Flattening fixed-angle chains is strongly NP-hard

Citation	Demaine, Erik D., and Sarah Eisenstat. “Flattening Fixed-Angle Chains Is Strongly NP-Hard.” Algorithms and Data Structures. Ed. Frank Dehne, John Iacono, & Jörg-Rüdiger Sack. LNCS Vol. 6844. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. 314–325.
As Published	http://dx.doi.org/10.1007/978-3-642-22300-6_27
Publisher	Springer Berlin / Heidelberg
Version	Author's final manuscript
Accessed	Sun Oct 21 08:32:48 EDT 2018
Citable Link	http://hdl.handle.net/1721.1/73923
Terms of Use	Creative Commons Attribution-Noncommercial-Share Alike 3.0
Detailed Terms	http://creativecommons.org/licenses/by-nc-sa/3.0/

Flattening Fixed-Angle Chains Is Strongly NP-Hard

Erik D. Demaine⋆ and Sarah Eisenstat⋆

MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA
{edemaine,seisenst}@mit.edu

Abstract. Planar configurations of fixed-angle chains and trees are well studied in polymer science and molecular biology. We prove that it is strongly NP-hard to decide whether a polygonal chain with fixed edge lengths and angles has a planar configuration without crossings. In particular, flattening is NP-hard when all the edge lengths are equal, whereas a previous (weak) NP-hardness proof used lengths that differ in size by an exponential factor. Our NP-hardness result also holds for (nonequilateral) chains with angles in the range $[60^\circ - \varepsilon, 180^\circ]$, whereas flattening is known to be always possible (and hence polynomially solvable) for equilateral chains with angles in the range $(60^\circ, 150^\circ)$ and for general chains with angles in the range $[90^\circ, 180^\circ]$. We also show that the flattening problem is strongly NP-hard for equilateral fixed-angle trees, even when every angle is either 90° or 180°. Finally, we show that strong NP-hardness carries over to the previously studied problems of computing the minimum or maximum span (distance between endpoints) among non-crossing planar configurations.

Keywords: geometric folding, linkages, hardness, polymers

1 Introduction

Molecular geometry (also called *stereochemistry*) studies the 3D geometry of the atoms (and the bonds between them) that constitute a molecule [5]. If we represent an atom by a vertex and a bond by an edge, we obtain a graph structure; this structure comes equipped with fixed edge (bond) lengths, making a *linkage*, and fixed (bond) angles between incident edges, making a *fixed-angle linkage*. In general, a *fixed-angle linkage* is a geometrically embedded graph that can reconfigure (change embedding) so long as it preserves the fixed edge lengths and angles [4]. Typical edge (bond) lengths in polymers are 100–270 picometers, and typical (bond) angles are around 72°, 90°, 109°, 120°, and 180°.

Most large (macro)molecules are *polymers*, and many are *nonbranching polymers*, meaning that the graph structure decomposes into a chain of substructures of small size. Examples of nonbranching polymers include proteins, DNA

⋆ Research supported in part by NSF grant CDI-0941538.

strands, and RNA strands. Motivated by this reality, the computational study of fixed-angle linkages [4] usually focuses on *fixed-angle chains*, where the graph is a path (representing the *backbone* of the polymer), and on *fixed-angle trees*, where the graph is a tree, especially caterpillars, representing a small amount of additional structure attached to the backbone.

Motivated by these applications to polymer science, Soss and Toussaint [8,9] introduced several computational problems, three of which we study here:

Flattening: Given a fixed-angle linkage, decide whether it has a continuous non-crossing motion that results in a flat configuration (lying in the plane).
Min flat span: Compute the flat configuration of a fixed-angle linkage with minimum possible *span*—distance between the two endpoints.
Max flat span: Compute the flat configuration of a fixed-angle linkage with maximum possible span.

They proved that all three of these problems are weakly NP-hard, by reducing from the integer partition problem. Because the integer partition problem is weakly NP-hard, it is only hard when the numbers in the problem are exponentially large. Therefore, the reductions given by Soss and Toussaint show only that the flattening and span problems are hard when the edge lengths differ by exponential factors, or when the angles have polynomially many bits of precision. Neither of these assumptions is realistic in the case of polymers: the edge lengths are all within a factor of 2 or 3 of each other, and the geometric models (and hence the angles) have some small imprecision (from other forces and quantum imprecision).

Our results. In this paper, we prove that all three problems are strongly NP-hard, and thus hard when the edge lengths are all very close (or even identical) and a constant number of different angles are used. More specifically, we prove the following special cases to be strongly NP-hard:[1]

Problem	Linkage	Edge lengths	Angle range	Theorem
Flattening	fixed-angle chain	equilateral	$[16.26°, 180°]$	4
Flattening	fixed-angle chain	$\Theta(1)$	$[60-\varepsilon°, 180°]$	5
Flattening	fixed-angle caterpillar tree	equilateral	$\{90°, 180°\}$	3
Min flat span	fixed-angle chain	equilateral	$[16.26°, 180°]$	6
Max flat span	fixed-angle chain	equilateral	$[16.26°, 180°]$	7

The $16.26° \approx \arcsin \frac{7}{25}$ angle bound can easily be improved to around $22.6°$, and perhaps further to $30°$ or $45°$.

Overview. Our proofs start in Section 3 with an artificial problem, flattening *semi-rigid* fixed-angle chains, as a building block for the more interesting results above. In a semi-rigid chain, some sections of the chain can be marked *rigid*, meaning that the vertices in the section cannot move relative to each other.

[1] A linkage is *equilateral* if all edge lengths are equal. "$\Theta(1)$" denotes that all edge lengths are within constant factors of each other.

Naturally, this additional set of constraints makes flattening more difficult, and we show that the problem is NP-hard even for equilateral chains with all angles in $\{90°, 180°\}$. Then we show in Section 4 how to remove the semi-rigidity constraint using either sharper angles or fixed-angle trees. Finally, in Section 5, we show how to transform the flat state into one with an especially small or large span, and guarantee that the chain has a flat state in all cases (of possibly suboptimal span).

2 Definitions

2.1 Linkages

Definition 1. *A* linkage *consists of a graph* $G = (V, E)$ *and edge lengths* $\ell : E \to \mathbb{R}_{\geq 0}$. *$G$ is called the* structure graph *of the linkage. A* configuration *of a linkage in d dimensions is a mapping* $C : V \to \mathbb{R}^d$ *satisfying the constraint* $\ell(u, v) = \|C(u) - C(v)\|$ *for each edge* $(u, v) \in E$. *A configuration is* non-crossing *if any two edges* $e_1, e_2 \in E$ *intersect only if the two edges are incident in the structure graph, and intersect only at their shared vertex.*

Definition 2. *A* fixed-angle linkage *is a linkage with an additional set of constraints specifying an angle function* $\theta_i : \mathcal{N}(v_i) \times \mathcal{N}(v_i) \to [0°, 180°]$ *for each vertex v_i, where $\mathcal{N}(v_i)$ is the set of neighbors of v_i. In addition to satisfying the length constraints of the linkage, any configuration of the linkage has the property that for each vertex $v_i \in V$ and each pair of its neighbors $v_j, v_k \in \mathcal{N}(v_i)$, the angle $\angle v_j v_i v_k$ has measure $\theta_i(v_j, v_k)$.*

Definition 3. *A* chain *of length n is a linkage whose structure graph is* $G = (V, E)$ *where* $V = \{v_1, v_2, \ldots, v_n\}$ *and* $E = \{(v_1, v_2); (v_2, v_3); \ldots; (v_{n-1}, v_n)\}$.

Definition 4. *A linkage is* equilateral *if $\ell(e) = 1$ for all $e \in E$.*

Definition 5. *A fixed-angle linkage is* orthogonal *if all angles $\theta_i(v_j, v_k)$ are either $90°$ or $180°$.*

Definition 6. *A* flat state *of a fixed-angle linkage is a non-crossing 2D configuration of the linkage. A 3D configuration of a fixed-angle chain can be* flattened *if there exists a continuous sequence of non-crossing configurations starting at the current configuration and ending in a flat state.*

Definition 7. *The* span *of a flat state of a fixed-angle chain is the distance between v_1 and v_n in that configuration.*

Finally, we define a new kind of fixed-angle chain, which places an additional constraint on the locations of the vertices in a configurations.

Definition 8. *A* semi-rigid chain *of length n is a fixed-angle chain of length n with constraints to ensure that parts of the chain are rigid. These constraints are specified in two parts: a sequence $s_0 < s_1 < \ldots < s_\ell$ such that $s_0 = 1$ and $s_\ell = n$; and the distance functions $d_1, \ldots, d_\ell$, where each d_i gives all pairwise distances between the vertices $\{v_{s_{i-1}}, v_{s_{i-1}+1}, \ldots, v_{s_i}\}$. The* articulation points *of a semi-rigid chain are the vertices $v_{s_0}, v_{s_1}, \ldots, v_{s_\ell}$.*

The additional restrictions imposed by the semi-rigid chain make it easier to prove that flattening the chain is NP-hard. The relationship between semi-rigid chains and fixed-angle chains makes it possible to give a reduction from one to the other.

2.2 Rectilinear Planar Monotone 3-SAT

One variant of the standard 3-SAT problem is *planar 3-SAT*, where the graph of the variables and clauses, with edges between variables and the clauses that contain them, has a planar embedding. Planar 3-SAT is known to be NP-complete [7]. One variant of planar 3-SAT, *rectilinear planar 3-SAT*, places three additional restrictions on the planarity of the graph:

1. All variables and clauses are rectangles.
2. All of the variables lie along a single horizontal line.
3. All edges lie along vertical lines.

Rectilinear planar 3-SAT is also known to be NP-complete [6]. In 2010, de Berg and Khosravi introduced an even more restricted version of rectilinear planar 3-SAT [3]: an instance of the *rectilinear monotone planar 3-SAT problem* is a rectilinear planar 3-SAT instance such that every clause is either all positive or all negative, all positive clauses lie above the line of variables, and all negative clauses lie below the line of variables. They proved the following theorem:

Theorem 1. *It is NP-complete to decide whether an instance of rectilinear monotone planar 3-SAT is satisfiable.*

3 Flattening Semi-Rigid Chains

In this section, we begin by constructing gadgets for a semi-rigid chain which have a limited number of flat states. Then in Theorem 2, we use those gadgets to show that it is NP-hard to find a flat state for an equilateral orthogonal semi-rigid chain.

Lemma 1. *Up to reflection, the semi-rigid chain depicted in Figure 1(a) has three possible flat states, depicted in Figures 1(a), 1(b), and 1(c).*

Lemma 2. *Given the location of the section of chain between a_0 and a_1, each flat state of the semi-rigid chain depicted in Figure 2 has the following properties:*

1. *The point a_{17} has coordinates $(3, 0)$.*
2. *The y-coordinate of at least one of b_1, b_2, or b_3 must be negative.*

Lemmas 1 and 2 can both be proved by case analysis. We now use the results of Lemma 2 to show the following theorem.

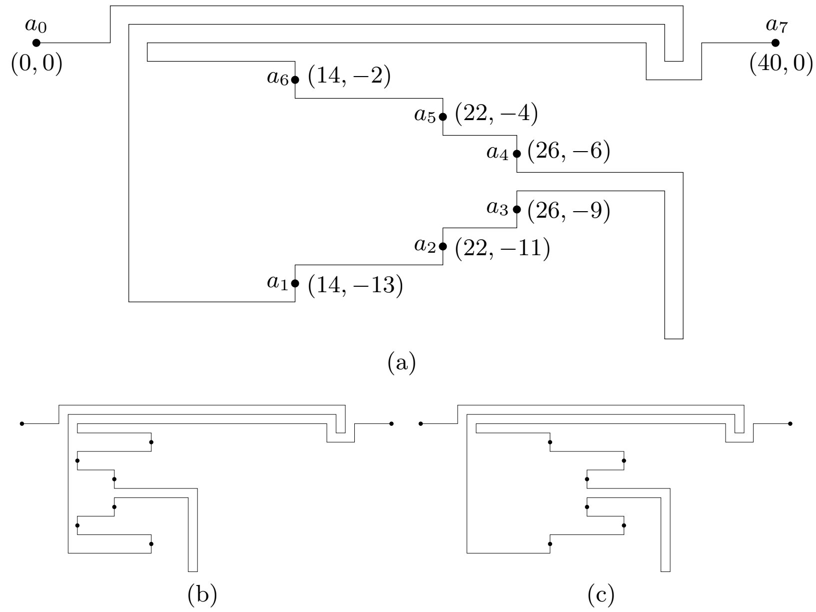


Fig. 1. The three possible flat states for the semi-rigid chain given in Fig. 1(a). Each labeled point is an articulation point; all other sections of the chain are rigid.

Theorem 2. *There exists a polynomial-time algorithm reducing from an instance ϕ of rectilinear planar monotone 3-SAT to an orthogonal equilateral semi-rigid chain which can be flattened if and only if ϕ is satisfiable.*

Proof. The *pins* of the clause gadget depicted in Fig. 2 are the rigid chains between between articulation points a_8 and a_9, between a_{11} and a_{12}, and between a_{14} and a_{15}. Lemma 2 shows that all possible flat states for that clause gadget have the property that at least one of b_1, b_2, or b_3 must lie below a certain line. A clause has the property that at least one of its literals must be true. So to set up the reduction from one to the other, our literals should be pieces of a semi-rigid chain such that, if the literal is false, the chain will intersect with the corresponding pin when it protrudes below the line. That way, if there is a flat state, then at least one of the literals for that clause must be true. We will accomplish this using gadgets like those depicted in Fig. 3. If the pin extends below the line, then the literal gadget must also dip below the line. If the pin does not extend below the line, then the literal gadget can go either way.

Because we are reducing from monotone rectilinear planar 3-SAT, we know that each clause will contain either all negative or all positive literals, and that all positive clauses will lie above the variables while all negative clauses will lie below the variables. Our choice of gadget for the literal allows us to construct a clause involving the literal's negation by mirroring a clause gadget over the horizontal line and making the pins point upwards instead of downwards.

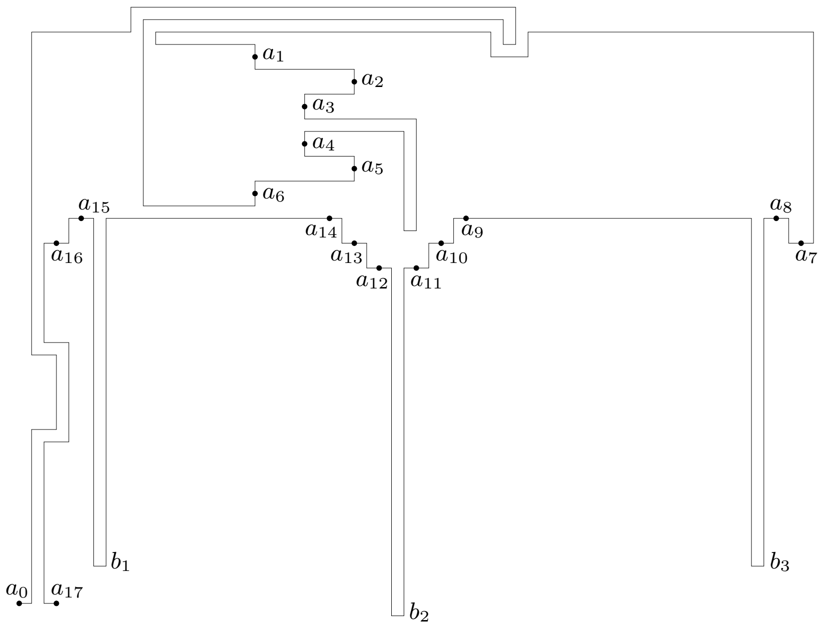


Fig. 2. The semi-rigid chain used for a clause gadget. The articulation points are $a_0, \ldots, a_{17}$. The coordinates of all labelled points are given in Table 1. The parameter h adjusts the height of the gadget; the parameters w_1 and w_2 adjust the distances between the points b_1, b_2, and b_3.

Unfortunately, there are two problems with the idea we have sketched. The first is direction. In the rigid chain which is partially depicted in Fig. 3, it would be equally valid to have a flat state where the clause gadget is mirrored across the line so that its pins point up. If there is only one clause gadget, then we may say without loss of generality that the clause gadget will fall above the line. However, as soon as there is more than one clause gadget, we may have to consider the possibility of flat states where one clause gadget is in the right position while the other is in the wrong position. Hence, we need a way to make sure that each clause gadget extends in the right direction. The second problem we must consider is consistency. In order to correctly convert from the rectilinear structure to our fixed-angle chain, we must be able to have a clause gadget appear in between two literal gadgets for the same variable. But if the two literal gadgets are independent, there is no way to ensure that the two gadgets will take on the same value.

The modification we make will solve both of these problems. We will have three separate sections of the chain running parallel to each other. The chain in the middle will consist of a number of long variable gadgets, one for each variable in the original formula. The chain above the middle chain will contain the clause gadgets for all of the positive clauses, as well as smaller literal gadgets. This will ensure that each clause gadget must extend above the chain; if it extended below,

Table 1. The coordinates for the labeled points in Fig. 2.

label	x-coord	y-coord	label	x-coord	y-coord	label	x-coord	y-coord
a_0	0	0	a_7	$w_1 + w_2 + 10$	h	a_{14}	$w_1 + 1$	$h + 2$
a_1	$w_1 - 5$	$h + 15$	a_8	$w_1 + w_2 + 8$	$h + 2$	a_{15}	5	$h + 2$
a_2	$w_1 + 3$	$h + 13$	a_9	$w_1 + 12$	$h + 2$	a_{16}	3	h
a_3	$w_1 - 1$	$h + 11$	a_{10}	$w_1 + 10$	h	a_{17}	3	0
a_4	$w_1 - 1$	$h + 8$	a_{11}	$w_1 + 8$	$h - 2$	b_1	7	3
a_5	$w_1 + 3$	$h + 6$	a_{12}	$w_1 + 5$	$h - 2$	b_2	$w_1 + 7$	-1
a_6	$w_1 - 5$	$h + 4$	a_{13}	$w_1 + 3$	h	b_3	$w_1 + w_2 + 7$	3

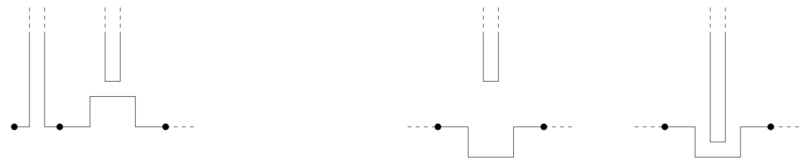


Fig. 3. The interaction between the pins in the clause gadget and the gadgets used for the literals.

it would intersect with the chain in the middle. The chain below the middle chain will contain the clause gadgets for all of the negative clause gadgets, as well as a number of smaller literal gadgets. A sample of this is depicted in Fig. 4. We will connect the three chains as depicted in Fig. 5.

We say that a variable is true if the long gadget for that variable in the middle chain dips below the center line of the middle chain; the variable is false otherwise. Hence, if a variable is false, then all of the smaller gadgets for that variable in the top chain must rise above the center line. So a clause containing positive literals cannot lower one of its pins for a variable which is false. If on the other hand a variable is true, then any smaller gadgets for that variable in the bottom chain must also dip below the center line. Hence, any clause with all negative literals cannot raise the pin for a variable which is true. In other words, for any clause, the pin which is lowered (or raised, depending on the clause type) cannot correspond to a false literal. So the only way to get a non-intersecting flat state is to have at least one true literal in each clause. □

4 Flattening Fixed-Angle Chains and Trees

In this section, we give reductions from semi-rigid chains to several kinds of fixed-angle linkages. In Theorem 3, we provide a reduction to orthogonal equilateral fixed-angle trees. In Theorem 4, we provide a reduction to equilateral fixed-angle chains with minimum angle $\arcsin \frac{7}{25} \approx 16.26°$. In Theorem 5, we provide a reduction to general fixed-angle chains with edge lengths $\Theta(1)$ and angles $> 60° - \varepsilon$.

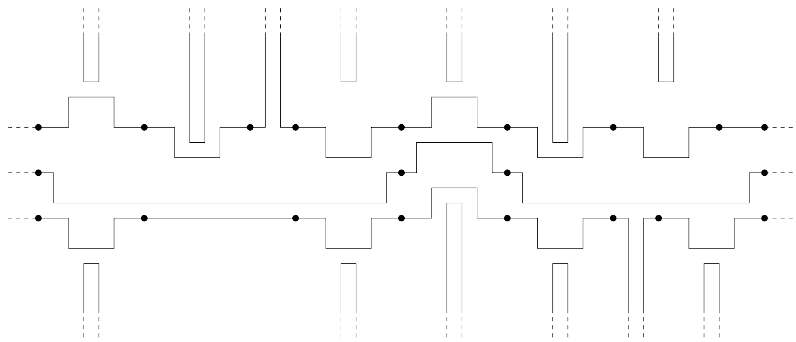


Fig. 4. A sample of the three pieces of the semi-rigid chain which will be used for the reduction from rectilinear planar monotone 3-SAT.

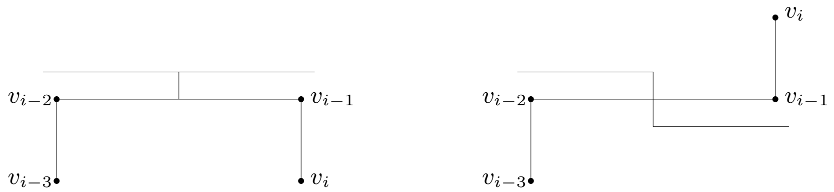
Fig. 5. The way in which we connect the three semi-rigid chains from Fig. 4. Once the locations of the top and bottom chains are fixed, we know that clause gadgets will protrude upwards from the top chain and downwards from the bottom chain. So the only possible location for the middle chain is between the two other chains.

Theorem 3. *There exists a polynomial-time algorithm which takes as input an orthogonal equilateral semi-rigid chain, and outputs an orthogonal equilateral fixed-angle tree that can be flattened if and only if the semi-rigid chain can be flattened.*

Proof. The first step in the conversion process is to merge adjacent edges within the same rigid piece which have an angle of 180° between them. This means that our semi-rigid chain is no longer equilateral, and instead has integer lengths which are between 1 and n, where n is the length of the original chain. We then scale up our semi-rigid chain by a factor of 6. Our goal is to replace the rigidity constraints of the original chain with some new structure.

Say that points $v_{i-3}, v_{i-2}, v_{i-1}$, and v_i all lie within the same rigid piece. If we have the locations of v_{i-1} and v_{i-2}, then there are two possible locations for v_i. To determine which location is correct, it is sufficient to know whether v_{i-3} lies above or below the line between v_{i-1} and v_{i-2}. Hence, to impose the rigidity constraints, it is sufficient to create two types of local gadgets: one gadget which can only be flattened if v_{i-3} and v_i lie on the same side of the line between v_{i-1} and v_{i-2}; and one gadget which can only be flattened if v_{i-3} and v_i lie on different sides of the line between v_{i-1} and v_{i-2}. Those gadgets are depicted in Fig. 6. We attach each gadget halfway down the edge between v_{i-2} and v_{i-1}.

Any flat state of the fixed-angle tree can be converted to a flat state of the original semi-rigid chain by removing the new gadget edges. Each gadget can be



(a) To keep v_i and v_{i-3} on the same side of the line between v_{i-1} and v_{i-2}.

(b) To keep v_i and v_{i-3} on different sides of the line between v_{i-1} and v_{i-2}.

Fig. 6. The gadgets used in Theorem 3.

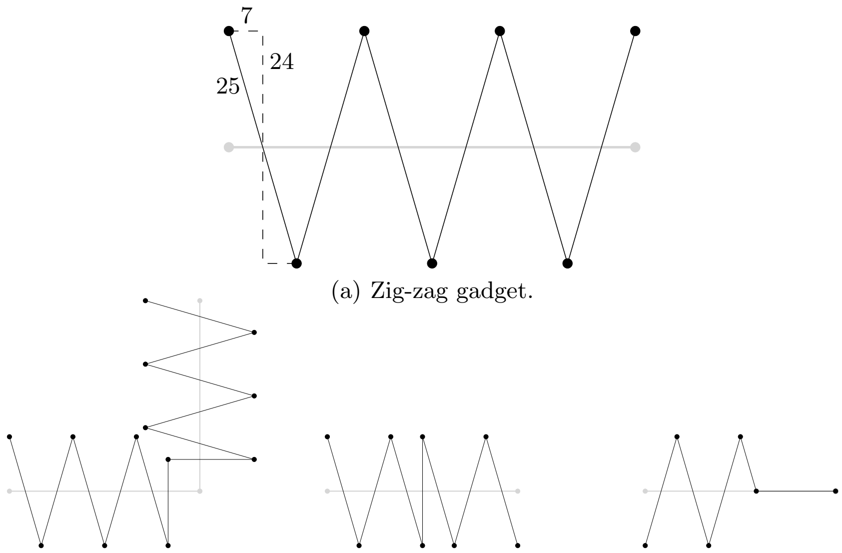
thought of as thickening the edge between v_{i-1} and v_{i-2}, because the gadgets can lie above or below the edge. The original edges were infinitely thin; the new edges have thickness 4. Because each edge was scaled up by a factor of 6, any non-crossing flat state of the original semi-rigid chain will not become self-intersecting when the gadgets are added. Therefore, any flat state of the original semi-rigid chain is a flat state of the fixed-angle tree we have created. □

Theorem 4. *There exists a polynomial-time algorithm which takes as input an orthogonal equilateral semi-rigid chain, and outputs an equilateral fixed-angle chain such that each flat state of one chain corresponds to a flat state of the other chain, and the spans differ by a fixed constant factor c.*

Proof. We begin by replacing each edge in the original semi-rigid chain with three edges connected with a fixed angle of 180°. Next, we introduce several types of gadgets, each of which can be used to replace a section of the semi-rigid chain. The first gadget is used to replace any interior edge in a rigid piece which has fixed-angle 180° with the edges on either side. The gadget we use will zig-zag across the original location of the edge, as depicted in Fig. 7(a). Each edge in the depicted gadget has length 50, so we can consider each such edge to be a sequence of 50 smaller equilateral edges.

The second gadget used is known as the turn gadget, which is depicted in Fig. 7(b). It is used to cause the zig-zag to turn by a total of 90°. The depicted flat state for the turn gadget is the only possible flat state, barring reflection of the whole gadget. If the turn gadget is connected to a zig-zag with a fixed-angle of $2\arcsin(7/25)$, then the direction that the zig-zag goes in (that is, whether the final point in the zig-zag is up or down) determines the direction of rotation for the turn gadget.

If we were to use only zig-zags and turn gadgets, the result would be a spiral, because each turn gadget would cause a rotation in the same direction. So in order to allow us to switch directions, we use the gadget depicted in Fig. 7(c), which is known as a switch gadget. When a switch gadget is used, it changes the direction of the zig-zag. This means that when the next turn gadget is used, the turn will go in the opposite direction to previous turns. Because we scaled up the original chain, there will always be room to place a switcher gadget between adjacent turns.



(a) Zig-zag gadget.

(b) Turn gadget. (c) Switch gadget. (d) Articulation gadget.

Fig. 7. The gadgets used for the proof of Theorem 4.

Together, these three gadgets ensure that if the first three points in our chain are fixed, then there is only one way to arrange the rest of the chain. This lets us enforce the rigidity constraints for each rigid piece of the original semi-rigid chain. To join these rigid pieces together, we use the articulation gadget depicted in Fig. 7(d). The articulation gadget has only one possible flat state, barring reflection. It is used to replace the edges adjacent to an articulation point. The right half of the articulation gadget lies in the same location as the end of the replaced edge. Therefore, when we apply the fixed-angle constraint from the original semi-rigid chain, it places the correct restriction on the angle between the two rigid pieces. In addition, the use of this gadget for each articulation point (including the ends of the original chain) means that when we transform a flat state of the original semi-rigid chain to a flat state of the new fixed-angle chain, the distances between articulation points will be scaled up by a constant factor.

Each of these gadgets replaces a single edge of length 1 with a gadget whose flat state has length 84 and width 48. Just as in Theorem 3, the fact that we scaled up the original chain by a factor of 3 means that the substitution of these gadgets for the original edges of the tree will not create intersections. □

In 2002, Aloupis et al. showed that every fixed-angle chain with angles between 90° and 180° has a canonical flat state [1]. In 2006, Aloupis and Meijer showed that every equilateral fixed-angle chain with angles strictly between 60° and 150° has a canonical flat state [2]. We have shown that it is NP-hard to compute a flat state for some equilateral fixed-angle chains with angles between 16.26° and 180°. This naturally leads to the question of how large the minimum

(a) For the min span problem.

(b) For the max span problem.

Fig. 8. How to connect the three semi-rigid chains from Theorem 2 when reducing to the minimum and maximum span problems. Note the new articulation point a_i.

angle can be while still ensuring that flattening is NP-hard. In our next result, we show that it is NP-hard to compute a flat state for some fixed-angle chains with angles between $\theta < 60°$ and $180°$. This result does not use equilateral chains, but all edges used in this reduction have length $\Theta(1)$.

Theorem 5. *Given any constant $\theta < 60°$, there exists a polynomial-time algorithm which takes as input an orthogonal equilateral semi-rigid chain, and outputs a fixed-angle chain with minimum angle $\geq \theta$ that can be flattened if and only if the semi-rigid chain can be flattened.*

Proof. Insert proof sketch here. $\square$

5 Flat Span

In this section, we adapt the proof of Theorem 2 to show the NP-hardness of the related problems of minimum and maximum flat span.

Theorem 6. *There exists a polynomial-time algorithm which takes as input a rectilinear planar monotone 3-SAT instance ϕ, and outputs an equilateral fixed-angle chain and a distance d such that the minimum span of the chain in any flat state is less than d if and only if ϕ is satisfiable.*

Proof. In Theorem 2, we saw a reduction that involved constructing three separate chains and connecting them as in Fig. 5. For this reduction, we connect the same three chains as depicted in Fig. 8(a). In the depicted flat state, which is non-crossing if and only if ϕ is satisfiable, a_0 is at $(0, 0)$, a_i is at $(w + 2, -1)$, and a_n is at $(w, 3)$, so the span is $\sqrt{w^2 + 9}$. There are two other flat states, both of which can be made non-crossing regardless of whether ϕ is satisfiable. In the first flat state, we reflect the middle chain over the articulation point a_i, which moves a_0 to coordinates $(2w + 4, 0)$. In the second flat state, we flip the middle chain over the articulation point a_i and then over the articulation point a_j, which moves a_0 to coordinates $(2w + 4, -6)$. The span of either flat state will be $> \sqrt{w^2 + 9}$. By applying Theorem 4, we get an equilateral chain whose minimum span depends on the satisfiability of ϕ. $\square$

Theorem 7. *There exists a polynomial-time algorithm which takes as input a rectilinear planar monotone 3-SAT instance ϕ, and outputs an equilateral fixed-angle chain and a distance d such that the maximum span of the chain in any flat state is greater than d if and only if ϕ is satisfiable.*

Proof. The argument is similar to Theorem 6, but with the three chains from Theorem 2 arranged as depicted in Fig. 8(b), so that a_n is at $(w + 12, 3)$. □

Acknowledgments. This research was initiated during the open-problem sessions organized around MIT class 6.849: Geometric Folding Algorithms in Fall 2010. We thank the other participants of these sessions — Zachary Abel, Martin Demaine, Isaac Ellowitz, Jason Ku, Jayson Lynch, Tom Morgan, Jie Qi, TB Schardl, and Tomohiro Tachi — for their helpful ideas and for providing a conducive research environment.

References

1. Greg Aloupis, Erik D. Demaine, Vida Dujmović, Jeff Erickson, Stefan Langerman, Henk Meijer, Joseph O'Rourke, Mark H. Overmars, Michael A. Soss, Ileana Streinu, and Godfried T. Toussaint. Flat-state connectivity of linkages under dihedral motions. In *Proceedings of the 13th International Symposium on Algorithms and Computation*, volume 2518 of *Lecture Notes in Computer Science*, pages 369–380, 2002.
2. Greg Aloupis and Henk Meijer. Reconfiguring planar dihedral chains. In *Abstracts from the 22nd European Workshop on Computational Geometry*, Delphi, Greece, March 2006.
3. Mark de Berg and Amirali Khosravi. Optimal binary space partitions in the plane. In *Proceedings of the 16th Annual International Conference on Computing and Combinatorics*, volume 6196 of *Lecture Notes in Computer Science*, pages 216–225, 2010.
4. Erik D. Demaine and Joseph O'Rourke. *Fixed-Angle Linkages.* Cambridge University Press, July 2007.
5. Ronald J. Gillespie and Paul L. A. Popelier. Molecular geometry and the vsepr model. In *Chemical Bonding and Molecular Geometry: From Lewis to Electron Densities*, chapter 4. Oxford University Press, 2001.
6. Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992.
7. David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.
8. Michael Soss. *Geometric and Computational Aspects of Molecular Reconfiguration.* PhD thesis, School of Computer Science, McGill University, 2001.
9. Michael Soss and Godfried T. Toussaint. Geometric and computational aspects of polymer reconfiguration. *Journal of Mathematical Chemistry*, 27(4):303–318, 2000.