Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research 33 (2006) 1117–1128

computers & operations research

www.elsevier.com/locate/cor

Computing obnoxious 1-corner polygonal chains

J.M. Díaz-Báñez[a,*], F. Hurtado[b]

[a]Departamento de Matemática Aplicada II, US, Virgen de Africa, 7, 41011 Sevilla, Spain
[b]Departamento de Matemática Aplicada II, UPC, Barcelona, Spain

Abstract

We consider an obnoxious facility location problem in which the facility is a trajectory consisting of a bounded length polygonal chain of two edges having extremes anchored at two given points. In other words, given a set S of points in the plane and a positive value l_0, we want to compute an anchored 1-corner polygonal chain having length at most l_0 such that the minimum distance to the points in S is maximized. We present non-trivial algorithms based on geometric properties of each possible configuration providing a solution. More specifically, we give an $O(n \log n)$-time algorithm for finding a 1-corner obnoxious polygonal chain whose length is exactly l_0, and an $O(n^2)$-time algorithm when the length of the optimal chain is at most the given bound l_0.

Keywords: Facility location; Computational geometry; Maximin; Transportation

1. Introduction

A classical operations research problem that has also been considered in the computer science community is the *facility location* problem. The task is to position an object (*the facility*) in an underlying space such that a distance measure between the facility and some given points (*the demand points*) is minimized or maximized. Most of the problems described in the literature are concerned with finding an optimal location for a "desirable" facility, where the goal is to minimize a distance function between the facility and the sites. Just as important is the case of locating an "undesirable" or "obnoxious" facility. In this case, instead of minimizing the largest distance between the facility and the given points, we would like to maximize the smallest distance.

* Corresponding author.
E-mail addresses: dbanez@us.es (J.M. Díaz-Báñez), ferran.hurtado@upc.es (F. Hurtado).

0305-0548/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2004.09.006

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; see Díaz-Báñez et al. [1] for a recent survey on the current state-of-art of these problems.

In this paper, we deal with the placement of an undesirable facility modelled by a polygonal chain with one corner, amidst existing installations or facilities. It is clear that the location of this trajectory must be constrained, as otherwise the route may be simply removed to infinity. We also consider an additional constraint on the length of the chain, because when the chain represents a facility to be constructed or routed, the length is related to some cost, and becomes clearly relevant. We also add the restriction that the chain must start and end at specified anchor points a and b, corresponding to given origin and destination.

Due the geometric nature of the problem we address the resolution from the point of view of the computational geometry. There is a vast literature on location theory and there exist actually a lot of papers based on its connection with computational geometry. In fact, with many practical motivations, geometric instances of facility location problems have attracted a significant amount of the research to date. Some applications of the computation of an obnoxious route include urban, industrial and military task planning when the transportation of some kind of obnoxious material is addressed. The proposed problem combines the computation of a short path with risks issues as actually done in real-world applications within hazardous material logistic area. See Erkut and Verter [2,3] where a discrete underlying space is considered and Drezner and Wesolowsky [4], Melachrinoudis and Xanthopulos [5] and Díaz-Báñez et al. [6] for the continuous case.

On the other hand, applications of these problems go well beyond the field of location science. For instance, the problem to compute a connecting path avoiding collisions is one of the most important tasks in robotics. In [7], a path allowing right-angle turns is considered. In order to minimize the cost, to consider a bound on the length of the path is a logical constraint. In this sense, our problem gives a path with maximal clearance.

There has been considerable activity in the computational geometry community on facility location problems that involve computing non-single facilities of various types. Several optimization problems dealing the location of a 1-corner chain using a minimax criterion have been posed by Glozman et al. [8] and Díaz-Báñez et al. [9]. On the other hand, maximin criteria have been investigated for the optimal positioning of points [10–12], lines [13], line segments [14], circumferences [15], and planes in 3-D [16].

An outline of the paper is as follows. In Section 2, we study the configuration cases that may determine an optimal route depending on whether the length of the polygonal chain is equal to or at most l_0. We show that an optimal 1-corner route for the problem must be at minimum distance from two or three points; all these configurations may be explored with a brute-force $O(n^4)$-time algorithm. In Section 3, we present an $O(n \log n)$-time algorithm for the maximin 1-corner polygonal route problem when the length of the route is exactly l_0. When l_0 is an upper bound for the length, we propose different algorithms, described in Section 4, depending on the position of the points that determine an optimal chain. All these cases can be solved within $O(n^2)$ worst case running time.

2. Overview

We start by introducing some notations. Hereafter we denote by $S=\{p_1, p_2, \ldots, p_n\}$ the given finite set of points. A *1-corner route* $\mathcal{R}$ is a polygonal chain with one corner (its intermediate vertex) that starts and ends at specified anchor points a and b. When the corner point is q, we also use the notation $\mathcal{R}=a-q-b$.

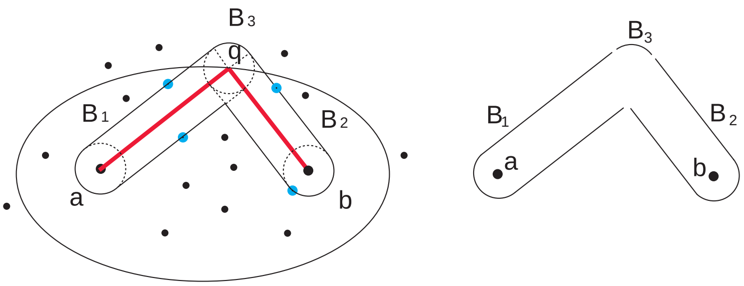

Fig. 1. An empty anchored boomerang and its three parts.

We denote the line segment connecting p and q by $\overline{pq}$. The Euclidean distance between two points p and q is denoted $d(p, q)$. If p is a point in $\mathbb{R}^2$, and C is a closed subset of $\mathbb{R}^2$, then the distance between p and C is defined as $d(p, C) = \min\{d(p, q) : q \in C\}$. Thus, $d(p, \mathscr{R}) = \min\{d(p, \overline{aq}), d(p, \overline{pq})\}$. The maximin 1-corner polygonal chain problem can be now stated as follows.

Given a set S of points in the plane and a positive value l_0, find a 1-corner polygonal route, $\mathscr{R}$ with Euclidean length $l(\mathscr{R}) \leqslant l_0$, such that $\min_{p_i \in S} d(p_i, \mathscr{R})$ *is maximized among all possible chains fulfilling the conditions.*

It is clear that the solution to this problem might not be unique. For example when the anchor points are outside the convex hull of S there may exist an infinite number of solutions. Although our algorithms can be handled for detecting all optimal solutions when there exist a finite number of them, in this paper we focus on finding *one* optimal configuration.

Let us observe that the restriction on the length implies that the point q cannot be exterior to an ellipse with focus at a and b. This defines a continuous search space; however, we can generate a discrete set of candidate placements as follows.

Definition 1. Given a 1-corner route $\mathscr{R} = a - q - b$, the locus of points at distance r from $\mathscr{R}$ is called a *boomerang* centered at $\mathscr{R}$ and radius r. We call a boomerang an *empty boomerang* if it does not contain any point of S. An empty boomerang is *critical* when some points from S lie on its boundary; in this case enlarging its radius would result in a non-empty boomerang.

Equivalently, a boomerang is the area swept by a disk whose center describes the route. Thus, in a geometric setting, the problem asks for finding the largest empty boomerang anchored at a and b (refer to Fig. 1).

Definition 2. The points in S that determine a tentative placement of an optimal route $\mathscr{R} = a - q - b$ by the fact that they lie on the boundary of its critical boomerang are called the *critical points*.

Let B_1, B_2 and B_3 be the parts of the boomerang where the critical points may lie, corresponding to points whose closest point in the route belongs to segment $\overline{aq}$, $\overline{qb}$, or is the point q, respectively (Fig. 1). We classify the cases for critical points cases according to their location on the parts of the boomerang, as shown in Fig. 2. We briefly describe the way for obtaining the cases. We begin with the obvious observation that the boundary of an optimal boomerang must contain at least one point of S as, otherwise,

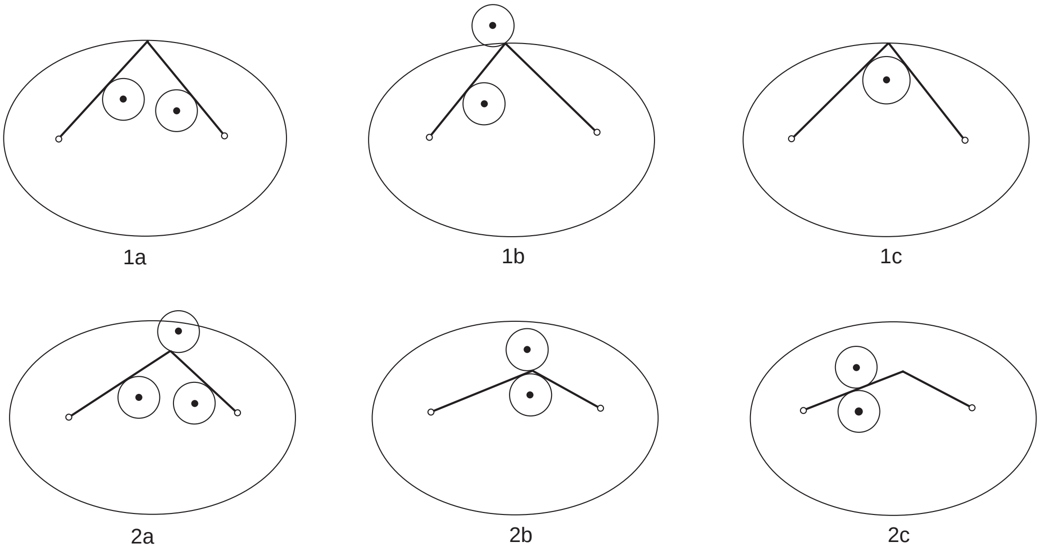

Fig. 2. Cases of critical points.

the radius can be increased. Then, the idea is to use the freedom left in order to move the chain anchored at a and b and to increase its radius until at least one more point of S is encountered. All of the six cases in Fig. 2 are candidates or critical configurations because every movement of the corner q decreases the minimum distance to the chain. An exhaustive but straightforward analysis of the situations that may arise gives immediately the following result, which we state formally for future reference.

Lemma 1. *Let $\mathscr{R}$ be a solution for the maximin 1-corner polygonal chain and B the critical boomerang generated by $\mathscr{R}$. Then the possible positions of the critical points for $\mathscr{R}$ are as follows.*

1. *If q is on the ellipse*:
   (a) *One point in B_1 and one point in B_2, or*
   (b) *one point in B_1 (or B_2) and one point in B_3, or*
   (c) *one point in $B_1 \cap B_2$.*
2. *If q is inside the ellipse*:
   (a) *One point in B_1, one point in B_2 and one point in B_3, or*
   (b) *one point in $B_1 \cap B_2$, and one point in B_3, or*
   (c) *two points in B_1 (or B_2).*

Using this lemma we could exhaustively consider all possible cases for the placement of a critical boomerang according to its critical points, and then find the optimal one, which would lead to a naive $O(n^4)$-time algorithm.

In the next sections, we show how to solve the problem more efficiently.

3. Finding the best corner on the ellipse

In this section, we give an $O(n \log n)$-time algorithm for the version of the maximin 1-corner route problem in which the length of the route is exactly a number l_0, which is equivalent to force the corner q to lie on an ellipse E with focus a and b. In other words, we want to compute an anchored route $\mathscr{R}=a-q-b$, with $q \in E$, such that $\min_{p_i \in S} d(p_i, \mathscr{R})$ is maximized.

Our algorithm uses the lower envelope of a suitable family of functions. The key idea is to divide the ellipse into a linear number of arcs such that in each arc we know which point of S minimizes the distance to the articulated configuration that models the route. Notice that when this point of S is unique, it has the same property minimizing the distance in a neighborhood, and therefore that the positions on the ellipse in which there are ties are finite in number.

Denote by $\mathscr{P}$ the partition of E into arcs $A_j = \{q \in E : \min_{p_i \in S} d(p_i, \mathscr{R}) = d(p_j, \mathscr{R})\}$, where we assume that no two consecutive arcs belong to the same point of S (otherwise we would merge the two arcs).

We can prove that $\mathscr{P}$ has a linear number of arcs A_j. In order to simplify the explanation, notice first of all that we can obtain the same kind of partitions $\mathscr{P}_a$ and $\mathscr{P}_b$ of E corresponding, respectively, to the segments $\overline{aq}$ and $\overline{bq}$, independently and see that their size is linear. Then, by using the definition of the distance function, we can easily compute $\mathscr{P}$, since $d(p_i, \mathscr{R}) = \min\{d(p_i, \overline{aq}), d(p_i, \overline{bq})\}$, and therefore $\mathscr{P}$ can be computed in linear time by comparing distances in each sub-interval, as we have $|\mathscr{P}| \leqslant |\mathscr{P}_a| + |\mathscr{P}_b|$. Hence, we see that it is enough to show how to compute the partition of the ellipse E when the left segment $\overline{aq}$ rotates with q describing E, as the partition $\mathscr{P}_b$ can be computed in an analogous way.

Without loss of generality, we consider that the origin of coordinates o is the anchor point a. Let θ be the polar angle of $q \in E$; for each $p_i \in S$ we denote by $d_i(\theta) = d(p_i, \overline{oq})$ the distance between the point p_i and the segment $\overline{oq}$. Then, it is sufficient to compute the lower envelope of the n univariated continuous functions $d_i(\theta)$, $p_i \in S$, in order to determine the arcs A_j of the partition of the ellipse E.

Lemma 2. *Let p_i and p_j be two distinct points of S. Then, the graphs of the functions d_i and d_j intersect at most twice.*

Proof. Given two points p_i and p_j of S, $i \neq j$, let $L(i,j) = \{x \in \mathbb{R}^2; d(p_i, \overline{ox}) = d(p_j, \overline{ox})\}$ be the *locus* of points x in the plane such that the distance from p_i and p_j to the line segment $\overline{ox}$ are equal. We show below that $L(i,j)$ and E intersect at most twice; consequently, there exist at most two points x, y on E for which $d(p_i, \overline{ox}) = d(p_j, \overline{ox})$and $d(p_i, \overline{oy}) = d(p_j, \overline{oy})$.

Barcia et al. [14] give an exhaustive description of the locus $L(i,j)$. For the sake of simplicity, we only describe one of the cases (the most common one); the other situations are similar. Suppose that $d(p_i, o) > d(p_j, o)$ and p_i, p_j and o are not collinear. Then $L(i,j)$ is a differentiable 2-dimensional curve consisting of two half-lines joined by an arc of curve of maximum degree four (as illustrated in Fig. 3). The locus $L(i,j)$ dissects the plane into two domains $D(i,j)$ and $D(j,i)$ having both of them $L(i,j)$ as complete separating boundary. More precisely, denote $D(i,j) = \{x \in \mathbb{R}^2 : d(p_i, \overline{ox}) < d(p_j, \overline{ox})\}$ and $D(j,i) = \{x \in \mathbb{R}^2 : d(p_j, \overline{ox}) < d(p_i, \overline{ox})\}$. Barcia et al. [14] give a complete proof of the fact that $D(i,j)$ is a convex region and that, furthermore, $L(i,j)$ is contained in a wedge with angle $\theta < (\pi/2)$ and apex in o (see Fig. 3). As the origin o lies inside the elipse, this implies that $L(i,j)$ intersects the ellipse E at most twice, which proves the claim. □

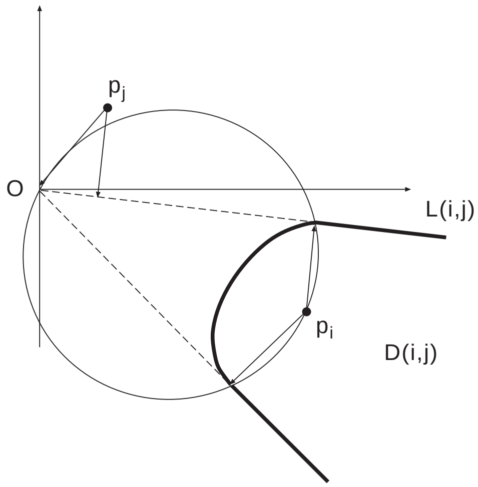

Fig. 3. The locus $L(i, j)$.

Let L_o^a be the lower envelope of the graphs of the functions d_i. Traverse this envelope from left to right, and form an ordered sequence with the labels of the points defining the pieces of functions encountered. Lemma 2 implies that this sequence is a Davenport–Schinzel sequence of order two [17]. Therefore, the number of intervals in the partition is $O(n)$ [17], and that L_o can be computed using a standard divide-and-conquer approach in $O(n \log n)$ time.

In a similar way, let L_o^b be the lower envelope of the graphs of the functions d_i when the origin o is used instead of the anchor b. In $O(n)$ time we can compute the lower envelope L of both functions L_o^a and L_o^b, which consists of $O(n)$ vertices. In fact, the merge step simply scans two sorted lists of intervals. Finally, traversing L from left to right, we can identify its maximum. In summary, we have proven the following result.

Theorem 1. *The obnoxious 1-corner route problem with corner on the ellipse E can be solved in* $O(n \log n)$ *time*.

4. Finding the corner inside the ellipse

In this section, we study the situations in which the corner of the trajectory may be placed in the interior of the ellipse, giving rise to the possibilities described in Lemma 1.

Case 2a: By definition, there are three points at minimum distance from the chain $\mathscr{R} = a - q - b$, each one in a different part of the associated boomerang. In order to explore all these configurations, for every point p_i of S we construct the largest radius boomerang that has p_i in its part B_3.

Lemma 3. *Given a set of points S and their radial orderings around a and around b, for each point* p_i *of S we can find in linear time the largest radius critical boomerang having* p_i *in its part* B_3*, and two additional critical points, one in* B_1 *and another one in* B_2.

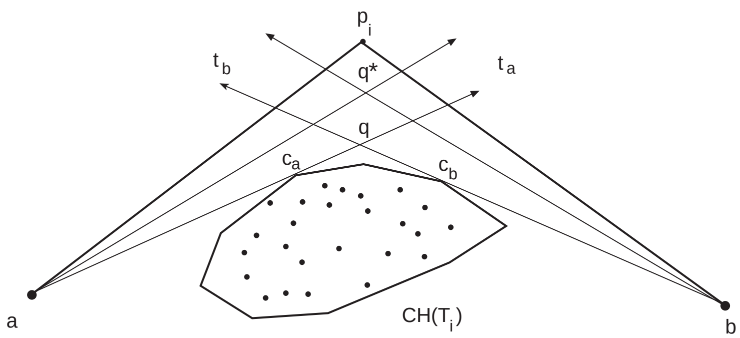

Fig. 4. Finding an optimal route for p_i in Case 2a.

Proof. Suppose without loss of generality that p_i is above the line containing a and b as in Fig. 4; let us see how to obtain an optimal configuration for this case. Let T_i be the set of points of S inside the triangle of vertices a, p_i and b. Let t_a and t_b, respectively, be the left supporting line to $CH(T_i)$ from a and the right supporting line from b, and let us denote by q the point where t_a and t_b intersect, and by c_a and c_b the respective contact points of these supporting lines with $CH(T_i)$. If q is outside the triangle, then p_i would not generate a candidate configuration for this case, thus we assume that q is inside the triangle.

Since the order of the points in T_i around a (as well as around b) is given, $CH(T_i)$ can be obtained in linear time. Besides that, we can obtain an ordered list of the nearest points of $CH(T_i)$ to the tangent t_a (resp. t_b) when it rotates counterclockwise (resp. clockwise), which is simply the list of vertices of $CH(T_i)$ as they appear counterclockwise on its boundary starting at c_a (resp. clockwise starting at c_b). In this way, in $O(n)$ time we can find the first points of each list p_a, p_b such that $d(p_a, \overline{aq^*}) = d(p_b, \overline{bq^*}) = d(p_i, q^*)$, where q^* is the intersection point between the rays from a and b which satisfies the preceding equalities. This point q^* gives the tentative optimal configuration; the emptyness of the rest of the boomerang can be checked in a final step. □

If we compute in a first step, using $O(n \log n)$ time, the radial order of the points in S, both around a and b, and then tentatively apply the preceding construction to every point p_i, we see that Lemma 3 gives an $O(n^2)$ time algorithm in order to find an optimal 1-corner route for Case 2a.

Case 2b: Suppose that there is a critical point p_i in $B_1 \cap B_2$ and another critical point p_j in B_3. We now consider the point p_i as fixed and compute the largest empty boomerang corresponding to this configuration (Fig. 5).

Lemma 4. *Given the Voronoi diagram of* S, $V(S)$, *for each point* $p_i \in S$ *we can obtain in linear time an optimal 1-corner route* $\mathscr{R} = a - q - b$ *among those such that* $d(p_i, \overline{aq}) = d(p_i, \overline{bq}) = d(p_j, q)$, *for some* $p_j \in S \backslash \{p_i\}$.

Proof. Consider a point p_i in S and let L_i be the locus of points q such that $d(p_i, \overline{aq}) = d(p_i, \overline{bq})$. It is easy to prove that the locus L_i is a line passing through p_i. Then, we must find the first point $p_j \in S \backslash \{p_i\}$ with the following properties: (1) There exists a point q in L_i such that $d(p_j, q) = d(p_i, \overline{aq}) = d(p_i, \overline{bq})$; and (2) no point p_k in $S \backslash \{p_i, p_j\}$ with $d(p_k, a - q - b) < d(p_i, a - q - b) = d(p_j, a - q - b)$.

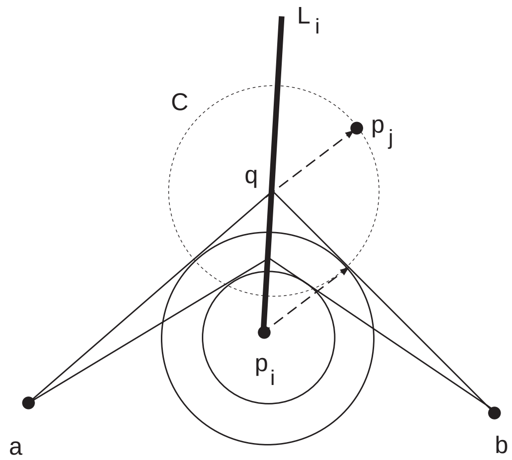

Fig. 5. Finding an optimal route for p_i in Case 2b.

As the circle C through p_j centered at q must be empty of points of S, q must lie in the Voronoi region of p_j. Therefore, we can intersect L_i with $V(S)$, which gives a linear number of segments, and find the solution for the points q satisfying the equalities in (1) above, in constant time in every such segment. If we walk along L_i, starting at p_i, and process the segments in the order in which they are encountered, we see that we can find the point p_j in linear time, which proves the claim. □

If we compute in a first step, using $O(n \log n)$ time, the Voronoi diagram $V(S)$ of the points in S, and then tentatively apply the preceding construction in Lemma 4 to every point p_i, we obtain an $O(n^2)$ time algorithm in order to find an optimal 1-corner route for Case 2b.

Case 2c: Without loss of generality, we assume that the critical points p_i and p_j determining the optimal 1-corner $a-q-b$ are closer to $\overline{aq}$ than to $\overline{bq}$. A technical result is required.

Lemma 5. *Let* $a_1, a_2, \ldots, a_n$ be *n ordered real numbers and let* $r_1, r_2, \ldots, r_n$ *be n non-negative values. Then, the set* $\bigcup_{1 \leqslant i \leqslant n} [a_i - r_i, a_i + r_i]$ *can be computed in* $O(n)$ *time.*

Proof. Let us construct the set $\bigcup_{1 \leqslant i \leqslant n} I_i$ incrementally following the increasing order of the centers $a_1, a_2, \ldots, a_n$. We maintain the current union in a list $\mathscr{L}$ of disjoint intervals $J_1, \ldots, J_t$, (in general different from the intervals I_i), stored in the order given by their centers.

Initially, we set the list to be $\mathscr{L} := \{J_1 = I_1\}$. If $I_1 \cap I_2 = \emptyset$ we simply add $J_2 = I_2$ to $\mathscr{L}$. Otherwise we update the list to contain the single interval $J_1 = I_1 \cup I_2$. Observe that, in this case, the center of J_1 lies between a_1 and a_2, which in particular implies that it lies to the left of a_3.

For the general step, assume that the union $I_1 \cup I_2 \cup \cdots \cup I_i$ has been computed and stored in a list $\mathscr{L} = \{J_1, \ldots, J_k\}$. Recall that the intervals in $\mathscr{L}$ are disjoint, their centers are increasingly ordered and all lie to the left of a_{i+1}; then we proceed as follows.

If $I_{i+1} \cap J_k = \emptyset$, we simply add $J_{k+1} = I_{i+1}$ to the current list, with cost $O(1)$ which we charge to the interval I_{i+1}, that has just entered the list; furthermore notice that $I_{i+1} \cap J_k = \emptyset$ implies that $I_{i+1} \cap J_1 = \cdots = I_{i+1} \cap J_{k-1} = \emptyset$.

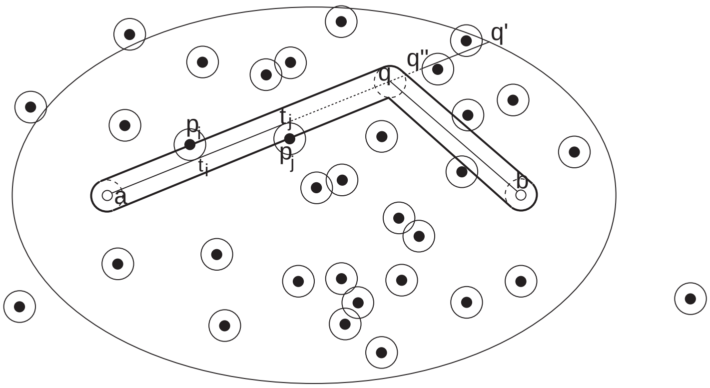

Fig. 6. Configuration for Case 2c.

If $I_{i+1} \cap J_k \neq \emptyset$ we update the list by making $J_k := J_k \cup I_{i+1}$, in $O(1)$ time which we charge to the "old" J_k, which has been just replaced. Next we check the intersection $J_k \cap J_{k-1}$. If it is empty, the cost of its computation is charged to I_{i+1}, which has entered the list after a suitable merging step, and we proceed to the next iteration with I_{i+2}. Otherwise, i.e., if $J_k \cap J_{k-1} \neq \emptyset$, we update J_{k-1} to be $J_{k-1} := J_{k-1} \cup J_k$ and remove J_k from the list, actions with cost $O(1)$ which is charged to the removed J_k. Next we would check whether $J_{k-1} \cap J_{k-2}$ is empty or not, and so on.

As in the whole process there are at most n intervals entering $\mathscr{L}$ and at most $n - 1$ intervals removed from the list, the overall running time is $O(n)$, as claimed. □

Remark. Notice that the preceding lemma also applies when, instead of points on a line, we have a circle (or an arc of circle), with points on it whose radial order around the center of the circle is given, and the union of intervals centered at these points has to be computed.

We are now ready for dealing with Case 2c. For the ease of description we are assuming general position hereafter, that is, no two points in S are collinear with a or are exactly at the same distance from a (these degenerate cases can be handled by the method but require many case details).

Lemma 6. *Suppose that the radial order of S around b is known. Then we can obtain in $O(n^2)$ time an optimal configuration $\mathscr{R}=a-q-b$ characterized by two critical points in the part B_1 of the corresponding boomerang.*

Proof. For each $\delta \geqslant 0$ and each point $p_i \in S$, let D_i^δ denote the disk with center p_i and radius δ. Let R_i^δ and L_i^δ be the rays anchored at a that are right tangent and left tangent, respectively, to the disk D_i^δ. When δ grows, the tangents R_i^δ and L_i^δ rotate clockwise and counterclockwise, respectively, for each p_i.

Let us denote by q' the point in which the ray through a point q, and having a as origin, intersects the ellipse E. Any configuration $a - q - b$ in situation 2c, with the critical points in B_1 and radius δ, fulfills three conditions (refer to Fig. 6).

(1) The disks of radius δ centered at a and b are empty of points from S.

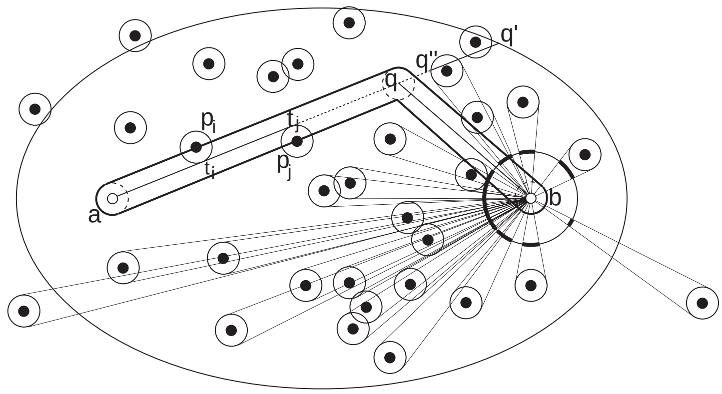

Fig. 7. Checking condition (3).

(2) If p_i and p_j are the critical points, the disks D_i^δ and D_j^δ are tangent at aq at points t_i and t_j, respectively. If t_j (for example) is the contact point furthest away from a, no disk D_k^δ is tangent or overlaps the segment at_j, for $k \neq i, j$.
(3) If we consider the disks as obstacles, b can see a point (namely q) in the segment t_jq' such that the disk of radius δ centered at q contains no point from S and no disk D_k^δ overlaps t_jq.

The first condition can be guaranteed a fortiori by taking values of δ bounded above by $\delta_{\min} = \min\{\min\{d(a, p_i), p_i \in S\}, \min\{d(b, p_i), p_i \in S\}\}$. If in addition to that we have a pair of critical points p_i and p_j in the situation of condition (2) (and henceforth a fixed value of δ), we can check condition (3) in $O(n)$ time: We first shrink the segment t_jq' to t_jq'' to ensure that no disk D_k^δ overlaps t_jq'', and then we treat the disks D_k^δ with centers in the halfplane of the line at_j in which b lies as obstacles when seen from b, that we can project onto intervals on a circle centered at b, and then compute their union as described in the remark after Lemma 5; the complement of the union gives the windows for fitting the arm $q - b$ of $a - q - b$ (Fig. 7).

Therefore, we are left with the task of determining how many pairs of points of S give candidate configurations in the situation of condition (2), and which is the cost of finding all such pairs.

Notice that for a fixed pair of critical points p_i, p_j as in Figs. 6 and 7, the tangents R_i and L_j coincide for some value of δ. For larger values of δ the disk D_i^δ will always overlap the segment between a and the contact point of L_j with D_j^δ, therefore L_j can no longer appear in a candidate configuration. Hence, if we charge the starting pair p_i, p_j to L_j, i.e., to the tangent with contact point further away from a, we have a charging scheme proving that the number of candidate configurations is linear.

In order to find all of them let us proceed as follows. For everyone of the tangents L_k (as well as for R_k), $k = 1, \ldots, n$, let us find all the values of δ making the tangent coincide with some other tangent. These values are stored in a doubly linked list $\mathscr{L}(L_k)$, each value with a pointer to its copy in the list for the other tangent giving rise to the value. Each list is scanned and rearranged in such a way that the smallest value appears in first position. The whole process for the set of tangents takes $O(n^2)$ time.

Next, we select the minimum value δ_1 among the first values from each list. If it corresponds to two left tangents L_i and L_j, with respective contact points t_i and t_j, and t_j is further away from a than t_i, the tangent L_j cannot participate in a candidate configuration for larger values of δ, because the disk D_i^δ

would overlap the segment at_j. Therefore we can discard the tangent L_j, as well as the list $\mathscr{L}(L_j)$ and the associated copies of the values in the other lists, remove δ_1 from $\mathscr{L}(L_i)$, and scan the updated list $\mathscr{L}(L_i)$ in order to put in the first position the smallest value. The whole process takes $O(n)$ time. If δ_1 corresponds to two right tangents, we proceed analogously. If it corresponds to a right tangent and a left tangent, we get a candidate configuration, but we have already seen in the charging scheme that one of the tangents can be discarded for larger values of δ; for the other one we remove δ_1 from its associated list and update the smallest value.

In any case, we see that we can always discard one of the tangents in $O(n)$ time, and are then ready to iterate. Therefore, the whole process of identifying the $O(n)$ candidates can be carried out in $O(n^2)$ time. As each one can be checked for condition (3) in linear time, and we can maintain the best found solution, this gives the claimed complexity. □

If we compute in a first step, using $O(n \log n)$ time, the radial order of the points in S around a and b, and then apply Lemma 6, we obtain an $O(n^2)$ time algorithm in order to find an optimal 1-corner route for Case 2c.

Combining the results from the three preceding cases, we arrive at the main result of this section.

Theorem 2. *The obnoxious 1-corner route problem whose corner is not external to an ellipse with focus a and b can be solved in* $O(n^2)$ *time*.

5. Conclusion and remarks

In this paper, we have considered a problem consisting of locating an undesirable facility amidst demand-points, where the facility is an anchored 1-corner polygonal chain with length constrained to be at most a given number. Algorithms based on geometric concepts and techniques have been proposed, beating substantially the brute-force approach.

The methods for the case in which the corner is constrained to lie on the ellipse can be extended to some other types of curves. In fact, if the corner q is on a curve intersecting the locus $L(i, j)$ at most twice, the approach can be applied in a quite similar way.

Finally, let us mention that the techniques we have used would lead to very high complexities when k corners are allowed, hence a new approach would have to be developed in order to obtain exact and efficient algorithms for general polygonal chains.

Acknowledgements

The first author is partially supported by Project MCYT BFM2000-1052-C02-01 and BFM2003-04062. The second author is partially supported by Projects DGES-SEUID PB98-0933, MCYT BFM2001-2340, MCYT-FEDER BFM2002-0557, MCYT BFM2003-00368 and Gen. Cat 2001SGR00224.

References

[1] Díaz-Báñez JM, Mesa JA, Shöbel A. Continuous location of dimensional structures. European Journal of Operational Research 2004;152:22–44.

[2] Erkut E, Verter V. Hazardous materials logistics. In: Drezner Z., editor. Facility location: a survey of applications and methods. Berlin: Springer; 1995. p. 467–506.

[3] Erkut E, Verter V. Modelling of transport risk for hazardous materials. Operations Research 1998;46(5):625–42.
[4] Drezner Z, Wesolowsky GO. Location of an obnoxious route. Journal Operational Research Society 1989;40:1011–8.
[5] Melachrinoudis E, Xanthopulos Z. Semi-obnoxious single facility location in Euclidean space. Computers and Operations Research 2003;30:2191–209.
[6] Díaz-Báñez JM, Gómez F, Toussaint GT. Computing shortest paths for transportation of hazardous materials in continuous spaces. Journal of Food Engineering 2004, in press.
[7] Chen S-W. Widest empty L-shaped corridor. Information Processing Letters 1996;58:277–83.
[8] Glozman A, Kedem K, Shpitalnik G. Computing a double-ray center for a planar point set. International Journal of Computational Geometry & Applications 1999;9:109–24.
[9] Díaz-Báñez JM, Gómez F, Hurtado F. Approximation of point sets by 1-corner polygonal chains. INFORMS Journal on Computing 2000;12:317–23.
[10] Toussaint GT. Computing largest empty circles with location constraints. International Journal of Computer and Information Sciences 1983;12:347–58.
[11] Ben-Moshe B, Katz MJ, Segal M. Obnoxious facility location: complete services with minimal harm. International Journal of Computational Geometry & Applications 2000;10:581–92.
[12] Katz MJ, Kedem K, Segal M. Improved algorithms for placing undesirable facilities. Computers and Operations Research 2002;29(13):1859–72.
[13] Follert F, Schömer E, Sellen J, Smid M, Thiel C. Computing a largest empty anchored cylinder and related problems. International Journal of Computational & Geometry Applications 1997;7:563–80.
[14] Barcia JA, Díaz-Báñez JM, Lozano A, Ventura I. Computing an obnoxious anchored segment. Operations Research Letters 2003;31:293–300.
[15] Díaz-Báñez JM, Hurtado F, Meijer H, Rappaport D, Sellarès JA. The largest empty annulus problem. International Journal of Computational Geometry & Applications 2003;13:317–25.
[16] Díaz-Báñez JM, López M, Sellarès JA. Computing largest empty slabs. Lecture Notes in Computer Science 2004;3045: 99–108.
[17] Sharir M, Agarwal PK. Davenport–schinzel sequences and their geometric applications. Cambridge: Cambridge University Press; 1995.