Computing the Wingspan of a Butterfly
(Extended Abstract)

BINAY BHATTACHARYA*
PETER EGYED†
GODFRIED TOUSSAINT†

Abstract

A simple polygon P is a butterfly polygon provided it contains exactly four convex vertices labeled a, b, c, d in order such that the line segments $[a, c]$ and $[b, d]$ intersect. In this paper we consider a fundamental geometric minimization problem which we call computing the wingspan of a butterfly. The problem involves computing a shortest line segment joining a pair of opposite concave chains of an n vertex butterfly polygon, where the line segment is constrained to lie inside the polygon. We propose an $O(\log^2 n)$ time algorithm for computing such a line segment. This result finds application in computing shortest transversals of sets, minimal sets of external visibility, and shortest lines-of-sight.

1 Introduction

Several geometric optimization problems involving the computation of shortest distances or shortest line segments have been considered in the computing literature. For example, Edelsbrunner showed how to compute the shortest distances between two convex polygons with a total of n vertices in $O(\log n)$ time [7]. Dobkin and Kirkpatrick proposed a solution to the three dimensional version of this problem. They gave an $O(n)$ time algorithm for computing the shortest distance between a pair of convex polyhedra with a total of n vertices [6]. A similar problem was considered by McKenna and Toussaint who showed that the minimum vertex distance between a pair of convex polygons with a total of n vertices could be computed in $O(n)$ time [8]. In this paper we consider a closely related problem which we call *computing the wingspan of a butterfly*. A *butterfly polygon* is a simple polygon containing exactly four convex vertices labeled a, b, c, d in order such that the line segments $[a, c]$ and $[b, d]$ intersect. The problem is stated as follows: given an n vertex butterfly polygon P and a pair of opposite concave chains of P, compute a shortest line segment joining the chains that lies inside P. We propose an $O(\log^2 n)$ time algorithm that solves this problem. Note that our algorithm also gives an $O(\log^2 n)$ time solution to the problem of computing the shortest distance between two convex polygons that is constrained to pass through a point.

*School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada V5A 1S6.

†School of Computer Science, McGill University, Montreal, Quebec, Canada H3A 2A7.

2 Geometric Theory

Consider an n vertex, butterfly polygon P with convex vertices labeled v_1, v_2, v_3, v_4 in counterclockwise (CCW) order. Without loss of generality, assume v_1, v_2, v_3 and v_4 are located in the 1st, 2nd, 3rd and 4th quadrants of the plane, respectively. Furthermore, label the concave chains of P by T, B, L and R where $T = (v_1, \ldots, v_2)$, $L = (v_2, \ldots, v_3)$, $B = (v_3, \ldots, v_4)$ and $R = (v_4, \ldots, v_1)$. We concentrate on computing a shortest line segment lying inside P that joins L to R. Obviously, not every line segment joining L to R lies inside P. However, if r is a shortest line segment joining L to R and r also lies inside P, then r is a solution to our problem.

Let l_1 and l_2 be the common separating tangents of T and B where $slope(l_1) < slope(l_2)$. As well, let a and c denote the intersection points of l_2 with R and L, respectively, and let b and d denote the intersection points of l_1 with L and R, respectively. Finally, let a' and c' denote the tangent points of l_2 with T and B, respectively, and let b' and d' denote the tangent points of l_1 with T and B, respectively. Observe that every line segment joining L to R that has an endpoint on any of the CCW chains $(a, \ldots, v_1)$, $(v_2, \ldots, b)$, $(c, \ldots, v_3)$ or $(v_4, \ldots, d)$ intersects either T or B. Hence, we may replace the CCW chains $(a, \ldots, a')$, $(b', \ldots, b)$, $(c, \ldots, c')$ and $(d', \ldots, d)$ of P by the line segments $[a, a']$, $[b', b]$, $[c, c']$ and $[d', d]$, respectively, without affecting the solution to our problem. Since this operation can be carried out in $O(\log n)$ time, we will assume, without loss of generality, that the line segments $[v_1, v_3]$ and $[v_2, v_4]$

are contained in P.

Suppose there is a unique line segment realizing the shortest distance between L and R. This line segment must have as an endpoint a vertex of either L or R. Now suppose there is more than one line segment realizing the shortest distance between L and R. In this case, there must exist parallel edges, e_L and e_R, such that every shortest line segment joining L to R also joins e_L to e_R and is perpendicular to e_L and e_R. Furthermore, exactly two of the shortest line segments joining L to R must have as an endpoint a vertex of either L or R.

Let $s = [a, b]$ be a shortest line segment joining L to R where $a \in L$, $b \in R$ and either a is a vertex of L or b is a vertex of R. Furthermore, if s intersects $T(B)$ then s is the highest (lowest) such line segment. We show later that if s intersects $T(B)$ then every shortest line segment that lies inside P and joins L to R is tangent to $T(B)$. This suggests that an appropriate strategy for solving our problem would be to break the problem into two subproblems, as follows. Determine whether s lies inside P, and if it does, report it as the solution. If s does not lie inside P, then over all line segments joining L to R and tangent to $T(B)$, compute the shortest. The first subproblem is equivalent to computing a shortest line segment joining two disjoint convex polygons, a problem that has been solved by Edelsbrunner in $O(\log n)$ time [7]. We now show how the second problem can be converted into an instance of a fundamental geometric optimization problem.

A *concave-cone* C is composed of two concave chains $C_L = (p_1, p_2, \ldots, p_L)$ and $C_R = (q_1, q_2, \ldots, q_R)$ where $p_1 = q_1$. Furthermore, as the edges of C_L and C_R are traversed in order, their slopes are strictly increasing over the range $(-\infty, 0)$ and strictly decreasing over the range $(0, +\infty)$, respectively. The vertex that C_L and C_R share is called the *apex* of C. A *convex-bridge* of C is a convex chain $P = (w_1, w_2, \ldots, w_k)$ where $w_1 \in C_L$, $w_k \in C_R$, P is the upper half of some convex polygon and P lies below $[p_L, q_R]$. Let u and w denote the slopes of the common tangents of P with C_L and C_R, respectively. For each slope $m \in [u, w]$ there is a line segment $h(m)$ with slope m that joins C_L to C_R and is tangent to P. Consider a function f of slope defined over the range $[u, w]$ so that $f(m)$ for $m \in [u, w]$ is the length of $h(m)$. The following lemma characterizes the function f over the range $[u, w]$.

Lemma 1 *The function f over the range $[u, w]$ is unimodal.*

Given that s does not lie inside P, we would like to apply Lemma 1 to the problem of computing the shortest line segment tangent to $T(B)$ and joining L to R. Consider the following lemma.

Lemma 2 *Every line segment joining L to R either lies inside P or intersects exactly one of T and B.*

Proof: Consider an arbitrary line segment $r = [p, q]$ that joins L to R and observe that r meets L and R only at p and q. Let v be the intersection of the line segments $[v_1, v_3]$ and $[v_2, v_4]$. Clearly, T is contained in $\Delta v_1 v_2 v$ and B is contained in $\Delta v_3 v_4 v$, however, both T and B may not contain v. Finally, observe that r either passes through v or intersects exactly one of $\Delta v_1 v_2 v$ and $\Delta v_3 v_4 v$. Hence, r either lies inside P or intersects exactly one of T and B. □

In what follows, we assume neither v_1, v_2, v_3 nor v_4 is an endpoint of s. This assumption guarantees that s always decomposes L and R into four chains and serves only to eliminate from consideration the simple cases where the resulting number of chains is less than four. Let α_1, α_4 and α_2, α_3 be the pairs of angles formed at the intersections of s with L and R, respectively, where α_1 and α_2 lie above s and α_3 and α_4 lie below s. Consider the following lemma.

Lemma 3 *If s intersects $T(B)$ then $\alpha_1 \geq \pi/2$, $\alpha_2 \geq \pi/2$, $\alpha_3 \geq \pi/2$, $\alpha_4 \geq \pi/2$ and $\alpha_3 + \alpha_4 > \pi$ ($\alpha_1 + \alpha_2 > \pi$).*

Proof: Suppose s intersects T. If $\alpha_1 < \pi/2$ then there must exist some $b' \in R$ such that the line segment $[a, b']$ is shorter than s, which contradicts our assumption that s was shortest possible. Hence $\alpha_1 \geq \pi/2$. Similar arguments demonstrate that $\alpha_2 \geq \pi/2$, $\alpha_3 \geq \pi/2$ and $\alpha_4 \geq \pi/2$. To complete the proof observe that our choice of s rules out the possibility that $\alpha_3 = \alpha_4 = \pi/2$. Hence $\alpha_3 + \alpha_4 > \pi$. An analogous proof holds for the case when s intersects B. □

Suppose that s intersects $T(B)$. Let $s_a = [a, b']$ be the line segment tangent to $T(B)$ and joining a to R. Similarly, let $s_b = [b, a']$ be the line segment tangent to $T(B)$ and joining b to L. Observe that by Lemma 2 both s_a and s_b lie inside P. The following lemma characterizes every shortest line segment that lies inside P and joins L to R.

Lemma 4 *If s intersects $T(B)$ then of the line segments that lie inside P and join L to R, every shortest one is tangent to $T(B)$ and lies below (above) s.*

Proof: Suppose s intersects T. Observe that every line segment that joins L to R and has both endpoints lying either above s_a or above s_b, intersects T. Consider a pair $p \in L$, $q \in R$ of points where p lies on or above s_a and q lies on or below s_b. Let $r = [p, q]$ and recall Lemma 3. If $r \neq s_a$ then r is longer than s_a. Similarly, if p lies on or below s_a and q lies on or above s_b and

$r \neq s_b$, then r is longer than s_b. Hence, if s intersects T then of the line segments that lie inside P and join L to R, every shortest one lies below s. Consider a line segment r that joins L to R, that does not intersect T and whose endpoints lie on or below s. Let r' be the line segment parallel to r that joins L to R and is tangent to T. Again, recall Lemma 3. If both endpoints of r' lie on or below s and $r \neq r'$, then r is longer than r'. On the other hand, if either endpoint of r' lies on or above s and $r \neq s_a$ and $r \neq s_b$, then r is longer than one of s_a and s_b. Hence, if s intersects T then of the line segments that lie inside P and join L to R, every shortest one is tangent to T and lies below s. A similar proof holds for the case where s intersects B. □

Let m_a and m_b denote the slopes of s_a and s_b, respectively. For each slope $m \in [m_a, m_b]$ there is a line segment $s(m)$ with slope m that joins L to R and is tangent to $T(B)$. Consider a function d of slope defined over the range $[m_a, m_b]$ so that $d(m)$ for $m \in [m_a, m_b]$ is the length of $s(m)$.

Lemma 5 *If s intersects $T(B)$ then the function d over the range $[m_a, m_b]$ is unimodal.*

Proof: Suppose s intersects T. Let e_L and e_R be the edges of L and R, respectively, that intersect s and extend below s. Recall Lemma 3. If we extend the edges e_L and e_R above s they will intersect at some point v above s. The two chains $L' = (v, a, \ldots, v_3)$ and $R' = (v, b, \ldots, v_4)$ together form a concave-chain C. Furthermore, the portion T' of T that lies inside C, is a convex-bridge of C. Let u and w be the slopes of the common tangents of T' with L' and R', respectively. Combining Lemma 1 with the fact that $[m_a, m_b]$ is a subrange of $[u, w]$, we get that the function d over the range $[m_a, m_b]$ is unimodal. An analogous proof holds for the case where s intersects B. □

3 Algorithm

Observe that the line segments $s(m)$ for $m \in [m_a, m_b]$ are exactly those that join L to R, are tangent to $T(B)$ and lie below (above) s. This suggest the following algorithm that takes as input the butterfly polygon P and returns a shortest line segment that joins L to R and lies inside P. First determine the line segment s. If s lies inside P then return s, otherwise find the line segment that corresponds to the minimum of the function d over the range $[m_a, m_b]$ and return the line segment. The correctness of the algorithm follows directly from Lemma 5.

It remains to be shown that there exists an $O(\log^2 n)$ time implementation of the algorithm. We assume P is stored in a data structure, such as an array or balanced tree, that in $O(\log n)$ time supports binary searches on the chains L, R, T and B, and can report the vertices adjacent to a given vertex.

Using an algorithm of Edelsbrunner [7] that computes the shortest distance between two disjoint convex polygons, s can be computed in $O(\log n)$ time. Given either a line or a line segment and a convex polygon Q with n vertices, whether the line or line segment intersects Q can be determined in $O(\log n)$ time using an algorithm of Chaselle and Dobkin [5]. Hence, in $O(\log n)$ time it is possible to determine if s lies inside P, since, if s does not lie inside P then s must intersect either T or B. Clearly, if s lies inside P, the algorithm computes the desired line segment in $O(\log n)$ time.

Suppose s intersects either T or B. We now turn our attention to computing the line segment that corresponds to the minimum of the function d over the range $[m_a, m_b]$, which we call the optimal line segment. In order to compute the optimal line segment it is necessary to first compute the line segments s_a and s_b, which requires $O(\log n)$ time. Suppose, without loss of generality, that s intersects T. Let p and q denote the points of s_a and s_b, respectively, that are tangent to T, and let T' denote the portion of T between p and q in the clockwise sense. Recall that the function d over the range $[m_a, m_b]$ is unimodal. Suppose we are given some slope $r \in [m_a, m_b]$. The line segment $s(r)$ can be found in $O(\log n)$ time by first computing the line l with slope r that is tangent to T, and then the intersection of l with L and R. Furthermore, given $s(r)$, it is possible to decide in $O(\log n)$ time whether the optimal line segment has slope in the range $[m_a, r]$ or $[r, m_b]$. Hence, using binary search on T', it is possible to isolate, in $O(\log n)$ time, the vertex $v \in T'$ to which the optimal line segment is tangent.

Given v and some point $c \in L$ such that the line l through v and c is tangent to T', and has slope in the range $[m_a, m_b]$, it is possible to determine the intersection point of l with R in $O(\log n)$ time. Furthermore, let r be the slope of l, then it is possible to decide in $O(\log n)$ time whether the optimal line segment has slope in the range $[m_a, r]$ or $[r, m_b]$. Hence, having isolated v, the optimal line segment can be determined in $O(\log^2 n)$ using binary search on L. Clearly, the algorithm runs in $O(\log^2 n)$ time. Therefore, we have established the following theorem, which is the main result of this paper.

Theorem 1 *Given an n vertex butterfly polygon P, the wingspan of P can be computed in $O(\log^2 n)$ time.*

4 Applications

We have presented several results which find application in a variety of contexts. One approach to the problem of computing a shortest transversal of a family of n line segments requires solving n different subproblems [3, 4]. Each subproblem involves determining the wingspan of a butterfly polygon. A similar application arises in computing a shortest transversal of a family of convex polygons [4]. Combining our results with those found in [2], it is possible to obtain the shortest line segment from which an n vertex convex polygon is weakly externally visible in $O(n)$ time. Finally, given two edges of a simple n vertex polygon, the shortest line-of-sight between the two edges (shortest line segment internal to the polygon and joining the two edges) can be computed in $O(n)$ time using our results in conjunction with those in [1].

5 Concluding Remarks

In this paper we presented an $O(\log^2 n)$ time algorithm for computing the wingspan of an n vertex butterfly polygon. We also described several applications of the algorithm. It remains an open problem as to whether the wingspan of a butterfly polygon can be computed in $O(\log n)$ time. The existence of such an algorithm would imply that the shortest distance between a pair of convex polygons that is constrained to pass through a point could also be computed in $O(\log n)$ time.

References

[1] Avis, D., Gum, T. and Toussaint, G. T., 'Visibility between two edges of a simple polygon', *The Visual Computer* **2**, pp. 342-357 (1986).

[2] Bhattacharya, B. K., Kirkpatrick, D. and Toussaint, G. T., 'Determining sector visibility of a polygon', *Proc. 5th ACM Symposium on Computational Geometry*, pp. 247-254 (1989).

[3] Bhattacharya, B. K. and Toussaint, G. T., 'Computing shortest transversals', *Computing*, to appear.

[4] Bhattacharya, B. K., Czyzowicz, J., Egyed, P., Stojmenovic, I., Toussaint, G. T. and Urrutia, J., 'Computing shortest transversals of sets', *Proc. 7th ACM Symposium on Computational Geometry* (1991).

[5] Chazelle, B. and Dobkin, D., 'Intersection of convex objects in two and three dimensions', *Journal of the ACM* **34**, pp. 1-27 (1987).

[6] Dobkin, D. and Kirkpatrick, D., 'A linear algorithm for determining the separation of convex polyhedra', *Journal of Algorithms* **6**, pp. 381-392 (1985).

[7] Edelsbrunner, H., 'Computing the extreme distances between two convex polygons', *Journal of Algorithms* **6**, pp. 213-224 (1985).

[8] McKenna, M. and Toussaint, G. T., 'Finding the minimum vertex distance between two disjoint convex polygons in linear time', *Computers and Mathematics with Applications* **11**, pp. 1227-1242 (1985).