

1-1-2009

# Locating and Obnoxious Line among Planar Objects

D. Z. Chen

Haitao Wang  
*Utah State University*

---

## Recommended Citation

Chen, D. Z. and Wang, Haitao, "Locating and Obnoxious Line among Planar Objects" (2009). *ECE Faculty Publications*. Paper 72.  
[http://digitalcommons.usu.edu/ece\\_facpub/72](http://digitalcommons.usu.edu/ece_facpub/72)

This Article is brought to you for free and open access by the Electrical and Computer Engineering, Department of at DigitalCommons@USU. It has been accepted for inclusion in ECE Faculty Publications by an authorized administrator of DigitalCommons@USU. For more information, please contact [becky.thoms@usu.edu](mailto:becky.thoms@usu.edu).



International Journal of Computational Geometry & Applications  
 © World Scientific Publishing Company

## LOCATING AN OBNOXIOUS LINE AMONG PLANAR OBJECTS\*

DANNY Z. CHEN

*Department of Computer Science and Engineering  
 University of Notre Dame  
 Notre Dame, IN 46556, USA  
 dchen@cse.nd.edu*

HAITAO WANG<sup>†</sup>

*Department of Computer Science and Engineering  
 University of Notre Dame  
 Notre Dame, IN 46556, USA  
 hwang6@nd.edu*

29 May 2010

Revised 05 April 2012

Communicated by JSB Mitchell

### ABSTRACT

Given a set  $P$  of  $n$  points in the plane such that each point has a positive weight, we study the problem of finding an *obnoxious* line that intersects the convex hull of  $P$  and maximizes the minimum weighted Euclidean distance to all points of  $P$ . We present an  $O(n^2 \log n)$  time algorithm for the problem, improving the previously best-known  $O(n^2 \log^3 n)$  time solution. We also consider a variant of this problem whose input is a set of  $m$  polygons with a total of  $n$  vertices in the plane such that each polygon has a positive weight and whose goal is to locate an obnoxious line with respect to the weighted polygons. An  $O(mn + n \log^2 n \log m + m^2 \log n \log^2 m)$  time algorithm for this variant was known previously. We give an improved algorithm of  $O(mn + n \log^2 n + m^2 \log n)$  time. Further, we reduce the time bound of a previous algorithm for the case of the problem with unweighted polygons from  $O((m^2 + n \log m) \log n)$  to  $O(m^2 + n \log m)$ .

*Keywords:* Obnoxious line; facility location; separation problems; algorithms design.

### 1. Introduction

Determining the locations of undesirable or obnoxious facilities among a set of geometric objects has been an important research topic. In this paper, the target obnoxious facility is a line in the plane. Formally, given a set of  $n$  points in the plane,  $P = \{p_1, p_2, \dots, p_n\}$ , where each point  $p_i$  has a weight  $w_i > 0$ , a line  $l$  is said to be *obnoxious* if there is at least one point of  $P$  lying on each side of  $l$  (i.e.,

\*This research was supported in part by NSF under Grant CCF-0916606.

<sup>†</sup>Corresponding author.

$l$  intersects the convex hull of  $P$ ) and the value of  $\min\{w_i \cdot d(l, p_i) \mid p_i \in P\}$  is maximized, where  $d(l, p_i)$  denotes the Euclidean distance between the line  $l$  and the point  $p_i$ . The *obnoxious line location* (OLL) problem seeks an obnoxious line with respect to the weighted points of  $P$ . When all the point weights are equal, we call it the *unweighted* OLL problem.

In a variant of OLL, a set of  $m$  (possibly intersecting) polygons,  $\mathcal{P} = \{P_1, P_2, \dots, P_m\}$ , with a total of  $n$  vertices in the plane is given, where each polygon  $P_i$  has a weight  $w_i > 0$ , and the goal is to compute an obnoxious line  $l$  with respect to  $\mathcal{P}$  such that there is at least one polygon of  $\mathcal{P}$  lying on each side of  $l$  and the value of  $\min\{w_i \cdot d(P_i, l) \mid P_i \in \mathcal{P}\}$  is maximized, where  $d(P_i, l)$  is the minimum distance of any point in  $P_i$  to  $l$ . We call this variant the *OLL problem among weighted polygons*, denoted by OLLP. When all the polygon weights are equal, we call it the *unweighted* OLLP problem.

These problems are motivated by applications of locating a linear route through existing facilities such that the route is noxious or hazardous to its surroundings or of obtaining the maximum clearance of a route with respect to the existing facilities. The weights may represent various importance of the facilities. For examples, one may need to build or find pipelines, roads, railway lines, or sailing routes to transport noxious materials.

Clearly, an algorithm for OLL can be used to solve OLLP with slight modifications, and vice versa. Since  $m = o(n)$  in many applications, a good algorithm for OLLP whose performance depends on both  $m$  and  $n$  can be much better than that for OLL. Because OLL can be viewed as a special case of OLLP (with  $m = n$ ), a good algorithm for OLLP ideally should also be no worse than a good algorithm for OLL for any  $m \leq n$ .

### 1.1. *Previous and related work*

The unweighted OLL problem is equivalent to the widest empty corridor problem<sup>1</sup>. By using topological sweeping and duality, the problem was solved in  $O(n^2)$  time and  $O(n)$  space<sup>1</sup>. By using the similar method, the unweighted OLLP is solvable in  $O((m^2 + n \log m) \log n)$  time<sup>2</sup>. The OLL problem was first studied in<sup>3</sup>, with an  $O(n^3)$  time algorithm. Based on parametric search, the OLLP problem was solved in  $O(mn + n \log^2 n \log m + m^2 \log n \log^2 m)$  time and  $O(m^2 + n)$  space<sup>2</sup>. Thus, by treating each input point as a special polygon and using the algorithm in<sup>2</sup> (with  $m = n$ ), the problem OLL is solvable in  $O(n^2 \log^3 n)$  time and  $O(n^2)$  space. The authors in<sup>4</sup> derives an algorithm for the problem OLL with less space  $O(n)$  but more time  $O(n^3 \log n)$ .

The dynamic version of the widest empty corridor problem was also studied<sup>5</sup>. Other problem variants include  $k$ -dense corridors<sup>5,6,7</sup>,  $L$ -shaped corridors<sup>8</sup>, and curved corridors<sup>9</sup>. The “dual” problem of OLL, which seeks a line minimizing the maximum weighted distance between the line and a given set of points in the plane, has been studied as well. The unweighted version is equivalent to the set

width problem which is solvable in  $O(n \log n)$  time<sup>10,11</sup>. The weighted version was solved in  $O(n^2 \log n)$  time<sup>11</sup>. It turns out that solving the OLL problem resorts to considerably different algorithmic techniques than its “dual” problem.

## 1.2. Our contributions

In this paper, we present an  $O(n^2 \log n)$  time algorithm for solving the OLL problem, improving the  $O(n^2 \log^3 n)$  time solution in<sup>2</sup>. For the OLLP problem, we give an  $O(mn + n \log^2 n + m^2 \log n)$  time algorithm, improving the  $O(mn + n \log^2 n \log m + m^2 \log n \log^2 m)$  time result in<sup>2</sup>. For the unweighted OLLP problem, by a slight modification of the  $O((m^2 + n \log m) \log n)$  time algorithm in<sup>2</sup>, we reduce its running time to  $O(m^2 + n \log m)$ .

Note that our results are “ideal”; namely, when  $m = o(n)$ , our algorithm for OLLP (resp., unweighted OLLP) is always faster than the corresponding algorithm for OLL (resp., unweighted OLL), and when  $m = n$ , their asymptotic time bounds are the same, respectively.

Both our OLL and OLLP algorithms are based on the parametric search approach, which is similar to the scheme discussed in<sup>2</sup>. We derive new and more efficient algorithms to solve the decision problems in the parametric search for both problems OLL and OLLP. In addition, the authors in<sup>2</sup> used Megiddo’s parametric search<sup>12</sup>; instead, we show that Cole’s parametric search can be applied, which is faster than Megiddo’s approach.

For the problem OLL, we model the decision problem of the parametric search as the following *disks separability problem*: Given  $n$  disks in the plane, either find a line  $l$  in the plane such that  $l$  does not intersect the interior of any disk and there is at least one disk on each side of  $l$ , or report that there exists no such line. We solve the problem efficiently by using the visibility complex<sup>13,14</sup>. Note that other objects separability problems (known as the *shattering problem*) have also been studied, e.g.,<sup>15,16,17,18</sup>.

Since the problem OLL can be solved based on some kind of sorting, Cole’s parametric search<sup>19</sup> can be applied. By using our disks separability algorithm to make decisions in the parametric search, we solve the problem OLL in  $O(n^2 \log n)$  time.

For the problem OLLP, the decision problem of the parametric search can be viewed as a generalization of the disks separability problem. To solve it, based on new observations, we manage to extend our disks separability algorithm and give an improved algorithm over the one in<sup>2</sup>. In addition, we present an improved preprocessing procedure and show that Cole’s parametric search<sup>19</sup> can be applied, where our improved OLLP decision algorithm is used to make decisions. All these efforts lead to a more efficient algorithm for OLLP.

Finally, our improvement on the unweighted OLLP algorithm is due to the application of the linear time algorithm for overlaying simply connected planar subdivisions<sup>20</sup>.

The paper is organized as follows. In Section 2, we discuss our algorithm for the disks separation problem. We give our solution for the OLL problem in Section 3. Our algorithm for the OLLP problem is presented in Section 4. Section 5 concludes the paper.

## 2. The Disks Separability Problem

Given a set of disks in the plane,  $\mathcal{D} = \{D_i \mid 1 \leq i \leq n\}$ , where each disk  $D_i$  is centered at the point  $(x_i, y_i)$  with a radius  $r_i$ , the disks separability problem seeks to either find a line  $l$  in the plane such that  $l$  does not intersect the interior of any disk in  $\mathcal{D}$  and there is at least one disk of  $\mathcal{D}$  on each side of  $l$ , or report that there exists no such line. We call a line that satisfies these requirements a *separation line* of  $\mathcal{D}$ . We say that  $\mathcal{D}$  is *separable* if there exists a separation line. In this paper, when we say that a line *intersects* a disk, we mean that the line intersects the interior of the disk (and thus the line is not tangent to the disk); similarly, when we say that two disks *intersect* each other, we mean that the intersection in their interior is not empty (and thus the two disks are not mutually tangent to each other). A *common tangent* of two disks is a line segment that is tangent to both disks at its endpoints; the common tangent is *free* if it does not intersect any disk in  $\mathcal{D}$ .

To prove the above theorem, we begin with the following obvious but critical lemma. Similar observations were also given in <sup>17</sup>.

**Lemma 1.** *Given  $\mathcal{D}$ , a separation line exists if and only if there is a separation line that contains a free common tangent of two disks in  $\mathcal{D}$ .*

For any line segment  $s$  that does not intersect any disk in the plane, we obtain a new segment  $s'$  by extending  $s$  along its two directions until it either intersects a disk or goes into infinity, and we call  $s'$  the *maximal segment* of  $s$ .

Based on Lemma 1, to solve the disks separability problem, it suffices to compute the maximal segments of all free common tangents of the disks in  $\mathcal{D}$ , and for each maximal segment, check whether its two endpoints are at the infinity. To this end, we make use of the visibility complex <sup>13,14</sup>. But since the visibility complex <sup>13,14</sup> is for interior-disjoint convex objects of constant size in the plane, it is not directly applicable to our problem since the disks in  $\mathcal{D}$  may intersect each other. In the following, we do some preprocessing on the disks in  $\mathcal{D}$ .

We first compute the union of the disks in  $\mathcal{D}$ , which can be done in  $O(n \log n)$  time <sup>21</sup>. Let  $\mathcal{D}^*$  denote the union of the disks.

The second step is to partition  $\mathcal{D}^*$  into  $O(n)$  interior-disjoint, convex, and constant-size objects, as follows (refer to Fig. 1 for an example). Denote by  $\partial\mathcal{D}^*$  the boundary of  $\mathcal{D}^*$ .

The boundary  $\partial\mathcal{D}^*$  consists of a set of cycles. Each cycle is a sequence of arcs from one or more disks, with reflex vertices at the junctions between consecutive arcs. A cycle with only one arc is just a single disk. Such a cycle is already convex and constant-size, so we will not consider such cycles further. Let  $e$  be an arc on

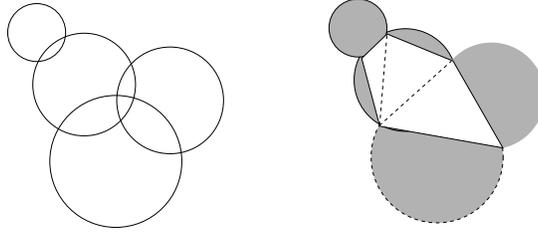


Fig. 1. Illustrating our desired partition (right) of the union of a set of disks (left): The ears in the partition are shaded and the dashed segments are triangulation diagonals.

$\partial\mathcal{D}^*$ , with reflex vertices  $u$  and  $v$  as its endpoints. Then the line segment  $\overline{uv}$  is contained in  $\mathcal{D}^*$ . This holds because both  $u$  and  $v$  lie on the boundary of a single disk  $D \in \mathcal{D}$  (the one where  $e$  is located). It follows that  $\overline{uv} \subseteq D \subseteq \mathcal{D}^*$ . We call the region bounded by  $e$  and the segment  $\overline{uv}$  an *ear*. An ear is convex and has constant size. Any two ears are interior-disjoint. We separate all the ears from the rest of  $\mathcal{D}^*$  by slicing off each ear with its bounding segment. This partitions  $\mathcal{D}^*$  into  $O(n)$  ears and  $O(n)$  polygonally bounded regions with a total of  $O(n)$  vertices. We finish the partition by triangulating each polygonal region, cutting  $\mathcal{D}^*$  into  $O(n)$  ears and triangles. See Fig. 1 for an example.

We have obtained a set of  $O(n)$  interior-disjoint, convex, and constant-size objects. Let  $S$  be the set of the common tangents of all these objects. Clearly,  $|S| = O(n^2)$ . Let  $S'$  be the set of the common tangents of the disks in  $\mathcal{D}$ . It is easy to know that  $S' \subseteq S$ . However, it is possible that a common tangent in  $S$  does not belong to  $S'$ . Recall that our goal is to compute the maximal segments of the common tangents in  $S'$ . To this end, we first compute the visibility complex of those objects in  $O(n \log n + |S|)$  time<sup>13,14</sup>. The visibility complex gives us the maximal segments of the common tangents in  $S$ <sup>13,14,22</sup>. Note that for any common tangent in  $S$ , we can determine in constant time whether it is in  $S'$ . Therefore, the maximal segments of the common tangents in  $S'$  are also obtained. Consequently, as discussed before, the disk separation problem is solved.

**Theorem 1.** *Given a set  $\mathcal{D}$  of  $n$  disks in the plane, we can solve the disks separability problem in  $O(n \log n + |S|)$  time, where  $|S| = O(n^2)$  and  $|S|$  is sensitive to the positions of the disks in  $\mathcal{D}$ .*

### 3. Obnoxious Line Location among Weighted Points (OLL)

In this section, we present our algorithm for the problem OLL. Our approach is parametric search which uses our disks separability algorithm in Section 2 as a decision making procedure. Below is the main result of this section.

**Theorem 2.** *Given a set of  $n$  weighted points in the plane, an obnoxious line can be determined in  $O(n^2 \log n)$  time.*

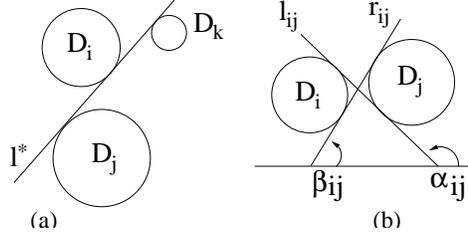


Fig. 2. (a) The line  $l^*$  is the inner common tangent of  $D_i$  and  $D_j$  (resp.,  $D_i$  and  $D_k$ ); the tangent point of  $D_i$  on  $l^*$  is between those of  $D_j$  and  $D_k$  on  $l^*$ . (b) Illustrating the two inner common tangents of two disks.

In the following, in Section 3.1, we discuss some geometric properties for the problem OLL. Our main algorithm for OLL is given in Section 3.2.

### 3.1. Preliminaries

Given a point set  $P = \{p_1, p_2, \dots, p_n\}$ , where each  $p_i$  has a weight  $w_i > 0$ , for any  $\epsilon > 0$  and each  $1 \leq i \leq n$ , define  $D_i(\epsilon)$  to be a disk centered at  $p_i$  with radius  $\frac{\epsilon}{w_i}$ . Let  $\mathcal{D}(\epsilon)$  be the disk set  $\{D_i(\epsilon) \mid 1 \leq i \leq n\}$ . Suppose  $l^*$  is an obnoxious line for  $P$  and  $\epsilon^*$  is the corresponding error, i.e.,  $\epsilon^* = \min\{w_i \cdot d(l^*, p_i) \mid p_i \in P\}$ . For any  $\epsilon > 0$ , it is easy to see that if  $\mathcal{D}(\epsilon)$  is separable, then  $\epsilon \leq \epsilon^*$  must hold; otherwise,  $\epsilon^* < \epsilon$ . Based on the results in <sup>3</sup>, an obnoxious line  $l^*$  has the following (almost self-evident) properties. We say that two disks are *outer tangent* to each other if they are mutually tangent and no disk is contained by the other.

**Lemma 2.** *One of the following two cases must hold for  $l^*$ : (1) There exist two disks in  $\mathcal{D}(\epsilon^*)$  outer tangent to each other and  $l^*$  passes through their tangent point (thus  $l^*$  is perpendicular to the line segment connecting the two disk centers); (2) there exist three disks  $D_i, D_j$ , and  $D_k$  in  $\mathcal{D}(\epsilon^*)$  such that  $l^*$  is the inner common tangent of  $D_i$  and  $D_j$ ,  $l^*$  is also the inner common tangent of  $D_i$  and  $D_k$ , and the tangent point of  $D_i$  on  $l^*$  is between the tangent points of  $D_j$  and  $D_k$  on  $l^*$  (see Fig. 2(a)).*

Let  $\epsilon_{ij}$  be the error such that  $D_i(\epsilon_{ij})$  and  $D_j(\epsilon_{ij})$  are outer tangent to each other, which corresponds to case (1) in Lemma 2. Denote by  $E_1$  the set of all such  $O(n^2)$  possible  $\epsilon_{ij}$ 's. Similarly, a corresponding error in case (2) of Lemma 2 is determined by three disks, and we denote by  $E_2$  the set of all these errors. Thus, we have  $|E_1| = O(n^2)$  and  $|E_2| = O(n^3)$ . According to Lemma 2,  $\epsilon^*$  must be in  $E_1 \cup E_2$ . By making use of our disks separability algorithm in Section 2 and the linear time selection algorithm <sup>23</sup>, a straightforward approach can find  $\epsilon^*$  and  $l^*$  in  $O(n^3)$  time. A more efficient algorithm based on parametric search is given in the next subsection.

### 3.2. The parametric search algorithm

We first compute the two elements  $\epsilon_1 < \epsilon_2$  in  $E_1$ , where  $\epsilon_1$  is the largest error such that  $\mathcal{D}(\epsilon_1)$  is separable and  $\epsilon_2$  is the smallest error such that  $\mathcal{D}(\epsilon_2)$  is not separable. Both  $\epsilon_1$  and  $\epsilon_2$  can be computed in  $O(n^2 \log n)$  time by utilizing the disks separability algorithm. Note that it must be  $\epsilon_1 \leq \epsilon^* < \epsilon_2$ . Thus, if  $\epsilon_1 < \epsilon^*$ , then  $\epsilon^*$  is attained in a situation corresponding to case (2) of Lemma 2. Furthermore, since the interval  $(\epsilon_1, \epsilon_2)$  contains no value in  $E_1$ , for any  $\epsilon', \epsilon'' \in (\epsilon_1, \epsilon_2)$  and any  $i \neq j$ ,  $D_i(\epsilon') \cap D_j(\epsilon') = \emptyset$  if and only if  $D_i(\epsilon'') \cap D_j(\epsilon'') = \emptyset$  (i.e., the intersection relation between  $D_i(\epsilon)$  and  $D_j(\epsilon)$  remains unchanged as the value of  $\epsilon$  varies in  $(\epsilon_1, \epsilon_2)$ ); in other words,  $D_i(\epsilon')$  and  $D_j(\epsilon')$  have inner common tangents if and only if  $D_i(\epsilon'')$  and  $D_j(\epsilon'')$  have inner common tangents. In the following discussion,  $\epsilon$  is always restricted to be inside the interval  $(\epsilon_1, \epsilon_2)$ .

For any two disks  $D_i(\epsilon)$  and  $D_j(\epsilon)$  that have inner common tangents, denote by  $l_{ij}$  (resp.,  $r_{ij}$ ) the inner common tangent that rotates clockwise (resp., counter-clockwise) when we increase the value of  $\epsilon$ . Denote by  $\alpha_{ij}(\epsilon)$  (resp.,  $\beta_{ij}(\epsilon)$ ) the angle defined by  $l_{ij}$  (resp.,  $r_{ij}$ ) and the  $x$ -axis (see Fig. 2(b)). To adapt to the forthcoming parametric search, we restrict all the angles  $\alpha_{ij}(\epsilon)$  and  $\beta_{ij}(\epsilon)$  in the interval  $[0, \pi)$ . In <sup>2</sup>,  $\alpha_{ij}(\epsilon)$  is decreasing and  $\beta_{ij}(\epsilon)$  is increasing as  $\epsilon$  is increased. However, in our problem setting, this is not always the case (due to different domain definitions). Precisely, initially,  $\alpha_{ij}(\epsilon)$  is decreasing, but after  $\alpha_{ij}(\epsilon) = 0$  (if ever),  $\alpha_{ij}(\epsilon)$  starts to decrease from  $\pi$ . Thus, the curve defined by  $\alpha_{ij}(\epsilon)$  consists of at most two disjoint continuous decreasing pieces. Fig. 3 shows two curves defined by two different  $\alpha$  functions. By a slight abuse of notation, we also use  $\alpha_{ij}(\epsilon)$  to denote the curve defined by  $\alpha_{ij}(\epsilon)$  with  $\epsilon \in (\epsilon_1, \epsilon_2)$ . For each continuous piece of the curve  $\alpha_{ij}(\epsilon)$ , define the *curve piece span* as the difference of the largest value of  $\alpha_{ij}(\epsilon)$  and the smallest value of  $\alpha_{ij}(\epsilon)$  in that curve piece. We then have the next lemma.

**Lemma 3.** *For any curve  $\alpha_{ij}(\epsilon)$ , if it has one piece, then its piece span is less than  $\pi/2$ ; if it has two pieces, then the sum of its two piece spans is less than  $\pi/2$ . Further, there exist no two  $\epsilon'$  and  $\epsilon''$  in  $(\epsilon_1, \epsilon_2)$  such that  $\epsilon' \neq \epsilon''$  and  $\alpha_{ij}(\epsilon') = \alpha_{ij}(\epsilon'')$ .*

**Proof.** When  $\epsilon = 0$ , each of the two disks  $D_i(\epsilon)$  and  $D_j(\epsilon)$  degenerates into a point that is the center of the disk. Denote by  $l_1$  the line containing the two centers, and by  $\alpha_1$  the angle defined by  $l_1$  and the  $x$ -axis. Let  $\epsilon_{ij}$  be the error such that  $D_i(\epsilon_{ij})$  and  $D_j(\epsilon_{ij})$  are outer tangent to each other and neither of them is contained by the other. Let  $p'$  be the tangent point of  $D_i(\epsilon_{ij})$  and  $D_j(\epsilon_{ij})$ . Denote by  $l_2$  the line passing through  $p'$  and perpendicular to  $l_1$ , and by  $\alpha_2$  the angle defined by  $l_2$  and the  $x$ -axis. Refer to Fig. 4 for an example. Note that when we increase  $\epsilon$  from 0 to  $\epsilon_{ij}$ ,  $l_{ij}$  will rotate from  $l_1$  to  $l_2$ . Thus,  $l_{ij}$  rotates exactly  $\pi/2$  if we increase  $\epsilon$  from 0 to  $\epsilon_{ij}$ . By the definitions of  $\epsilon_1$  and  $\epsilon_2$ , we have  $0 \leq \epsilon_1$  and  $\epsilon_2 \leq \epsilon_{ij}$ . Hence  $l_{ij}$  rotates less than  $\pi/2$  when  $\epsilon \in (\epsilon_1, \epsilon_2)$ . Note that the amount of angle rotated by  $l_{ij}$  is exactly the piece span if the curve  $\alpha_{ij}(\epsilon)$  has one piece or the sum of the two piece spans if the curve has two pieces. The first part of the lemma thus follows.

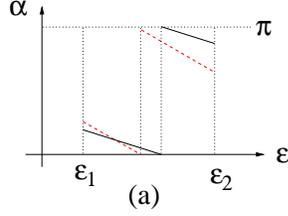


Fig. 3. Illustrating two  $\alpha$  curves, one solid (black) and one dotted (red). Each curve has two pieces. The two curves can intersect each other at most once.

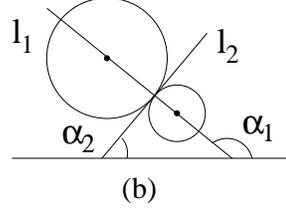


Fig. 4. Illustrating two outer tangent disks and the corresponding  $l_1, l_2, \alpha_1$ , and  $\alpha_2$ .

If  $\alpha_{ij}(\epsilon)$  has only one piece, then the second part is obviously true since the curve  $\alpha_{ij}(\epsilon)$  is strictly decreasing. In the following, we assume that  $\alpha_{ij}(\epsilon)$  has two pieces. Suppose there are  $\epsilon'$  and  $\epsilon''$  in  $(\epsilon_1, \epsilon_2)$  such that  $\epsilon' \neq \epsilon''$  and  $\alpha_{ij}(\epsilon') = \alpha_{ij}(\epsilon'')$ . Without loss of generality, assume  $\epsilon' < \epsilon''$ . Note that the point  $(\epsilon', \alpha_{ij}(\epsilon'))$  must be on the left piece and the point  $(\epsilon'', \alpha_{ij}(\epsilon''))$  must be on the right piece. Thus, the curve piece span of the left piece is at least  $\alpha_{ij}(\epsilon')$  and the curve piece span of the right piece is at least  $\pi - \alpha_{ij}(\epsilon'')$  (refer to Fig. 3 for an example). But this implies that the sum of the two curve piece spans is at least  $\pi$ , contradicting with the fact that their sum is less than  $\pi/2$ .  $\square$

The curve  $\beta_{ij}(\epsilon)$  has similar properties as in Lemma 3.

Note that if all  $O(n^2)$  inner common tangent angles  $\alpha_{ij}$  and  $\beta_{ij}$  between the  $D_i(\epsilon)$ 's are sorted at  $\epsilon = \epsilon^*$  ( $\epsilon^*$  is not yet known), then we have  $\alpha_{ij}(\epsilon^*) = \beta_{ik}(\epsilon^*)$  if three disks  $D_i, D_j$ , and  $D_k$  determine  $l^*$  as shown in case (2) of Lemma 2 (e.g., see Fig. 2(a)). Thus,  $l^*$  can be found if we sort all  $\alpha_{ij}(\epsilon^*)$ 's and  $\beta_{ij}(\epsilon^*)$ 's. Although we do not know the value of  $\epsilon^*$ , the sorting can still be done by parametric search, in which the disks separability algorithm is utilized to make decision for each comparison. Further, although each  $\alpha$  curve or  $\beta$  curve may not be strictly increasing or decreasing, we can still make our parametric search work by using the special properties in Lemma 3, as explained below.

To compare any two different  $\alpha$  angles  $\alpha_{ij}(\epsilon^*)$  and  $\alpha_{tk}(\epsilon^*)$ , we claim that the two curves  $\alpha_{ij}(\epsilon)$  and  $\alpha_{tk}(\epsilon)$  have at most one intersection if they do not overlap (Lemma 4 in <sup>2</sup> shows a similar result). Intuitively, one curve is decreasing “faster” than the other, and thus, although each of them may have two pieces, they still have at most one intersection (see Fig. 3). We omit the formal proof here. If the two curves overlap (this can be determined in constant time), then we have  $\alpha_{ij}(\epsilon^*) = \alpha_{tk}(\epsilon^*)$ ; otherwise, we need to compute the value  $\epsilon'$  (if any) such that  $\alpha_{ij}(\epsilon') = \alpha_{tk}(\epsilon')$ . Although both  $\alpha_{ij}(\epsilon)$  and  $\alpha_{tk}(\epsilon)$  may have two pieces, we can still compute their intersection in constant time. After obtaining  $\epsilon'$ , we can follow the standard parametric search technique to determine the comparison result of  $\alpha_{ij}(\epsilon^*)$  and  $\alpha_{tk}(\epsilon^*)$  and possibly shrink the interval  $(\epsilon_1, \epsilon_2)$ , which uses the disks separability algorithm.

The comparison of any two different  $\beta$  angles  $\beta_{ij}(\epsilon^*)$  and  $\beta_{tk}(\epsilon^*)$  can be performed similarly. To compare any two values  $\alpha_{ij}(\epsilon^*)$  and  $\beta_{tk}(\epsilon^*)$ , by Lemma 3 and similar observations for the  $\beta$  curves, we can show that the two curves  $\alpha_{ij}(\epsilon)$  and  $\beta_{tk}(\epsilon)$  have at most one intersection. We omit the proof here. Similarly, their intersection can be computed in constant time. By following the standard parametric search technique, we can determine the comparison result of  $\alpha_{ij}(\epsilon^*)$  and  $\beta_{tk}(\epsilon^*)$  and possibly shrink the interval  $(\epsilon_1, \epsilon_2)$ . In summary, we can sort all  $O(n^2)$  angles  $\alpha_{ij}(\epsilon^*)$  and  $\beta_{ij}(\epsilon^*)$  by the standard parametric search for sorting, in which each comparison takes  $O(n^2)$  time. By Cole's parametric search<sup>19</sup>, we obtain  $l^*$  and  $\epsilon^*$  in  $O(n^2 \log n)$  time. Theorem 2 thus follows.

#### 4. Obnoxious Line Location among Weighted Polygons (OLLP)

In this section, we present an improved algorithm for the OLLP problem. The previously best-known algorithm solves this problem in  $O(mn + n \log^2 n \log m + m^2 \log n \log^2 m)$  time<sup>2</sup>. We note that by slight modifications on our OLL algorithm, OLLP is solvable in  $O(n^2 \log n)$  time. Also in<sup>2</sup>, the unweighted version of OLLP was solved in  $O((m^2 + n \log m) \log n)$  time. By a slight modification on that algorithm, we reduce the running time. Our results in this section are summarized below.

**Theorem 3.** *The OLLP problem is solvable in  $O(mn + n \log^2 n + m^2 \log n)$  time; its unweighted version is solvable in  $O(m^2 + n \log m)$  time.*

In the following, we first present in Section 4.1 an algorithm for the decision version of the OLLP problem, which works as a crucial subroutine in the parametric search for OLLP. The algorithm improves the previous work in<sup>2</sup> by roughly an  $O(\log n)$  factor. In Section 4.2, we give the parametric search for OLLP. Finally, the unweighted OLLP is discussed in Section 4.3.

##### 4.1. The decision version of OLLP

Let  $\mathcal{P} = \{P_1, P_2, \dots, P_m\}$  be a set of  $m$  polygons with a total of  $n$  vertices in the plane such that each polygon  $P_i$  has a weight  $w_i > 0$ . Without loss of generality, we assume that all polygons in  $\mathcal{P}$  are convex since a closest point of a polygon  $P_i$  to a line not intersecting  $P_i$  is always at a vertex of the convex hull of  $P_i$ . (Thus, if any  $P_i$  is not convex, then we simply replace it by its convex hull; this takes  $O(n)$  time for the set  $\mathcal{P}$ .)

For any value  $\epsilon > 0$ , let  $\hat{P}_i(\epsilon)$  be the region that is the Minkowski sum of  $P_i$  and the disk centered at the origin with radius  $\frac{\epsilon}{w_i}$ , for each  $1 \leq i \leq m$  (see Fig. 5). Clearly,  $\hat{P}_i(\epsilon)$  is convex. Note that the boundary of  $\hat{P}_i(\epsilon)$  consists of line segments and arcs, and each arc corresponds to a vertex of  $P_i$ . Let  $\hat{\mathcal{P}}(\epsilon) = \{\hat{P}_i(\epsilon) \mid 1 \leq i \leq m\}$ . We call a line  $l$  a *separation line* for  $\hat{\mathcal{P}}(\epsilon)$  if  $l$  does not intersect the interior of any region in  $\hat{\mathcal{P}}(\epsilon)$  and there is at least one region of  $\hat{\mathcal{P}}(\epsilon)$  on each side of  $l$ . If there exists a separation line for  $\hat{\mathcal{P}}(\epsilon)$ , then we say that  $\hat{\mathcal{P}}(\epsilon)$  is *separable*. Suppose  $l^*$  is

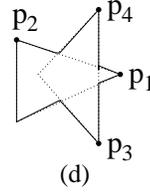
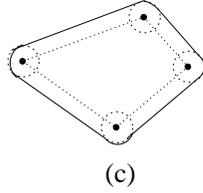


Fig. 5. Illustrating the Minkowski sum (the solid curve) of a polygon (the dashed one) and a disk centered at the origin.

Fig. 6. Illustrating two convex hulls whose boundaries  $\partial_i$  and  $\partial_j$  intersect transversally four times. The solid segments are the boundary of their union;  $p_1$  and  $p_2$  are on  $\partial_i \cap \partial$  and  $p_3$  and  $p_4$  are on  $\partial_j \cap \partial$ .

an obnoxious line for  $\mathcal{P}$  and  $\epsilon^*$  is the corresponding error. Given  $\epsilon > 0$ , if  $\hat{\mathcal{P}}(\epsilon)$  is separable, then it must be  $\epsilon \leq \epsilon^*$ ; otherwise,  $\epsilon^* < \epsilon$ .

The decision version of OLLP is equivalent to the following problem: Given  $\epsilon > 0$ , determine whether  $\hat{\mathcal{P}}(\epsilon)$  is separable, and if yes, find a separation line. The algorithm in <sup>2</sup> solves this problem in  $O((m^2 + n \log m) \log n)$  time. We give the following improved result.

**Theorem 4.** *The decision version of OLLP is solvable in  $O(n \log n + m^2)$  time.*

Note that the approach in <sup>18</sup> can find a separation line for  $m$  polygons of totally  $n$  vertices in  $O(n^2)$  time. Our result in Theorem 4 is faster for small  $m$ .

Our algorithm can be viewed as an extension of our disks separability solution, with an additional preprocessing procedure. In the following, we first give in Lemma 4 an algorithm under the assumption that every two different regions in  $\hat{\mathcal{P}}(\epsilon)$  have at most four common tangents, and then show that we can preprocess  $\hat{\mathcal{P}}(\epsilon)$  so that this assumption holds while the solution for the original problem is not affected.

**Lemma 4.** *If any two different regions in  $\hat{\mathcal{P}}(\epsilon)$  have at most four common tangents, then in  $O(n \log n + m^2)$  time, we can either find a separation line for  $\hat{\mathcal{P}}(\epsilon)$  or report that no separation line exists.*

**Proof.** The algorithm can be viewed as an extension of our disks separability algorithm, which we sketch below. Recall that every region in  $\hat{\mathcal{P}}(\epsilon)$  is convex. Similar to Lemma 1, there exists a separation line for  $\hat{\mathcal{P}}(\epsilon)$  if and only if there is a separation line that contains a common tangent of two regions in  $\hat{\mathcal{P}}(\epsilon)$ . Again, we can first compute the maximal segments of all common tangents and then check whether there exists a maximal segment that has both endpoints at the infinity. For simplicity of discussion, we assume no two convex regions in  $\hat{\mathcal{P}}(\epsilon)$  are tangent to each other. Since any two convex regions have at most four common tangents, the boundaries of any two regions intersect at most twice.

Again, the first step is to compute the union of the regions in  $\hat{\mathcal{P}}(\epsilon)$ , denoted by  $\hat{\mathcal{P}}^*$ . For this, we can use the standard divide-and-conquer approach. Similar idea has

been used elsewhere, e.g.,<sup>24,25</sup>. Here, we use the linear time algorithm in<sup>20</sup> as the merge step. Since the boundaries of any two convex regions intersect at most twice, the combinatorial size of the overlay of all convex regions in  $\hat{\mathcal{P}}(\epsilon)$  is  $O(n + m^2)$ . Therefore, computing the union  $\hat{\mathcal{P}}^*$  takes  $O(n \log m + m^2)$  time.

The second step is to partition the union  $\hat{\mathcal{P}}^*$  into  $O(n)$  interior-disjoint and convex objects. To this end, by using similar idea as in our disks separation algorithm, we can compute  $O(m)$  ears that are convex. However, in this case, each ear may not be constant-size because its curved boundary may have multiple vertices. By separating all ears from  $\hat{\mathcal{P}}^*$ , we can obtain  $O(m)$  polygons of totally  $O(n)$  vertices. Triangulating these polygons gives  $O(n)$  triangles. In summary, we have partitioned  $\hat{\mathcal{P}}^*$  into  $O(m)$  ears and  $O(n)$  triangles. Let  $S$  be the set of common tangents of all these ears and triangles. We claim that  $|S| = O(n + m^2)$ . Indeed, each common tangent in  $S$  is either a common tangent of two ears or a side of a triangle because all these triangles are in the interior of  $\hat{\mathcal{P}}^*$ . Since there are  $O(m)$  ears and  $O(n)$  triangles,  $|S| = O(n + m^2)$  holds. The algorithm in<sup>13</sup> can also compute the visibility complex of non-constant convex objects, and it takes  $O(n \log n + |S|)$  time to compute the visibility complex of all ears and triangles.

Let  $S'$  be the set of common tangents of the convex regions in  $\hat{\mathcal{P}}(\epsilon)$ . Clearly,  $S' \subseteq S$ . The visibility complex gives the maximal segments of the common tangents in  $S$ , and thus, the maximal segments of the common tangents in  $S'$  are also available. Consequently, we can either find a separation line for  $\hat{\mathcal{P}}(\epsilon)$  or report that no separation line exists. The overall running time of the algorithm is  $O(n \log n + m^2)$ . The lemma thus follows.  $\square$

To satisfy the conditions of Lemma 4, we perform a preprocessing on  $\hat{\mathcal{P}}(\epsilon)$ , which uses a simple sweeping algorithm to find all connected components of  $\hat{\mathcal{P}}(\epsilon)$  and then for each connected component, computes its convex hull. It should be noted that in this setting, two regions in  $\hat{\mathcal{P}}(\epsilon)$  are considered to be *connected* if and only if they intersect in their interior. Since all regions in  $\hat{\mathcal{P}}(\epsilon)$  are convex, this preprocessing can be done in  $O(n \log n)$  time by using the algorithm in<sup>26</sup>. Clearly, the number of connected components in  $\hat{\mathcal{P}}(\epsilon)$  is no more than  $m$ . Let  $\hat{\mathcal{P}}'$  be the set of convex hulls thus resulted, with  $|\hat{\mathcal{P}}'| \leq m$ . The next lemma establishes the fact that  $\hat{\mathcal{P}}'$  can be used as the input to the algorithm in Lemma 4 to solve our original problem, which yields the result of Theorem 4.

**Lemma 5.** *A line is a separation line for  $\hat{\mathcal{P}}(\epsilon)$  if and only if it is a separation line for  $\hat{\mathcal{P}}'$ . Further, any two different convex hulls in  $\hat{\mathcal{P}}'$  have at most four common tangents.*

**Proof.** First of all, it is easy to see that a separation line for  $\hat{\mathcal{P}}'$  must also be a separation line for  $\hat{\mathcal{P}}(\epsilon)$ . On the other hand, suppose  $l$  is a separation line for  $\hat{\mathcal{P}}(\epsilon)$ . Denote by  $\hat{\mathcal{P}}_{(1)}(\epsilon)$  the set of regions in  $\hat{\mathcal{P}}(\epsilon)$  on one side of  $l$  and by  $\hat{\mathcal{P}}_{(2)}(\epsilon)$  the set of regions in  $\hat{\mathcal{P}}(\epsilon)$  on the other side of  $l$ . Since  $\hat{\mathcal{P}}_{(1)}(\epsilon)$  and  $\hat{\mathcal{P}}_{(2)}(\epsilon)$  are separated by

$l$ , any connected component of  $\hat{\mathcal{P}}(\epsilon)$  cannot contain both a region in  $\hat{\mathcal{P}}_{(1)}(\epsilon)$  and a region in  $\hat{\mathcal{P}}_{(2)}(\epsilon)$ . Let  $\hat{\mathcal{P}}'_{(1)}$  (resp.,  $\hat{\mathcal{P}}'_{(2)}$ ) be the set of convex hulls of the connected components of the regions in  $\hat{\mathcal{P}}_{(1)}(\epsilon)$  (resp.,  $\hat{\mathcal{P}}_{(2)}(\epsilon)$ ). Since  $l$  does not intersect the interior of any region in  $\hat{\mathcal{P}}_{(1)}(\epsilon)$  (resp.,  $\hat{\mathcal{P}}_{(2)}(\epsilon)$ ),  $l$  cannot intersect the interior of any convex hull in  $\hat{\mathcal{P}}'_{(1)}$  (resp.,  $\hat{\mathcal{P}}'_{(2)}$ ). Thus,  $l$  is also a separation line for  $\hat{\mathcal{P}}'$ .

Obviously, two different convex hulls in  $\hat{\mathcal{P}}'$  have at most four common tangents if one of the following cases holds: (1) They do not intersect in their interior; (2) one convex hull contains the other. The only remaining case to consider is that the two convex hulls intersect in their interior and neither of them contains the other. Let  $\hat{P}'_i$  and  $\hat{P}'_j$  be any two such convex hulls in  $\hat{\mathcal{P}}'$ . Denote by  $\partial_i$  and  $\partial_j$  their boundaries, respectively. We claim that  $\partial_i$  and  $\partial_j$  can intersect each other transversally at most twice. Let  $\partial$  be the boundary of the union of  $\hat{P}'_i$  and  $\hat{P}'_j$  (see Fig. 6). Thus,  $\partial$  consists of parts of  $\partial_i$  and parts of  $\partial_j$ . Suppose  $\partial_i$  and  $\partial_j$  intersect each other transversally more than twice. Then there must exist two distinct vertices  $p_1$  and  $p_2$  of  $\partial_i$  lying on  $\partial - \partial_j$  and two distinct vertices  $p_3$  and  $p_4$  of  $\partial_j$  lying on  $\partial - \partial_i$ , such that their cyclical order around  $\partial$  is  $p_1, p_3, p_2$ , and  $p_4$  (see Fig. 6). But this implies that the two connected components of  $\hat{\mathcal{P}}(\epsilon)$  corresponding to  $\hat{P}'_i$  and  $\hat{P}'_j$  intersect in their interior, a contradiction. Since  $\partial_i$  and  $\partial_j$  can intersect each other transversally at most twice, there can be at most two common tangents between  $\hat{P}'_i$  and  $\hat{P}'_j$ . The lemma thus follows.  $\square$

#### 4.2. Solving OLLP (the optimization version)

We follow the parametric search scheme in <sup>2</sup>, but use our improved OLLP decision algorithm to make decisions. Cole's parametric search is applied instead of Megiddo's one used in <sup>2</sup>. An improved preprocessing step for the parametric search is also given.

Let  $l^*$  be an obnoxious line for  $\mathcal{P}$  and  $\epsilon^*$  be its corresponding error. Lemma 2 for the OLL problem can be applied to the OLLP problem here if we replace the disks of  $\mathcal{D}(\epsilon)$  by the convex regions of  $\hat{\mathcal{P}}(\epsilon)$ . For any two different regions  $\hat{P}_i(\epsilon)$  and  $\hat{P}_j(\epsilon)$  in  $\hat{\mathcal{P}}(\epsilon)$ , define  $l_{ij}(\epsilon)$  (resp.,  $r_{ij}(\epsilon)$ ) as the inner common tangent that rotates clockwise (resp., counterclockwise) when the value of  $\epsilon$  is increased. Denote by  $\alpha_{ij}(\epsilon)$  (resp.,  $\beta_{ij}(\epsilon)$ ) the angle defined by  $l_{ij}(\epsilon)$  (resp.,  $r_{ij}(\epsilon)$ ) and the  $x$ -axis (similar to Fig. 2(b)). As noted in <sup>2</sup>, if the  $O(m^2)$  angles  $\alpha_{ij}(\epsilon^*)$  and  $\beta_{ij}(\epsilon^*)$  are sorted, then for candidates of case (1) in Lemma 2, we have  $\alpha_{ij}(\epsilon^*) = \beta_{ij}(\epsilon^*)$ ; for candidates of case (2),  $\alpha_{ij}(\epsilon^*) = \beta_{ik}(\epsilon^*)$  holds (similar to Fig. 2(a)).

Although the value of  $\epsilon^*$  is not yet known, we can sort all angles  $\alpha_{ij}(\epsilon^*)$  and  $\beta_{ij}(\epsilon^*)$  by parametric search. But before doing so, as in <sup>2</sup>, we need a preprocessing step. For any two  $\hat{P}_i(\epsilon)$  and  $\hat{P}_j(\epsilon)$  with  $w_i \neq w_j$ , when increasing the value of  $\epsilon$ , the arcs supporting their inner common tangents may change. Precisely, suppose when  $\epsilon = \epsilon_1$ ,  $l_{ij}(\epsilon_1)$  is supported by the arc of  $\hat{P}_i(\epsilon_1)$  corresponding to the vertex  $v_i$  of  $P_i$  and the arc of  $\hat{P}_j(\epsilon_1)$  corresponding to the vertex  $v_j$  of  $P_j$ , and when

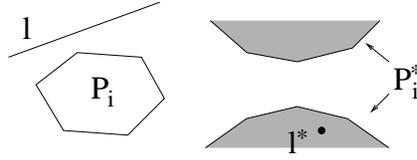


Fig. 7. Illustrating the dual  $P^*$  of a polygon  $P$ . The point  $l^*$  is the dual of the line  $l$ .

$\epsilon = \epsilon_2$ ,  $l_{ij}(\epsilon_2)$  is supported by the arc of  $\hat{P}_i(\epsilon_2)$  corresponding to the vertex  $v_{i''}$  of  $P_i$  and the arc of  $\hat{P}_j(\epsilon_2)$  corresponding to the vertex  $v_{j''}$  of  $P_j$ . Then it is possible that  $v_{i'} \neq v_{i''}$  and  $v_{j'} \neq v_{j''}$ . The goal of the preprocessing is to determine those arcs supporting all inner common tangents at  $\epsilon = \epsilon^*$ . Lemma 3 in <sup>2</sup> gives an  $O(mn + (m^2 + n \log m) \log^2 n)$  time algorithm to accomplish this task, whose search space consists of  $O(mn)$  values of  $\epsilon$  and an algorithm for the decision version of OLLP is utilized to prune these values (refer to <sup>2</sup> for more details). By using our improved OLLP decision algorithm, the preprocessing step can be performed in  $O(mn + (m^2 + n \log n) \log n)$  time.

After the preprocessing step, we have the arcs that support all inner common tangents at  $\epsilon = \epsilon^*$ . Thus, we can use the same parametric search method as for the OLL problem in Section 3.2, which applies Cole’s parametric search <sup>19</sup>. However, instead of  $O(n^2)$  tangents involved in the disks case, here we have  $O(m^2)$  inner common tangents. By using our improved OLLP decision algorithm to make decisions, the parametric search then solves the OLLP problem in  $O((m^2 + n \log n) \log m)$  time. Hence, the first part of Theorem 3 follows.

### 4.3. The unweighted OLLP

The unweighted OLLP was solved in  $O((m^2 + n \log m) \log n)$  time in <sup>2</sup>. We first briefly describe the algorithm in <sup>2</sup> and then explain how we improve it by a slight modification.

The algorithm in <sup>2</sup> is based on duality. For each polygon  $P_i$  in  $\mathcal{P}$ , denote by  $U_i$  (resp.,  $L_i$ ) the dual of all lines above (resp., below)  $P_i$  (see Fig. 7). Note that  $U_i$  and  $L_i$  are disjoint convex polygonal regions (one unbounded from above and the other unbounded from below). Hence,  $P_i^* = U_i \cup L_i$  is the set of all points dual to the lines not intersecting (the interior of)  $P_i$ . To solve the unweighted OLLP, it suffices to compute  $\mathcal{P}^* = \bigcap_{i=1}^m P_i^*$  since any point in  $\mathcal{P}^*$  corresponds to, in the primal plane, a line that does not intersect any polygon in  $\mathcal{P}$  (see Fig. 7). The complexity  $|\mathcal{P}^*|$  of  $\mathcal{P}^*$  is defined as the number of its vertices (which is proportional to the number of its edges and faces). To analyze  $|\mathcal{P}^*|$  and compute  $\mathcal{P}^*$ , the authors in <sup>2</sup> used the results in <sup>24</sup>, where  $|\mathcal{P}^*|$  was shown to be  $O(m^2 + n)$  and a divide-and-conquer algorithm was given to compute  $\mathcal{P}^*$  in  $O((m^2 + n \log m) \log n)$  time. This divide-and-conquer algorithm makes use of the algorithm in <sup>27</sup> in the merge step to merge two planar subdivisions. If we replace this merging algorithm by the linear time algorithm in <sup>20</sup> for computing the overlay of two simply connected

planar subdivisions, then  $\mathcal{P}^*$  can be computed in  $O(m^2 + n \log m)$  time. As noted in <sup>2</sup>, after  $\mathcal{P}^*$  is obtained, the unweighted OLLP can be solved by exploring  $\mathcal{P}^*$  in  $O(m^2 + n)$  time. In summary, solving the unweighted OLLP takes  $O(m^2 + n \log m)$  time. The second part of Theorem 3 thus follows.

## 5. Conclusion

In this paper, we gave new algorithms for the obnoxious line location problems that improve the previous work. A remaining issue is that our algorithms utilize Cole's parametric search <sup>19</sup>, which relies on a sorting network. All known constructions for such networks are fairly complicated and involve large constants, and thus the parametric search approach is mainly of theoretical interest. Therefore, it would be interesting to have practical and efficient algorithms for the obnoxious line location problems.

## 6. Acknowledgment

The authors would like to thank Joseph S.B. Mitchell for his helpful suggestions on solving the disks separability problem and on improving the presentation of the paper.

## References

1. M. Houle and A. Maciel. Finding the widest empty corridor through a set of points. In G.T. Toussaint, editor, *Snapshots of Computational and Discrete Geometry*. TR SOCS-88.11, Dept. of Computer Science, McGill University, Montreal, Canada, 1988.
2. J.M. Díaz-Báñez, P.A. Ramos, and P. Sabariego. The maximin line problem with regional demand. *European Journal of Operational Research*, 181(1):20–29, 2007.
3. Z. Drezner and G.O. Wesolowsky. Location of an obnoxious route. *Journal of Operational Research Society*, 40(11):1011–1018, 1989.
4. M. Al-Bow, C. Durso, M.A. Lopez, and Y. Mayster. Locating an obnoxious line through a set of weighted points. In *Proc. of 26th European Workshop on Computational Geometry*, pages 217–220, 2010.
5. R. Janardan and F.P. Preparata. Widest-corridor problems. *Nordic Journal of Computing*, 1(2):231–245, 1994.
6. S. Chattopadhyay and P. Das. The  $k$ -dense corridor problems. *Pattern Recognition Letters*, 11(7):463–469, 1990.
7. C. Shin, S.Y. Shin, and K. Chwa. The widest  $k$ -dense corridor problems. *Information Processing Letters*, 68(1):25–31, 1998.
8. S. Cheng. Widest empty  $L$ -shaped corridor. *Information Processing Letters*, 58(6):277–283, 1996.
9. S. Bereg, J.M. Díaz-Báñez, C. Seara, and I. Ventura. On finding widest empty curved corridors. *Computational Geometry: Theory and Applications*, 38(3):154–169, 2007.
10. M.E. Houle and G.T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):761–765, 1988.
11. D.T. Lee and Y.F. Wu. Geometric complexity of some location problems. *Algorithmica*, 1(1–4):193–211, 1986.

12. N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM*, 30(4):852–865, 1983.
13. M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete and Computational Geometry*, 16(4):419–453, 1996.
14. M. Pocchiola and G. Vegter. The visibility complex. *International Journal of Computational Geometry and Applications*, 6(3):279–308, 1996.
15. A. Efrat, G. Rote, and M. Sharir. On the union of fat wedges and separating a collection of segments by a line. *Computational Geometry: Theory and Applications*, 3(5):277–288, 1993.
16. A. Efrat and O. Schwarzkopf. Separating and shattering long line segments. *Information Processing Letters*, 64(6):309–314, 1997.
17. R. Freimer, J.S.B. Mitchell, and C.D. Piatko. On the complexity of shattering using arrangements. In *Proc. of 2nd Canadian Conference on Computational Geometry*, pages 218–222, 1990.
18. S.C. Nandy, T. Asano, and T. Harayama. Shattering a set of objects in 2D. *Discrete Applied Mathematics*, 122:183–194, 2002.
19. R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *Journal of the ACM*, 34(1):200–208, 1987.
20. U. Finke and K.H. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proc. of the 11th Annual ACM Symposium on Computational Geometry*, pages 119–126, 1995.
21. H. Imai, M. Iri, and K. Murota. Voronoi diagram in the Laguerre geometry and its applications. *SIAM Journal on Computing*, 14(1):93–105, 1985.
22. Olaf Hall-Holt. *Kinetic Visibility*. PhD thesis, Stanford University, 2002.
23. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
24. P.K. Agarwal and M. Sharir. Ray shooting amidst convex polygons in 2D. *Journal of Algorithms*, 21(3):508–519, 1996.
25. K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete and Computational Geometry*, 1(1):59–71, 1986.
26. F. Nielsen and M. Yvinec. Output-sensitive convex hull algorithms of planar convex objects. *International Journal of Computational Geometry and Applications*, 8(1):39–66, 1998.
27. B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM*, 39(1):1–54, 1992.