# An algorithmic study of manufacturing paperclips and other folded structures 

Esther M. Arkin ${ }^{\text {a }}$, Sándor P. Fekete ${ }^{\text {b,* }}$, Joseph S.B. Mitchell ${ }^{\text {a }}$<br>${ }^{\text {a }}$ Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794-3600, USA<br>${ }^{\text {b }}$ Department of Mathematical Optimization, TU Braunschweig, Pockelsstr. 14, D-38106 Braunschweig, Germany

Received 20 June 2001; received in revised form 14 March 2002; accepted 30 July 2002
Communicated by H. Alt


#### Abstract

We study algorithmic aspects of bending wires and sheet metal into a specified structure. Problems of this type are closely related to the question of deciding whether a simple non-self-intersecting wire structure (a carpenter's ruler) can be straightened, a problem that was open for several years and has only recently been solved in the affirmative. If we impose some of the constraints that are imposed by the manufacturing process, we obtain quite different results. In particular, we study the variant of the carpenter's ruler problem in which there is a restriction that only one joint can be modified at a time. For a linkage that does not self-intersect or self-touch, the recent results of Connelly et al. and Streinu imply that it can always be straightened, modifying one joint at a time. However, we show that for a linkage with even a single vertex degeneracy, it becomes NP-hard to decide if it can be straightened while altering only one joint at a time. If we add the restriction that each joint can be altered at most once, we show that the problem is NP-complete even without vertex degeneracies. In the special case, arising in wire forming manufacturing, that each joint can be altered at most once, and must be done sequentially from one or both ends of the linkage, we give an efficient algorithm to determine if a linkage can be straightened.


© 2002 Elsevier Science B.V. All rights reserved.
Keywords: Linkages; Folding; Polygons; Manufacturing; Wire bending; NP-complete; NP-hard; Process planning

## 1. Introduction

The following is an algorithmic problem that arises in the study of the manufacturability of sheet metal parts: Given a flat piece, F, of sheet metal (or cardboard, or other bendable stiff sheet material),

[^0]0925-7721/02/\$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.


Fig. 1. Examples of paperclips: (a) and (b) are standard versions, which are readily straightened. (c) is a "butterfly" paperclip, which is not a planar structure and is not among the wire structures considered in our two-dimensional model. (d) shows a 5-link paperclip that cannot be straightened using complete bends in the plane. (e) shows a 6 -link structure that can be straightened, e.g., using the bend sequence animated below it for the bend sequence $\sigma=(1,5,4,3,2)$.
can a desired final polyhedral part, $P$, be made from it? The 2-dimensional version is the wirebending ("paperclip") problem: Given a straight piece, $F$, of wire, can a desired simple polygonal chain, $P$, be made from it? This problem also arises in the fabrication of hydraulic tubes, e.g., in airplane manufacturing. ${ }^{1}$ In both versions of the problem, we require that any intermediate configuration during the manufacture of the part be feasible, meaning that it should not be self-intersecting. In particular, the paperclips that we manufacture are not allowed to be "pretzels"-we assume that the wire must stay within the plane, and not cross over itself. See Fig. 1 for an illustration. We acknowledge that some real paperclips are designed to cross over themselves, such as the butterfly style of clip shown in the figure.

Our problem is one of automated process planning: Determine a sequence (if one exists) for performing the bend operations in sheet metal manufacturing. We take a somewhat idealized approach in this paper, in that we do not attempt to model here the important aspects of tool setup, grasp positions, robot motion plans, or specific sheet metal material properties which may affect the process. Instead, we focus on the precise algorithmic problem of determining a sequence for bend operations, on a given sheet

[^1]of material with given bend lines, assuming that the only constraint to performing a bend along a given bend line is whether or not the structure intersects itself at any time during the bend operation.

Note that the problem of determining if a bend sequence exists that allows a structure to unfold is equivalent to that of determining if a bend sequence exists that allows one to fold a flat (or straight) input into the desired final structure: the bending operations can simply be reversed. For the remainder of the paper, we will speak only of unfolding or straightening.

### 1.1. Motivation and related work

Our foldability problem is motivated from process planning in manufacturing of structures from wire, tubing, sheet metal and cardboard. The CAD/CAM scientific community has studied extensively the problem of manufacturability of sheet metal structures; see the thesis of Wang [34] for a survey. Systems have been built (e.g., PART-S [12] and BendCad [17]) to do computer-aided process planning in the context of sheet metal manufacturing; see also [3,10,19,35,36]. See [24] for a motion planning approach to the problem of computing folding sequences for folding three-dimensional cardboard cartons. Considerable effort has gone into the design of good heuristics for determining a bend sequence; however, the known algorithms are based on heuristic search (e.g., A*) in large state spaces; they are known to be worst-case exponential. (Wang [34] cites the known complexity as $\mathrm{O}\left(n!2^{n}\right)$.)

Our work is also motivated by the mathematical study of origami, which has received considerable attention in recent years. In mathematics of origami, Bern and Hayes [5] have studied the algorithmic complexity of deciding if a given crease pattern can be folded flat; they give an NP-hardness proof. Lang $[20,21]$ gives algorithms for computing crease patterns in order to achieve desired shapes in three dimensions. Other work on computational origami includes [1,14,15,18,27,28,31]. A closely related problem is that of flat foldings of polyhedra. It is a classic open question whether or not every convex polytope in three dimensions can be cut open along its edges so that it unfolds flat, without overlaps. Other variants and special cases have been studied; see [2,4,9,25,26].

Finally, we are motivated by the study of linkage problems; in fact, in the time since this paper was first drafted, the carpenter's ruler conjecture has been resolved by Connelly, Demaine and Rote [11] and Streinu [32]: Any (strongly) simple polygonal linkage with fixed length links and hinged joints, can be straightened while maintaining strong simplicity (i.e., without the linkage crossing or touching itself). (They also show related facts about linkage systems, e.g., that any simple polygonal linkage can be convexified.) In fact, Streinu [32] gives an algorithmic solution that bounds the complexity of the unfolding and is somewhat more general than the slightly earlier results of [11]. These results imply that any (strongly simple) paperclip can be manufactured if one has a machine that can perform a sufficiently rich set of bending operations. For a recent overview of folding and unfolding, see the thesis of Demaine [13]. Earlier and related work on linkages includes [7,8,22,23,29,30,33]. Our hardness results are particularly interesting and relevant in light of these new developments, since we show that even slight changes in the assumptions about the model or the allowed input results in linkages that cannot be straightened, and it is NP-hard to decide if they can be straightened.

### 1.2. Summary of results

(1) We show that it is (weakly) NP-complete to determine if a given rectilinear polygonal linkage can be straightened, under the restriction that only one joint at a time is altered and each joint can be altered
only once (so the joint must be straightened in a single bend operation). A consequence is that the more general sheet metal bending problem is hard as well, even in the case of parallel bend lines and an orthohedral structure $P$.
(2) We prove that it is (weakly) NP-hard to determine if a given polygonal linkage can be straightened if there is a vertex degeneracy, in which two vertices coincide. Here we again assume that only one joint can be altered at a time, but we do not assume that a joint is altered only once, so we may make any number of bends at any particular joint.
(3) We give efficient algorithms for determining if a given bend sequence is feasible, assuming only one joint is altered at a time, and for determining if certain special classes of bend sequences are feasible. In particular, we give an efficient $\left(\mathrm{O}\left(n \log ^{2} n\right)\right)$ algorithm for determining if a polygonal linkage can be straightened using a sequential strategy, in which the joints are completely straightened, one by one, in order along the linkage. We also give efficient polynomial-time algorithms for deciding whether there is a feasible bend sequence that straightens joints in an order "inwards" from both ends or "outwards" towards both ends. (Such constrained bend sequences may be required for automated wire-bending machines.) These results will be made more precise in Section 4.

## 2. Preliminaries

The input to our problem is a simple polygonal chain (linkage), $P$, with vertex sequence $\left(b_{0}, b_{1}, b_{2}, \ldots, b_{n+1}\right)$. The points $b_{0}$ and $b_{n+1}$ are the endpoints of the chain, and the $n$ vertices $b_{1}, \ldots, b_{n}$ are the bends (or joints). The line segments $b_{i} b_{i+1}$ are the edges (or links) of $P$. The edge $b_{i} b_{i+1}$ is a closed line segment; i.e., it includes its endpoints. We consider the chain $P$ to be oriented from $b_{0}$ to $b_{n+1}$, and we consider each edge of $P$ to have a left and a right side. Each bend $b_{i}$ has an associated bend angle $\theta_{i} \in(0,2 \pi]$, measured between the right sides of the two edges incident on $b_{i}$.

The chain $P$ is strongly simple if any two edges, $b_{i} b_{i+1}$ and $b_{j} b_{j+1}$, of $P$ that are not adjacent $(i \neq j)$ are disjoint and any two adjacent edges share only their one common endpoint. We say that $P$ is simple if it is not self-crossing but it possibly is self-touching, with a joint falling exactly on a non-incident edge or another joint; i.e., $P$ is simple if it is strongly simple or an infinitesimal perturbation of it is strongly simple.

We consider the chain $P$ to be a structure consisting of rigid rods as edges, whose lengths cannot change, connected by hinged joints. When a bend operation is performed at joint $b_{i}$, the bend angle $\theta_{i}$ is changed. Throughout this paper, we assume that the only bend operations allowed are single-joint bends, in which only one bend angle is altered at a time. We establish the convention that when a bend operation occurs at $b_{i}$, the subchain containing the endpoint $b_{0}$ remains fixed in the plane, while the subchain containing $b_{n+1}$ rotates about the joint $b_{i}$. This convention allows us to have a unique embedding of a partially or fully straightened chain in the plane.

A bend operation is complete if, at the end of the operation, the bend angle is $\pi$; we then say that the joint has been straightened. A bend operation that is not complete is called a partial bend. A sequence of bend operations is said to be monotonic if no bend operation increases the absolute deviation from straightness, $\left|\theta_{i}-\pi\right|$, for a joint $b_{i}$. If all joints of $P$ have been straightened, the resulting chain is a straight line segment, $F$, of length $\sum_{i=0}^{n}\left|b_{i} b_{i+1}\right|$, where $\left|b_{i} b_{i+1}\right|$ denotes the Euclidean length of segment $b_{i} b_{i+1}$. By our bend operation convention, one endpoint of $F$ is $b_{0}$, and $F$ contains the segment $b_{0} b_{1}$ (which never moves during bend operations).

For $S \subseteq B$, we let $P(S)$ denote the partially straightened polygonal chain having each of the bends $b_{i} \in S$ straightened (to bend angle $\pi$ ), while each of the other bends, $b_{i} \notin S$, is at its original bend angle $\theta_{i}$. Thus, in this notation $P(B)=F$ and $P(\emptyset)=P$. We let $P(S ; i, \theta)$, for $1 \leqslant i \leqslant n$ with $i \notin S$, denote the chain in which each bend $b_{j} \in S$ is at bend angle $\pi$, bend $b_{i}$ is at angle $\theta$, and all other bends $b_{j} \notin S$ are at their original bend angles $\theta_{j}$. We say that chain $P(S)$ or $P(S ; i, \theta)$ is feasible if it is a simple chain.

We say that bend $b_{i}$ is foldable (or is a feasible fold) for $P(S)$ if $P(S ; i, \theta)$ is feasible for all $\theta$ in the range between $\pi$ and $\theta_{i}$ (more precisely, for all $\theta_{i} \leqslant \theta \leqslant \pi$, if $\theta_{i}<\pi$, or for all $\pi \leqslant \theta \leqslant \theta_{i}$, if $\left.\theta_{i}>\pi\right)$. If $b_{i}$ is foldable, then it is possible to make a complete bend at $b_{i}$, meaning that the joint can be straightened in a single operation without causing the chain to self-intersect. We say that a permutation $\sigma=\left(i_{1}, i_{2}, \ldots, i_{n}\right)$ of the indices $\{1,2, \ldots, n\}$ is foldable for $P$ if, for $j=1,2, \ldots, n$, joint $b_{i_{j}}$ is foldable for $P\left(\left\{b_{i_{1}}, \ldots, b_{i_{j-1}}\right\}\right)$, i.e., if $P$ can be unfolded into the straight segment $F$ using the bend sequence $\sigma$ (so that, by reversing the operations, $P$ can be manufactured from $F$ using the reverse of the bend sequence).

The Wire Bend Sequencing problem can be formally stated as: Determine a foldable permutation $\sigma$, if one exists, for a given chain $P$.

This paper studies the Wire Bend Sequencing problem for polygonal chains in the plane. We note, however, that our results have some immediate implications for the Sheet Metal Bend Sequencing problem, which is defined analogously for a polyhedral surface $P$ having a pattern $B$ of bend lines (creases), each of which must be straightened in order to flatten $P$ into a flat polygon $F$. Specifically, the hardness of the Sheet Metal Bend Sequencing follows from the hardness of the Wire Bend SEQUENCING, which can be seen as a special case of the sheet metal problem in which $F$ is a rectangle and the bend lines $B$ are all segments parallel to two of the sides of $F$ and extending all the way across $F$.

We give an example in Fig. 1 of some common paperclip shapes, (a)-(c). We also show an example, (d), of a 5-link paperclip that cannot be straightened using complete bends, for any permutation $\sigma$ of the bends. Finally, we show an example of a 6 -link paperclip for which the foldable permutations are $\{(1,5,4,3,2),(1,5,4,2,3)\}$; we show the sequence of bends, with the intermediate structures, for the permutation $\sigma=(1,5,4,3,2)$.

## 3. Hardness results

### 3.1. Complete bends

Our first result shows that if we require bends to be complete, as in our specification of the WIRE BEND SEQUENCING problem, the problem of deciding if there is a feasible bend sequence is NP-complete.

Theorem 1. Wire Bend Sequencing is (weakly) NP-complete, even if $P$ is rectilinear.
Proof. We prove NP-completeness, even in the case that we are restricted to a special class of bend sequences, namely, those that can be written as the concatenation of up to four monotone subsequences of the index set $\{1, \ldots, n\}$. Below, we refer to each subsequence of bends as a monotone pass over the chain, going from one end to the other, performing a specified subset of complete bends.

Our reduction is from Partition: Given a set $S$ of $n$ integers, $a_{i}$, which sum to $A=\sum_{i} a_{i}$, determine if there exists a partition of the set into two subsets each of which sums to $A / 2$.


Fig. 2. Proving hardness of the WIRE BEND SEQUENCING problem for rectilinear chains: frame and key.

The key idea of our construction uses two components, as shown in Fig. 2: One is a rigid "frame" that can only be unfolded if one end of the chain can be removed from within this frame. The other component is a "key" that encodes the partition instance. Collapsing the key is possible if and only if there is a partition of the integers into two sets of equal sum. The total number of segments will be $\ell=26+4 n$; we write $b_{i}(i=0,1, \ldots, 26+4 n)$ for the vertices, and $s_{i}=\left(b_{i-1}, b_{i}\right)$ for the segments. For any point in time, we refer to the position of a joint $b_{i}$ by its coordinates ( $x_{i}, y_{i}$ ). When discussing some of the relative distances, we use $d_{\infty}\left(b_{i}, b_{j}\right)=\max \left\{\left|x_{i}-x_{j}\right|,\left|y_{i}-y_{j}\right|\right\}$.

More precisely, the frame consist of 13 segments, $s_{1}=\left(b_{0}, b_{1}\right), \ldots, s_{13}=\left(b_{12}, b_{13}\right)$, as shown in the figure. Segment lengths are chosen such that the size of the frame is $\Theta(L)$, with minimal coordinate differences $d_{\infty}\left(b_{0}, b_{13}\right), d_{\infty}\left(b_{1}, b_{12}\right), d_{\infty}\left(b_{2}, b_{11}\right), d_{\infty}\left(b_{3}, b_{10}\right), d_{\infty}\left(b_{4}, b_{9}\right), d_{\infty}\left(b_{5}, b_{8}\right), d_{\infty}\left(b_{6}, b_{7}\right)$ being $\Theta(\varepsilon)$, where $\varepsilon=1 /\left(n^{3} L^{2}\right)$. The "key" consists of $13+4 n$ segments, $s_{14}=\left(b_{13}, b_{14}\right), \ldots, s_{26+4 n}=$ $\left(b_{25+4 n}, b_{26+4 n}\right)$. For $i=0, \ldots, 4 n+2$, the "auxiliary" segments $s_{15+4 i}, s_{16+4 i}, s_{17+4 i}$ have length $\varepsilon$, while the "partition" segments $s_{18+4 i}$ have length $a_{i}$. The long "positioning" segments $s_{14}, s_{15}, s_{25+4 n}, s_{26+4 n}$ have lengths $L, L / 3, L / 3$ and $L-3 n \varepsilon$, respectively; they guarantee that the partition segments must have a particular relative position when removing the key. We choose the scale to be such that $L / 4>A$, for technical reasons that will become clear later in the proof. As indicated in the figure, the initial position of each key segment $s_{i}, i=14, \ldots, 26+4 n$ has $x$-coordinate $x_{13}$ or $x_{13}+\varepsilon$, with a horizontal distance of $x_{i}-x_{4}=\varepsilon$ or $x_{i}-x_{4}=2 \varepsilon$ from $s_{4}$. Moreover, $b_{14}$ is positioned at a vertical distance of $y_{14}-y_{4}=n^{2} \varepsilon=1 /\left(n L^{2}\right)$ above $b_{4}$.

The purpose of the auxiliary segments is as follows. As shown in Fig. 2, we have two types of joints in the figure: the "ordinary" ones (indicated by solid black dots) form the frame and can only be accessed once. The "quadruple" ones (indicated by hollow dots in Fig. 2) consist of the four simple joints at three consecutive auxiliary segments; they are found along the key as described. These quadruple joints make it possible to simulate opening and closing such a joint a limited number of times.

Now assume that there is a partition $S=S_{1} \cup S_{2}$, such that $\sum_{i \in S_{1}} a_{i}=\sum_{i \in S_{2}} a_{i}$. In order to see that the key can be removed from the frame we first convert it into the "stair" configuration shown in


Fig. 3. Turning the key into a stair. (a) An intermediate stage of the monotone pass. (b) The stair configuration at the end of the monotone pass, with details of the state of quadruple joints.

Fig. 3: We make one monotone pass over the chain towards the key end, and straighten one ordinary joint per quadruple joint whenever this joint separates two segments from different $S_{i}$. Thus, segments corresponding to numbers in $S_{1}$ will be horizontal, while those for numbers in $S_{2}$ will be vertical. In order to keep the number of monotone passes limited to four, during this first pass we also straighten two ordinary joints per quadruple joint separating two segments from the same $S_{i}$, as shown in Fig. 3(b).

Making a similar monotone pass, we can convert the stair into a "flat harmonica", as shown in Fig. 4, with segments from $S_{2}$ pointing "down", i.e. $y_{i}<y_{i-1}$, and segments from $S_{1}$ pointing "up", i.e. $y_{i}>y_{i-1}$. By assumption about the partition, the positions of endpoints $b_{18}$ and $b_{18+4 n}$ satisfy $d_{\infty}\left(b_{18}, b_{18+4 n}\right)<3 n \varepsilon$ and $2 L / 3-3 \varepsilon<y_{13}-y_{18}<2 L / 3+3 \varepsilon$, i.e., both $b_{18}$ and $b_{18+4 n}$ are roughly $2 L / 3$ below $b_{13}$. Altogether, the position of the last segment $s_{26+4 n}$ of length $L$ in the chain will differ by at most $\mathrm{O}(n \varepsilon)$ from the vertical position of segment $s_{14}$, with all other segments strictly in-between. This collapsed structure can be rotated about $b_{13}$ without colliding with any frame segments. Then it is easy to open up the remaining frame (by straightening $b_{12}, b_{11}, b_{10}, b_{8}, b_{7}, b_{6}, b_{5}, b_{4}, b_{3}, b_{2}, b_{1}$ as one monotone pass, skipping $b_{9}$ ). Finally, the resulting monotone chain can be straightened in one last monotone pass.

Conversely, assume now that the chain can be straightened. See Fig. 5. It is clear that $b_{13}$ must be straightened before any other joint in the set $\left\{b_{1}, \ldots, b_{12}\right\}$. In order to avoid hitting vertex $b_{4}$ during this motion, any part of the key to the right and below $b_{13}$ must be strictly within the circle $C$ of radius $r=\sqrt{\left(L+1 /\left(n L^{2}\right)\right)^{2}+\varepsilon^{2}}<L+2 /\left(n L^{2}\right)$ around $b_{13}$, where $r$ is the distance between $b_{13}$ and $b_{4}$ (see Fig. 4). The following technical arguments show that at this time, segment $s_{26+4 n}$ has to be in a vertical position that basically coincides with $s_{14}$, which is only possible in case of a feasible partition.


Fig. 4. Turning the stair into a harmonica of small width and length $L$. (The horizontal width is not drawn to scale in order to show details.)


Fig. 5. When straightening joint $b_{13}$, the key must be fully contained in the shaded circle of radius $r<L+2 /\left(n L^{2}\right)$. This forces a particular position of segment $s_{26+4 n}$.

When starting the rotation about $b_{13}, s_{26+4 n}$ is an axis-parallel segment of length $L-3 n \varepsilon>L-$ $1 /\left(n L^{2}\right)$. The rigid frame and the closeness of $b_{14}$ and $b_{4}$ ensure that segment $s_{26+4 n}$ cannot lie to the left of $s_{14}$, implying that $s_{26+4 n}$ can only lie within the quarter circle of radius $r$ below and to the right of $b_{13}$ when $b_{13}$ is straightened.

Let $b_{\text {min }}$ be one of the two points in $\left\{b_{25+4 n}, b_{26+4 n}\right\}$ that is not further from $b_{13}$ than the other, and let $b_{\max }$ be the other point. If the vertical distance $y_{13}-y_{\text {min }}$ is greater than $\sqrt{7 /(n L)}$, it follows that the Euclidean distance between $b_{\text {max }}$ and $b_{13}$ is at least

$$
\sqrt{\frac{7}{n L}+\left(L-\frac{1}{n L^{2}}\right)^{2}}=\sqrt{\left(L+\frac{2}{n L^{2}}\right)^{2}+\frac{1}{n L}-\frac{3}{n^{2} L^{4}}}>L+\frac{2}{n L^{2}}>r
$$

a contradiction to the assumption that $s_{26+4 n}$ is fully contained in $C$. Now, using our assumption that $L / 4>A$, we know that $b_{25+4 n}$ and $b_{14}$ are connected by a polygonal chain of length strictly less than $L / 3+L / 4+L / 3=11 L / 12$, implying that $b_{25+4 n}$ has Euclidean distance at least $L / 12$ from $b_{13}$, so $b_{26+4 n}=b_{\min }$ and $b_{25+4 n}=b_{\max }$. As $b_{26+4 n}$ is within $\sqrt{7 /(n L)}$ of $b_{13}$, it follows that $b_{26+4 n}$ has Euclidean distance at least $L-\sqrt{7 /(n L)}$ from $b_{14}$. If $s_{26+4 n}$ were horizontal, then the Euclidean distance between $b_{25+4 n}$ and $b_{14}$ would be at least

$$
\sqrt{(L-\sqrt{7 /(n L)})^{2}+\left(L-1 /\left(n L^{2}\right)\right)^{2}}>11 L / 12
$$

a contradiction. Hence, $s_{26+4 n}$ must be vertical. Just as we derived for the vertical distance between $b_{13}$ and $b_{26+4 n}$, it follows for the horizontal distance that $x_{26+4 n}-x_{13} \leqslant \sqrt{7 /(n L)} \ll 1$.

Now observe that when starting the rotation about $b_{13}$, all partition segments must be strictly between $s_{14}$ and the narrow strip between $s_{14}$ and $s_{26+4 n}$, meaning that they are all vertical. Let $S_{1}$ be the set of "upwards" partition segments $s_{i}$ with $y_{i}>y_{i-1}$, and $S_{2}$ be the set of "downwards" partition segments $s_{i}$ with $y_{i}<y_{i-1}$. As $\left|y_{24+4 n}-y_{15}\right|=\Theta(n \varepsilon)$ and $\left|y_{25+4 n}-y_{14}\right|=\Theta\left(n^{2} \varepsilon\right)$, we conclude that the integral total length of upwards segments equals the integral total length of downwards segments.

This means that $\sum_{i \in S_{1}} a_{i}=\sum_{i \in S_{2}} a_{i}$, and we have a feasible partition. This completes the proof.

### 3.2. Partial bends

Now we consider the case in which each joint may be changed an arbitrary number of times during the straightening operations, while still making single-joint bends (bending only one joint at a time). This version of the problem is closely related to the carpenter's ruler problem studied by $[11,32]$. In the context of our study on folding, there may be the additional requirement of using only monotonic bend operations, e.g., to avoid work-hardening the wire, possibly causing it to break. We begin with the following observation about the sufficiency of monotonic single-joint bends; see also the discussion on p. 9 of Demaine's thesis [13].

Theorem 2. Any strongly simple polygonal chain $P$ can be straightened using a finite number of monotonic single-joint bends.

Proof. Consider the set $\mathcal{S}$ of points in $n$-dimensional joint-angle space that correspond to strongly simple embeddings of the linkage. A single-joint bend corresponds to axis-parallel motion in joint-angle space. If self-touching is prohibited, $\mathcal{S}$ is an open set; note too that $\mathcal{S}$ is bounded. By Streinu's result [32], there is an opening motion of the chain that consists of a finite number of individual monotonic moves. Such an opening motion corresponds to a path, $\Pi$, in $\mathcal{S}$, comprised of a finite number of arcs, each corresponding to a monotonic move. Let $\varepsilon$ be the Euclidean distance between path $\Pi$ and the boundary of $\mathcal{S}$; since $\mathcal{S}$ is open, we know that $\varepsilon>0$. Then we can replace each arc of the path $\Pi$ with a finite sequence of axis-parallel moves of size $\varepsilon / 2$, yielding a straightening that uses single-joint bends.

We will refer to a sequence of small individual moves that mimics an overall large-scale motion of several joints as "wiggly", since the overall motion may be achieved through back-and-forth motions of individual segments that gradually change individual angles.

The following results show that allowing even a single point of self-incidence along the linkage changes the overall situation quite drastically.

Lemma 3. There are polygonal chains $P$ with a single vertex-to-vertex incidence that cannot be straightened using partial single-joint bends.

Proof. See Fig. 6. The chain has eight joints (labeled $b_{0}, \ldots, b_{7}$ ) and seven segments (of the form $\left.s_{i}=\left(b_{i-1}, b_{i}\right)\right)$. The endpoint $b_{0}$ coincides with joint $b_{5}$. It is easily checked that none of the joints $b_{1}, \ldots, b_{4}$ can be changed without causing a self-intersection: Assume that there is a feasible motion of a joint $b_{i}$ with $0<i<5$. Then the points $b_{0}$ and $b_{5}$ would move away from each other along a circle around $b_{i}$. Without loss of generality, assume that $b_{5}$ remains in place, while $b_{0}$ is moving. Now consider the first such rotation that starts with $b_{0}$ and $b_{5}$ coinciding, and that avoids a crossing of $s_{1}$ with both $s_{5}$ and $s_{6}$. If $b_{0}$ moves clockwise around $b_{i}$, it is easy to see that the angle between $\left(b_{0}, b_{i}\right)$ and $s_{5}$ must be at least $\pi / 2$ when starting the motion, or else $s_{1}$ and $s_{5}$ intersect. If $b_{0}$ moves counterclockwise around $b_{i}$, the same follows for the angle between $\left(b_{0}, b_{i}\right)$ and $s_{6}$. Therefore, the center of rotation must lie within the shaded region shown in the figure. (The cone to the left of $b_{5}$ is feasible for clockwise rotation, while the cone to the right of $b_{5}$ is feasible for counterclockwise roation.) However, none of the joints $b_{1}, \ldots, b_{4}$ lies inside of this feasible region. It follows that $b_{0}, \ldots, b_{5}$ form a rigid frame, as long as the angle at $b_{5}$ stays smaller than $\pi / 2$.


Fig. 6. A polygonal chain that cannot be opened with single-joint moves.

On the other hand, it is easy to see that $b_{7}$ cannot be removed from the pocket formed by $b_{1}, b_{2}$ and $b_{3}$ if only the two remaining "free" joints $b_{5}$ and $b_{6}$ can be changed. The claim follows.

If $b_{0}$ and $b_{5}$ have some positive distance, then the frame can be opened along the lines of the approach in [11] or [32] by gradually straightening $b_{5}, b_{6}, b_{1}, b_{2}$ and $b_{3}$, so that the "zig-zagging" part between $b_{0}$ and $b_{3}$ pushes left, while $b_{6}$ swings around $b_{5}$.

Using the frame as a gadget, we can show the following:
Theorem 4. It is NP-hard to decide if a polygonal chain $P$ with a single vertex-to-vertex incidence can be straightened by arbitrary partial single-joint bends.

Proof. The basic idea is similar to the one in Theorem 1 and also establishes a reduction of Partition. (Refer to Fig. 7 for an overview.) As before, we write $b_{i}$ for the joints, and $s_{i}=\left(b_{i-1}, b_{i}\right)$ for the segments. We use the idea of the construction from Lemma 3 to construct a rigid frame, with the key corresponding to the free end of that chain. The frame has one end, $b_{0}$, of the polygonal chain wedged into the corner $b_{13}$, which has angle $\varphi \ll \pi / 2$. Because of the degeneracy at $b_{13}$, none of the joints $b_{1}, \ldots, b_{12}$ can be moved individually without causing a self-intersection between $b_{0}$ and the chain in the neighborhood of $b_{13}$ :


Fig. 7. Illustration of the proof of Theorem 4. Note that lengths are not drawn to scale, in order to show sufficient details; in particular, the dimensions of the bottleneck are much smaller than the edges encoding the partition instance.

Just like in the proof of Lemma 3, none of the joints $b_{1}, \ldots, b_{12}$ lies in the area of possible locations of feasible rotations. This continues to be the case while $\phi+\psi<\pi$, i.e., while the sum of angles at $b_{13}$ and at $b_{17}$ does not change significantly.

Again, the "key" contains the $n$ segments $s_{19}, \ldots, s_{19+n}$ of integral lengths $a_{1}, \ldots, a_{n}$ that encode an instance of Partition. As before, let $S$ denote the set of integers for the Partition instance. We also use "long" auxiliary segments of lengths $L / 2$ and $L$, where $L \gg \sum_{i} a_{i}=A$. Here segments $s_{19}$ and $s_{20+n}$ have length $L / 2$, while $s_{21+n}$ has length $L$.

The critical dimensions of the frame are chosen such that the key can just be removed from the frame if and only if it can be collapsed to a length of $L$. Removing the key consists in pulling it through the narrow bottleneck formed by the segments $s_{3}$ and $s_{9}$ by extending the "spring" formed by $s_{14}$ and $s_{15}$, while moving the "keyholder" $s_{18}$ down by a distance of $L+\varepsilon$. This is possible if and only if there is a feasible partition.

More precisely, let the angle at $b_{17}$ be $\psi \ll \pi$. Let $\varepsilon=\mathrm{O}(1 / n L)$. We assume that the dimensions of the frame are chosen sufficiently large to guarantee that moving $b_{17}$ down by a vertical distance of $L+\varepsilon$ increases its distance from $b_{13}$ by $L \cos \psi+\Theta(\varepsilon)$, i.e., the angles at $b_{17}$ and at $b_{13}$ do not change much. The segments $s_{14}$ and $s_{15}$ forming the spring have length $L / 2 \cos \psi+\Theta(\varepsilon)$, so extending the spring will just suffice to move the keyholder $s_{18}$ down by $L+\varepsilon$, but not more. The vertical "height" of the bottleneck, i.e., the length of the segments $s_{3}$ and $s_{9}$, is $\varepsilon / 3$, while the horizontal "width" $x_{8}-x_{3}=x_{9}-x_{2}=\varepsilon^{4}$ is significantly smaller. (As we will discuss below, this forces the keyholder to be roughly vertical throughout the motion.) For the initial position of the key inside of the frame, we assume $L<y_{18}-y_{3}<L+\varepsilon / 3$ and $L<y_{20+n}-y_{3}<L+\varepsilon / 3$. Finally, $y_{6}-y_{7}=L+\Theta(\varepsilon)$ and $x_{6}-x_{5}=\Theta(A)$, so the dimensions of the rectangle formed by $b_{4}, b_{5}, b_{6}, b_{7}$ are not large enough to change the basically vertical orientation of the location segments $s_{19}, s_{19+n}$ and $s_{20+n}$.

Now assume that there is a feasible partition, $S=S_{1} \cup S_{2}$, such that $\sum_{i \in S_{1}} a_{i}=\sum_{i \in S_{2}} a_{i}$. By performing a (finite) "wiggly" sequence of moves, we can move the partition segments such that (1) any segment $s_{19+i}$ representing $a_{i} \in S_{1}$ satisfies $y_{i}-y_{i-1}=a_{i}+\mathrm{O}\left(\varepsilon^{5}\right)$ and $\left|x_{i}-x_{i-1}\right|=\mathrm{O}\left(\varepsilon^{5}\right)$, so that $s_{19+i}$ is pointing up; (2) any segment $s_{19+i}$ representing $a_{i} \in S_{2}$ satisfies $y_{i-1}-y_{i}=a_{i}+\mathrm{O}\left(\varepsilon^{5}\right)$ and $\left|x_{i}-x_{i-1}\right|=\mathrm{O}\left(\varepsilon^{5}\right)$, so that $s_{19+i}$ is pointing down; and (3) we end up placing $b_{20+n}$ within Euclidean distance $\mathrm{O}\left(\varepsilon^{4}\right)$ from $b_{18}$ and placing $b_{21+n}$ at distance $\varepsilon / 3+\mathrm{O}\left(\varepsilon^{5}\right)$ from $b_{3}$ and $b_{8}$. Thus, extending the spring by an appropriate wiggly motion moves the key through the bottleneck. Now it is easy to open the joint $b_{13}$, and unfold the whole chain.

Conversely, assume that the chain can be unfolded. As discussed above, the sum of angles at $b_{13}$ and at $b_{17}$ has to change significantly before the frame ceases to be rigid. Now note that the dimensions of the bottleneck force the keyholder segment to be roughly vertical, i.e., to have slope within $\mathrm{O}\left(1 / \varepsilon^{3}\right)$ of vertical. (See Fig. 8.) Furthermore, we noted above that any feasible vertical motion of $b_{17}$ does not change the angle at $b_{17}$ by a significant amount; it is clear that this also prevents the angle at $b_{13}$ from changing much. Therefore, the frame remains rigid until the key has been removed from the lock.

Now consider the positions of segments $s_{18}$ and $s_{21+n}$ when $b_{21+n}$ crosses the horizontal line $y=y_{2}$. By the dimensions of the bottleneck, $s_{21+n}$ must have a slope within $\Omega\left(1 / \varepsilon^{3}\right)$ of vertical. Furthermore, by construction of the rectangle $b_{4} b_{5} b_{6} b_{7}$, we are assured that $y_{18}-y_{20+n} \leqslant \mathrm{O}(\varepsilon)$, i.e., $s_{20+n}$ cannot be significantly below $s_{18}$. On the other hand, $b_{20+4 n}$ must be below the horizontal line $y=y_{3}$ when $b_{18}$ has been moved down by a vertical distance of $L+\varepsilon$. Since no segment within the key can change its


Fig. 8. All segments have to be inside of the shaded region when moving through the bottleneck, i.e., must be close to being vertical. (Horizontal scale and size of the bottleneck are vastly exaggerated to allow sufficient resolution. In scale, the shaded region is basically a vertical line.)
vertical slope significantly while $s_{21+n}$ is within the bottleneck (they must all remain wedged between $s_{18}$ and $s_{21+n}$ and be strictly contained in the shaded region in Fig. 8), we conclude that $y_{20+n}-y_{18} \leqslant \mathrm{O}(\varepsilon)$ upon leaving the bottleneck, i.e., $s_{20+n}$ cannot be significantly above $s_{18}$.

Therefore, the sets $S_{1}=\left\{i \in 1, \ldots, n \mid y_{i} \geqslant y_{i-1}\right\}$ and $S_{2}=\left\{i \in 1, \ldots, n \mid y_{i}<y_{i-1}\right\}$ upon entering the bottleneck from above and leaving it from below must satisfy $\sum_{i \in S_{1}} a_{i}=\sum_{i \in S_{2}} a_{i}+\Theta(\varepsilon)$. Since $\varepsilon \ll 1$, this implies that there is a feasible partition.

For the case of monotonic bend operations, the above proof can be easily modified:

Corollary 5. It is NP-hard to decide if a polygonal chain $P$ with a single vertex-to-vertex incidence can be straightened by monotonic partial single-joint bends.

Proof. The joints in the construction shown in Fig. 7 that may not be changed monotonically are $b_{14}, \ldots, b_{20+n}$, the ones that are not part of the frame. By using small quadruple joint gadgets as in the construction for Theorem 1, we get a chain that can be opened with monotonic moves, if and only if it can be opened.

## 4. Algorithms

In this section, we turn our attention to positive algorithmic results, giving efficient algorithms for deciding if particular bend sequences are feasible. We consider here only the case of complete bends.

Consider an arbitrary permutation, $\sigma=\left(i_{1}, \ldots, i_{n}\right)$, of the bends along a wire. In order for $\sigma$ to be a foldable sequence, it is necessary and sufficient that for each $j=1, \ldots, n$ the bend $b_{i_{j}}$ is foldable. Recall that in our notation $P\left(\left\{i_{1}, \ldots, i_{j-1}\right\}\right)$ denotes the partially bent chain after the bends at $b_{i_{1}}, \ldots, b_{i_{j-1}}$ have been straightened. The point $b_{i_{j}}$ splits $P\left(\left\{i_{1}, \ldots, i_{j-1}\right\}\right)$ into two subchains; let $P_{0}$ (respectively $\left.P_{n+1}\right)$ denote the subchain containing the endpoint $b_{0}$ (respectively $b_{n+1}$ ). Now, $b_{i_{j}}$ is foldable if the joint at $b_{i_{j}}$ can be straightened without causing a collision to occur between $P_{0}$ and $P_{n+1}$ at any time during the rotation about $b_{i_{j}}$. We can assume, without loss of generality, that $P_{0}$ is fixed and that $P_{n+1}$ is pivoted about $b_{i_{j}}$. During this bend operation at $b_{i_{j}}$, each point, $u$, on $P_{n+1}$ moves along a circular arc, $A_{u}$, subtending an angle $\theta_{i_{j}}$, centered on $b_{i_{j}}$. It is clear that in order for the bend to be feasible, none of these $\operatorname{arcs} A_{u}$ may cross the chain $P_{0}$, for all choices of points $u$ on $P_{n+1}$.

If the perpendicular projection of $b_{i_{j}}$ onto the line containing an edge $e$ of $P_{n+1}$ lies on the edge, let $w_{e} \in e$ denote the projection point. (Each edge of $P_{n+1}$ has at most one projection point.) Let $U$ denote the union of the set of vertices of $P_{n+1}$ and the set of projection points on edges of $P_{n+1}$. In the lemma below, we observe that, in order to test feasibility of straightening the bend $b_{i_{j}}$, it suffices to consider only the feasibility of the final position of the chain $P_{n+1}$ and to test $P_{0}$ for intersection with the discrete set $\mathcal{A}=\left\{A_{u}: u \in U\right\}$. See Fig. 9 .

Lemma 6. Joint $b_{i_{j}}$ is foldable if and only if (1) no arc of $\mathcal{A}$ intersects $P_{0}$, and (2) after the bend, no segment of $P_{n+1}$ intersects a segment of $P_{0}$.


Fig. 9. Foldability of the joint $b_{i}$ : The subchain $P_{n+1}$ is shown with thicker lines (two dashed copies show it after different stages of rotation about $b_{i_{j}}$ ). Each vertex and each projection point (shown as black disks) of $P_{n+1}$ moves along a circular arc, shown using a thin dashed arc. In this example, the rotation shown is not feasible, as it fails both conditions (1) and (2) of the lemma.

Proof. If joint $b_{i_{j}}$ is foldable, then, by definition, there can be no intersection of $P_{n+1}$ with $P_{0}$ during its rotation about $b_{i_{j}}$. This implies conditions (1) and (2).

If conditions (1) and (2) hold, then we claim that there can be no intersection of $P_{n+1}$ with $P_{0}$ during the rotation. Consider a subsegment, $s$, of $P_{n+1}$ whose endpoints are consecutive points of $U$. (Note that at least one endpoint of $s$ must be a vertex of $P_{n+1}$.) During the rotation, it sweeps a region $R_{s}$ that is bounded by two circular arcs centered at $b_{i_{j}}$, corresponding to the trajectories of its endpoints during the rotation, and two line segments, corresponding to the positions of $s$ before and after the rotation. Here, we are using the fact that the distance from $b_{i_{j}}$ to a point $q \in s$ monotonically changes as a function of the position of $q$ on $s$. (The projection points were introduced in order to assure this property.) Our claim follows from the fact that $P_{0}$ is a simple, connected chain: It cannot intersect $s$ at some intermediate stage of the rotation unless it intersects the boundary of the region $R_{s}$. Such an intersection is exactly what is being checked with conditions (1) and (2).

Lemma 7. For any $S \subseteq B$, and any $b_{i} \notin S$, one can decide in $\mathrm{O}(n \log n)$ time if joint $b_{i}$ is foldable for the chain $P(S)$.

Proof. Using standard plane sweep methods for segment intersections, adapted to include circular arcs, we can check in $\mathrm{O}(n \log n)$ time both conditions ((1) and (2)) of Lemma 6. Events in the sweep algorithm correspond to joints and to vertical points of tangency of circular arcs, assuming we use a vertical sweep line. During the sweep, we keep track of the vertical ordering of the segments and arcs that cross the sweep line; we check for intersection between any two objects that become adjacent in this ordering, stopping if a crossing is detected. Since we process $\mathrm{O}(n)$ events, each at a cost of $\mathrm{O}(\log n)$, the time bound follows.

Remark 8. Condition (2) can be tested in linear time, by Chazelle's triangulation algorithm. We suspect that condition (1) can also be tested in linear time. Condition (1) involves testing for rotational separability of two simple chains about a fixed center point $\left(b_{i}\right)$, which is essentially a polar coordinate variant of translational separability (which is easily tested for simple chains using linear-time visibility (lower envelope) calculation). The issue that must be addressed for our problem, though, is the "wraparound" effect of the rotation; we believe that this can be resolved and that this idea should lead to a reduction in running time of a factor of $\log n$.

Corollary 9. The foldability of a permutation $\sigma$ can be tested in $\mathrm{O}\left(n^{2} \log n\right)$ time.
We obtain improved time bounds for testing the feasibility of a particularly important folding sequence: the identity permutation. Many real tube-bending and wire-bending machines operate in this way, making bends sequentially along the wire/tube. (Such is the case for the hydraulic tube-bending machines at Boeing's factory, where this problem was first suggested to us.) Of course, there are chains $P$ that can be straightened using an appropriate folding sequence but cannot be straightened using an identity permutation folding sequence; see Fig. 1(e). However, for this special case of identity permutations, we obtain an algorithm for determining feasibility that runs in nearly linear time:

Theorem 10. In time $\mathrm{O}\left(n \log ^{2} n\right)$ one can verify if the identity permutation $(\sigma=(1,2, \ldots, n)$ ) is a foldable permutation for $P$.

Proof. For notational convenience, we consider the equivalent problem of verifying if it is feasible, in the order $b_{1}, b_{2}, \ldots, b_{n}$, to bend the joints $b_{i}$ from joint angle $\pi$ to final angle $\theta_{i}$, thereby transforming a straight wire into the final shape $P$, rather than our convention until now of considering the problem of performing bend operations to straighten the chain $P$.

Thus, consider performing the bends in the order given by the identity permutation $\sigma$, and consider the moment when we are testing the foldability of $b_{i}$. The subchain $P_{n+1}$, from $b_{i}$ to $b_{n+1}$, is a single line segment, $b_{i} b_{n+1}$, since the joints $b_{i+1}, \ldots, b_{n}$ are straight at the moment. Thus, verifying the foldability of bend $b_{i}$ amounts to testing if the segment $b_{i} b_{n+1}$ can be rotated about $b_{i}$ by the desired amount, without colliding with any other parts of the subchain $P_{0}$ of $P$, from $b_{0}$ to $b_{i}$. In other words, we must do a wedge emptiness query with respect to $P_{0}$, defined by $b_{i}$, segment $b_{i} b_{n+1}$ and the angle $\theta_{i}$. Since $P_{0}$ is connected, emptiness can be tested by verifying that the boundary of the wedge does not intersect $P_{0}$. (See Fig. 10.) Thus, we can perform this query by using (straight) ray shooting and circular-arc ray shooting in $P_{0}$; the important issue is that $P_{0}$ is dynamically changing as we proceed with more bends. However, in order to avoid the development of potentially complex dynamic circular-arc ray shooting data structures, we devise a simple and efficient method that "walks" along portions of $P_{0}$, testing for intersection with the circular arc, $\gamma$, from $b_{n+1}$ to $b_{n+1}^{\prime}$, where $b_{n+1}^{\prime}$ is the location of $b_{n+1}$ after the bend at $b_{i}$ has been performed.

In particular, we keep track of a "painted" portion of $P_{0}$, which corresponds to the subset of $P_{0}$ that has been "walked over". We consider the chain $P_{0}$ to be a degenerate simple polygon, having two sides which form a counterclockwise loop around $P_{0}$. We consider the case in which the bend at $b_{i}$ is a rotation of the segment $b_{i} b_{n+1}$ clockwise to the segment $b_{i} b_{n+1}^{\prime}$; the case of a counterclockwise bend at $b_{i}$ is handled similarly. When we perform a bend at $b_{i}$, we walk (counterclockwise) along the unpainted portions of $P_{0}$, between two points, $a$ and $a^{\prime}$, on the boundary of $P_{0}$, where $a$ and $a^{\prime}$ are defined according to cases that depend on the outcomes of two ray-shooting queries:


Fig. 10. Testing the foldability of the joint $b_{i}$. This example is intended to illustrate a generic step in the algorithm; for this particular chain, note that it is not feasible to make the bends $b_{1}, \ldots, b_{i-1}$ to get to the state shown.


Fig. 11. Case (a): Both of the rays $b_{i} b_{n+1}$ and $b_{i} b_{n+1}^{\prime}$ miss $P_{0}$ and go off to infinity.


Fig. 12. Case (b): Both of the rays $b_{i} b_{n+1}$ and $b_{i} b_{n+1}^{\prime}$ hit $P_{0}$. The walk extends from $a$ to $a^{\prime}$ over the highlighted portion of $P_{0}$, painting any previously unpainted portion of it.
(a) If both of the rays $\overrightarrow{b_{i} b_{n+1}}$ and $\overrightarrow{b_{i} b_{n+1}^{\prime}}$ miss $P_{0}$ (and go off to infinity), then there is nothing more to check: the rotation at $b_{i}$ can be done without interference with $P_{0}$, since $P_{0}$ is a (connected) polygonal chain lying in the complement of the wedge defined by $\overrightarrow{b_{i} b_{n+1}}$ and $\overrightarrow{b_{i} b_{n+1}^{\prime}}$. See Fig. 11.
(b) If both of the rays $\overrightarrow{b_{i} b_{n+1}}$ and $\overrightarrow{b_{i} b_{n+1}^{\prime}}$ hit $P_{0}$, then we let $a$ and $a^{\prime}$ (respectively) be the points on the boundary of $P_{0}$ where they first hit $P_{0}$. See Fig. 12.


Fig. 13. Case (c): Exactly one of the rays $b_{i} b_{n+1}$ and $b_{i} b_{n+1}^{\prime}$ hits $P_{0}: b_{i} b_{n+1}$ hits $P_{0}$ (left) or $b_{i} b_{n+1}^{\prime}$ hits $P_{0}$ (right). The walk extends from $a$ to $a^{\prime}$ over the highlighted portion of $P_{0}$, painting any previously unpainted portion of it.
(c) If exactly one of the rays $\overrightarrow{b_{i} b_{n+1}}$ and $\overrightarrow{b_{i} b_{n+1}^{\prime}}$ hits $P_{0}$ while the other misses $P_{0}$ (and goes off to infinity), then we define $a$ and $a^{\prime}$ as follows. Assume that the ray $\overrightarrow{b_{i} b_{n+1}}$ hits $P_{0}$ (and the ray $\overrightarrow{b_{i} b_{n+1}^{\prime}}$ misses $P_{0}$ ); the other case is handled similarly (see Fig. 13, right). Then, we define $a$ to be the point on the boundary of $P_{0}$ where the ray $\overrightarrow{b_{i} b_{n+1}}$ hits $P_{0}$, and we define $a^{\prime}$ to be the point on the boundary of $P_{0}$ where a ray from infinity in the direction $\overrightarrow{b_{n+1} b_{i}}$ (towards $b_{i}$ ) hits $P_{0}$. See Fig. 13, left.

During the walk from $a$ to $a^{\prime}$ along the boundary of $P_{0}$, we test each segment for intersection with the circular arc $\gamma$ in time $\mathrm{O}(1)$. Whenever we reach a portion of the boundary that is already painted, we skip over that portion, going immediately to its end. Already painted portions have endpoints that were determined by rays in previous steps of the painting procedure. Since there are only a total of $\mathrm{O}(n)$ rays (one per edge of $P$ ), this implies only $\mathrm{O}(n)$ endpoints of painted portions. As we walk, we mark the corresponding portions over which we walk as "painted". Since, by continuity, it is easy to see that the painted portion of any one segment of $P_{0}$ is connected, we know that we must encounter at least one vertex of $P_{0}$ between the time that the walk leaves a painted portion and the time that the walk enters the next painted portion. Thus, during a walk, we charge the tests that we do for intersection with $\gamma$ off to the vertices that are being painted. The remainder of the justification of the algorithm is based on two simple claims:

Claim 11. There is no need to walk back over a painted portion in order to check for intersections with an arc $\gamma$ at some later stage.

Proof. The fact that we need not walk over a painted portion testing again for intersections with $\gamma$ follows from the fact that with each bend in the sequence, the length of the segment $b_{i} b_{n+1}$ that we are rotating goes down by the length of the last link. Thus, if the motion of the tip, $b_{n+1}$, sweeps an arc $\gamma$ that does not reach a portion $\mu$ of the boundary of $P_{0}$ when the link $b_{i} b_{n+1}$ is straight, it cannot later be that a link $b_{j} b_{n+1}(j>i)$ can permit the tip $b_{n+1}$ to reach the same portion $\mu$ when pivoting is done about $b_{j}$; this is a consequence of the triangle inequality.

Claim 12. In testing for intersection with $\gamma$, we check enough of the chain $P_{0}$ : if any part of it intersects $\gamma$, then it must lie on the portion between $a$ and $a^{\prime}$ over which we walk.

Proof. In case (a), there is nothing to check. In case (b), the closed Jordan curve from $b_{i}$ to $a$ (along a straight segment), then along the boundary of the simple polygon $P_{0}$ to $a^{\prime}$, then back to $b_{i}$ (along a straight segment) forms the boundary of a region whose only intersection with $P_{0}$ is along the shared boundary from $a$ to $a^{\prime}$; thus, if $\gamma$ lies within this region (i.e., does not intersect the boundary of $P_{0}$ from $a$ to $a^{\prime}$ ), then $\gamma$ does not intersect any other portion of $P_{0}$. In case (c) we argue similarly, but we use the Jordan region defined by the segment from $b_{i}$ to $a$, the boundary of $P_{0}$ from $a$ to $a^{\prime}$, the ray from $a^{\prime}$ to infinity (in the direction of $\overrightarrow{b_{i} b_{n+1}}$ ), then the reverse of the ray $\overrightarrow{b_{i} b_{n+1}^{\prime}}$ back to $b_{i}$.

The total time for walking along the chain $P_{0}$ can be charged off to the vertices of $P$, resulting in time $\mathrm{O}(n)$ for tests of intersection with arcs $\gamma$, exclusive of the ray shooting time. The final time bound is then dominated by the time to perform $n$ straight ray shooting queries in a dynamic data structure for the changing polygonal chain $P_{0}$; these ray shooting queries are utilized both in testing for intersection with the segment $b_{i} b_{n+1}^{\prime}$ and in determining the points $a$ and $a^{\prime}$ that define the walk. These ray-shooting queries and updates are done in time $\mathrm{O}\left(\log ^{2} n\right)$ each, using existing techniques [16], leading to the claimed overall time bound.

Next, we turn to two other important classes of permutations. Again, for notational convenience, we consider the problem of verifying if it is feasible, in the order given by the permutation, to bend the joints $b_{i}$ from joint angle $\pi$ to final angle $\theta_{i}$, thereby transforming a straight wire into the final shape $P$. We say that a permutation is an outwards folding sequence (respectively, inwards folding sequence) if at any stage of the folding, the set of bends that have been completed, and therefore are not straight, is a subinterval, $b_{i}, b_{i+1}, \ldots, b_{j}$ (respectively, a pair of intervals $b_{1}, b_{2}, \ldots, b_{i}$ and $b_{j}, b_{j+1}, \ldots, b_{n}$ ); thus, the next bend to be performed is either $b_{i-1}$ or $b_{j+1}$ (respectively, $b_{i+1}$ or $b_{j-1}$ ). Inwards and outwards folding sequences are a subclass of permutations that model a constraint imposed by some forming machines. See Fig. 14. The identity permutation is a folding sequence that is a special case of both an inwards and an outwards folding sequence.


Fig. 14. An intermediate state ( $i, j$ ) in the bending of an outwards (left) and an inwards (right) folding sequence. For the outwards folding sequence on the left, the next bend is either $b_{i-1}$ or $b_{j+1}$; the new positions of the chain are shown dashed. For the inwards folding sequence on the right, the next bend is either $b_{i+1}$ or $b_{j-1}$.

We show that one can efficiently search for a folding sequence that is inwards or outwards. Our algorithms are based on dynamic programming.

First, consider the case of outwards folding sequences. We keep track of the state as the pair $(i, j)$ representing the interval of bends $\left(b_{i}, b_{i+1}, \ldots, b_{j}\right)$ already completed. We construct a graph $\mathcal{G}$ whose $\mathrm{O}\left(n^{2}\right)$ nodes are the states $(i, j)$ (with $1 \leqslant i<j \leqslant n$ ) and whose edges link states that correspond to the action of completing a bend at $b_{i-1}$ or $b_{j+1}$ (if the starting state is $(i, j)$ ). Thus, each node has constant degree. Our goal is to determine if there is a path in this graph from some $(i, i)$, for $i \in\{1,2, \ldots, n\}$ to $(1, n)$. We augment this graph with a special node $v_{0}$, linked to each node $(i, i)$. Then, our problem is readily solved in $\mathrm{O}\left(n^{2}\right)$ time once we have the graph constructed, since it is simply searching for a path from node $v_{0}$ to node $(1, n)$. (Alternatively, we can construct the graph as we search the graph for a path.) In order to construct the graph, we need to test whether bend $b_{i-1}$ or $b_{j+1}$ can be performed without intersecting the folded chain, $P^{\prime}$, linking $b_{i-1}$ to $b_{j+1}$. This is done in a manner very similar to that we described above for the case of identity permutations: we perform ray-shooting queries in time $\mathrm{O}\left(\log ^{2} n\right)$ and then use a "painting" procedure to keep track of the states of $2 n-2$ "walks" that determine circular-arc ray shooting queries. In particular, there is a separate painting procedure corresponding to each of the $n-1$ choices of $i$ and to each of the $n-1$ choices of $j$. For example, for a fixed choice of $i$, the painting procedure will consider each of the possible bends $b_{i+1}, \ldots, b_{n}$ in order, allowing us to amortize the cost of checking for intersections with the circular arc $\gamma$ associated with each bend. In total, the cost of the walks is $\mathrm{O}\left(n^{2}\right)$, while there may also be $\mathrm{O}\left(n^{2}\right)$ ray shooting queries (in a dynamically changing polygon). Thus, the total cost is dominated by the ray shooting queries, giving an overall time bound of $\mathrm{O}\left(n^{2} \log n\right)$.

For the case of an inwards folding sequence, we build a similar state graph and search it. However, the cost of testing if a bend is feasible is somewhat higher, as we do not have an especially efficient procedure for testing the foldability of a polygonal chain. (Our painting procedures exploit the fact that the link being folded is straight.) Thus, we apply the relatively naive method of testing feasibility given in Lemma 7, at a cost of $\mathrm{O}(n \log n)$ per test (which potentially improves to $\mathrm{O}(n)$ time, if our conjecture mentioned in the remark after the Lemma is true). Thus, the overall cost of the algorithm is dominated by the $\mathrm{O}\left(n^{2}\right)$ feasibility tests, at a total cost of $\mathrm{O}\left(n^{3} \log n\right)$. In summary, we have:

Theorem 13. In time $\mathrm{O}\left(n^{2} \log ^{2} n\right)$ one can determine if there is an outwards folding sequence; in time $\mathrm{O}\left(n^{3} \log n\right)$ one can determine if there is an inwards folding sequence.

## 5. Conclusion

We conclude with some open problems that are suggested by our work:
(1) Is the bend sequencing problem for wire folding strongly NP-complete, or is there a pseudo-polynomial-time algorithm? If not in wire bending, is it strongly NP-complete for the 3-dimensional sheet metal folding problem?
(2) Is it NP-hard to decide if a polygonal chain in three dimensions can be straightened? In [6] simple examples of locked chains in three dimensions are shown; can these be extended to a hardness proof for the decision problem?
(3) In practice, in order to make a bend using a punch and die on a press brake, it is necessary to consider accessibility constraints. For each bend operation, the die is placed on one side of the material, while the punch is placed on the other side. The bend is formed by pushing the punch into the die (which has a matching shape), with the material in between. (See Wang [34].) In the simplest model of this operation on a wire, we can consider the punch and the die to be oppositely directed rays that form a bend by coming together (from opposite sides of the wire) so that their apices meet at the bend point. The accessibility constraint in this simple model is that the rays representing the punch and die must be disjoint from the wire structure both at the initial placement of these "tools" and during the bend operation itself.
(4) Can the foldability of a permutation be decided in subquadratic time for wire bending? This would be possible if one had a dynamic data structure that will permit efficient (sublinear) queries for the foldability of a vertex.

## Acknowledgements

We thank an anonymous referee for pointing out some errors in an earlier draft and for many valuable suggestions to improve the presentation. We thank Steve Skiena for valuable input in the early stages of this research and, in particular, for contributing ideas to the hardness proof of Theorem 1. We also thank Erik Demaine for discussions leading to the current version of Theorem 2. Estie Arkin acknowledges support from the National Science Foundation (CCR-9732221, CCR-0098172) and HRL Laboratories. Sándor Fekete acknowledges support from the National Science Foundation (ECSE-8857642, CCR-9204585) during his time at Stony Brook (1992-93), when this research was initiated. Joe Mitchell acknowledges support from HRL Laboratories, the National Science Foundation (CCR-9732221, CCR-0098172), NASA Ames Research Center, Northrop-Grumman Corporation, Sandia National Labs, Seagull Technology, and Sun Microsystems.

## References

[1] E.M. Arkin, M.A. Bender, E.D. Demaine, M.L. Demaine, J.S.B. Mitchell, S. Sethia, S.S. Skiena, When can you fold a map?, in: F. Dehne, J.-R. Sack, R. Tamassia (Eds.), Proceedings of the 7th Workshop on Algorithms and Data Structures (WADS 2001), in: Lecture Notes in Computer Science, Vol. 2125, Providence, RI, 2001, pp. 401-413.
[2] B. Aronov, J. O'Rourke, Nonoverlap of the star unfolding, Discrete Comput. Geom. 8 (1992) 219-250.
[3] V. Ayyadevara, D. Bourne, K. Shimada, R.H. Sturges, Determining near optimal interference-free polyhedral configurations for stacking, in: Proceedings of IEEE International Symposium on Assembly and Task Planning, 1999, pp. 286-293.
[4] M. Bern, E.D. Demaine, D. Eppstein, E. Kuo, A. Mantler, J. Snoeyink, Ununfoldable polyhedra with convex faces, Computational Geometry 24 (2003) 51-62.
[5] M. Bern, B. Hayes, The complexity of flat origami, in: Proc. 7th ACM-SIAM Sympos. Discrete Algorithms, 1996, pp. 175183.
[6] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, S. Whitesides, Locked and unlocked polygonal chains in 3D, in: Proc. 10th ACM-SIAM Sympos. Discrete Algorithms, 1999, pp. 866-867.
[7] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, S. Whitesides, Locked and unlocked polygonal chains in 3D, Discrete Comput. Geom. 26 (3) (2001) 269-281.
[8] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, S. Robbins, I. Streinu, G. Toussaint, S. Whitesides, A note on reconfiguring tree linkages: Trees can lock, Discrete Appl. Math. 117 (1-3) (2002) 293-297.
[9] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, J. O’Rourke, M. Overmars, S. Robbins, S. Whitesides, Unfolding some classes of orthogonal polyhedra, in: Proc. 10th Canad. Conf. Comput. Geom., 1998, pp. 70-71. Fuller version in Elec. Proc. http://cgm.cs.mcgill.ca/cccg98/proceedings/welcome.hml.
[10] D. Bourne, C.-H. Wang, Design and manufacturing of sheet metal parts: Using features to resolve manufacturability problems, in: A. Busnaina (Ed.), Computer in Engineering, ASME, New York, 1995, pp. 745-753.
[11] R. Connelly, E. Demaine, G. Rote, Every polygon can be untangled, in: Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci., 2000, pp. 432-442.
[12] L.J. de Vin, J. de Vries, A.H. Streppel, H.J.J. Kals, PART-S, a CAPP system for small batch manufacturing of sheet metal components, in: Proc. of the 24th CIRP International Seminar on Manufacturing Systems, 1992, pp. 171-182.
[13] E.D. Demaine, Folding and Unfolding, PhD Thesis, Department of Computer Science, University of Waterloo, 2001.
[14] E.D. Demaine, M.L. Demaine, A. Lubiw, Folding and one straight cut suffice, in: Proc. 10th Annu. ACM-SIAM Sympos. Discrete Alg., 1999, pp. 891-892.
[15] E.D. Demaine, M.L. Demaine, J.S.B. Mitchell, Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami, Computational Geometry 16 (1) (2000) 3-21.
[16] M.T. Goodrich, R. Tamassia, Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations, J. Algorithms 23 (1997) 51-73.
[17] S.K. Gupta, D.A. Bourne, K.H. Kim, S.S. Krishnan, Automated process planning for sheet metal bending operations, J. Manufacturing Systems 17 (5) (1998) 338-360.
[18] T. Hull, On the mathematics of flat origamis, Congr. Numer. 100 (1994) 215-224.
[19] K.K. Kim, D. Bourne, S. Gupta, S.S. Krishnan, Automated process planning for robotic sheet metal bending operations, J. Manufacturing Syst. 17 (5) (1998) 338-360.
[20] R.J. Lang, Mathematical algorithms for origami design, Symmetry: Culture and Science 5 (2) (1994) 115-152.
[21] R.J. Lang, A computational algorithm for origami design, in: Proc. 12th Annu. ACM Sympos. Comput. Geom., 1996, pp. 98-105.
[22] W.J. Lenhart, S.H. Whitesides, Turning a polygon inside-out, in: Proc. 3rd Canad. Conf. Comput. Geom., 1991, pp. 66-69.
[23] W.J. Lenhart, S.H. Whitesides, Reconfiguring closed polygonal chains in Euclidean $d$-space, Discrete Comput. Geom. 13 (1995) 123-140.
[24] L. Lu, S. Akella, Folding cartons with fixtures: A motion planning approach, in: Proc. 1999 IEEE International Conference on Robotics and Automation Detroit, MI, 1999, pp. 1570-1576.
[25] A. Lubiw, J. O’Rourke, When can a polygon fold to a polytope? Technical Report 048, Dept. Comput. Sci., Smith College, June 1996. Presented at AMS Conf., 5 October 1996.
[26] M. Namiki, K. Fukuda, Unfolding 3-dimensional convex polytopes: A package for Mathematica 1.2 or 2.0, Mathematica Notebook, Univ. of Tokyo, 1993.
[27] J. O'Rourke, Computational geometry column 33, Internat. J. Comput. Geom. Appl. 8 (1999) 381-384; Also in SIGACT News 29 (2) (1998) 12-16, Issue 107.
[28] J. O'Rourke, Folding and unfolding in computational geometry, in: J. Akiyama, M. Kano, M. Urabe (Eds.), Proc. Japan Conf. Discrete and Computational Geometry, Tokyo, Japan, 9-12 December 1998, in: Lecture Notes in Computer Science, Vol. 1763, Springer-Verlag, Berlin, 2000, pp. 258-266.
[29] N. Pei, S. Whitesides, On the reachable regions of chains, in: Proc. 8th Canad. Conf. Comput. Geom., 1996, pp. 161-166.
[30] N. Pei, S. Whitesides, On folding rulers in regular polygons, in: Proc. 9th Canad. Conf. Comput. Geom., 1997, pp. 11-16.
[31] C. Schevon, J. O’Rourke. A conjecture on random unfoldings, Technical Report JHU-87/20, Johns Hopkins Univ., Baltimore, MD, 1987.
[32] I. Streinu, A combinatorial approach to planar non-colliding robot arm motion planning, in: Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci., 2000, pp. 443-453.
[33] M. van Kreveld, J. Snoeyink, S. Whitesides, Folding rulers inside triangles, Discrete Comput. Geom. 15 (1996) 265-285.
[34] C.-H. Wang, Manufacturability-driven decomposition of sheet metal products, PhD Thesis, Carnegie Mellon University, The Robotics Institute, 1997.
[35] C.-H. Wang, D. Bourne, Concurrent decomposition for sheet metal products, in: ASME Design Engineering Technical Conference, Sacramento, 1997.
[36] C.-H. Wang, R.H. Sturges, Bendcad: A design system for concurrent multiple representations of parts, J. Intelligent Manufacturing 7 (1996) 133-144.


[^0]:    * Corresponding author.

    E-mail addresses: estie@ams.sunysb.edu (E.M. Arkin), sandor.fekete@tu-bs.de (S.P. Fekete), jsbm@ams.sunysb.edu (J.S.B. Mitchell).

[^1]:    ${ }^{1}$ We thank Karel Zikan for introducing to us the hydraulic tube bending problem at Boeing's factory.

