Locating an obnoxious plane

J.M. Díaz-Báñez * *M.A. López* † *J.A. Sellarès* ‡

22nd July 2004

Abstract

Let S be a set of n points in three-dimensional Euclidean space. We consider the problem of positioning a plane π intersecting the convex hull of S such that $\min\{d(\pi, p); p \in S\}$ is maximized. In a geometric setting, the problem asks for the widest empty slab through n points in space, where a slab is the open region of $\mathbb{R}^3$ that is bounded by two parallel planes that intersect the convex hull of S. We give a characterization of the planes which are locally optimal and we show that the problem can be solved in $O(n^3)$ time and $O(n^2)$ space. We also consider several variants of the problem which include constraining the obnoxious plane to contain a given line or point and computing the widest empty slab for polyhedral obstacles. Finally, we show how to adapt our method for computing a largest empty annulus in the plane, improving the known time bound $O(n^3 \log n)$ [8].

Keywords: Location; Computational geometry; Maximin; Duality.

1 Introduction

Location science is a classical field of operations research that has also been considered in the computational geometry community. A class of problems from this field, often referred to as maximin facility location, deals with the placement of undesirable or obnoxious facilities. In these problems the objective is to maximize the minimal distance between the facility and a set of input points. Furthermore, in order to ensure that the problems are well-defined the facility is normally constrained to go through some sort of bounding region, such as the convex hull or bounding box of the input points. Applications of these problems go well beyond the field of location science. For instance, splitting the space using cuts that avoid

*Corresponding author: Departamento. de Matemática Aplicada II, Escuela Universitaria Politécnica, Virgen de África 7, 41011, Sevilla, SPAIN (dbanez@us.es).

†(University of Denver, USA (mlopez@cs.du.edu).

‡Universitat de Girona, SPAIN (sellares@ima.udg.es).

the input points is useful in areas like cluster analysis, robot motion-planning and computer graphics.

Maximin facility location problems have recently been considered in computational geometry. Maximin criteria have been investigated in 2-d for the optimal positioning of of points [22, 4], lines [13], anchored lines [12], and circumferences [8]. When the facility is a line, the problem is equivalent to that of computing a *widest empty corridor*, i.e., a largest empty open space bounded by two parallel lines. Variants of the problem have also been considered and include corridors containing k input points [16, 20, 6], dynamic updates [16, 20] and L-shaped corridors [7]. Most of the results to date are two-dimensional and, with a few exceptions (e.g., [12]), little progress has been reported in three dimensions. In a recent work [14], a bichromatic separating problem has been solved in three dimensions.

The most classical versions of facility location problems consider the positioning of one or several point-like facilities. Nowadays, there is a growing body of research on the location of non-point facilities. For example, line location problems have been extensively studied both in the plane [21] and in the three dimensional space [5]. See [9] for a recent survey on the current state-of-art of these problems. In this paper, we deal with the maximin location of a plane in 3-d. We formulate the *obnoxious plane problem*, **OPP**, as follows: Given a set S of n points in $\mathbb{R}^3$, find a plane π intersecting the convex hull of S which maximizes the minimum Euclidean distance to the points.

Notice that, in 2-d, our problem reduces to that of computing the *widest empty corridor* through a set of points in the plane. This problem has been solved in $O(n^2)$ time and $O(n)$ space [13]. We extend the definition of corridor through a point set from $\mathbb{R}^2$ to $\mathbb{R}^3$ as follows: a *slab* through S is the open region of $\mathbb{R}^3$ that is bounded by two parallel planes that intersect the convex hull of S. The *width* of the slab is the distance between the bounding planes. Thus, we are interested in finding *the widest empty slab*.

It is natural to consider a "dual" version of our problem, where the goal is to minimize the maximal distance between the plane and a set of input points. This minimax location problem was solved in [15] for 3-d using techniques different from our own. Even for 2-d, the approaches used to solve the minimax [17] and the maximin [13] versions are very different.

The rest of the paper is organized as follows. In Section 2, we present some notation and preliminary results. In Section 3, we describe an algorithm to compute an obnoxious plane in $O(n^3)$ time and $O(n^2)$ space. Other variants, obtained by constraining the optimal plane π to go through a given line or given point, are described in Section 4, and solved in $O(n \log n)$ and $O(n^{2+\varepsilon})$ time, respectively. In Section 5, we compute the widest empty slab through a set of polyhedral obstacles within the same bounds as the **OPP**. Section 6 presents a reduction of the largest empty annulus problem to our problem. Finally, Section 7 contains some concluding remarks and open problems.

2 Characterization of candidate planes

In this section we describe a simple formula to compute the width of a slab and derive necessary conditions for slab optimality.

Observation 1. Let π and σ be two distinct parallel planes with (common) unit normal $\overrightarrow{n}$. Let p and q be arbitrary points on π and σ, respectively. Then, $\text{dist}(\pi, \sigma) = |\overrightarrow{n} \cdot (q-p)|$.

The following lemma characterizes candidate solutions for the **OPP**,

Lemma 1 *Let π^* be a solution to an instance of* **OPP** *and let π_1 and π_2 be the bounding planes of the slab generated by π^*. Then, exactly one of the following conditions must hold:*

(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^*.

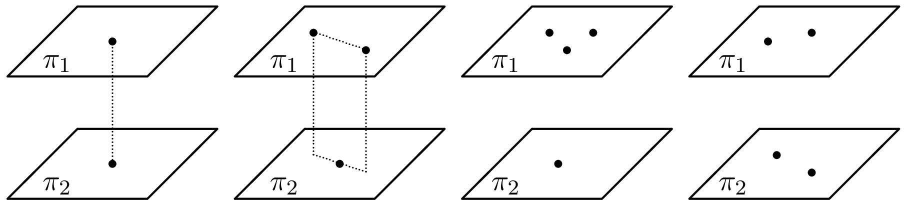
(c) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 3$, $k \geq 1$, S_1 are not collinear, and $S_1 \cup S_2$ are not coplanar.

(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.

Proof: We begin with the obvious observation that both π_1 and π_2 must contain at least one point of S as, otherwise, $\text{dist}(\pi_1, \pi_2)$ can be increased. In the sequel, let $\overrightarrow{n}$ be a unit normal to π^* (hence, also normal to π_1 and π_2) chosen so that $(q-p) \cdot \overrightarrow{n} > 0$ for any points q on π_2 and p on π_1. Conceptually, we find π_2 (resp. π_1), by translating a copy of π^* in direction $\overrightarrow{n}$ (resp. $-\overrightarrow{n}$), parallel to itself, until at least one point of S is encountered. The cases described in the lemma exhaustively cover all possibilities for the number of points encountered when performing this translation.

First, consider case (a). Suppose π^* is not orthogonal to $p_2 - p_1$. Then π_1 and π_2 can be rotated simultaneously around p_1 and p_2, respectively, so as to decrease the angle between $\overrightarrow{n}$ and $p_2 - p_1$, while keeping the slab empty. This, in turn, increases $\overrightarrow{n} \cdot (p_2 - p_1) = \text{dist}(\pi_1, \pi_2)$, contradicting the optimality of π^*.

Consider now case (b) and assume that the plane τ through p_{11}, p_{12} and p_{21} is not orthogonal to π^*, so that the angle ϕ between $\overrightarrow{n}$ and τ is strictly positive. We show that a small rotation of π_1 around the line $\overline{p_{11}p_{12}}$ (and a simultaneous rotation of π_2 around p_{12} that keeps the two planes parallel) can be performed so as to decrease the angle between $\overrightarrow{n}$ and $p_{21} - p_{11}$ while keeping the slab empty. In order to assess the effect of the rotation let $\overrightarrow{u}$ denote a unit normal to the rotated slab. Furthermore, let $\overrightarrow{m}$ be a unit normal to τ chosen so that $\overrightarrow{m} \cdot \overrightarrow{n} > 0$. (Note that $\overrightarrow{m} \cdot \overrightarrow{n} \neq 0$, as $\phi > 0$.) Let $\overrightarrow{u} = \overrightarrow{n} - (\alpha \overrightarrow{n} \cdot \overrightarrow{m})\overrightarrow{m}$ such that $0 < \alpha \leq 1$ and the



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

Figure 1: Types of candidate slabs according to Lemma 1.

slab π' with bounding planes π'_1 and π'_2 and unit normal $\overrightarrow{u}/|\overrightarrow{u}|$ is empty. First, we observe that $0 < |\overrightarrow{u}| < 1$. This follows from the fact that $|\overrightarrow{u}|^2 = \overrightarrow{u} \cdot \overrightarrow{u} = 1 - \alpha(2-\alpha)(\overrightarrow{m} \cdot \overrightarrow{n})^2$, $0 < \alpha(2-\alpha) \leq 1$ and $0 < \overrightarrow{m} \cdot \overrightarrow{n} = \cos\phi < 1$. Then,

$$\begin{aligned}
\mathrm{dist}(\pi'_1, \pi'_2) &= (p_{21} - p_{11}) \cdot \overrightarrow{u}/|\overrightarrow{u}| \\
&= (p_{21} - p_{11}) \cdot (\overrightarrow{n} - (\alpha \overrightarrow{n} \cdot \overrightarrow{m})\overrightarrow{m})/|\overrightarrow{u}| \\
&= (p_{21} - p_{11}) \cdot \overrightarrow{n}/|\overrightarrow{u}| - (\alpha \overrightarrow{n} \cdot \overrightarrow{m})(p_{21} - p_{11}) \cdot \overrightarrow{m}/|\overrightarrow{u}| \\
&= (p_{21} - p_{11}) \cdot \overrightarrow{n}/|\overrightarrow{u}| > (p_{21} - p_{11}) \cdot \overrightarrow{n} = \mathrm{dist}(\pi_1, \pi_2),
\end{aligned}$$

contradicting the optimality of π^*.

The remaining cases correspond to input sets where the points of $S_1 \cup S_2$ are not coplanar, and logically cover all possibilities not yet covered by cases (a) or (b). For algorithmic purposes it is useful to distinguish between inputs where the points of S_1 are collinear and those where they are not. Whenever (c) or (d) applies there is no rotation of the slab planes that preserves point incidences, so no additional information on the orientation of π_1 and π_2 can be derived.

It is not difficult to construct instances where each of the four cases occurs (see Figure 1). This shows that all cases are necessary and completes the proof. $\square$

As a consequence of the preceding lemma we can restrict our search to slabs C that satisfy one of the four conditions. We will denote by C_{11}, C_{21}, C_{31}, C_{22} the set of candidate slabs that satisfy the conditions of cases (a), (b), (c) and (d), respectively. Representatives from each set are shown in Figure 1.

Lemma 1 guarantees an $O(n^4)$ upper bound on the number of candidates planes for all the types of optimal slabs. Nevertheless, it is possible to substantially reduce the number of candidates by using a dual transformation.

3 Computing the candidates

The optimal slab in C_{11} can be solved separately in $O(n^3)$ time by brute force. We describe how to compute optimal slabs in C_{21}, C_{31} and C_{22}.

Our approach is based on topological sweeps over the arrangement of planes corresponding to a dual representation of the points in S. We need to reinterpret the conditions (b), (c), and (d) of Lemma 1 in the dual space in order to find the solution using the arrangement. We use the transformation $\mathcal{D}$ 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.

The following result is easy to prove and describes some useful properties of the duality transform.

Lemma 2 *Let $\mathcal{D}$ be the mapping defined above. The following properties hold:*

1. *$\mathcal{D}$ is its own inverse, i.e., $\mathcal{D}(\mathcal{D}(x)) = x$ where x is either a point or a plane.*
2. *$\mathcal{D}$ is a one-to-one correspondence between all non-vertical planes and all points in $\mathbb{R}^2$.*
3. *$\mathcal{D}$ preserves point-plane incidence.*
4. *Point p lies above plane π iff point $\mathcal{D}(\pi)$ lies above plane $\mathcal{D}(p)$.*

Since the dual transformation cannot handle vertical planes, we first discuss how to solve the special case in which the optimal plane is vertical.

Lemma 3 *The optimal vertical obnoxious plane can be computed in $O(n^2)$ time and $O(n)$ space.*

Proof: For a vertical plane π let $\ell(\pi)$ denote its intersection with the plane $z = 0$. Also, For a point $p \in S$, let p^* denote its orthogonal projection onto the plane $z = 0$. Note that, for any parallel vertical planes π_1 and π_2, $\text{dist}(\pi_1, \pi_2) = \text{dist}(\ell(\pi_1), \ell(\pi_2))$. Furthermore, a point $p \in \pi_i$ iff $p^* \in \ell(\pi)$. These facts, allow us to reduce the case of vertical slabs to the widest empty slab in 2-d. We build a set $S^* = \{p^*, p \in S\}$ and apply the algorithm of [13] to S^*. This algorithm runs in $O(n^2)$ time and $O(n)$ space. The equation of the optimal line for S^*, when interpreted in 3-d, is precisely the equation of the optimal vertical plane. $\square$

As a consequence of this lemma, we can restrict our attention to non-vertical slabs. Moreover, we assume that the points in S are in general position. In other words, we assume that, in dual space, every two planes intersect in a line, every three meet in a single point, and no four planes have a point in common.

Let H denote the set of planes $\{\pi_p = \mathcal{D}(p), p \in S\}$, and $\mathcal{A}(H)$, the arrangement of $\mathbb{R}^3$ induced by H. The properties of the duality transform can be used to characterize in $\mathcal{A}(H)$ the sets of slabs C_{21}, C_{31}, and C_{22}.

Let C be a slab with bounding planes π' and π''. The width of C can be computed using Observation 1. Since π' and π'' are parallel, the points $\mathcal{D}(\pi')$ and $\mathcal{D}(\pi'')$ lie on a vertical line in the dual space. Thus the slab C is represented in the dual space by the vertical segment $\mathcal{D}(C)$ with endpoints $\mathcal{D}(\pi')$ and $\mathcal{D}(\pi'')$. In fact, an empty slab in C_{31} corresponds to a vertical segment inside a cell of $\mathcal{A}(H)$ that connects a vertex and a face of that cell. Similarly, the empty slabs of C_{21} and C_{22} correspond to vertical segments inside cells of $\mathcal{A}(H)$ that connect an edge with a face, and an edge with an edge, respectively. By systematically examining these vertical segments we can report the overall widest empty slab. We now explain how to do this.

3.1 Finding the solution in the arrangement

In this section we describe a simple method, based on topological sweep in 3-d, to compute the optimal non-vertical slab in $O(n^3)$ time and $O(n^2)$ space. The idea is to sweep over $\mathcal{A}(H)$ while at any given time only storing a portion of it.

In dual space, for each cell of $\mathcal{A}(H)$ we examine all of the vertical segments that connect a vertex with a face, all of those that connect two edges and, by taking advantage of the orthogonality condition of Lemma 1(b), a selected subset of those that connect an edge with a face. To this end, we adapt the topological sweep algorithm of [2]. This algorithm requires $O(n^3)$ time and $O(n^2)$ working space when the planes of H are in general position.

We briefly review the mechanics of the topological sweep. The approach followed by [2] generalizes to 3-d the method proposed in [11] for sweeping an arrangements of lines in 2-d. Since $\mathcal{A}(H)$ may contain $\Theta(n^3)$ vertices and $\Theta(n^2)$ lines, the 3-d algorithm is optimal with respect to time complexity.

The idea is to sweep with a continuous unbounded surface that shares a point with each of the $O(n^2)$ lines of $\mathcal{A}(H)$. The *cut* is defined to be the set of segments or rays of $\mathcal{A}(H)$ intersected by the sweeping surface. Initially the surface is a plane perpendicular to the x-axis, and positioned to the left of the leftmost vertex of $\mathcal{A}(H)$. The sweep surface then advances from vertex to vertex. The transition of the surface from the left of a vertex to its right is called a *3-d elementary step*. Such a step consists of three 2-d steps, one on each of the three defining planes of the vertex.

The algorithm can perform an elementary step provided there exists at least one vertex with all three of the left-going edges in the current cut. Since this condition is always satisfied, the algorithm can perform elementary steps until all the vertices have been swept. To discover where in a cut an elementary step can be applied, a data structure based on the horizon tree [11] is used. This data structure stores information about the cells intersected by the sweep surface. The data structure requires $O(n^2 \log n)$ time for initialization and $O(1)$ amortized time per elementary step. Consequently, the overall sweep takes $O(n^3)$ time. The space complexity is $O(n^2)$ due to the use of a "local" data structure that requires $O(n)$ space for each plane of the arrangement.

In order to solve our problem, we perform a topological sweep of $\mathcal{A}(H)$. When leaving a cell c, we test every vertex-face, edge-edge, and edge-face pair of c in order to identify and compute the width of all pairs that are vertically aligned, i.e., all pairs that can be joined by a vertical segment interior to c. These pairs correspond to candidates from C_{31}, C_{22} and C_{21}, respectively, associated with c. As described below, each candidate can be processed in $O(1)$ amortized time. While performing the sweep, we keep the vertices, edges and faces of all active cells. This can be done using $O(n^2)$ space as described in [2]. At this point, we should note that not all cells of the arrangement are considered. In fact, we discard the points (in the dual) corresponding to planes (in the primal) that do not intersect the convex hull of S.

The details on how to process a candidate slab depend on its type. We now elaborate on this.

C_{31}: When leaving a cell c, we compute the width of each vertex-face pair associated with c and update the maximum every time a better candidate is found. In order to do this, for each vertex v of c we identify the face of c intersected by a vertical segment, interior to c, emanating from v. This is done by comparing the vertex against all faces of c. We then compute the width of the slab associated with this segment by using Observation 1.

C_{22}: The edge-edge pairs of C_{22} can be identified and reported as in the C_{31} case. We omit the details.

C_{21}: The width of the edge-face pairs of C_{21} can be computed as in C_{31}. Identifying the candidates, however, is more difficult. This is due to the fact that the number of vertical segments associated with an edge-face pair is not finite. Each such segment corresponds to an empty slab. Fortunately, the orthogonality condition of Lemma 1(b) can be used to identify the desired candidates as follows. Suppose that in dual space we find a vertical segment s connecting a point p_e on the edge e to a point p_f on the face f of a cell c. The parallel planes π_1 and π_2 in primal space that correspond to points p_e and p_f, respectively, have common normal vector $\overrightarrow{n_s} = (x(s), y(s), -1)$. Let p_{1i} and p_{1j} be the input points associated with the dual planes incident on e. and let p_{2k} be the input point associated with dual face f. The plane π passing through p_{1i}, p_{1j} and p_{2k} has normal vector $\overrightarrow{n} = (p_{1i} - p_{2k}) \times (p_{1j} - p_{2k})$. Edge e and face f determine a candidate slab if plane π is orthogonal to planes π_1 and π_2, i.e., if $\overrightarrow{n_s} \cdot \overrightarrow{n} = 0$. By parameterizing edge e in terms of its endpoints it is straightforward to determine a point on e that satisfies the orthogonality condition or to conclude that such a point does not exist. This computation takes $O(1)$ time.

The following lemma allows us to compute an upper bound on the total number of candidates in $C_{31} \cup C_{22} \cup C_{21}$ as well as a bound on the time required to identify those candidates.

Lemma 4 *[3] Let $\mathcal{A}(H)$ be the arrangement of a collection of n planes in R^3. For each cell c of $\mathcal{A}(H)$ let $f_i(c)$ denote the number of i-dimensional faces of the boundary of c, for $i = 0, 1, 2$, and let $f(c) = \Sigma_{i=0}^{2} f_i(c)$. Then $\Sigma f(c)^2 = O(n^3)$ where the sum extends over all cells of $\mathcal{A}(H)$.*

The result below is now a simple consequence of the previous discussion and the fact that the total number of vertex-face, edge-edge and edge-face pairs inside the cells of $\mathcal{A}(H)$ is bounded by $\Sigma f(c)^2$. Note, in particular, that Lemma 4 allows us to identify all candidate slabs for all cells in $O(n^3)$ time.

Theorem 5 *An obnoxious plane though a set of n points in $\mathbb{R}^3$ can be computed in $O(n^3)$ time and $O(n^2)$ space.*

Clearly, if degeneracies are present and if a topological sweep algorithm that handles them is not available, one can first construct $\mathcal{A}(H)$ explicitly. This should be done using a robust algorithm, such as the incremental solution coupled with simulation of simplicity, described in [10]. When doing this, the space complexity increases to $O(n^3)$ while the time complexity remains the same.

4 The constrained problems

In this section we consider constrained versions of the obnoxious plane problem where the optimal plane is required to pass through a fixed line or point.

4.1 The line-constrained obnoxious plane problem

The line-constrained version can be stated as follows. Given a set S of n points and a line ℓ in $\mathbb{R}^3$, compute a plane π passing through ℓ such that $\min_{p \in S} d(p, \pi)$ is maximal.

Without loss of generality, we assume that the line ℓ is the x-axis. We seek an optimal obnoxious plane π through this axis. Let π_α denote the plane whose normal $\overrightarrow{n}$ makes an angle α with the y-axis. Thus, we are looking for the value of $\alpha \in [0, \pi)$ such that $\min_{p \in S} d(p, \pi_\alpha)$ is maximal.

The proposed algorithm partitions the interval $[0, \pi)$ into subintervals such that all (rotated) planes in the same subinterval have the same point $p \in S$ as the nearest point. To compute the optimal value of α, it suffices to compute the lower envelope of n univariate functions $d(p, \pi_\alpha)$, $p \in S$.

The following result is crucial for computing the lower envelope efficiently.

Lemma 6 *Let p and q be two distinct points of S. Then, the functions $d(p, \pi_\alpha)$ and $d(q, \pi_\alpha)$ have at most two points of intersection.*

Proof: Let π be a plane passing trough the x-axis. The vector $\overrightarrow{n} = (0, \cos\alpha, \sin\alpha)$, $\alpha \in [0, \pi)$, is normal to the plane π_α. In other words, π_α has equation $\cos\alpha\ y + \sin\alpha\ z = 0$. Observe that for any two points $p = (p_1, p_2, p_3)$ and $q = (q_1, q_2, q_3)$ in S, the intersection of

$d(p, \pi_\alpha)$ with $d(q, \pi_\alpha)$ satisfies $|p_2 \cos\alpha + p_3 \sin\alpha| = |q_2 \cos\alpha + q_3 \sin\alpha|$. Thus, the distance functions have common points for $\alpha = \arctan(\frac{q_2 - p_2}{p_3 - q_3})$ and $\alpha = \arctan(\frac{q_2 + p_2}{p_3 + q_3})$. Furthermore, the functions coincide when $p_2 = q_2 = 0$ and $|p_3| = |q_3|$. This proves the claim. □

Let L_S be the lower envelope of the graphs of $d(p, \pi_\alpha)$, $p \in S$. Lemma 6 implies that the identifiers of the points corresponding to the edges of L_S, when traversing $L_\mathcal{P}$ from left to right, form a Davenport-Schinzel sequence of order two ([18]). Then, by divide-and-conquer, we can compute L_S in $O(n \log n)$ time. Furthermore, the number of intervals in the partition is in $O(n)$ ([18]). Thus, by traversing L_S, from left to right, we can identify the highest vertex, which corresponds to the optimal direction for π_α. This leads an $O(n \log n)$-time algorithm. A lower bound $\Omega(n \log n)$ for this problem can be obtained by reducing the *largest empty anchored cylinder* problem of [12] to our problem. Given n points on a plane Π, and an anchor point $O \in \Pi$, we consider the line l through O perpendicular to Π. The optimal obnoxious plane constrained to l solves the 2-d problem of [12]. Since any solution to the problem of [12] requires $\Omega(n \log n)$ time under the algebraic computation tree model, our algorithm is optimal under this model. In summary, we have proven the result below.

Theorem 7 *The line-constrained obnoxious plane can be computed in optimal $O(n \log n)$ time and $O(n)$ space.*

4.2 The point-constrained problem

The point-constrained problem can be stated as follows. Given a set S of n points and a point p_o, all in $\mathbb{R}^3$, compute a plane π_o through p_o such that $\varepsilon_o = \min_{p \in S} d(p, \pi_o)$ is maximal.

Without loss of generality, we assume that p_o is the origin. Our goal is to compute a normal $\overrightarrow{n_o}$ to π_o such that ε_o is maximal. Let $(1, \theta, \phi)$ be the spherical coordinates of $\overrightarrow{n_o}$, with zenith $0 \le \phi \le \pi$ and azymuth $0 \le \theta < 2\pi$. Given a point $p = (x_1, y_1, z_1) \in S$, the distance d_p between p and π_o can be expressed as

$$d_p(\phi, \theta) = |\sin\phi \cos\theta \, x_1 + \sin\phi \sin\theta \, y_1 + \cos\phi \, z_1|.$$

We can extend the approach described in the previous section by considering a finite collection of surfaces in 3-d space. To compute the lower envelope of the n bivariate functions we can use the divide-and-conquer deterministic approach of [1]. Thus, assuming an appropriate model of computation, we establish the following result.

We are interested in computing the lower envelope of the n bivariate functions $\mathcal{F} = \{d_p : p \in S\}$. Various randomized and deterministic algorithms for doing this in $O(n^{2+\varepsilon})$ time have been proposed (see [19], for a recent review). We can use, for example, the divide-and-conquer deterministic approach of [1].

The combinatorial complexity of the minimization diagram of $\mathcal{F}$ is $O(n^{2+\varepsilon})$. Since the maximum value of the function associated with a cell can be found in $O(1)$ time, the problem can be solved in $O(n^{2+\varepsilon})$ time. This assumes, of course, an appropriate model of computation – the usual real RAM model augmented with primitives to compute the intersection between two functions d_p and d_q and the maximum of a function d_p in constant time–.

Furthermore, we can avoid the trigonometric operations by rewriting the distance function as

$$d_p = |xyp_1 + x\sqrt{1-y^2}p_2 + \sqrt{1-x^2}p_3|,\ x \in [-1, 1].$$

We have established the following result.

Theorem 8 *The point-constrained obnoxious plane can be computed in $O(n^{2+\varepsilon})$ time and space.*

5 The obnoxious plane through polyhedral objects

Let $\mathcal{O}$ be a set of simple polyhedral objects in $\mathbb{R}^3$ with a total of n vertices. The **OPP** for $\mathcal{O}$ consists of finding a plane which maximizes the minimum distance to the objects. An empty slab C through $\mathcal{O}$ is an open region that intersects no objects from $\mathcal{O}$ and is enclosed by two parallel planes that intersect the convex hull of $\mathcal{O}$. Note that for a given $\mathcal{O}$ an empty slab may not exist. It is not difficult to prove that the bounding planes of a widest empty slab through $\mathcal{O}$ satisfy one of the four conditions of Lemma 1, except that the points p_{ij} are now vertices of polyhedra in $\mathcal{O}$.

Our method extend the approach of [16] to the three dimensional space. Let E be the set of edges of the polyhedra in $\mathcal{O}$. The dual representation of edge $e \in E$ with endpoints p and p' is the double-wedge $W(e)$ formed by planes $\mathcal{D}(p)$ and $\mathcal{D}(p')$ that does not contain the vertical plane through the line $\mathcal{D}(p) \cap \mathcal{D}(p')$. A plane π intersects e if and only if point $\mathcal{D}(\pi)$ lies inside $W(e)$.

Let $\mathcal{A}(H)$ be the arrangement of the n dual planes of the vertices of $\mathcal{O}$. (This arrangement is the same as the arrangement of the planes bounding the double-wedges $W(e)$ for $e \in E$.) Let π_1 and π_2 be two planes in the primal space. Planes π_1 and π_2 intersect the same edges of E (and therefore the same number of edges) if and only if $\mathcal{D}(\pi_1)$ and $\mathcal{D}(\pi_2)$ lie in the same cell of $\mathcal{A}(H)$. Let *count*(c) denote the number of edges of E intersected by any plane whose dual lies inside cell c of $\mathcal{A}(H)$. When *count*$(c) = 0$, the points in c correspond to planes in the primal that do not intersect any edge of E. Consequently, an open vertical segment whose endpoints lie on the boundary of a cell c with *count*$(c) = 0$ is the dual of an empty slab through $\mathcal{O}$.

To find the widest empty slab through $\mathcal{O}$, we use the topological sweep algorithm described in Section 3.1, but consider only the cells c of $\mathcal{A}(H)$ for which *count*$(c) = 0$. To identify these cells, we adapt a technique of [2] to compute *count*(c) in $O(1)$ time for each cell c of $\mathcal{A}(H)$. This computation is done when a cell is first encountered during the sweep. At the start of the algorithm, we compute the count for each of the $O(n^2)$ cells cut by the initial topological plane. Since we have $O(n)$ edges, this takes $O(n)$ time per cell and $O(n^3)$ time altogether. Consider now the computation of *count*(c) during the sweep. Suppose that the sweep plane first encounters c at a vertex formed by the intersection of planes v_1^*, v_2^*, v_3^*, corresponding to vertices v_1, v_2, v_3 of $\mathcal{O}$, respectively. Let c' be the cell of $\mathcal{A}(H)$ left behind by the sweep plane when c is first encountered. To compute *count*(c) from *count*(c') we consider only the double-wedges that may change the count at c. Initially, *count*(c) is set to *count*(c'). Then, we increment the count for each double-wedge that contains c but not c', and decrement it for each double-wedge that contains c' but not c. The time to do this is proportional to the number of edges from E incident on v_1, v_2, or v_3. This number is at most nine, as the worst case occurs when the plane through v_1, v_2, v_3 does not contain any edges from E. Consequently, *count*(c) can be computed from *count*(c') in $O(1)$ time, and the result below follows.

Theorem 9 *An obnoxious plane through a set of polyhedral objects in* $\mathbb{R}^3$ *with a total of* n *vertices can be computed in* $O(n^3)$ *time and* $O(n^2)$ *space.*

6 Computing a largest empty annulus

In [8] it is shown that given a set of n points S in $\mathbb{R}^2$, an empty annulus A (open region between two concentric circles) with largest width that partitions S into two subsets of points can be computed in $O(n^3 \log n)$ time an $O(n)$ space. We present an alternative algorithm to solve this problem in $O(n^3)$ time and $O(n^2)$ space.

Let us borrow the notation of [8]. Let $o(A)$ and $O(A)$ denote the inner and outer boundary of the circles defining A. Let $w(A)$, the width of A, be the positive difference between the radii of $O(A)$ and $o(A)$. An empty annulus of greatest width is a *syzygy* annulus if there are points p, q, with $p \in S \cap o(A)$ and $q \in S \cap O(A)$, and p is contained in the open segment whose endpoints are the center of the inner circle and q. As pointed out in [8], there always exist a largest empty annulus A such that (1) A is not a sizygy annulus and $|S \cap o(A)| \geq 2$ and $|S \cap O(A)| \geq 2$, or (2) A is a sizygy annulus and $|S \cap o(A)| \geq 2$ and $|S \cap O(A)| \geq 1$.

We first transform the set S from $\mathbb{R}^2$ to $\mathbb{R}^3$ by the well known paraboloid transformation $\mathcal{P} : p = (p_x, p_y) \rightarrow p^* = (p_x, p_y, p_x^2 + p_y^2)$. The point $p^\star$ is the vertical projection of point p onto the unit paraboloid $U : z = x^2 + y^2$ of $\mathbb{R}^3$. There is a one-to-one correspondence between circles in the original space and non-vertical planes in the transformes space.

It can be easily verified that the mapping $\mathcal{P}$ raises the annulus A of inner circumference $o(A) : x^2 + y^2 + ax + by + c = 0$ and outer circumference $O(A) : x^2 + y^2 + ax + by + d = 0$, with $c > d$, to the slab $A^\star$ bounded by the parallel planes $o(A)^\star : z + ax + by + c = 0$ and $O(A)^\star : z + ax + by + d = 0$. Reciprocally, any non vertical slab C bounded by planes $\pi : z + ax + by + c = 0$ and $\Pi : z + ax + by + d = 0$, with $c > d$, and both intersecting the unit paraboloid U, transforms to an annulus $C^\star$ whit inner and outer circumferences $\pi^\star : x^2 + y^2 + ax + by + c = 0$ and $\Pi^\star : x^2 + y^2 + ax + by + d = 0$ respectively, and with width $w(C^\star) = 1/2\left[\sqrt{a^2+b^2-4d} - \sqrt{a^2+b^2-4c}\right]$. Observe also that a point p lies on (respectively inside, outside) a circle c if and only if the dual hyperplane $c^\star$ contains (respectively passes above, below) the dual point $p^\star$. Thus, the largest empty annulus problem in the plane reduces to the largest empty slab problem in the space. In fact, there are two cases to be considered. (1) The optimal slab corresponds to a non sizygy annulus (the candidate slabs can be determined adapting the C_{22} case) or (2) the optimal slab corresponds to a sizygy annulus (the candidate slabs can be determined adapting the C_{21} case). Consequently, we have the following result.

Theorem 10 *Given a set of n points in $\mathbb{R}^2$, a largest empty annulus can be computed in $O(n^3)$ time and $O(n^2)$ space.*

7 Concluding remarks

In this paper, we have presented algorithms for solving the obnoxious plane problem (or equivalently, the largest empty slab problem) and several of its variants. For the case where the input consists of points or polyhedra, the problem can be solved in $O(n^3)$ time and $O(n^2)$ space. The solutions proposed are based on a geometric transformation that maps points to planes, planes to points, and edges to double-wedges. We also consider cases where the solution plane is constrained to contain a given line or a given point. The former can be solved in $O(n \log n)$ time and $O(n)$ space; the latter, in $O(n^{2+\epsilon})$ time and space. The solutions to the constrained problems are obtained by reducing the problem to a simple task of computing maxima on lower envelopes. Finally, we have considered a connection between the obnoxious plane problem and largest empty annulus problem of [8]. An adaptation of our method for solving the latter problem in the plane has also been presented, improving the known time-complexity.

As open problem are left other variants of the problem, which include dynamic updates, k-dense slabs, approximation algorithms with time complexity $o(n^3)$ and the problem of estimating tight lower bounds.

Acknowledgments

The first author was partially supported by Projects BFM2000-1052-C02-01 and BFM2003-04062. The second author was supported in part by the National Science Foundation under grant DMS-0107628. The third author was supported in part by grants TIC2001-2392-C03-01.

References

[1] Agarwal P., Schwarzkopf O. and Sharir M.: The overlay of lower envelopes and its applications. Discrete and Computational Geometry 15 (1996) 1-13.

[2] Anagnostou, E. G., Guibas, L. J., and Polimenis V. G.: Topological sweeping in three dimensions. Proceedings of the International Symposium on Algorithms (SIGAL). Lecture Notes in Computer Science 450 (1990) 310-317.

[3] Aronov, B., Matousek, J. and Sharir M.: On the sum of Squares of cell Complexities in Hyperplane Arrangements. J. Combin. Theory Ser A 65 (1994) 311-321.

[4] Ben-Moshe B., M.J., Katz, M.J. and Segal, M.: Obnoxious Facility Location: Complete Services with Minimal Harm. Int. J. Comp. Geom. and App. 10 (2000) 581-592.

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

[6] Chattopadhyay S. and Das P.: The k-dense corridor problems. Pattern Recogn. Lett. 11 (1990) 463-469.

[7] Cheng. S.-W.: Widest empty L-shaped corridor. Inf. Proc. Lett. 58 (1996) 277-283.

[8] Díaz-Báñez J.M., F. Hurtado F., Meijer H., Rappaport D. and Sellarès T.: The largest empty annulus problem. International Journal of Computational Geometry and Applications 13(4) (2003) 317-325.

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

[10] Edelsbrunner H.: *Algorithms in Combinatorial Geometry*. Springer-Verlag (1987).

[11] Edelsbrunner H. and Guibas L.: Topologically sweeping an arrangement. Journal of Computer and System Sciences 38 (1989) 165-194.

[12] Follert, F., Schömer, E., Sellen, J., Smid, M., Thiel, C.: Computing a largest empty anchored cylinder and related problems. International Journal of Computational Geometry and Applications 7, (1997) 563-580.

[13] Houle, M., Maciel A.: Finding the widest empty corridor through a set of points, In G.T. Toussaint, ed., *Snapshots of computational and discrete geometry*, 210-213. TR SOCS-88.11, dept of Computer Science, McGill University, Canada, 1988.

[14] Hurtado F., Seara C. and Sethia S.: Red-Blue Separability Problems in 3D, Lecture Notes in Computer Science 2669 (2003) 766-775.

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

[16] Janardan R., Preparata F.P.: Widest-corridor problems. Nordic Journal of Computing 1 (1994) 231-245.

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

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

[19] Sharir M.: Recent Developments in Theory of Arrangements of Surfaces. Lecture Notes in Computer Science 1738 (2000) 1-21.

[20] Shin C., Shin S. Y., Chwa K.: The widest k-dense corridor problems. Information Processing Letters 68(1) (1998) 25-31.

[21] Schöbel A. Locating least distant lines in the plane. European Journal of Operational Research 106 (1998) 152-159.

[22] Toussaint G. T.: Computing Largest Empty Circles with Location Constraints. International Journal of Computer and Information Sciences 12 (1983) 347–358.