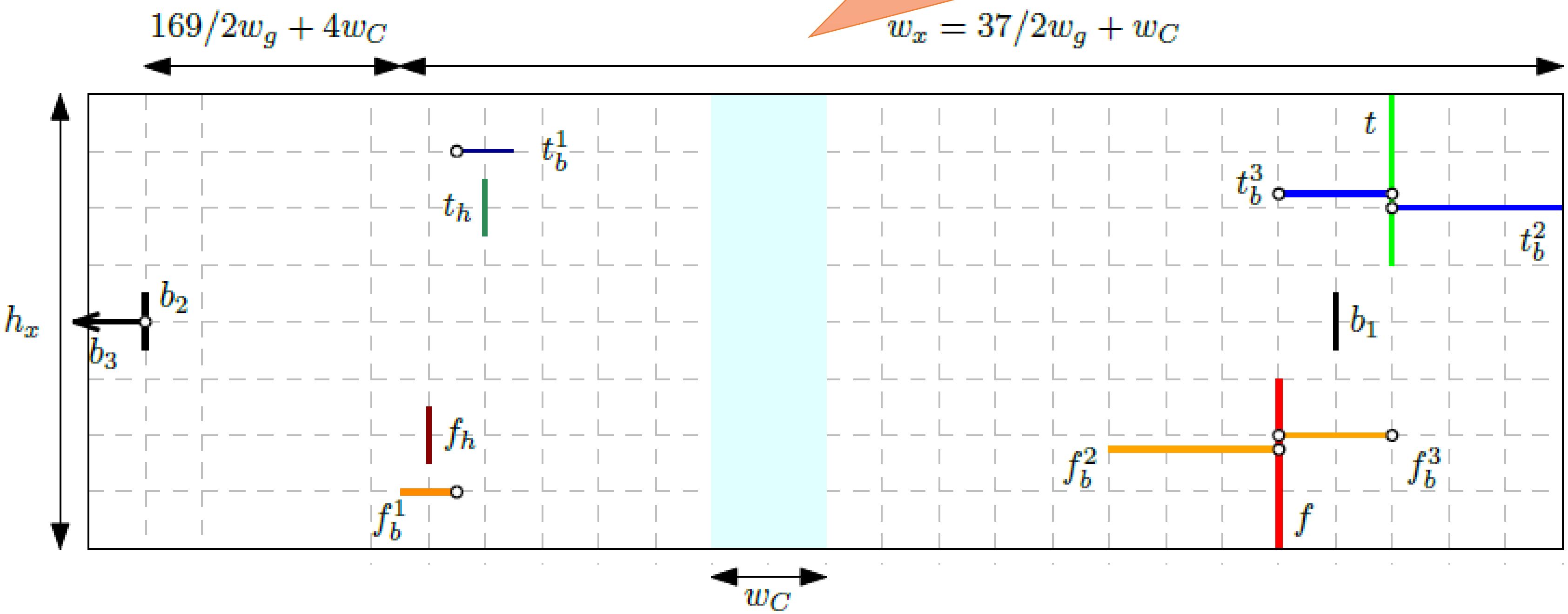
(Variable gadget)

Lemma 2: Only possible ways are

$t \to t_b^1 \to t_h \to t_b^2/t_b^3 \to f \to f_b^1 \to f_h \to f_b^2/f_b^3 \to b_1 \to b_2 \to b_3$

or

$f \to f_b^1 \to f_h \to f_b^2/f_b^3 \to t \to t_b^1 \to t_h \to t_b^2/t_b^3 \to b_1 \to b_2 \to b_3,$



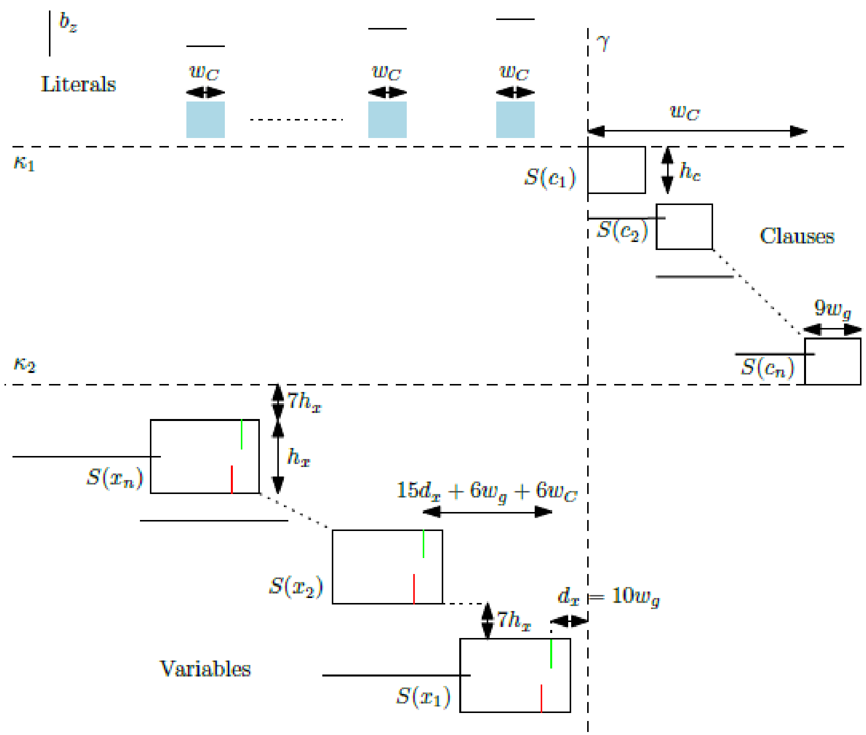
Main result and proof

Theorem: The line segment folding problem is NP-complete.

(Key Issue) When you fold along a line, never go through other.

[Proof] Reduction from 3-SAT.

(Overview)



Open problems (1)

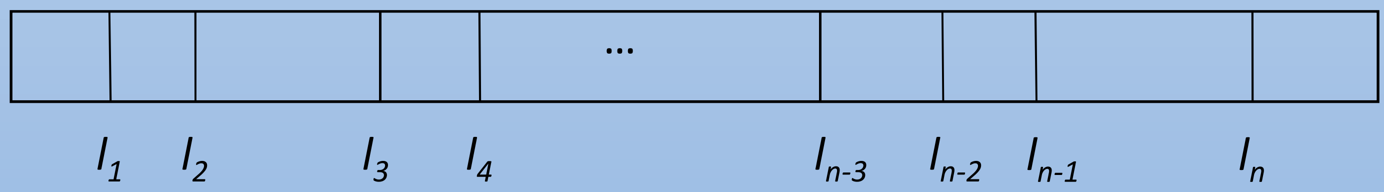
1. Simple-fold-and-cut problem (with cut):
   1. Improve the time complexity for the simple-fold-and-cut problem for line segments.
   2. On other simple folding models, we will have a different problem for the simple-fold-and-cut problem for line segments
   3. General simple-fold-and-cut problem is still open
      - We conjecture that finding a shortest sequence of simple foldings for general two-dimensional case is NP-complete

Open problems (2)

2. Simple-fold problem (without cut):
   1. The number of simple folding can be infinite?
   2. Is there any reasonable and nontrivial restriction that the problem is polynomial-time solvable?

- for parallel line segments?

Input: Parallel line segments



Output: Finding the way of overlapping all line segments (without cut, i.e., we can overlap the other paper).

Computational ORIGAMI=

Geometry＋Algorithm＋Computation

- Mathematics
- Theoretical Computer Science
- Real High Performance Computing

- Many Applications from micro-size to space-size
  - Bioinfomatics (e.g., DNA folding),
  - Robotics, packaging,
  - Architecture

Let’s join it!

- Many young researchers;
  - even undergrad students, highschool students!