An Optimal Algorithm for Solving the Restricted Minimal Convex Nested Polygonal Separation Problem

D.P. Wang and R.C.T. Lee

Department of Computer Science

National Tsing–Hwa University

Hsinchu Taiwan 30043

Republic of China

Abstract

Assume that convex polygon A contains convex polygon C. In this paper, we propose a linear time algorithm for finding a convex polygon B nested between these two polygons under the condition that the vertices of B must be a subset of vertices of A and the number of vertices of B is minimized. Our algorithm is optimal.

Section 1 Introduction

In [1], the following problem was discussed: Given two convex polygons A and C such that C is contained inside of A, determine a convex polygon B which is nested between A and C and the number of vertices of B is minimum.

In [1], a greedy algorithm was proposed to solve the above problem in $O(n \log n)$ time, where n is the total number of vertices of A and C. Until now, as far as the authors are aware of, no linear time algorithm of the above problem has been found.

In this paper, we further require that the vertices of the solution polygon B must be a subset of vertices of A. Under this constraint, we can solve the above problem in linear time which is optimal. Such a solution polygon B is called a restricted minimal convex nested polygonal separator of A and C.

Now, let us formally state our problem: We are given two convex polygons

A and C such that C is contained in A. Find a convex polygon B such that B is nested between A and C under the condition that the vertices of B must be a subset of vertices of A and the number of vertices of B is minimized. Let the numbers of vertices of A and C be m and k respectively. We use $a_0, a_1, \ldots, a_{m-1}$ and $c_0, c_1, \ldots, c_{k-1}$ to denote the vertices of A and C respectively, in a counterclockwise order. Let n be the sum of m and k. Furthermore, we assume that the vertices of A and C are wrapped around at modulo m and k respectively.

Section 2 The Algorithm

Consider Fig. 1. Our algorithm starts from finding the farthest vertex which any vertex of A can reach without going through the convex polygon C. For instance, the farthest vertex of a_1 is a_3 and the farthest vertex of a_3 is a_5. Formally, the farthest vertex of a vertex of A is defined by first defining the right tangents of C: **For any vertex a_i of A, there are two tangents of C with respect to a_i [3]. Assume that we stand at a vertex a_i and look at the direction $\overrightarrow{a_i c_j}$. We say that $\overrightarrow{a_i c_j}$ is a right tangent of C with respect to a_i if $\overrightarrow{a_i c_j}$ is a tangent of C and C lies to the left–hand side of $\overrightarrow{a_i c_j}$. Let a_i be a vertex of A and $\overrightarrow{a_i c_j}$ be the right tangent of C with respect to a_i. We say that a_p is a farthest vertex of a_i if $\overrightarrow{a_i c_j}$ intersects A at edge $\overline{a_p a_{p+1}}$.**

A chord of a convex polygon X is an edge inside X which connects two vertices of X. For our case, we also require any chord of A does not intersect polygon C. Suppose $\overline{a_i a_j}$ is a chord of A. We say a_i is a starting vertex of $\overline{a_i a_j}$ if polygon C is at the left–hand side of $\overline{a_i a_j}$. If a_i is a starting vertex of $\overline{a_i a_j}$, a_j is an ending vertex of $\overline{a_i a_j}$. Suppose that the farthest vertex of a_i is a_j, and we connect a_i to a_j. Then $\overline{a_i a_j}$ is a chord of A, a_i is a starting vertex of $\overline{a_i a_j}$ and a_j is an ending vertex of $\overline{a_i a_j}$. For example, for the case shown in Fig. 1, all of the chords created by connecting vertices of A and their farthest vertices are shown in Fig. 2.

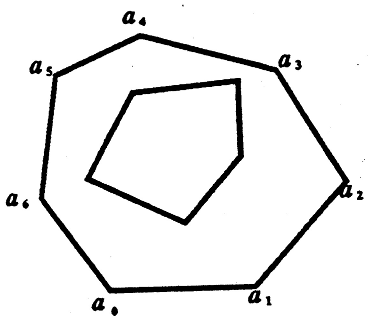

Fig. 1

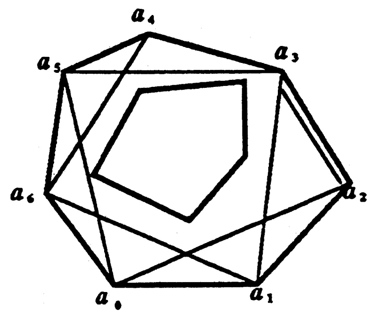

Fig. 2

Our original problem then becomes the following problem: Given two convex polygons A and C and a set of chords $\mathscr{L}_1, \mathscr{L}_2, \ldots, \mathscr{L}_m$ where each chord is specified by an ordered pair $<x_i, y_i>$ where x_i and y_i denote, respectively, the starting and ending vertex of the chord with y_i following x_i in the counterclockwise direction. Our minimal chord covering problem is to find a minimal number of chords whose union covers polygon A. For instance, for the problem instance of Fig. 2, $\{\overline{a_1a_3}, \overline{a_3a_5}, \overline{a_0a_2}, \overline{a_5a_0},\}$ is a solution, as shown in Fig. 3(a). The set of the starting vertices of $\{\overline{a_1a_3}, \overline{a_3a_5}, \overline{a_0a_2}, \overline{a_5a_0},\}$ is $\{a_1, a_3, a_0, a_5\}$. Let us order these start vertices into a circular sequence $a_0 \rightarrow a_1 \rightarrow a_3 \rightarrow a_5 \rightarrow a_0$, as shown in Fig. 3(b). And connect the starting vertices of the circular sequence into a cyclic fashion to form a convex polygon $B = a_0 \rightarrow a_1 \rightarrow a_3 \rightarrow a_5 \rightarrow a_0$, as shown in Fig. 3(c). Then, in the nest section, we will prove that B is a restricted minimal convex nested polygon of A and C.

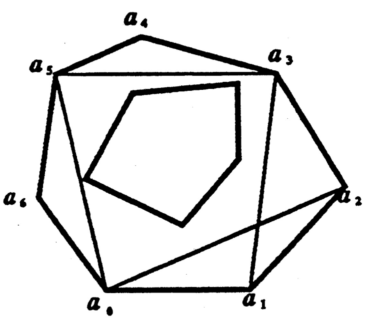

Fig. 3(a)

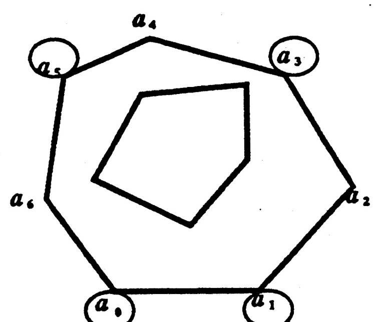

Fig. 3(b)

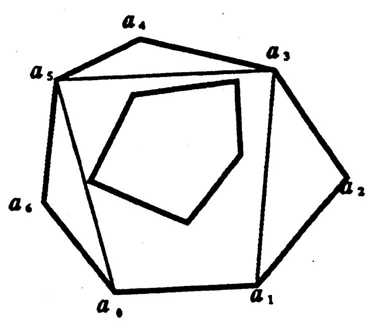

Fig. 3(c)

To solve the above minimal chord covering problem, we may use the algorithm suggested by Lee and Lee [2]. In [2], the problem considered is as follows: Given a set of m arcs $\mathscr{L}_1, \mathscr{L}_2, \ldots, \mathscr{L}_m$ on a circle and each arc $\mathscr{L}_i$ is specified by an ordered pair $<x_i, y_i>$, where x_i and y_i denote, respectively, the two endpoints of the arc with y_i following x_i in the counterclockwise direction. The circle–cover minimization problem is to find a minimum number of arcs whose union covers the circle. In [2], they presented an $O(m\log m)$ time algorithm to solve it. Furthermore, if the endpoints of these arcs ,i.e., $\{x_1, x_2, \ldots, x_m\}$ and $\{y_1, y_2, \ldots, y_m\}$ have been sorted, then linear time is sufficient.

The following is our algorithm for solving the restricted minimal convex nested polygonal separation problem.

Algorithm A: Finding a restricted minimal convex nested polygonal separator of *A* and *C*.

Input: Two convex polygons *A* and *C* where *C* is inside of *A* and the vertices of *A* and *C* are $a_0, a_1, \ldots, a_{m-1}$ and $c_0, c_1, \ldots, c_{k-1}$ respectively, in counterclockwise order.

Output: A restricted minimal convex nested polygonal separator of *A* and *C*.

Step 1: For each vertex of *A*, find its farthest vertex.

Step 2: Connect each vertex of *A* to its farthest vertex.

Step 3: Solve the minimal chord covering problem with *A* and the chords produced in Step 2 as the input.

Step 4: Order the starting vertices of the chords found by Step 3 in circular sequence.

Step 5: Connect the starting vertices of the sequence found by Step 4 in a cyclic fashion to form a convex polygon *B*.

Step 6: Return convex polygon *B*.

In order to prove the correctness of Algorithm A, we need the following lemma.

Lemma 1: The number of edges of the restricted minimal convex nested polygonal separator of *A* and *C* is equal to the number of a minimal set of chords covering *A*.

Proof: Assume that t_c and t_s are the number of a minimal set of chords covering *A* and the number of edges of the restricted minimal convex nested polygonal separator of *A* and *C* respectively. First, we want to prove that $t_c \geq t_s$. Consider a minimal set of chords covering *A* whose size is t_c. Order the starting vertices of these chords into a circular sequence on polygon *A* and connect these starting vertices into a cyclic fashion to form a convex polygon. The result is a restricted convex nested polygon of *A* and *C* with t_c edges. Since t_s is the number of edges of the restricted **minimal** convex nested polygon of *A* and *C*, $t_c \geq t_s$.

Now, we prove that $t_s \geq t_c$. Consider a restricted minimal convex nested polygon of *A* and *C* whose size is t_s. It is also a chord cover of polygon *A* with t_s chords. Since t_c is the **minimum** number of chords covering *A*, $t_s \geq t_c$.

Q.E.D.

Since Algorithm A solve the minimal chord covering problem. Lemma 1 implies that the polygon produced by Algorithm A is an optimal one.

Step 1 of Algorithm A is critical. A naive way of finding all farthest vertices requires $O(n^2)$ time [3]. In [4], we can use greedy strategy to solve it in O(n) time.

Now let us discuss the time complexity of Algorithm A. It needs O(n) time to perform Step 1 and 2. It needs O(n) time to perform Step 3 by applying the circle–cover minimization algorithm [2]. Step 4 can be done in O(n) time by a linear scanning of vertices in A. Step 5 needs the same time–complexity. Therefore, we have the following Theorem.

Theorem 2: The restricted minimal convex nested polygonal separator of A and C can be found in O(n) time.

Section 3 Conclusions

We have presented a linear–time algorithm for obtaining a restricted minimal convex nested polygonal separator of two nested polygons. Our algorithm makes an efficient search in finding the farthest vertices of all vertices of a polygon A and use the circle–cover minimization algorithm as a subroutine to obtain our solution polygon B. And we think that this approach is useful in solving some other geometric problems.

References

[1] A. Aggarwal and H. BooTH and J. O'RouRke and S. Suri and C. K.Yap "Finding Minimal Convex Nested Polygons" **Information and Computation** Vol 83, pp.98–110 (1989).

[2] C. C. Lee and D. T. Lee " On a Circle–Cover Minimization Problem", **Information Processing Letters** Vol 18, pp109–115 (1984).

[3] F. P. Preparata and M. I. Shamos **Computational Geometry – an Introduction.** Springer Verlag, New York, (1985).

[4] D. P. Wang and R. C. T. Lee "An Optimal Algorithm for Solving the Restricted Minimal Convex Nested Polygonal Separation Problem" **Technical Report** (1992).