Computational Geometry 47 (2014) 90–109

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Minimum-area enclosing triangle with a fixed angle ☆

Prosenjit Bose *, Jean-Lou De Carufel **

School of Computer Science, Carleton University, 5302 Herzberg Laboratories, 1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada

ARTICLE INFO

Article history:
Received 24 September 2010
Accepted 11 July 2013
Available online 16 July 2013
Communicated by J. Mitchell

Keywords:
Geometric optimization
Enclosing problems
Computational geometry

ABSTRACT

Given a set S of n points in the plane and a fixed angle $0 < \omega < \pi$, we show how to find in $O(n \log n)$ time all triangles of minimum area with one angle ω that enclose S. We prove that in general, the solution cannot be written without cubic roots. We also prove an $\Omega(n \log n)$ lower bound for this problem in the algebraic computation tree model. If the input is a convex n-gon, our algorithm takes $\Theta(n)$ time.

1. Introduction

In geometric optimization, the goal is often to find an optimal object or optimal placement of an object subject to a number of geometric constraints. Examples include finding the smallest circle enclosing a point set [8,10] or finding the smallest circle enclosing at least k points of a point set of n points ($k \leqslant n$) [4,7]. In our setting, we study the following problem: given a set S of n points in the plane, find all the triangles of minimum area with a fixed angle ω, $0 < \omega < \pi$, that enclose S. When no constraint is put on the angles, Klee and Laskowski [5] gave an $O(n \log^2 n)$ time algorithm for finding the minimum-area enclosing triangle. This was later improved to $O(n \log n)$ by O'Rourke et al. [9]. Bose et al. [2] provided optimal algorithms for the setting where one wishes to find the minimum-area isosceles triangles with a fixed angle. The setting we explore here is in between the two. We place a restriction on the angle but do not insist on the triangle to be isosceles. Our solution, which we outline below, uses ideas from the solutions of Klee and Laskowski and Bose et al.

The five main steps of the algorithm are presented in Sections 2, 3, 4, 5 and 6. Each section outlines one step, proves the mathematical formulas involved and gives its time complexity. As we present in Section 5, at some point, the algorithm needs to calculate the roots of a fourth degree polynomial. This is unfortunate when it comes to numerical robustness. However, in Section 7, we show that we cannot dodge such algebraic expressions. Finally, we prove an $\Omega(n \log n)$ lower bound for this problem in the algebraic computation tree model.

2. Overview and preliminaries

Since the solution to the general problem only needs to consider the vertices of the convex hull of S, the first step is to compute the convex hull. In the remainder of the paper, we assume that the input is a convex n-gon with vertices given in clockwise order.

☆ An extended abstract of this paper appeared in Bose and De Carufel (2010) [1]. This research was partially supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and FQRNT (Fonds québécois de la recherche sur la nature et les technologies).

* Corresponding author.
** Main corresponding author. Tel.: +1 613 520 4333; fax: +1 613 520 4334.
E-mail addresses: jit@scs.carleton.ca (P. Bose), jdecaruf@cg.scs.carleton.ca (J.-L. De Carufel).

0925-7721/$ – see front matter
http://dx.doi.org/10.1016/j.comgeo.2013.07.002

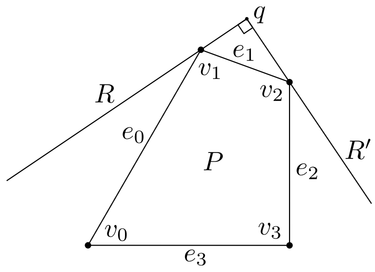

Fig. 1. In this example, P is a quadrilateral and $\omega = \frac{1}{2}\pi$.

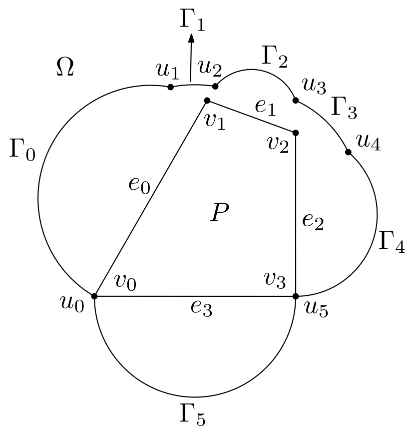

Fig. 2. Step 1: Ω is the $\frac{1}{2}\pi$-cloud of P ($n' = 6$).

Let P be a convex n-gon (refer to Fig. 1). We denote the edges and the vertices of P in clockwise order by e_i and v_i for $0 \leqslant i \leqslant n-1$ (all index manipulation is modulo n). Here and in the following sections, as we present the algorithm, we trace each step through the example of Fig. 1.

We begin with two definitions.

Definition 1 *(ω-wedge).* Let q be a point in the plane and ω be an angle ($0 < \omega < \pi$). Let R and R' be two rays emanating from q such that the angle between R and R' is ω. We say that the closed set formed by q, R, R' and the points between R and R' creates an *ω-wedge*, denoted by $\mathcal{W}(\omega, q, R, R')$. The point q is called the *apex* of the ω-wedge. An ω-wedge W *touches* a polygon P when $P \subseteq W$ and both R and R' touch P, i.e. $R \cap P \neq \varnothing$ and $R' \cap P \neq \varnothing$ (refer to Fig. 1).

For the rest of the paper, when looking at an ω-wedge facing down, R represents the left ray and R' represents the right ray (refer to Fig. 1). Also, when we make the ω-wedge turn around P, we do it clockwise.

Definition 2 *(ω-cloud).* Let P be a convex n-gon and ω be an angle ($0 < \omega < \pi$). By rotating an ω-wedge around P while continually touching P, the apex traces a sequence of circular arcs that we call an *ω-cloud* (refer to Fig. 2).

There are many technical details involved in the solution to this problem. Before getting too caught up in these details, let us first review the general approach to our solution.

Since we only consider enclosing triangles with an angle of ω, each optimal triangle can be constructed from an ω-wedge that touches P. Therefore, we consider all possible ω-wedges that touch P. The apices of these ω-wedges lie on an ω-cloud Ω which consists of a linear number of pieces of circular arcs (refer to [2]). Then, for each of these ω-wedges, it is possible to find the minimal triangle by identifying a third side. For the triangle to be optimal, the midpoint of this third side has to touch P (see Proposition 3 and Corollary 6). Hence, for each ω-wedge touching P, there is one and only one triangle to consider for optimality.

Moreover, when the apex q of an ω-wedge W moves clockwise along the ω-cloud Ω, the midpoint m of the third side of the optimal triangle moves clockwise along P (see Lemma 7). As W moves, we note the positions of q where m leaves an edge of P and where m enters a new edge of P. These positions of q are important event points. They divide Ω into a linear number of components (see Section 4). Let $\mathcal{C}$ be one of these components. We prove that the minimum-area triangle having a vertex (hence an angle ω) on $\mathcal{C}$ can be computed in constant time (see Lemmas 10, 11, 12 and 13). We then have a linear number of candidates (one for each piece of Ω) to consider rather than infinitely many if we had to take all possible