The Maximin Line Problem with Regional Demand

J. M. Díaz-Báñez* P.A. Ramos† P. Sabariego‡

September 22, 2005

Abstract

Given a family $\mathcal{P} = \{P_1, ..., P_m\}$ of m polygonal regions (possibly intersecting) in the plane, we consider the problem of locating a straight line ℓ intersecting the convex hull of $\mathcal{P}$ and such that $\min_k d(P_k, \ell)$ is maximal. We give an algorithm that solves the problem in time $O((m^2 + n \log m) \log n))$ using $O(m^2 + n)$ space, where n is the total number of vertices of $P_1, ..., P_m$. The previous best running time for this problem was $O(n^2)$. We also consider several variants of this problem which include a three dimensional version – the maximin plane problem –, the weighted problem and considering measuring distance different to the Euclidean one.

Keywords: Location, Computational Geometry, Linear Facility, Duality.

1 Introduction

The advances in computational geometry have given rise to the development of efficient algorithms to solve facility location problems. In fact, since its beginning, there has been a strong interaction between both fields. A proof of this is the fact that one of the problems which were solved in the origins of computational geometry, computing the *minimum spanning circle*, is a geometric interpretation of the best known facility location problem, the *1-center problem.*

This interaction has resulted in a wealth of papers and results of interest to researchers and practitioners in both fields. In this sense, the surveys [29] and [10] establish the current state-of-the-art for single and non-single facility location problems, respectively.

*Corresponding author: Departamento. de Matemática Aplicada II, Escuela Superior de Ingenieros, Avda. de los Descubrimiento s/n, 41092, Sevilla, SPAIN (dbanez@us.es), partially supported by Project BFM2003-04062.

†Universidad de Alcalá, SPAIN (pedro.ramos@uah.es), partially supported by Projects TIC2003-08933-C02-01 and BMF2002-04402-C02-01.

‡Universidad de Cantabria, SPAIN (pilar.sabariego@unican.es), supported in part by Project BFM2001-1153.

Linear facility location has been of great interest both in location theory [25, 26, 33, 3] and in computational geometry [23, 20, 22]. In this paper we deal with the placement of an undesirable facility modelled by a line, amidst polygonal regions. The computation of obnoxious routes has become a topic of increasing interest in recent years. In fact, there is a natural reason to obtain maximum 'clearance' in many applications as the design of channels for transportation of hazardous materials or paths avoiding obstacles in robotics [7, 8].

The problem of locating a linear route which maximizes the minimum weighted Euclidean distance to a set of points was first considered in [12]. In this paper, a naive $O(n^3)$-time algorithm was proposed. However, by using topological sweeping and duality, the unweighted version of this problem can be solved in $O(n^2)$ time and $O(n)$ space [19]. In fact, [19] address the problem of computing a widest empty corridor through a set of points in the plane, which is precisely an equivalent formulation of the maximin line problem.

On the other hand, although in the classical facility location problems the existing facilities are represented as a set of points, there exists a real interest in considering models involving regions as demand sites. In this way, different real world situations can be modelled better than the classical versions [11, 27]. In this case, the distance between the facility and the customer may be calculated as some forme of expected or average travel distance, for instance, see [5], or the distance to the closest point on the boundary of the region [4].

The problem of locating an obnoxious line in presence of polygonal regions was considered in [18, 17]. A brute-force $O(n^4)$-time algorithm was proposed both with Euclidean and polyhedral norms. However, because the problem is just the empty corridor problem, an efficient $O(n^2)$-time algorithm can be found in [21] for the unweighted Euclidean case. Given a set $\mathcal{P} = \{P_1, ..., P_m\}$ of m polygonal regions in the plane with a total of n edges, we wish to compute efficiently a maximin line with respect to $\mathcal{P}$ for a general case. Since in typical applications $m \ll n$, we would like to have an algorithm whose performance depends both on m and n and which is significantly better than the previous one when $m \ll n$. Our results include:

1. An algorithm to compute the unweighted maximin line through $\mathcal{P}$, both for the Euclidean metric and for a general metric, in $O((m^2 + n \log m) \log n)$ time and $O(m^2 + n)$ space. This result improves the $O(n^2)$-time algorithm of [21] if $m \ll n$.

2. The adaptation of the method to solve the general case in which the polygonal regions are weighted (the weights represent the number of inhabitants, for instance). This variant of the problem is solved in $O(nm^2 \log m + n \log^2 n \log m)$ time $O(m^2+n)$ space, which significantly improves the bound $O(n^4)$ of [18, 17].

3. An algorithm to solve an extension of the Euclidean unweighted version of the

problem to three dimensions, the *obnoxious plane problem in presence of polyhedra*, in $O(m^2 n \log(m^2 n))$ time and $O(m^2 n)$ space. This bound improves on the $O(n^3)$ time proposed in [9].

The rest of the paper is organized as follows. In Section 2 we state the problem, present some geometric preliminares and briefly describe known computational result. Our general approach is proposed in Section 3 for the Euclidean unweighted case. In Section 4 the method is adapted to solve a more general model. Section 5 address the three-dimensional extension of the problem.

2 Overview

We start by introducing some notation and considering a few geometric preliminaries. A summary of related results is also presented. Let $\mathcal{P} = \{P_1, \dots, P_m\}$ be a set of m polygonal regions in the plane with a total of n vertices. The distance between a polygon P and a straight line ℓ is given by the shortest distance based on the distance measure d, i.e. $d(P, \ell) = \min_{p \in P, x \in \ell} d(p, x)$, where $d(p, x)$ denotes the Euclidean distance between points p and x. First, we address the Euclidean case and then we adapt the approach to solve other versions, including weighted and arbitrary distances.

In the definition of an obnoxious facility location problem, the location of the facility must be constrained, as otherwise it may be simply removed to infinity. The facility is normally constrained to go through some sort of bounding region. As in [18], we are looking for a linear route inside the convex hull of the set $\mathcal{P}$.

The *maximin line through $\mathcal{P}$ problem* can be formalized as follows:

Given a set $\mathcal{P}$ of m (possibly intersecting and non-convex) polygons with a total of n vertices, compute a line ℓ such that

1. $\min_{P \in \mathcal{P}} d(P, \ell)$ *is maximal, and*

2. ℓ *divides $\mathcal{P}$ into two non-empty subsets.*

Because the closest point of a polygon P to a line ℓ not intersecting P is always a vertex of the convex hull of P, hereafter we consider that polygons are convex and, if this is not the case, we compute their convex hull as a preliminary step.

The problem can be reformulated equivalently as the computation of the widest empty corridor through polygonal obstacles [21]. In this geometric formulation, an empty corridor C, through $\mathcal{P}$, is the open region of the plane that is enclosed by two parallel straight lines intersecting the convex hull of $\mathcal{P}$ and such that the region does not intersect any polygon in $\mathcal{P}$.

Let us observe that a given set $\mathcal{P}$ may have no empty corridor through it and, if this is the case, the maximin line problem has no solution. Therefore, *decision* (deciding whether or not there exists a line ℓ through $\mathcal{P}$ not intersecting any polygon) and *optimization* (finding the farthest one) problems can be independently considered. This fact suggests trying to compute first a feasible set of lines and then find the optimal solution in that set.

The following lemma characterizes the solution to our problem. The proof is the same as in [19], where an analogous result is presented for the case of points instead of polygons.

Lemma 1 *Let ℓ^* be an optimal line and let ℓ_1 and ℓ_2 be the bounding lines of the corridor generated by ℓ^*. Then, one of the following conditions must hold:*

(a) ℓ_1 and ℓ_2 contain vertices v_1 and v_2, and the lines are perpendicular to the line segment connecting v_1 and v_2.

(b) There are two vertices on ℓ_1 and one vertex on ℓ_2 (or the opposite) and, furthermore, the vertex on ℓ_2 is between the vertices on ℓ_1 when viewed from a direction orthogonal to ℓ^*.

This lemma guarantees an $O(n^3)$ upper bound on the number of candidate lines for the two types of corridors. In [18, 17] , a straightforward $O(n^4)$-time algorithm is proposed by exhaustively considering all possible cases and finding the optimal one. A more efficient algorithm was proposed in [21].

The main idea is to interpret conditions (a) and (b) of Lemma 1 in the dual plane by using the duality transformation mapping the non vertical line ℓ with equation $y = mx - n$ to its dual point $\ell^* = (m, n)$ and the point $p = (a, b)$ to its dual line $p^* : y = ax - b$. We will use two properties of this transformation:

1. it preserves the above-below relation between points and lines: point p lies above line ℓ iff point ℓ^* lies above line p^*.
2. parallel lines are mapped to points with the same abscissa and, therefore, a slab is mapped to a vertical segment.

Using these properties, it follows that if pq is a segment, the dual of the set of lines intersecting pq is the double wedge defined by lines p^* and q^* and not containing the vertical line.

Let $\mathcal{H}$ be the set of lines dual to vertices of $\mathcal{P}$ and let $\mathcal{A}(\mathcal{H})$ be the arrangement in the plane induced by $\mathcal{H}$. The properties of the duality transform can be used to characterize in $\mathcal{A}(\mathcal{H})$ the sets of type-(a) and type-(b) corridors of Lemma 1.

A corridor C with bounding lines ℓ_1 and ℓ_2 is represented in the dual plane by the vertical segment with endpoints ℓ_1^* and ℓ_2^*. If C is an type-(a) empty corridor, then C

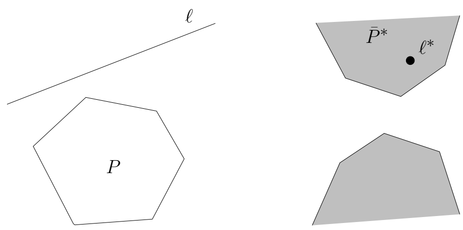


Figure 1: Dual interpretation of a free-collision line.

corresponds to a vertical segment inside a face of $\mathcal{A}(\mathcal{H})$ that connects a vertex and a edge of that cell. Similarly, a type-(b) empty corridor corresponds to a vertical segment inside a cell connecting an edge with an edge. In this case, the uniqueness of the segment follows from the perpendicularity condition.

Furthermore, the set of lines intersecting a given set of edges of $\mathcal{P}$ corresponds to a face of $\mathcal{A}(\mathcal{H})$ and, in particular, lines avoiding all the polygons of $\mathcal{P}$ correspond to some set of faces of the arrangement. As observed in [21], the topological sweep of [13] can be adapted to compute such set of faces and therefore we have:

Theorem 1 ([21]) *An optimal maximin line for the set of polygons $\mathcal{P}$ can be computed in $O(n^2)$ time and $O(n)$ space, where n is the total number of vertices of the polygons.*

3 Our approach

In this section we show that if m is small compared to n it is more convenient to avoid the construction of the whole arrangement of lines dual to vertices of $\mathcal{P}$. We show that the set of lines not intersecting any polygon in $\mathcal{P}$ has complexity $O(m^2 + n)$ and can be constructed in time $O((m^2 + n \log m) \log n)$. Therefore, if $m = o(n)$, the time complexity is improved perhaps with some extra memory cost, while if $m = O(\sqrt{n})$ time complexity is improved with the same space complexity.

Given a polygon $P \in \mathcal{P}$, let us denote by U_P and L_P the dual sets of the lines above P and bellow P, respectively. It is well known that U_P and L_P are disjoint convex polygons (one unbounded from above, the other unbounded from below), as shown in Figure 1. Then, $\bar{P}^* = U_P \cup L_P$ is the set of points dual to the lines not intersecting P. A vertex of $\bar{P}^*$ is the dual of a line supporting an edge of P.

We are interested in the set $\mathcal{A}^* = \bigcap_{P \in \mathcal{P}} \bar{P}^*$ because a point $\ell^* \in \mathcal{A}^*$ corresponds, in primal space, to a line ℓ that does not intersect any polygon in $\mathcal{P}$.

The complexity of this set is defined as the sum of its vertices, edges and faces and is proportional to the number of vertices. A vertex v of $\mathcal{A}^*$ is either a vertex of some polygonal region $\bar{P}^*$ or a point dual to a common tangent of two polygons in $\mathcal{P}$ that does not intersect any other polygon. Clearly, there are at most n vertices of the first type.

In [1] it is shown that the number of vertices of the second type is $O(m^2 + n)$ using an argument similar to the following one, that we include here for the sake of completeness. Let $Q_1, Q_2, \ldots, Q_t$, where $t \leq m$, be the connected components of $\bigcup_{P \in \mathcal{P}} P$. A vertex of the second type corresponds either to a line tangent to some Q_i – there are $O(n)$ of those – or to a common tangent of Q_i and Q_j. Because two disjoint polygons have at most four common tangents, we conclude that the number of vertices of the second type is $O(m^2+n)$. In the same paper it is shown that a simple divide-and-conquer algorithm which performs the conquer approach doing a sweep computes the set $\mathcal{A}^*$ in time $O((m^2 + n \log m) \log n)$. We summarize this discussion in the following result:

Lemma 2 *$\mathcal{A}^*$ has complexity $O(m^2+n)$ and can be computed in time $O((m^2+n \log m) \log n)$.*

Once $\mathcal{A}^*$ is computed, the decision problem reduces to check whether it has some face which correspond to a line having polygons on both sides, i. e. a face of the arrangement which has edges both above and bellow it. The optimization problem can be solved visiting all the faces of $\mathcal{A}^*$: within each face, we have to compute the width of the slabs which correspond in the dual plane to vertical segments connecting either two edges of $\mathcal{A}^*$ (type(a)-corridor) or a vertex and an edge (type(b)-corridor), as defined in Lemma 1. Clearly, this can be done in time proportional to the size of the face performing a sweep of the face. Therefore, we get the following result:

Theorem 2 *Let $\mathcal{P}$ be a set of m convex polygons with a total of n vertices. An optimal maximin line trough $\mathcal{P}$ can be found in time $O((m^2 + n \log m) \log n)$ using $O(m^2 + n)$ space.*

The approach in this section can be applied to the case when the obstacles are convex sets with constant description complexity, i. e. whose boundaries are algebraic curves with degree bounded by a certain constant. Again, we use duality and observe that the set of lines intersecting a convex set P corresponds in dual plane to the region between two convex curves, U and L, which are dual to the set of upper and lower tangents of P, respectively (see Figure 2). Therefore, the set of lines avoiding a family of convex sets $P_1, P_2, \ldots, P_m$ corresponds to a set of faces in the arrangement formed by those curves. The crucial parameter which bounds the complexity of the arrangement is the number of intersections between any two curves. If every two curves intersect in at most one point, then the complexity of the arrangement is $O(m^2)$ and can be constructed within the same asymptotic time (see [32]). Once the arrangement is constructed, the problem of computing the widest empty corridor can be easily solved. We observe that the condition for the number of intersections corresponds in primal plane to the fact that every two convex sets P_i and P_j have at most one common upper tangent, one common lower tangent and two inner tangents and is satisfied in a variety of situations, for instance, if the convex sets are pairwise disjoint or if we are dealing with a family of arbitrary disks.

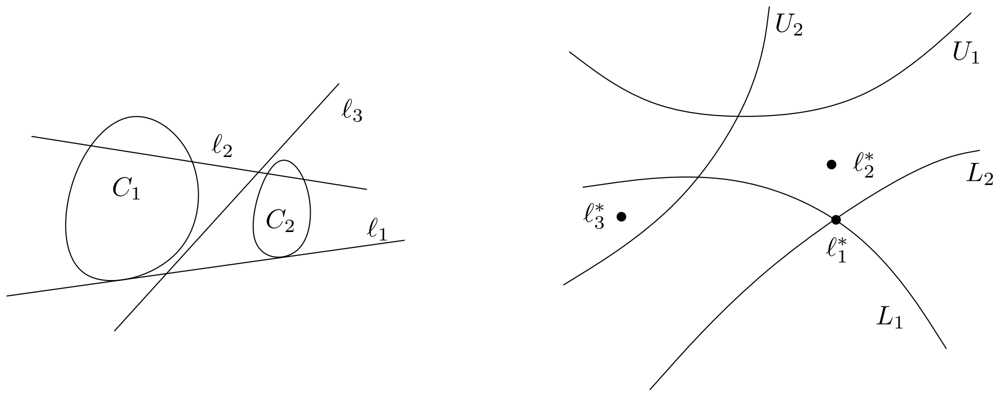


Figure 2: Duality for convex sets P_1 and P_2.

4 The weighted maximin line problem with arbitrary norms

In this section we generalize the problem both by considering distances different from the Euclidean and by adding weights to the sites. Facility location models usually consider given weights associated to the input, representing the importance of the existing facilities. Also, non-Euclidean norms to measure distances have been widely used in the literature [16, 33]. Let us start by introducing some notation that we borrow from [33].

Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be the family of convex polygonal regions and let w_k be the weight associated to the polygon P_k. Let $\mathcal{B}$ be a convex, compact, centrally symmetric set in the plane. The norm with unit ball $\mathcal{B}$ is defined by $\gamma_{\mathcal{B}}(x) := \min\{|\lambda| : x \in \lambda\mathcal{B}\}$. The induced distance between two points x and y is denoted $d_{\mathcal{B}}(x, y) := \gamma_{\mathcal{B}}(x - y)$. If A and B are two closed subsets in $\mathbb{R}^2$, then the distance between A and B is defined as $d_{\mathcal{B}}(A, B) := \min_{a\in A, b\in B} d_{\mathcal{B}}(a, b)$.

We consider both the decision and the optimization versions of the *weighted maximin line problem with arbitrary norm.* If we denote by $\mathcal{L}_P$ the set of lines in the plane intersecting $CH(\mathcal{P})$, the problems can be defined, respectively, as follows:

[Dec] Given $\delta > 0$, decide whether there exists a line $\ell \in \mathcal{L}_P$ such that $\displaystyle\min_{P_k\in\mathcal{P}} \frac{1}{w_k}\, d_{\mathcal{B}}(P_k, \ell) \geq \delta$

[Opt] Compute $\displaystyle\max_{\ell\in\mathcal{L}_P} \min_{P_k\in\mathcal{P}} \frac{1}{w_k}\, d_{\mathcal{B}}(P_k, \ell)$

Our plan is to give an efficient solution to the decision problem and then applying parametric search in order to solve the optimization problem. First we recall the concept of Minkowski sum: given two sets $A, B \subset \mathbb{R}^2$, the Minkowski sum of A and B is defined as

$$A \oplus B = \{a + b \,|\, a \in A, b \in B\},$$

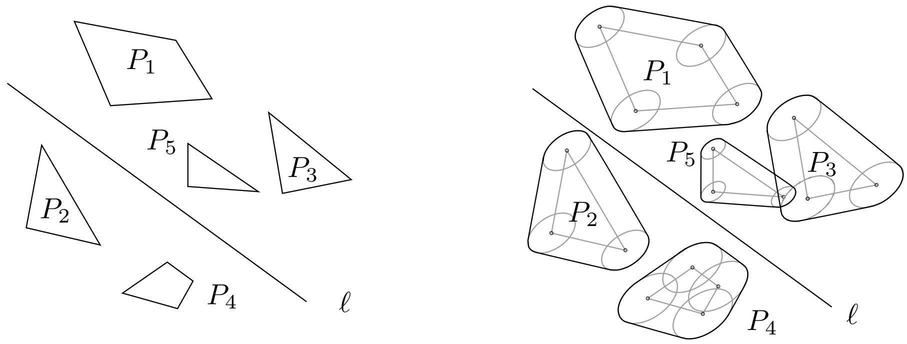


Figure 3: Decision problem interpreted via Minkowski sums.

where a and b are added up because they are interpreted as vectors once an origin has been fixed. Because

$$\frac{1}{w_k} d_{\mathcal{B}}(P_k, \ell) \geq \delta \quad \Leftrightarrow \quad \ell \cap (P_k \oplus \omega_k \delta \mathcal{B}) = \varnothing \tag{1}$$

the decision problem can be formulated in the following way: given $\delta > 0$, decide whether there exists a line $\ell \in \mathcal{L}_P$ that does not intersect the sets $P_i \oplus \omega_i \delta \mathcal{B}$, for $i = 1, \ldots, m$ (see Figure 3).

At this point we have to make some assumptions on the ball $\mathcal{B}$ that allows us to perform computations. For instance, we can assume that $\mathcal{B}$ is a convex polygon with a constant number of edges[1] or an algebraic curve with constant description complexity. In the later case, we assume that our model of computation is powerful enough to make the required operations (essentially, computing the common tangents of two homothetic copies of $\mathcal{B}$) in constant time.

We can solve the decision problem applying the same technique that was used to solve the Euclidean version of the problem. We use duality and observe that the set of lines intersecting a convex set $P_i \oplus \omega_i \delta \mathcal{B}$ corresponds in dual plane to the region between two convex curves, U_i and L_i, which are dual to the set of upper and lower tangents of $P_i \oplus \omega_i \delta \mathcal{B}$, respectively. Therefore, the set of lines which are at weighted distance at least δ from $P_1, P_2, \ldots, P_m$ corresponds to a set of faces in the arrangement formed by those curves. Using exactly the same arguments as in Lemma 2 it can be shown that the arrangement formed by the curves U_i and L_i has complexity $O(m^2 + n)$ and can be computed in time $O((m^2 + n \log m) \log n)$. Therefore, we have:

Theorem 3 *Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be a set of convex polygons with a total of n vertices and let $\omega_1, \ldots, \omega_m$ be the corresponding weights. Let $d_{\mathcal{B}}$ be the distance defined by the unit ball $\mathcal{B}$. The problem [Dec] can be solved in time $O((m^2 + n \log m) \log n)$ using $O(m^2 + n)$ space.*

[1] If the number of edges is not a constant, the same algorithm works, but the complexity is related to the number of edges of $\mathcal{B}$

4.1 The optimization problem

Parametric search is a well known technique in geometric optimization which originated in [24]. It can be used to solve optimization problems which are *monotone* with respect to a given parameter δ: if the answer to the corresponding decision problem is positive for a given δ_1, then it is also positive for every $\delta_2 < \delta_1$. Therefore, if we denote by δ^* the value of the parameter associated to the optimal solution, we know that if the answer to the decision problem is positive for δ_1 then $\delta^* \geq \delta_1$, while if it is negative, then $\delta^* < \delta_1$.

The main idea behind parametric search is trying to do binary search on the parameter δ; of course, we cannot do binary search on a real parameter, instead we try to reduce the search to a discrete set containing the value δ^* as follows: suppose that we have a sequential algorithm $\mathcal{D}$ solving the decision problem and that we run the algorithm using as input the (unknown) value δ^*. If the decisions made by the algorithm depend only on the sign of a polynomial $P(\delta)$, it becomes clear how to proceed: if $\delta_1, \ldots, \delta_k$ are the roots of $P(\delta)$, we run $\mathcal{D}$ for each of the roots and, in this way, we find that either $\delta^* < \delta_1$, $\delta^* > \delta_k$, $\delta^* \in (\delta_i, \delta_{i+1})$ or $\delta^* = \delta_i$ for some $i = 1, \ldots, k-1$. In the last case we are done, while in the rest we can proceed with $\mathcal{D}$ for δ^* because we know the sign of $P(\delta^*)$.

The approach outlined in the previous paragraph does not give a good complexity, but it can be improved if we have a parallel algorithm for the decision problem. The main idea is to use the parallel algorithm for collecting batches of roots of $P(\delta)$ that are *independent* – in the sense that we can perform one of them without knowing the result of the others – and do binary search on them. In this way, if the parallel algorithm solves the problem in T_p steps using Q processors and the sequential algorithm has complexity T_s, the final complexity of the optimization method is $O(QT_p + T_pT_s \log Q)$. The interested reader can find both theoretically oriented and applied oriented expositions of the parametric search technique in [2, 31].

Perhaps the main difficulty of the parametric search technique is that we need a parallel algorithm for the generic version of the decision problem, which is not always easy. Nevertheless, it can be observed that it is not necessary that the generic algorithm solves the problem under consideration: all we need is that the output of the generic algorithm changes combinatorially at δ^*. Actually, in quite some cases sorting can play the role of the generic algorithm and, in these cases, we can use one of the parallel sorting algorithms. It has been pointed out in [28] that Cole's algorithm presented in [6] may be specially appropriated in practice because it has good asymptotic complexity and small constants, and that quicksort can also give good results in practice. Let us see why sorting can also be used for our problem.

Let us consider $P_i^\delta = P_i \oplus \delta\omega_i\mathcal{B}$. The (curved) polygon P_i^δ is made up of arcs of copies of the unit ball $\mathcal{B}$ centered at the vertices of P_i and tangents between them. If u, v are vertices of P_i, we denote by u^δ and by v^δ, respectively, the corresponding arcs of P_i^δ. As δ increases, the two inner tangents to P_i^δ and P_j^δ rotate in opposite directions. We denote

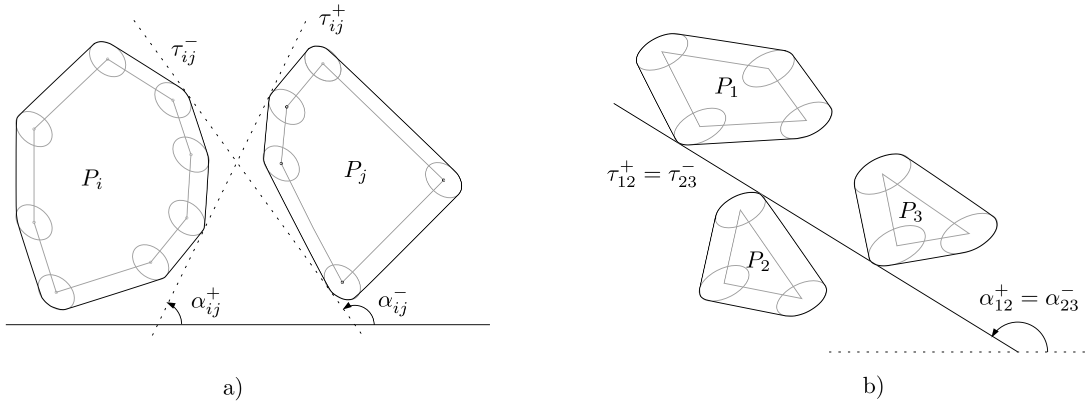


Figure 4: Reduction to sorting inner tangents.

by $\tau_{ij}^+(\delta)$ the inner tangent to P_i^δ and P_j^δ that rotates counterclockwise and by $\tau_{ij}^-(\delta)$ the inner tangent that rotates clockwise. Finally, $\alpha_{ij}^+(\delta)$ and $\alpha_{ij}^-(\delta)$ are, respectively, the angles determined by the inner tangents with the OX axis (see Figure 4.a).

The key observation is that, if we sort the $O(m^2)$ angles $\alpha_{ij}^+(\delta)$ and $\alpha_{ij}^-(\delta)$, the order changes in the candidate solutions to the maximin line problem. Actually, for candidates of type a) such that the vertices belong to polygons P_i and P_j we have that $\alpha_{ij}^+(\delta^*) = \alpha_{ij}^-(\delta^*)$ while for candidates of type b) we have that $\alpha_{ij}^+(\delta^*) = \alpha_{jk}^-(\delta^*)$ for a suitable labeling of the polygons (see Figure 4.b).

There is still one last caveat for using the generic sorting algorithm in the parametric search process. When the generic algorithm makes a comparison between $\alpha_{ij}^\pm(\delta^*)$ and $\alpha_{kl}^\pm(\delta^*)$, the result does not depend on the sign of a polynomial, but on the angle defined by the inner tangents to $P_i^{\delta^*}$ and $P_j^{\delta^*}$ and to $P_k^{\delta^*}$ and $P_l^{\delta^*}$. Therefore, in order to resolve the comparison, we have to compute the values of δ for which the angle is the same and afterwards run the sequential algorithm for some of those values of δ. In the following results we deal with this problem and give an upper bound on the number of such values.

Lemma 3 *Let $\tau_{ij}^\pm(\delta)$ be the set of inner tangents to the families of polygons P_i^δ and P_j^δ. The set $\tau_{ij}^\pm(\delta)$ has complexity $O(n_i + n_j)$ and can be computed within the same asymptotic time.*

Proof: We observe that the inner tangents to P_i and P_j, i.e. $\tau_{ij}^\pm(0)$, can be computed in time $O(\log n_i + \log n_j)$ (see [30]). Let us assume that we label the vertices of P_i and P_j counterclockwise and in such a way that $\tau_{ij}^+(0)$ is tangent to P_i and P_j at vertices u_1 and v_1, respectively (see Figure 5). Now, as δ increases, $\tau_{ij}^+(\delta)$ is tangent to two copies of the unit ball $\mathcal{B}$ centered at u_1 and v_1 and scaled according with ω_i and ω_j, the weights corresponding to the polygons. We recall that, in our model of computation, this set of tangents can be computed and described in constant time and we refer to it as an *elementary arc* of the curve $\alpha_{ij}^+(\delta)$. This elementary arc is completed either when the

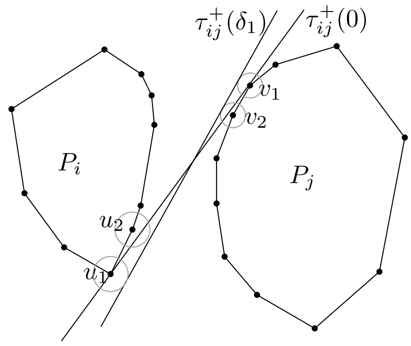


Figure 5: Computing $\tau_{ij}^{+}(\delta)$.

tangent to P_j^δ at the edge $v_1^\delta v_2^\delta$ is also tangent to P_i^δ (at u_1^δ) or when the tangent to P_i^δ at the edge $u_1^\delta u_2^\delta$ is also tangent to P_j^δ (at v_1^δ).

In this way, we can compute elementary arcs describing $\tau_{ij}^{+}(\delta)$ until we reach the value of δ for which polygons P_i^δ and P_j^δ are tangent. Clearly, during this process we always move forward on the boundary of the polygons, namely, counterclockwise for τ_{ij}^{+} and clockwise for τ_{ij}^{-} and thus the total complexity is $O(n_i + n_j)$. □

Given a point u_i with associated weight ω_i, consider $u_i^\delta = u_i \oplus \delta\omega_i\mathcal{B}$. In the following lemma we characterize the common tangents to a pair of such balls.

Lemma 4 *There exist two points m_{in} and m_{out} on the line defined by u_i and u_j such that for every $\delta > 0$ the inner and outer tangents to u_i^δ and u_j^δ pass through, respectively, m_{in} and m_{out}. Furthermore, for these points it holds*

$$\frac{d(u_i, m_{in})}{d(u_j, m_{in})} = \frac{d(u_i, m_{out})}{d(u_j, m_{out})} = \frac{\omega_i}{\omega_j}.$$

Points u_i, u_j, m_{in}, m_{out} are said to form an Harmonic system of points (see [15]).

Proof: Let m_{in} be the intersection point between an inner tangent and the line passing through u_i and u_j and let α_i and α_j the tangency points (see Figure 6). Because $\mathcal{B}$ is centrally symmetric, segments $u_i\alpha_i$ and $u_j\alpha_j$ are parallel and, therefore, triangles $u_i\alpha_i m_{in}$ and $u_j\alpha_j m_{in}$ are similar. Therefore,

$$\frac{d(\alpha_i, m_{in})}{d(\alpha_j, m_{in})} = \frac{d(\alpha_i, u_i)}{d(\alpha_j, u_j)} = \frac{\delta\omega_i}{\delta\omega_j}$$

For the outer tangents and m_{out} the argument is completely similar. □

It is worth noticing that point m_{in} is also known as the weighted midpoint of u_i and u_j.

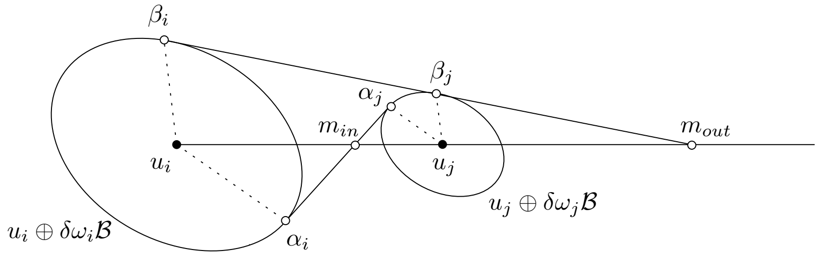


Figure 6: Illustration for the proof of Lemma 4.

We need to study the set of solutions of the equation $\alpha^{\pm}_{ij}(\delta) = \alpha^{\pm}_{kl}(\delta)$. Clearly, $\alpha^{+}_{ij}(\delta)$ and $\alpha^{-}_{kl}(\delta)$ intersect in at most one point, because the former one is an increasing function while the later is a decreasing function, and the same is true for $\alpha^{-}_{ij}(\delta)$ and $\alpha^{+}_{kl}(\delta)$. In the next result we study the set of solutions of the equation $\alpha^{+}_{ij}(\delta) = \alpha^{+}_{kl}(\delta)$ (the situation for the clockwise rotating tangents is identical).

Lemma 5 *Let $\alpha^{+}_{ij}(\delta)$ and $\alpha^{+}_{kl}(\delta)$ be elementary arcs describing the counterclockwise rotating inner tangents to u^{δ}_i and u^{δ}_j and to u^{δ}_k and u^{δ}_l, respectively. Let us assume that at least one of the indices k, l is different from i and j. If the equation $\alpha^{+}_{ij}(\delta) = \alpha^{+}_{kl}(\delta)$ has more than one solution, then the curves $\alpha^{+}_{ij}(\delta)$ and $\alpha^{+}_{kl}(\delta)$ are equal whenever both are defined.*

Proof: Without loss of generality, we can assume that k is different from i and j. Furthermore, according to Lemma 4, all common inner tangents to u^{δ}_i and u^{δ}_j and to u^{δ}_k and u^{δ}_l pass through, respectively, points that we denote by m_{ij} and m_{kl}.

Let us assume that α^{+}_{ij} and α^{+}_{kl} intersect twice. In Figure 7 the angles for which the common tangents are parallel are 0 and γ. Let α_h,β_h, for $h = 1, 2$, be the tangency points with balls u^{δ}_i and u^{δ}_k (see Figure 7). We observe that the triangles $\alpha_1\alpha_2 u_i$ and $\beta_1\beta_2 u_k$ are similar and, therefore, the segments $\alpha_1\alpha_2$ and $\beta_1\beta_2$ are parallel and such that $\frac{d(\alpha_1,\alpha_2)}{d(\beta_1,\beta_2)} = \frac{\omega_i}{\omega_k}$. Because the triangles $\alpha_1\alpha_2 m_{ij}$ and $\beta_1\beta_2 m_{kl}$ are also similar, it follows that $\frac{d(\alpha_1,m_{ij})}{d(\beta_1,m_{kl})} = \frac{\omega_i}{\omega_k}$ and, therefore, triangles $u_i\alpha_1 m_{ij}$ and $u_k\beta_1 m_{kl}$ are similar too. But then the edges $u_i m_{ij}$ and $u_k m_{kl}$ are parallel and their lengths are in the proportion $\frac{\omega_i}{\omega_k}$, which implies that curves $\alpha^{+}_{ij}(\delta)$ and $\alpha^{+}_{kl}(\delta)$ are equal whenever both are defined. □

Now we can give a bound on the number of solutions in the overall process which is crucial for the performance of the parametric search technique.

Lemma 6 *The total number of solutions of $O(m^2)$ equations of the form $\alpha^{\pm}_{ij}(\delta) = \alpha^{\pm}_{kl}(\delta)$ is $O(nm^2)$ and can be computed within the same asymptotic time.*

Proof: From Lemma 3 it follows that $\alpha^{\pm}_{ij}(\delta)$ has complexity $O(n_i + n_j)$ and can be computed within the same asymptotic time. Furthermore, from Lemma 5 we know that the

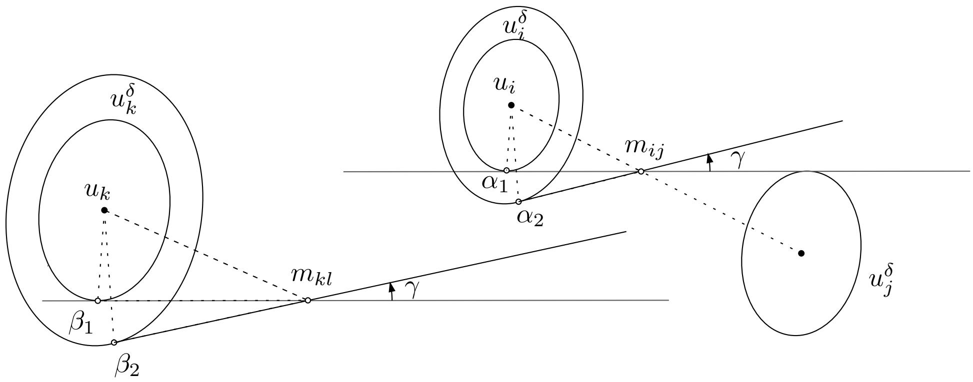


Figure 7: Illustration for the proof of Lemma 5

equations $\alpha_{ij}^{\pm}(\delta) = \alpha_{kl}^{\pm}(\delta)$ have at most $O(n_i + n_j + n_k + n_l) = O(n)$ solutions. Clearly, such solutions can be computed in the same asymptotic time using an standard sweep approach. □

We are now ready to describe the final procedure: we order the $O(m^2)$ inner tangents in $O(\log m)$ steps, and perform $O(m^2)$ comparisons in each step. Lemma 6 guarantees that the total number of solutions to the equations $\alpha_{ij}^{\pm}(\delta) = \alpha_{kl}^{\pm}(\delta)$ performed in each step is $O(nm^2)$ and, therefore, the complexity of the algorithm is $O((nm^2 + T_s \log(nm^2)) \log m)$, where $T_s = O((m^2 + n \log m) \log n)$. Putting all this together, we have:

Theorem 4 *Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be a set of convex polygons with a total of n vertices and let ω_i be the corresponding weights. Let $d_{\mathcal{B}}$ be the distance defined by the unit ball $\mathcal{B}$. The problem [Opt] can be solved in time $O((n \log m(m^2 + \log^2 n))$ using $O(m^2 + n)$ space.*

It may be argued that the algorithm is too involved and that the complexity is too high in order to be useful in practice. Nevertheless, we observe that the main contribution to the complexity bound comes from Lemma 6 and the fact that we do not make any assumptions on the relative size of the polygons. The complexity bound is much better in a variety of particular cases. For instance, if we take m as a constant, i.e., if we have a constant number of polygons, then we get an algorithm which complexity is close to linear. Perhaps more importantly, if we have a set of polygons with similar size we get the following:

Corollary 1 *Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be a set of convex polygons with a total of n vertices, let n_i be the number of vertices of P_i and assume that $m = \Theta(\sqrt{n})$ and $n_i = O(\sqrt{n})$ for $i = 1, \ldots, m$. Then, the problem [Opt] can be solved in time $O(n^{3/2} \log n)$ using $O(n)$ space.*

Proof: It is enough to observe that if $m = \Theta(\sqrt{n})$ and $n_i = O(\sqrt{n})$ for $i = 1, \ldots, m$, then the bound of Lemma 6 is $O(n^{3/2})$. □

5 The three-dimensional scene

In this section we deal with the problem of computing the plane which maximizes the minimum distance to a set of polyhedra in $\mathbb{R}^3$.

Given a set $\mathcal{P} = \{P_1, ..., P_m\}$ of m polyhedra in $\mathbb{R}^3$ with a total of n vertices, we want to find a plane π such that:

- $\pi \cap CH(\mathcal{P}) \neq \varnothing$.
- $\min_i d(P_i, \pi)$ is maximum.

This problem is named as *the obnoxious plane problem* in [9], where it is solved in $O(n^3)$ time and $O(n^2)$ space. The problem is equivalent to finding an empty region bounded by two parallel planes as wide as possible and defining a nontrivial partition in the set of polyhedra, that we refer as *the widest empty slab problem.* Let S be the set of vertices of the objects in $\mathcal{P}$.

We state a necessary condition for slab optimality in arbitrary dimension. We say that a hyperplane π strictly separates two sets of points if each of the sets is contained in one of the open halfspaces defined by π.

Theorem 5 *Let π^* be a solution to an instance of the obnoxious hyperplane problem and let π_1 and π_2 be the bounding hyperplanes of the slab generated by π^*. Then, the sets $S_1 = S \cap \pi_1$ and $S_2 = S \cap \pi_2$ cannot be strictly separated by a hyperplane orthogonal to π^*.*

Proof: Let us assume that $S_1 = \{p_i\}_{i \in I}$ and $S_2 = \{q_j\}_{j \in J}$. We denote by $\vec{n}$ the unitary vector orthogonal to π^*. Then, the distance between the parallel planes π_1 and π_2 is $\Delta = |\vec{n} \cdot \vec{p_i q_j}|$. First, observe that if S_1 and S_2 can be strictly separated by a hyperplane h orthogonal to π^*, then the unitary vector normal to h, denoted by $\vec{v}$, can be chosen such that

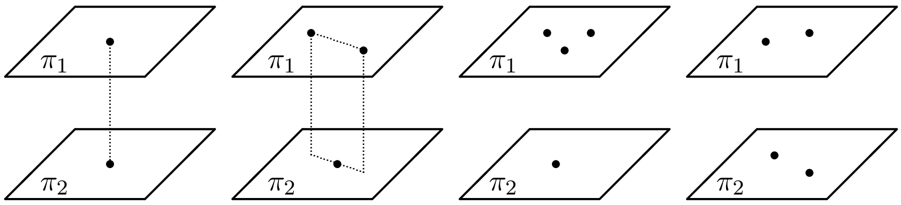
$$\min_{p_i \in S_1, q_j \in S_2} \vec{v} \cdot \vec{p_i q_j} = k > 0.$$

Now we consider an empty slab orthogonal to $\vec{n}_\varepsilon = \vec{n} + \varepsilon \vec{v}$. The width of the slab is

$$\Delta(\varepsilon) = \min_{p_i \in S_1, q_j \in S_2} \frac{\vec{n}_\varepsilon \cdot \vec{p_i q_j}}{\|\vec{n}_\varepsilon\|} = \min_{p_i \in S_1, q_j \in S_2} \frac{\Delta + \varepsilon \vec{v} \cdot \vec{p_i q_j}}{1 + \varepsilon^2} = \frac{\Delta + k\varepsilon}{1 + \varepsilon^2}.$$

Because $\Delta(0) = \Delta$ and $\Delta'(0) = k > 0$, we can guarantee that $\Delta(\varepsilon) > \Delta$ for $\varepsilon > 0$ small enough. □

As a consequence of the preceding Theorem we can restrict our search to slabs C that satisfy one of the four following conditions (see Figure 8) as in [9].



(a) C_{11} (b) C_{21} (c) C_{31} (d) C_{22}

Figure 8: Types of candidate slabs according to Theorem 5.

(a) Each of π_1 and π_2 contains exactly one point of S, p_1 and p_2 respectively, such that $p_2 - p_1$ is orthogonal to π^*.

(b) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 2$, $k \geq 1$ and $S_1 \cup S_2$ lie on a common plane τ that is orthogonal to π^*.

(c) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 3$, $k \geq 1$, S_1 are not collinear, and $S_1 \cup S_2$ are not coplanar.

(d) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 2$, $k \geq 2$, S_1 are collinear, S_2 are collinear, and $S_1 \cup S_2$ are not coplanar.

Following the same approach as in the two dimensional case, we solve the problem by exploring efficiently only the set of planes avoiding $\mathcal{P}$. We first give a bound for the combinatorial complexity of this set and show how it can be computed.

Let $\mathcal{D}$ be the transformation which maps a point $p = (a, b, c)$ to the plane $\mathcal{D}(p) : z = ax + by - c$ in the dual space, and maps a non-vertical plane $\pi : z = mx + ny - d$ to the point $\mathcal{D}(\pi) = (m, n, d)$ in the dual space. Given a polyhedron P_i, we denote by A_i the set of planes avoiding P_i and by A_i^* the set of points dual to planes in A_i.

Theorem 6 *Let $\mathcal{P} = \{P_1, ..., P_m\}$ be a set of m polyhedra with a total of n vertices. The set of planes avoiding $\mathcal{P}$ has complexity $O(m^2 n)$ and can be computed in time $O(m^2 n \log(m^2 n))$.*

Proof: We want to argue that the number of vertices of $\mathcal{A} = \cap_{i=1}^{m} A_i^*$ is $O(m^2 n)$. The vertices of $\mathcal{A}$ correspond to planes passing through three vertices of $\mathcal{P}$ and can be classified into three different types:

1. planes containing three vertices of a polyhedron (therefore, containing a face of its convex hull),

2. planes passing through an edge of a polyhedron and a vertex of another polyhedron,

3. planes passing through three vertices of three different polyhedra.

The number of vertices of the first type is clearly $O(n)$ and, for the second type, we observe that there are at most two planes tangent to a given edge and a given polyhedron and, therefore, the number of those vertices is $O(mn)$. Finally, for the third type of vertices, we claim that the number of planes tangent to polyhedra P_i, P_j and P_k, with n_i, n_j and n_k vertices respectively, is $O(n_i + n_j + n_k)$. In order to prove the claim, observe that A_i^* is the union of two unbounded convex polyhedra. Therefore, $A_i^* \cap A_j^* \cap A_k^*$ can be described as the union of at most eight disjoint sets, each of which is the intersection of three convex polyhedra and has complexity $O(n_i + n_j + n_k)$. The first part of the proof is finished because

$$\sum_{i \neq j \neq k} O(n_i + n_j + n_k) = O(m^2 n)$$

In order to compute $\mathcal{A}$ we use a divide and conquer approach. Assume that we partition $\mathcal{P}$ into two sets of $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$ polyhedra, denoted $\mathcal{R}$ and $\mathcal{B}$, with n vertices in total. Let $\mathcal{A}_r$ and $\mathcal{A}_b$ denote, respectively, the sets of points dual to planes avoiding $\mathcal{R}$ and $\mathcal{B}$, respectively. The crux of the method is observing that the merge step reduces to computing the intersection of $\mathcal{A}_r = \cup_{i=1}^{k} R_i$ and $\mathcal{A}_b = \cup_{j=1}^{l} B_j$, where R_i and B_j are convex polyhedra and the total complexity of $\mathcal{A}_r$ and $\mathcal{A}_b$ is $O(m^2 n)$.

We compute the intersection of $\mathcal{R}$ and $\mathcal{B}$ using a space sweeping approach. It is clear that the intersection can be easily computed if we are able to maintain the planar subdivision generated in the sweep plane by one of the sets, say $\mathcal{R}$, and perform point location in such subdivision when we encounter a new vertex of $\mathcal{B}$. These operations can be done efficiently by using the dynamic point location structure of Goodrich and Tamassia [14] which can manage monotone subdivisions (in our case, the subdivision is convex) and takes $O(\log n)$ per update and $O(\log^2 n)$ per point location query, where n is the total size of the subdivision. Because the total size of $\mathcal{R}$ and $\mathcal{B}$ is $O(m^2 n)$, it follows that the intersection can be computed in time $O(m^2 n \log(m^2 n))$. Therefore, if we denote by $T(m, n)$ the time required by the whole algorithm, we obtain the recursive formula

$$T(m, n) = T(m/2, n_1) + T(m/2, n - n_1) + O(m^2 n \log(m^2 n))$$

which solves to $T(m, n) = O(m^2 n \log(m^2 n))$. □

We now describe how to use the arrangement $\mathcal{A} = \cap_{i=1}^{m} A_i^*$ in order to solve the problem. As shown is the two dimensional case, the idea is to solve the optimization problem visiting all cells c in $\mathcal{A}$ and identifying the candidate slabs associated with c. By using the properties of the duality transform we look at open vertical segments whose endpoints lie on the boundary of each cell. We have to examine all the vertical segments inside a cell that correspond with candidates of type C_{11}, C_{21}, C_{31}, C_{22} as illustrated in Figure 8.

When leaving a cell c, we test every face-face, edge-face, vertex-face and edge-edge pair of c in order to identify and compute the width of all pairs that are vertically aligned, i.e., the widths of the candidate slabs in the primal space.

A detailed description of the detection of candidates within a cell in a three-dimensional arrangement is given in [9]. In each cell, each candidate can be processed in $O(1)$ amortized time. At this point, we should note that candidates type C_{11} and C_{21} differ from candidates C_{31} and C_{22}. In fact, the number of vertical segments associated with a face-face or edge-face pair is not finite. However, the orthogonality condition of the former can be used to identify those types of candidates in amortized $O(1)$ time per cell.

As a consequence of the above description, the overall time we need to obtain all the candidates in our arrangement and compute the optimal one is proportional to the size of the arrangement, and we have established the following result:

Theorem 7 *An obnoxious plane through a set of m polyhedral objects in $\mathbb{R}^3$ with a total of n vertices can be computed in $O(m^2 n \log(m^2 n))$ time and $O(m^2 n)$ space.*

Acknowledgments

The authors wish to thank Boris Aronov for his help in the proof of Theorem 6.

References

[1] P.K. Agarwal, and M. Sharir. Ray shooting Amidst Polygons in 2D, *Journal of Algorithms*, 21, (1996), 508–519.

[2] P.K. Agarwal, and M. Sharir. Efficient algorithms for geometric optimization, *ACM Computing Surveys*, 30, (1998), 412–458.

[3] J. Brimbeg , H. Juel, A. Schöbel. Linear facility location in three dimensions - models and solution methods. Operations Research, 50(6) (2002) 1050-1057.

[4] J. Brimberg, and G. O. Wesolowsky. Locating facilities by minimax relative to closest points of demand areas. *Computers & Operations Research*, 29:625–636, 2002.

[5] E. Carrizosa, M. Muñoz-Márquez, and J. Puerto. The Weber problem with regional demand. *European Journal of Operational Research*, 104(2):358–365, 1998.

[6] R. Cole. Parallel merge sort. *SIAM J. on Computing* 17: 770–785, 1988.

[7] J.M. Díaz-Báñez, F. Gómez, G. Toussaint. Computing shortest paths for transportation of hazardous materials in continuous spaces. *Journal of Food Engineering*, 70, 2005, pp. 293–298.

[8] J. M. Díaz-Báñez, F. Hurtado, Computing obnoxious 1-corner polygonal chains, Computers & Operations Research, 33, 2005.

[9] J. M. Díaz-Báñez, M. López, and J. A. Sellarès. Locating an obnoxious plane European Journal of Operational Research, in press, 2005.

[10] J. M. Díaz-Báñez, J. A. Mesa, and A. Shöbel. Continuous location of dimensional structures, *European Journal of Operational Research*, 152, 2004, 22-44.

[11] Z. Drezner. *Facility Location: A Survey of Applications and Methods.* Editor Zvi Drezner, Springer, 1995.

[12] Z. Drezner, and G.O. Wesolowsky. Location of an obnoxious route, *J. Operational Research Society*, 40, 1989. 1011–1018.

[13] H. Edelsbrunner, and L. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences* **38** 165-194, 1989.

[14] M. T. Goodrich, and R. Tamassia. Dynamic trees and dynamic point location. *SIAM Journal on Computing* **28** 612-636, 1998.

[15] G. H. Hardy. A Course of Pure Mathematics. Cambridge University Press, pp. 99–106, 1967.

[16] H. W. Hamacher, and S. Nickel. Multicriteria planar location problems. *European Journal Of Operational Research* **94** 66-86, 1996.

[17] Y. Hinojosa. Some problems in Location Theory. Ph.D. Thesis, University of Seville, 2000.

[18] Y. Hinojosa, and J. Puerto. The Polyhedral Norm Approach to the Problem of Locating Obnoxious Routes, *Studies in Locational Analysis*, 12, (1999), 49–65.

[19] 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*, pp 210-213. TR SOCS-88.11, dept of Computer Science, McGill University, Montreal, Canada, 1988.

[20] M. Houle, and G. T. Toussaint. Computing the width of a set, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, (1988), 760–765.

[21] J. Janardan, and F. P. Preparata. "Widest-corridor problems", *Nordic Journal of Computing*, 1, pp 231-245, 1994.

[22] N. M. Korneenko, and H. Martini. Hyperplane Approximation and Related Topics. In *New Trends in Discrete and Computational Geometry.* János Pach Ed., chapter 6, pages 135–162. Springer-Verlag, New York, 1993.

[23] D. T. Lee, and Y. F. Wu. Geometric Complexity of Some Location Problems. *Algorithmica*, 1, (1986), 193–211.

[24] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30, pp. 852-865, 1983.

[25] J. G. Morris, and J. P. Norback. A simple approach to linear facility location. *Transportation Science*, vol 14, 1, 1980, 1–8.

[26] J. G. Morris, and J. P. Norback. Linear Facility Location - Solving Extensions on the Basic Problems. *European Journal of Operational Research*, 12, 1983, 90–94.

[27] S. Nickel, J. Puerto, and A.M. Rodriguez-Chía. An approach to Location Models Involving Sets as Existing Facilities *Mathematics of Operational Research*, 28, 2003, 693–715.

[28] R. v. Oostrum, and R. C. Veltkamp. Parametric search made practical, *Computational Geometry*, 28, pp. 75-88, 2004.

[29] J. M. Robert, and G. T. Toussaint. Computational Geometry and facility location. Technical Report SOCS 90.20, McGill Univ., Montreal, PQ, 1990.

[30] H. Rohnert. Shortest paths in the plane with convex polygonal obstacles. *Inf. Processing Letters* (23), pp. 71-76, 1986.

[31] J. Salowe Parametric search. In *Handbook of Discrete and Computational Geometry.* J. E. Goodman and J. O'Rourke (Eds). pp. 683-698, 1997.

[32] M. Sharir, and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*, Cambridge-University Press, 1995.

[33] A. Shöbel. Locating least-distant lines in the plane. *European Journal Of Operational Research*, 106, (1998), 152-159.