International Journal of Computational Geometry & Applications

COMPUTING SIGNED PERMUTATIONS OF POLYGONS*

GREG ALOUPIS

Institute of Information Science, Academia Sinica, 128 Academia Road, Section 2
Nankang, Taipei 115, Taiwan
aloupis.greg@gmail.com

PROSENJIT BOSE

School of Computer Science, Carleton University, 5302 Herzberg Lab, 1125 Colonel By Drive
Ottawa, Ontario K1S 5B6, Canada
jit@scs.carleton.ca

ERIK D. DEMAINE

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology,
32 Vassar Street Cambridge, MA 02139, USA
edemaine@mit.edu

STEFAN LANGERMAN

Maître de Recherches du FRS-FNRS
Département d'Informatique, Université Libre de Bruxelles, Boulevard du Triomphe
CP 212 Bruxelles, Belgique.
stefan.langerman@ulb.ac.be

HENK MEIJER

Science Department, Roosevelt Academy, P.O. Box 94
Middelburg NL-4330 AB, Netherlands
h.meijer@roac.nl

MARK OVERMARS

Department of Information and Computing Sciences, Utrecht University, Padualaan 14
De Uithof, P.O. Box 80.089. Utrecht 3508 TB, Netherlands
m.h.overmars@cs.uu.nl

GODFRIED T. TOUSSAINT

School of Computer Science, McGill University, 3480 University Street
Montreal, Quebec H3A 2A7, Canada
godfried@cs.mcgill.ca

Received 16 April 2009
Revised 19 January 2010
Communicated by Christian Knauer

*This research was partly funded by NSERC and FCAR.

ABSTRACT

Given a planar polygon (or chain) with a list of edges $\{e_1, e_2, e_3, \ldots, e_{n-1}, e_n\}$, we examine the effect of several operations that *permute* this edge list, resulting in the formation of a new polygon. The main operations that we consider are: *reversals* which involve inverting the order of a sublist, *transpositions* which involve interchanging subchains (sublists), and *edge-swaps* which are a special case and involve interchanging two consecutive edges. When each edge of the given polygon has also been assigned a *direction* we say that the polygon is *signed*. In this case any edge involved in a reversal changes direction.

We show that a star-shaped polygon can be convexified using $O(n^2)$ edge-swaps, while maintaining simplicity, and that this is tight in the worst case. We show that determining whether a signed polygon P can be transformed to one that has rotational or mirror symmetry with P, using transpositions, takes $\Theta(n \log n)$ time. We prove that the problem of deciding whether transpositions can modify a polygon to fit inside a rectangle is weakly NP-complete. Finally we give an $O(n \log n)$ time algorithm to compute the maximum endpoint distance for an oriented chain.

Keywords: Computational geometry; polygonal reconfiguration; geometric permutation.

1. Introduction

The work in this paper belongs to the broad topic of *polygonal reconfiguration*. There are several operations found in the literature that will transform a given planar polygon P to a new one P'. Typically the polygons are simple and often the operations act only on small components of the input. Problems of interest include proving that a sequence of a particular operation suffices to reconfigure between any two polygons, and determining the computational complexity of doing so.

Restricting ourselves to finite-step reconfigurations, one of the earliest types of operations is the *Erdös flip* [1], where the chain between two vertices is rotated (flipped) about the axis joining them. Grenander et al. [2], as well as Guibas et al. [3], studied polygonal morphing by shifting and extending the length of edges. Pach and Tardos [4] studied the complexity of untangling a crossing polygon by changing vertex coordinates. Hernando et al. [5] defined an operation that transforms the boundary of a polygon embedded on a fixed point set.

Most related to our work is the well-studied *flipturn*, introduced by Joss and Shannon. According to their supervisor, Grunbaum [6], they never published their introductory work on this topic. Flipturns are similar to flips, except that the chain to be transformed must be between two convex hull vertices, and it is rigidly rotated by π so that the endpoints exchange positions. Our *reversal* operation removes the convex hull requirement for the vertices. Dubins et al. [7] worked on reconfiguring random self-avoiding closed lattice walks, with operations equivalent to the flip and flipturn. In fact their inspiration came from previous work on open random walks, from the physics community.

While studying polymer conformations, MacDonald et al. [8] described the method of *reptation*, where a chain is reconfigured by taking its last edge and placing it in front. This is a special case of our *single-edge transposition* operation, defined later on.

For more details on the history and algorithmic developments of the operations mentioned above, the reader may refer to [9].

We now turn to the reconfigurations studied in this paper. Specifically, we look at how a polygon can be permuted. Our inspiration comes from bioinformatics, where much focus has been placed on the problem of sorting a permutation of n integers by *reversals*.[10,11] As one might guess, a single reversal is applied to a consecutive set of these integers and the result is that their order is inverted. The key problem that arises is determining the minimum number of reversals necessary to sort a given permutation. This number is called the *reversal distance* of the permutation. A variation of this problem involves *signed* permutations.[12] In this case any integer affected by a given reversal also changes *parity*.

Each of these interesting combinatorial problems has its roots in bioinformatics and molecular biology.[10,11,12,13,14] Specifically, genomes have been modeled as linear or cyclic sequences, where each element in a sequence is a *block* of smaller elements that are never separated. A popular model for mutation involves inverting parts of these sequences. In order to determine the number of such mutations needed to transform one genome to another, one may compute the reversal distance of the associated permutations. An extension of this model is to consider the direction of each *block*. This leads to the study of signed permutations. We illustrate signed inversions in Figure 1 which has been modified from a figure found in the literature.[11]

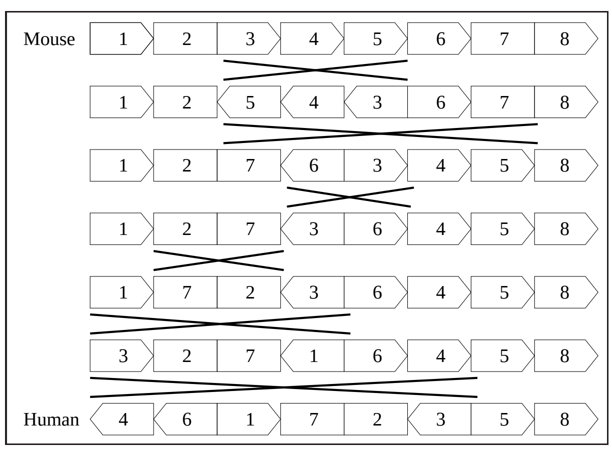

Fig. 1. A most parsimonious evolutionary scenario for the transformation of human into mouse chromosome assuming that the X chromosome evolves solely by inversions. Each block represents a conserved linkage group of genes. Reversal distance is equal to six.

The problem of computing *transposition distance* also stems from bioinformatics. In this case, a transposition involves exchanging two disjoint sets of consecutive integers in a permutation. Computing reversal distance has been shown to be NPhard [15] for unsigned permutations, but for the signed version a linear time algorithm exists.[12] Computing transposition distance is of unknown complexity.[14] The reader may also be interested in the following. [16,17,18]

In this paper we extend the ideas mentioned above from one dimension to two. Instead of considering permutations of integers, we consider permutations of *edges* which form polygons or chains. We define operations such as *edge-swaps*, *reversals* and *transpositions*, in analogy to $\mathbb{R}^1$. We introduce the notions of *signed permutations of polygons and chains*. These concepts give rise to a wide range of problems to be solved. Although there may be some links to bioinformatics, the main thrust of this paper is the study and classification of polygonal reconfigurations.

Since the original presentation of this work to the Computational Geometry community[19], some of our results have contributed to the inspiration of new research. For instance, Chan et al. used simple properties of transpositions in an unexpected application: computing the optimal trajectory of a blind search, that starts at an unknown location in a river of known width, and that is guaranteed to find the river shore.[20]

2. Definitions

First we introduce the notion of a *signed polygon* or *signed permutation* of a polygon. Any polygon P can be described by a list of edges $\{e_1, e_2, e_3, \ldots, e_{n-1}, e_n\}$. A signed polygon is no different, except that each edge is also assigned a direction. The same holds for chains. This is a generalization of the notion of parity that is used in $\mathbb{R}^1$. If the directions of all edges are consistent as we traverse a polygon or chain, then this polygon or chain is *oriented*. In Figure 2 we illustrate some signed polygons and chains.

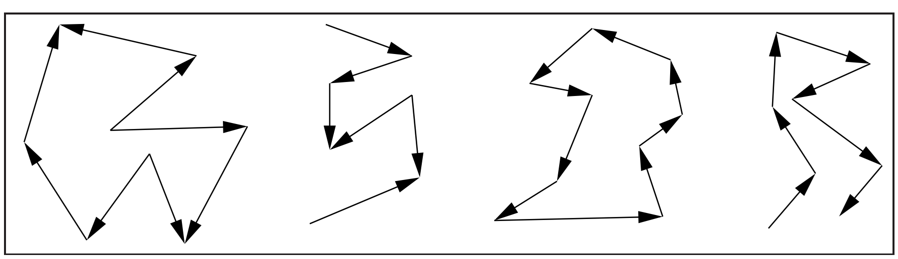

Fig. 2. From left to right, a signed polygon, signed chain, oriented polygon, oriented chain.

For the following operations defined on (possibly signed) polygons, edge directions are not critical. However the definitions become intuitive if we consider the polygons to be oriented, regardless of what the edge directions might be.

A *transposition* of two edges A and B involves interchanging their positions (in the cyclic sequence) so that the resulting polygon remains oriented. The resulting configuration is unique. This is illustrated in Figure 3. If A and B are consecutive, this operation is defined as an *edge-swap* or plainly *swap* (Figure 4a). It is not difficult to see that entire subchains may also be transposed. A *single-edge* transposition involves transposing an edge with an empty subchain. One may also think of this operation as a transposition between the single edge and one of its neighboring subchains (Figure 4b).

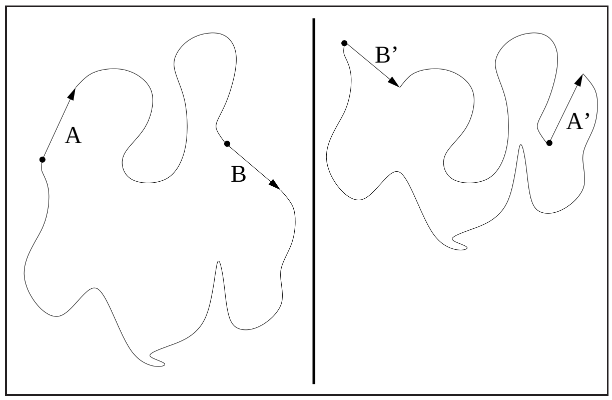

Fig. 3. Transposing two edges A and B.

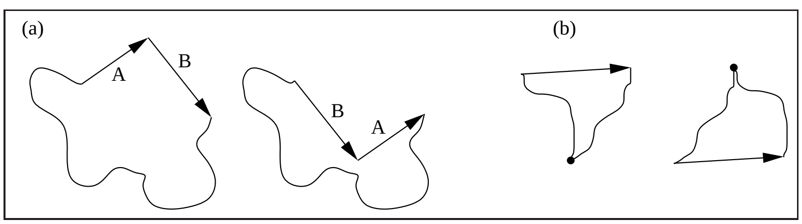

Fig. 4. (a) An edge swap. (b) A single-edge transposition.

A *reversal* of a subchain belonging to a polygon involves inverting the order of the edges in the subchain. Geometrically this rotates the subchain rigidly in the plane by an angle of π so that its endpoints are placed exactly at each other's original location. For unsigned polygons this is identical to the *flipturn*, mentioned earlier. A flipturn is defined only on a subchain defining a *pocket* of a polygon. Pockets are subchains linking two consecutive vertices of the convex hull. Here we allow reversals to take place on any subchain, not only on pockets. For signed polygons the direction of each edge involved in the reversal is switched, as is done for parity in $\mathbb{R}^1$. In Figure 5 we illustrate a reversal of subchain $\{e_i, \ldots, e_j\}$ for a signed (initially oriented) polygon.

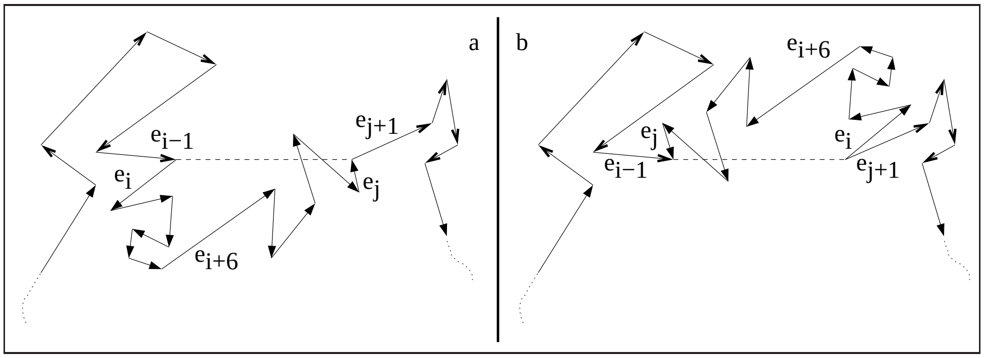

Fig. 5. Reversing a subchain of a signed polygon.

One can see that for unsigned polygons, an edge-swap is merely a transposition *or* a reversal of two consecutive edges. For signed polygons there is a difference in the resulting direction of each edge.

As mentioned, the operations mentioned above have been described for oriented polygons. In fact this orientation merely helps to describe the operations. The actual directions of edges in a signed polygon will not affect the operations or the resulting embedding. However, for chains alternate definitions exist. For example consider the oriented chain in Figure 6. We may choose to perform a reversal on edges (A, B, C)

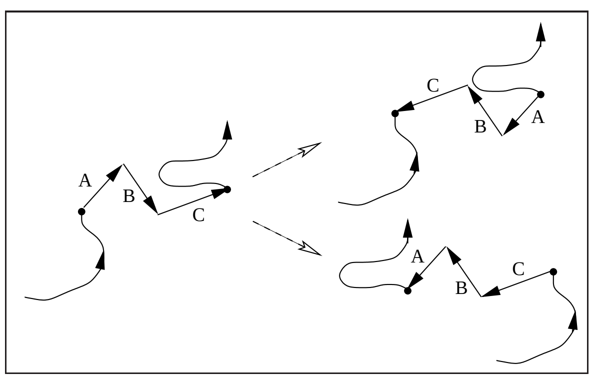

Fig. 6. Two ways that a reversal can be defined on a chain.

in at least two ways. One way (shown on top) is identical to what is done for polygons. This is convenient but also means that the endpoints of the chain will never move. A second way (shown at the bottom) is to preserve orientation. This may allow the chain to form more interesting configurations. We use the latter definition in Theorem 6 in the next section.

3. Permuting Polygons

Scott has shown that precisely two permutations of an edge list form oriented convex polygons, and these have maximal area.[21] It is also known that if the longest edge of a polygon has unit length, this polygon can be permuted to fit into a circle of radius $\sqrt{5}$.[22]

For the remainder of this section we present our results concerning permutations of polygons or chains. We impose the restriction that simplicity must be maintained at all times, unless mentioned otherwise.

In Figure $7a$ we show a polygon which does not admit any edge-swaps. Examples such as this one can be extended easily to create any n-gon which will not admit edge-swaps. In Figure $7b$ we show a polygon which does not admit single-edge transpositions, with the exception of a few edge-swaps for some edges that are almost collinear. These transpositions cannot change the basic shape. Thus we see that sometimes local permutations will not be sufficient to achieve desired reconfigurations.

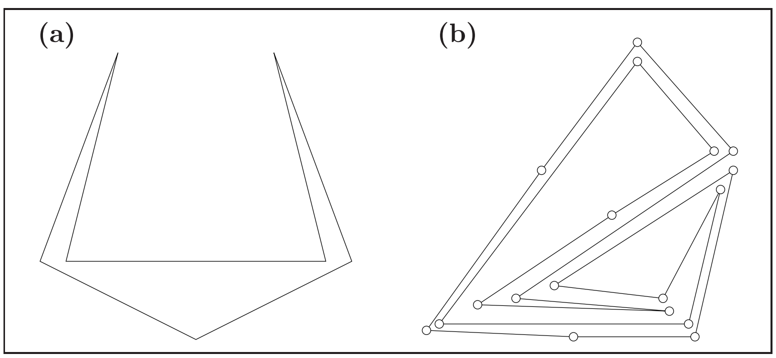

Fig. 7. Polygon (a) does not admit edge-swaps. Polygon (b) does not admit single-edge transpositions.

Theorem 1. *A simple polygon can be convexified with $O(n^2)$ reversals while maintaining simplicity after each reversal.*

Proof. This result holds for the more restricted reversal operation of flipturns.[23] □

Theorem 2. *A star-shaped polygon can be convexified with $O(n^2)$ edge-swaps while maintaining star-shapedness after each edge-swap, and this bound is tight in the worst case.*

Proof. Let k be a point in the kernel and without loss of generality suppose that the polygon is oriented clockwise. If the polygon is not convex, there must exist two successive edges $\overrightarrow{ab}$ and $\overrightarrow{bc}$ which form a left hand turn (see Figure 8a).

Since the polygon is star-shaped, b is the only vertex in the cone formed by the

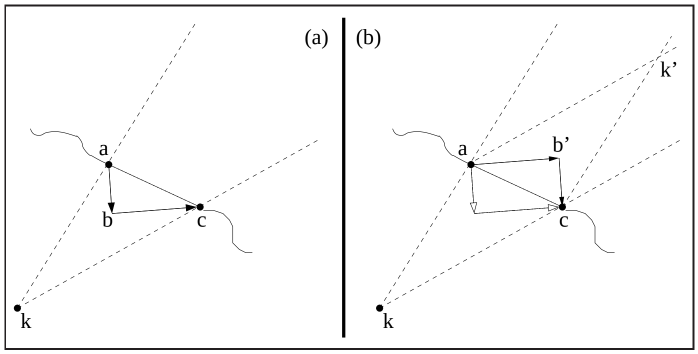

Fig. 8. Using edge-swaps to convexify a star-shaped polygon.

half-lines ka and kc. If we edge-swap $\overrightarrow{ab}$ and $\overrightarrow{bc}$, we obtain the configuration shown in Figure 8b. The new position of b (shown as b') must be somewhere in the triangle (a, c, k'), where k' is the unique point that completes a convex parallelogram with a, c and k. The swapped edges are still visible from k, and they do not interfere with the other edges of the polygon. Thus the polygon remains star-shaped. Furthermore any point in the kernel remains in the kernel and any point in the polygon remains in the polygon.

Every edge e can be found only within a halfplane determined by a line parallel to e that passes through k. Otherwise the "interior" part of e would not be visible from k. Now suppose that two edges, $\overrightarrow{ab}$ and $\overrightarrow{cd}$ form a right hand turn. This means that b and c coincide in the intersection of the two halfplanes, as shown in Figure 9. It is impossible to move these edges continuously within their respective halfplanes and into a left hand turn without obstructing visibility from k to either b or c. But

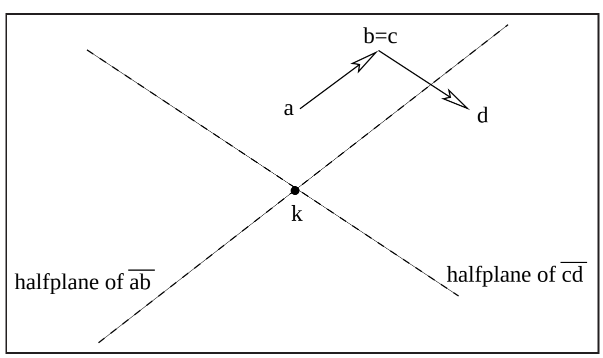

Fig. 9. Any pair of edges can be swapped at most once.

even without continuous motions, one edge cannot "jump" over the other during a swap. This would require the existence of another edge incident to b (or c), which would mean some part of the polygon is not visible from k. Thus once a pair of edges forming a left hand turn are swapped, they will never form a left hand turn again. The polygon will become convex only when there are no swaps to be made on left hand turns. Since any pair of edges can be swapped at most once, $O(n^2)$ swaps suffice to convexify a star-shaped polygon. In Figure 10 we show that this bound is tight. Every edge e_i $(2 \leq i \leq n-2)$ must be swapped with edges $e_1, \ldots, e_{i-1}$ for the polygon to become convex. □

Since our preliminary publication of this work in [19], we have learned that Ballinger [24] independently discovered Theorem 2.

For the following two theorems we do not enforce simplicity.

Theorem 3. *Determining whether a signed polygon can be permuted using transpositions so that its shape is rotated by an angle of* π *takes* $\Theta(n \log n)$ *time in the algebraic decision tree model of computation.*

Proof. Since only transpositions are allowed, each edge of the polygon must have its *opposite* also present in the polygon. This property also suffices if we do not impose the restriction of maintaining simplicity at all times. By "opposite" we mean an edge with the same angle, but opposite direction. For example, in Figure 11 edges a, b, c, d, e of the polygon on the left are matched by edges $\bar{a}, \bar{b}, \bar{c}, \bar{d}, \bar{e}$. This means that the shape of this polygon can be rotated by an angle of π (shown on right) with appropriate transpositions.

If we translate every edge to the origin (so that they are directed away from the origin), we obtain a set of n points. The shape of the given polygon can be rotated if and only if every such point has a reflection through the origin. This can be determined in $O(n \log n)$ time with a radial sort, and the matching lower bound

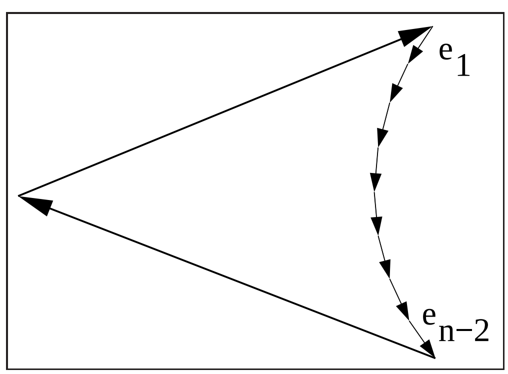

Fig. 10. A star-shaped polygon which requires $\Omega(n^2)$ edge-swaps to become convex.

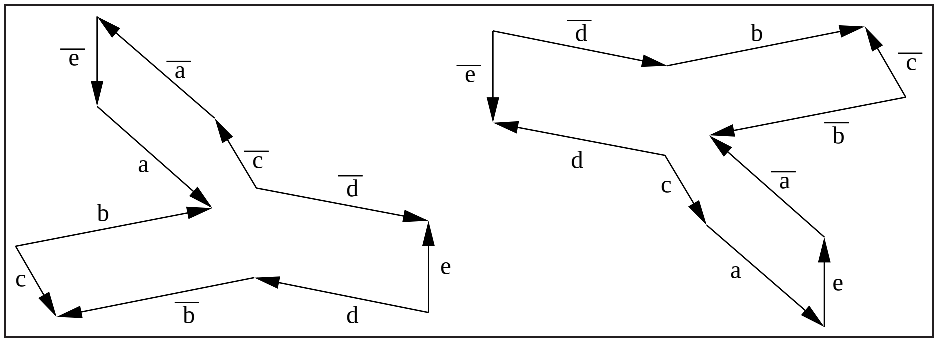

Fig. 11. Left: a signed polygon for which every edge is matched by an "opposite". Right: a permutation of the polygon with the same shape rotated by an angle of π.

is obtained by a reduction from *Set Equality*.[25] □

Theorem 4. *Determining whether we can obtain the mirror image of a signed polygon using transpositions takes* $\Theta(n \log n)$ *time in the algebraic decision tree model of computation.*

Proof. In order to be able to obtain a mirror image, there must exist an axis through which every edge has its reflection present (allowing translation). For example consider the polygon on the left in Figure 12. If we take a vertical line as an axis of symmetry, then edges d and j are reflections of each other. The same holds for pairs (b, h) and (f, k). Vertical edges do not need a matching edge. If such an axis exists, then a mirror image of the polygon can be obtained using transpositions. As in Theorem 3 we can place every edge at the origin to obtain a set of n points. The symmetries of this point set can be found using the Knuth-Morris-Pratt string matching algorithm.[26] The overall time complexity is $O(n \log n)$. This is pointed out by Eades who also mentions that such reflection tests have $\Omega(n \log n)$ lower bounds on fixed degree decision tree machines.[25] □

Theorem 5. *Given an oriented polygon P and a rectangle R, deciding whether P can be permuted by transpositions into an oriented polygon P' that can be drawn inside R is (weakly) NP-complete.* [a]

Proof. Consider an integer partition problem with $S = \{a_0, a_1, \ldots, a_{n-1}\}$ and $a_i > 0$ for all i. Let $A = \sum_{a \in S} a/2$. Deciding whether there is a subset S' of S with $\sum_{a \in S'} a = A$ is (weakly) NP-complete. Consider the following polygon P. Denote the edges of P in counter-clockwise order by $\{e_0, e_1, \ldots, e_{2n+3}\}$. The edges with

[a]This result also holds if P' is to be placed inside a strip or circle, instead of a rectangle.

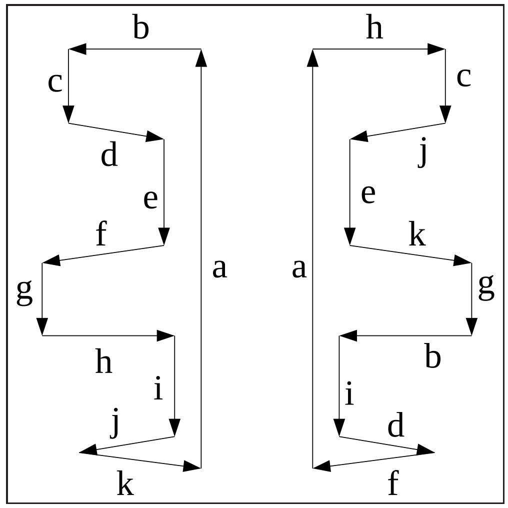

Fig. 12. Two polygons that are mirror images and have different permutations of the same edge list.

even indices are parallel to the x-axis; the edges with odd indices are parallel to the y-axis. Let ϵ be a positive number less than one. The edge e_0 has length $a_0 + \epsilon$. Edges e_i for $i = 2, 4, 6, \ldots, 2n-2$ have length $a_{i/2}$. Edge e_{2n} has length A. Edge e_{2n+2} has length $A+\epsilon$. Edges e_i for $i = 1, 3, 5, \ldots, 2n+1$ have length 1. Edge e_{2n+3} has length $n+1$.

We also assign directions to the edges, so that the edges form a counter-clockwise traversal of P. All edges of length 1 go up. The edges e_i for $i = 0, 2, 4, \ldots, 2n-2$ go from left to right. The few remaining edges go down and right to left, as illustrated in Figure 13(a) with n=7.

Let R be a rectangle of size $A+\epsilon$ by $n+1$. W.l.o.g assume that R has $(-\epsilon, 0)$ and $(A, n+1)$ as its left-bottom and right-top corner. Suppose P can be permuted into a polygon P' that can be drawn in the rectangle R. Again w.l.o.g assume that e_{2n+2} of P' lies along the top side of R and e_{2n+3} along the left side of R. This implies that the left and right endpoints of e_{2n} are $(0, y)$ and (A, y) for some value of y with $1 \leq y \leq n$. Moreover the edge e_0 lies below e_{2n}. The edges form a counter-clockwise traversal of P'. Since e_{2n} has a direction that goes from right to left, the horizontal edges above e_{2n} connect the left endpoint of e_{2n} with the right endpoint of e_{2n+2}, so their lengths must add up to A. Therefore the partition has a solution if and only if P can be permuted into a polygon P' that fits in R. Figure 13(b) shows a permutation of the polygon in Figure 13(a) that fits in rectangle R. □

Theorem 6. *The maximum endpoint distance over all permutations of an oriented chain can be computed in $O(n \log n)$ time.*

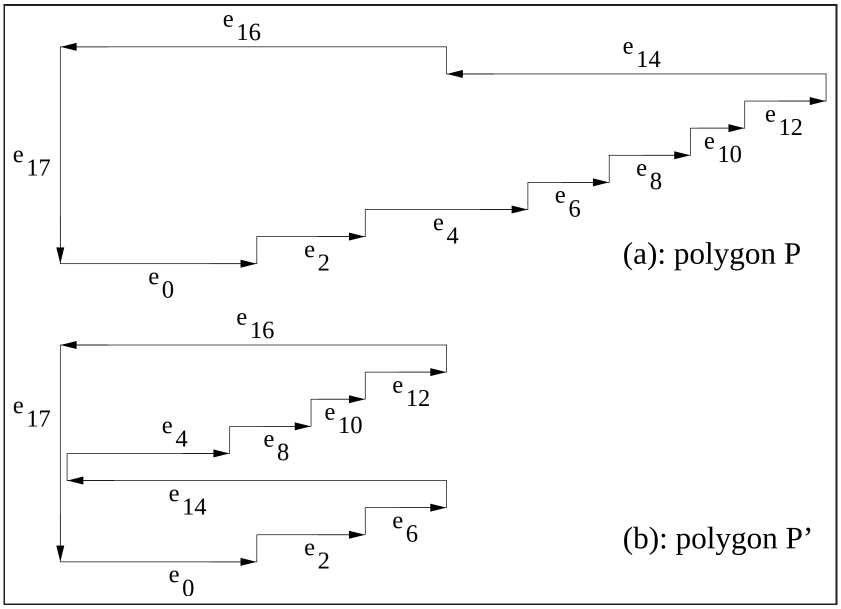

Fig. 13. Polygons P and P' with 18 vertices.

Proof. Fix one endpoint at the origin. Endpoint distance depends only on the direction of each edge. If we knew the direction in which to position the second endpoint, it would be a simple matter to select the direction of each edge in order to maximize the distance. Position two vectors at the origin for each edge, representing its possible directions. Sort the vectors radially and compute the sum of all vectors in one halfplane determined by a line ℓ through the origin. This represents the maximum distance in a direction perpendicular to ℓ. By rotating ℓ and updating the vector sum whenever a vector enters or exits the rotating halfplane, we obtain the endpoint distance over all directions. The time complexity is dominated by the sorting step, so the entire procedure takes $O(n \log n)$ time using $O(n)$ space. □

Acknowledgements

Much of this work was completed at the *Workshop on Geometric Aspects of Molecular Reconfiguration* organized by Godfried Toussaint at the Bellairs Research Institute of McGill University in Barbados, February 2002. We appreciate the helpful discussions with the other participants: David Bremner, Vida Dujmovic, Herbert Edelsbrunner, Jeff Erickson, Ferran Hurtado, Patrick Morin, Joseph O’Rourke, Suneeta Ramaswami, Ileana Streinu, Yusu Wang. We also appreciate the comments of an anonymous reviewer for the short version of this paper.[19]

References

1. Paul Erdős. Problem number 3763. *American Mathematical Monthly*, 42:627, 1935.
2. Ulf Grenander and Yunshyong Chow and Daniel M. Keenan. Hands: A Pattern Theoretic Study of Biological Shapes. Springer-Verlag, 1991.
3. Leonidas Guibas and John Hershberger and Subhash Suri. Morphing Simple Polygons. *Discrete and Computational Geometry*, 24:1–34, 2000.
4. Janos Pach and Gabor Tardos. Untangling a Polygon. In *Graph Drawing*, pages 154–161, 2001.
5. Carmen Hernando and Michael E. Houle and Ferran Hurtado. On local transformation of polygons with visibility properties. *Theoretical Computer Science*, 289(2):919–937, 2002.
6. Branko Grunbaum and Joseph Zaks. Convexification of polygons by flips and by flipturns. *Discrete Mathematics*, 241:333–342, 2001.
7. Lester E. Dubins and Alon Orlitsky and Jim A. Reeds and Lawrence A. Shepp. Self-avoiding random loops. *IEEE Transactions on Information Theory*, 34:1509–1516, 1988.
8. Bruce MacDonald and Naeem Jan and Douglas L. Hunter and Michael O. Steinitz. Polymer conformations through wiggling. *Journal of Physics A: Mathematical and General Physics*, 18:2627–2631, 1985.
9. Greg Aloupis. Reconfigurations of Polygonal Structures. Ph.D. thesis, McGill University, 2005.
10. Sridhar Hannenhalli and Pavel A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proc. 27th Annual ACM Symposium on the Theory of Computing*, pages 178–189, 1998.
11. Vineet Bafna and Pavel A. Pevzner. Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of X chromosome. *Mol. Biol. Evol.*, 12(2):239–246, 1995.
12. David A. Bader, Bernard M.E. Moret, and Mi Yan. A linear time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.
13. Sridhar Hannenhalli and Pavel A. Pevzner. To cut or not to cut (applications of comparative physical maps in molecular evolution). In *Seventh Anuual ACM-SIAM Symposium on Discrete Algorithms*, pages 304–313, 1996.
14. Mary E. Cosner, Robert K. Jansen, Bernard M.E. Moret, Linda A. Raubeson, Li-San Wang, Tandy Warnow, and Stacia Wyman. A new fast heuristic for computing the breakpoint phylogeny and experimental phylogenetic analyses of real and synthetic data. In *Proc. 8th Int'l Conf. on Intelligent Systems for Molecular Biology ISMB-2000, San Diego*, pages 104–115, 2000.
15. Alberto Caprara. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM J. Discrete Math.*, 12(1):91–110, 1999.
16. Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. 1.375-approximation algorithm for sorting by reversals. In *European Symposium on Algorithms (ESA)*, pages 200–210, 2002.
17. Vineet Bafna and Pavel A. Pevzner. Sorting by transpositions. *SIAM J. Discrete Math.*, 11(2):224–240, May 1998.
18. Henrik Eriksson, Kimmo Eriksson, Johan Karlander, Lars Svensson, and Johan Wästlund. Sorting a bridge hand. *Discrete Math.*, 241(1–3):289–300, October 2001.
19. Greg Aloupis, Prosenjit Bose, Erik D. Demaine, Stefan Langerman, Henk Meijer, Mark Overmars, and Godfried Toussaint. Computing signed permutations of polygons. In *Proceedings of the 14th Canadian Conference on Computational Geometry*, pages 68–71 (extended version available in electronic proceedings), August 2002.

20. Timothy Chan, Alexander Golynski, Alejandro Lopez-Ortiz, and Claude-Guy Quimper. Curves of width one and the river shore problem. In *Proceedings of the 15th Canadian Conference on Computational Geometry*, pages 73–75, 2003. See also: The asteroid surveying problem and other puzzles, *Proc. 19th ACM Symposium on Computational Geometry (SoCG)*, p.372-373, 2003.
21. Paul R. Scott. On a family of polygons. *Mathematics Magazine*, 55(2):104–105, March 1982.
22. Lidija I. Golovina and Isaak M. Yaglom. *Induction in Geometry*. Mir Publishers, Moscow, 1979.
23. Hee-Kap Ahn, Prosenjit Bose, Jurek Czyzowicz, Nicolas Hanusse, Evangelos Kranakis, and Pat Morin. Flipping your lid. *Geombinatorics*, X(2):57–63, 2000.
24. Brad Ballinger. Length-preserving transformations on polygons. Ph.D. thesis, University of California, Davis. 2003.
25. Peter Eades. Symmetry finding algorithms. In *Computational Morphology: A Computational Geometric Approach to the Analysis of Form, Godfried T. Toussaint (ed.), North-Holland*. 1988.
26. Donald E. Knuth, James H. Morris, and Vaughan R. Pratt. Fast pattern matching in strings. *SIAM Journal of Computing*, 6:322–350, 1977.