Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Operations Research Letters 34 (2006) 97–105
Operations Research Letters
www.elsevier.com/locate/orl

Locating two obnoxious facilities using the weighted maximin criterion

Arie Tamir

Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Ramat-Aviv, 69978 Tel-Aviv, Israel

Received 16 July 2004; accepted 17 February 2005
Available online 2 June 2005

Abstract

Given are a finite set of points P and a compact polygonal set S in R^2. The problem is to locate two new facilities in S, maximizing the minimum of all weighted distances between the points in P and the two new facilities, and the distance between the pair of new facilities. We present subquadratic algorithms.

Keywords: Obnoxious facilities; Maximin criteria; Planar location models

1. Introduction

The location of undesirable or obnoxious facilities has been a very important research area for more than twenty years. Location problems of this nature have been studied in a variety of metric spaces. Due to their potential applications the most popular models deal with planar and networks settings. (The reader is referred to [12,30,38,18,17,37,24,33,29,5,26] and the references cited therein, for further motivation and an extended list of applications.)

In this note we focus on the problem of locating two obnoxious facilities (new facilities) in the plane, where the objective is to maximize the minimum of all weighted distances between the customers (existing facilities) and the two new facilities, and the weighted distance between the pair of new facilities. The pair of obnoxious facilities is required to be located in some prespecified compact, polygonal domain.

We formulate the k-obnoxious facility model. Suppose that the input consists of a compact polygonal domain S in R^2, a set of demand points (existing facilities) $P=\{p_1,\dots,p_n\}$ in R^2, and two sets of nonnegative scalar weights: $\{w_1,\dots,w_n\}$ and $\{a_{i,j}: i,j=1,\dots,k\}$. Following [37,24,1], the problem is to find k points (new obnoxious facilities), $\{x_1,\dots,x_k\}$ in S, optimizing

The maximin weighted k-obnoxious facility location problem

max L

E-mail address: atamir@post.tau.ac.il.

0167-6377/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2005.02.004

s.t.

$$d(x_j, p_i) \geqslant w_i L, \quad i = 1, \ldots, n, \quad j = 1, \ldots, k,$$

$$d(x_s, x_t) \geqslant a_{s,t} L, \quad s, t = 1, \ldots, k, \quad s \neq t,$$

$$x_j \in S, \quad j = 1, \ldots, k.$$

To simplify the notation we assume that S has $m = \mathrm{O}(n)$ vertices and edges. When $w_i = 1$, for $i = 1, \ldots, n$, and $a_{i,j} = 1$, for $i, j = 1, \ldots, k$, we will call the model *unweighted*. Here we study only the 2-facility case, where $d(-,-)$ is either the rectilinear, L_1, or the Euclidean, L_2, distance function. (For the 2-facility problem, i.e., $k = 2$, we assume without loss of generality that $a_{1,2} = 1$.)

1.1. Previous and related results

Gianikkos and Appa [24,1] discuss the rectilinear version of the above weighted 2-obnoxious facility location problem, and present an algorithm based on binary search for its solution. For a prescribed value of L, they provide a geometric algorithm to test whether this value is feasible for the above problem. The complexity of this test is not explicitly stated, but it is certainly super quadratic in the number of points n. To find the maximum value of L, they use a binary search over a continuous range, known to contain the optimal value. This binary search does not produce L^*, the exact optimal solution value of L. (It only approximates the optimal value within any desirable accuracy level.)

We briefly survey some results in the literature, dealing with related maximin planar location problems. First, we refer the reader to [14,15,39,6,33,35,19,5] for algorithmic results on the simpler single facility model.

We are aware of only very few studies on planar multifacility maximin models. (See [11,30,17,37] for multifacility maximin problems on networks.)

Tamir [37] proved that when k, the number of obnoxious facilities, is part of the input, then the above maximin k-obnoxious facility location model is strongly NP-hard even in the one-dimensional case where S is the unit interval and $w_i = 0$ for $i = 1, \ldots, n$.

Katz et al. [26], improving upon earlier results in [9], consider the following decision model: given the set $P = \{p_1, \ldots, p_n\}$ of n demand points (existing facilities), a rectangular planar region S, and a pair of reals L and L', verify whether it is possible to locate k obnoxious facilities $\{x_1, \ldots, x_k\}$ in S, such that the minimum weighted distance between the demand points and the obnoxious facilities is at least L, and the distance between any pair of distinct obnoxious facilities is at least L'. (With the above notation their problem amounts to testing the feasibility of the multifacility problem, when $a_{i,j} = a$, for $i, j = 1, \ldots, k$, for some constant a. In particular, $L' = aL$.) For the rectilinear case they provide an $\mathrm{O}(n \log n)$ algorithm when $k = 2, 3$, and an $\mathrm{O}(n^{k-2} \log n)$ algorithm for the case $k \geqslant 4$. They also consider the optimization version, where L' is viewed as a fixed parameter, and the objective is to maximize the value of L such that the above decision problem is feasible. They present $\mathrm{O}(n \log^2 n)$ and $\mathrm{O}(n^{k-2} \log^2 n)$ algorithms for the cases $k = 2, 3$ and $k \geqslant 4$, respectively. (In the optimization version they consider only the unweighted case, but the weighted case can also be solved with the same complexity using the machinery developed in [32,13].) They also solve the decision problem corresponding to the Euclidean unweighted case when $k = 2$ in $\mathrm{O}(n \log n)$ time.

For the symmetric optimization problem, where L is viewed as a fixed parameter, and the objective is to maximize the value of L' such that the above decision problem is feasible, one can use the algorithms in [37], which are applicable to any norm and any value of k, to obtain $\frac{1}{2}$-approximation in polynomial time.

Ben-Moshe et al. [3] consider the following variant of the above model. The rectangular set S is replaced by $\{S_1, \ldots, S_m\}$, a set of $m = \mathrm{O}(n)$ translated copies of some axis-parallel rectangle. In their model, $L' = 0$, $w_i = 1$, for $i = 1, \ldots, n$, and the goal is to locate $\{x_1, \ldots, x_k\}$, such that $S_t \cap \{x_1, \ldots, x_k\}$ is nonempty for $t = 1, \ldots, m$, and the minimum distance between the demand points and the obnoxious facilities is at least L. They provide an $\mathrm{O}(n \log n)$ algorithm to resolve this variant of the decision problem when $k = 2$, for both, the rectilinear and the Euclidean distance functions.

The above multifacility models generalize classical packing problems, where the objective is to pack squares or discs into some prespecified compact planar set. These classical problems are known to be NP-complete, (see [20]). Note that unlike the classical packing models, where each neighborhood (e.g., square or disc) is required to be contained in the feasible set S, in the above obnoxious facility location problem only the centers of the neighborhoods, $\{x_1, \ldots, x_k\}$, must belong to S. Several studies have specifically addressed the problem of packing two Euclidean discs in a polygon. Bespamyatnikh [4] presents an $O(n \log^2 n)$ algorithm for packing two largest nonintersecting discs in a simple polygon S, (i.e., a polygon without holes). The algorithm is deterministic, and it is based on parametric search. Bose et al. [7] obtained a linear time algorithm for the case where the polygon is convex. Finally, Bose et al. [8] provide a simple $O(n \log n)$ randomized algorithm to solve this problem when S is a polygon, possibly with holes. We call this problem of packing two largest nonintersecting discs, the *2-disc packing problem.*

The 2-disc packing problems can be viewed as an unweighted case of our obnoxious facility location problem, where the set of demand points (existing facilities) is the continuum planar set consisting of all the boundary points of the set S. We focus on the more complicated weighted problem, where the given finite set of demand points P is not necessarily restricted to be in the set S. The methods used by Bespamyatnikh [4] and Bose et al. [8] rely heavily on the fact that the model is unweighted, and therefore are not applicable to our weighted problem. Specifically, the algorithms in [4,8] are based on finding the medial axis $M(S)$ of a polygon S which is the locus of all centers of discs that are contained in S and touch the boundary of S in two or more points. $M(S)$ is a portion of a Voronoi diagram, and it has a linear complexity. One can extend this concept to a weighted model, but the linear complexity of the respective weighted Voronoi diagram is not guaranteed anymore.

1.2. Our results

In this paper, we present $O(n \log^3 n)$ and $O(n \log^4 n)$ deterministic algorithms to solve the weighted 2-obnoxious facility location problem in the case where S is a convex compact polygon, for the rectilinear and Euclidean versions, respectively. The algorithm for the rectilinear norm is then extended to the case of a general compact polygonal domain, without increasing the complexity bound. The algorithms are based on the general parametric approach of Megiddo [31]. In Sections 2 and 3, we discuss the rectilinear and the Euclidean models, respectively, for the case where S is a convex compact polygon. In Section 4, we extend our algorithmic results for the rectilinear problem to the case when S is a general compact polygonal domain. The complexity remains $O(n \log^3 n)$.

We emphasize the main differences between our results and those in [8]. As explained above, the $O(n \log n)$ expected-time algorithm in [8] applies only to the unweighted case, where the feasible set is a polygon, possibly with holes. (They consider only the Euclidean model, but their approach seems to be extendable to the unweighted rectilinear case.) Our $O(n \log^3 n)$ algorithm for the weighted rectilinear case is deterministic, and it is applicable to more general regions. On the other hand, the randomized algorithm in [8] is quite simple, while our deterministic algorithm, which is based on the general parametric approach, is not that easy to implement.

1.3. Formal definitions and model presentation

For any planar set Y, we use $\delta(Y)$ and $CH(Y)$ to denote the boundary and the convex hull of Y, respectively.

Considering the above weighted 2-obnoxious facility location problem, to facilitate the discussion, we say that a positive scalar L is *feasible* if there exist x_1 and x_2 such that the triplet (x_1, x_2, L) satisfies the constraints of the above problem. The optimal solution value L^* is the largest feasible solution.

For any $L \geqslant 0$ define

$$Q(L) = \{x \in R^2 : d(x, p_i) \geqslant w_i L,\ i = 1, \ldots, n\}$$

and

$$S(L) = S \cap Q(L).$$

Then, a given positive L is feasible if and only if $S(L)$ contains two points x_1, x_2 with $d(x_1, x_2) \geqslant L$. L^* is then the largest value of L for which the diameter of $S(L)$ is at least L.

2. A parametric algorithm for the rectilinear case when S is convex

Since we deal with a planar problem, for convenience we replace the rectilinear norm L_1 by the equivalent maximum norm L_∞.

For the rectilinear case, we show how to solve the problem in (serial), subquadratic, $O(n \log^3 n)$ time, using the parametric approach of [31]. Specifically, we present an $O(\log n)$ time parallel algorithm with $O(n)$ processors, to test the feasibility of a given positive L.

In the rectilinear case, $Q(L)$ is the complement of the union of n open squares with edges parallel to the axes. (We have replaced the rectilinear norm by the equivalent maximum norm.) The boundary of $Q(L)$, $\delta(Q(L))$, consists of $O(n)$ edges and isolated points (see [34,27]).

$\delta(Q(L))$ can be constructed in $O(n \log n)$ serial time, and also in $O(\log n)$ parallel time, using $O(n)$ processors [25].

By definition, a given positive L is feasible if and only if there exist $x_1, x_2 \in S(L)$ such that $d(x_1, x_2) \geqslant L$. To verify feasibility we check whether the L_∞ diameter of $S(L)$ is at least L.

Due to the convexity of the distance function, the diameter of any compact planar set is attained at the extreme points of its convex hull. In particular, if $S(L)$ is a singleton its diameter is equal to zero. Otherwise the diameter of $S(L)$ is attained at two distinct extreme points of $CH(S(L))$, the convex hull of $S(L)$. The boundary of $S(L)$ is clearly piecewise linear with a finite number of corner points ("break points"). Hence, the extreme points of $CH(S(L))$ are also corner points of $\delta(S(L))$, the boundary of $S(L)$. Every corner point of $\delta(S(L))$ is either an extreme point of S, or an extreme point of one of the $O(n)$ edges of $\delta(Q(L))$, or an intersection point of an edge of $\delta(Q(L))$ with the boundary of S. (An isolated point of $\delta(Q(L))$ is viewed as a degenerate edge.) In particular, $\delta(S(L))$ has $O(n)$ corner points.

We now show how to identify all these corner points in parallel in $O(\log n)$ time, using $O(n)$ processors.

First, since S is fixed and independent of L, we assume that its extreme points are circularly presorted. Therefore, using binary search, for each open square, $\{x \in R^2 : d(x, p_i) < w_i L\}$, it takes $O(\log n)$ time to find the (at most four) circular sublists of extreme points of S which are in the open square. (By a circular sublist we refer to the list of extreme points on each one of the at most four connected piecewise linear arcs that we obtain on the boundary of S.) Hence, in $O(\log n)$ time with $O(n)$ processors, we can identify all the extreme points of S which are in $S(L)$. Since S is convex each edge of $\delta(Q(L))$ intersects the boundary of S in at most two points. Thus, in $O(\log n)$ time we can find the (at most) two corner points of $S(L)$ contributed by a given edge of $\delta(Q(L))$.

To summarize, all the $O(n)$ corner points of $\delta(S(L))$ can be found in parallel in $O(\log n)$ time, using $O(n)$ processors. The serial time is clearly $O(n \log n)$.

Let $V(L)$ denote the set of vertices, corner points, of $\delta(S(L))$. Then, the diameter of $S(L)$ is the largest dimension of the smallest closed rectangle with edges parallel to the axes, which contains $V(L)$. Hence, it will suffice to compute the largest and smallest horizontal and vertical coordinates of the $O(n)$ points of $V(L)$. The latter task can be achieved in $O(\log \log n)$ time in parallel, using only $O(n)$ processors [40]. The serial time is clearly $O(n)$.

Summarizing, we have given above an algorithm to test the feasibilty of a prescribed L in parallel, in $O(\log n)$ time, using $O(n)$ processors. The serial time is $O(n \log n)$. We now have all the necessary ingredients, and we can directly use the parametric approach in [31] to find L^*, the solution value to the rectilinear weighted 2-obnoxious facility location problem, in $O(n \log^3 n)$ (serial) time. This algorithm will also generate (x_1^*, x_2^*), an optimal location for the two obnoxious facilities. Alternatively, we can identify (x_1^*, x_2^*) by applying the above $O(n \log n)$ feasibility test routine to L^*.

Theorem 2.1. *Let S be a planar compact and convex polygon with $m = O(n)$ extreme points. Then, there is an $O(n \log^3 n)$ deterministic algorithm to solve the rectilinear 2-obnoxious facility problem in S.*

2.1. A binary search algorithm for rational data

Suppose that the input data (including the description of the facets of S), is rational, where each rational scalar is a ratio of two integers bounded above by an integer upper bound M. Then, in the rectilinear case L^* is rational with integer numerator and denominator bounded above by $O(M^{10})$. (The $O(M^{10})$ bound follows from the fact that the above problem is formulated in terms of five scalar variables: the four coordinates of x_1, x_2 and L. The optimal values of these five variables are defined by solving a subsystem of five linear equations with rational input data.) Hence, we can use the search over rationals approach [41] to find L^* in $O(n \log n \log M)$ time.

Theorem 2.2. *Let S be a planar compact and convex polygon with $m = O(n)$ extreme points. Suppose that the input data (including the description of the facets of S), is rational, where each rational scalar is a ratio of two integers bounded above by an integer upper bound M. Then, there is an $O(n \log n \log M)$ deterministic algorithm to solve the rectilinear 2-obnoxious facility problem in S.*

3. A parametric algorithm for the Euclidean case when S is convex

For the Euclidean case, we show how to solve the problem in serial subquadratic, $O(n \log^4 n)$ time, using the parametric approach of Megiddo [31]. Specifically, we present an $O(\log^2 n)$ time parallel algorithm with $O(n)$ processors, to test the feasibility of a prescribed positive L.

In the Euclidean case, $Q(L)$ is the complement of the union of n open discs. We next show that $S(L)$ can also be viewed as the complement of the union of $O(n)$ discs. The convex polygon S is assumed to have $m = O(n)$ facets. Each one of them can be viewed as a circle with infinite radius (centered "outside" S). Hence S is the complement of the union of m open discs. Moreover, $S(L)$ is the complement of the union of $n + m = O(n)$ open discs. From [27], we conclude that $\delta(S(L))$, the boundary of $S(L)$, consists of $O(n)$ circular arcs and isolated points, which we view as degenerate arcs. Following [20], we note that $\delta(S(L))$ can be constructed in parallel, in $O(\log^2 n)$ time using $O(n)$ processors. Also, the serial time to construct $\delta(S(L))$ is only $O(n \log n)$.

Consider $C_j(L)$, an arbitrary connected component of $\delta(S(L))$. We observe that $C_j(L)$ consists of "concave" circular arcs and has a finite number of corner points ("break points"). Moreover, the two-dimensional cell of $S(L)$ bounded by $C_j(L)$ is contained in the convex hull of the corner points of $C_j(L)$. Therefore, using the convexity of the distance function we conclude that when $S(L)$ is not a singleton, its diameter is attained at a pair of corner points of $\delta(S(L))$. We now use this observation to complete the description of the parallel algorithm to test the feasibility of L.

Suppose that $\delta(S(L))$ has been constructed by the above procedure. In particular, we now have $V(L)$, the set of all vertices, corner points, of $\delta(S(L))$. Clearly $|V(L)| = O(n)$.

We next use the algorithm in [23] to find the convex hull of $V(L)$ in parallel in $O(\log n)$ time using $O(n)$ processors. When this convex hull is specified in terms of its circular list of extreme points, for each extreme point it takes $O(\log n)$ time to identify the (extreme) point of the hull which is furthest away. In particular, it takes $O(\log n)$ time to compute in parallel the Euclidean diameter of $S(L)$, using only $O(n)$ processors [2]. The serial time is $O(n \log n)$.

Summarizing, we have given above an algorithm to test the feasibilty of a prescribed value of L in parallel, in $O(\log^2 n)$ time, using $O(n)$ processors. The serial time is $O(n \log n)$. We can now directly use the approach in [31] to find L^*, the solution value to the optimization problem for the Euclidean case, in $O(n \log^4 n)$ (serial) time. This algorithm will also generate (x_1^*, x_2^*), an optimal location for the two obnoxious facilities. Alternatively, we can identify (x_1^*, x_2^*) by applying the above $O(n \log n)$ feasibility test routine to L^*.

Theorem 3.1. *Let S be a planar compact and convex polygon with $m = O(n)$ extreme points. Then, there is an $O(n \log^4 n)$ deterministic algorithm to solve the Euclidean 2-obnoxious facility problem in S.*

4. Relaxing the convexity assumption on S in the rectilinear case

The above algorithms rely heavily on the fact that S is a compact convex polygon in the plane. Nevertheless, we can modify the algorithms for the rectilinear case, to handle general planar compact polygonal sets.

We use the following notions from [16,28]. A polygon is an open, connected and simply connected subset of the plane whose boundary can be partitioned into finitely many points (vertices) and open intervals (edges). A finite planar subdivision is a partition of the plane into polygonal regions, induced by a finite collection of finite intervals whose pairwise intersections are restricted to segment endpoints. Such a subdivision is indistinguishable from a straight-line embedding of a planar graph. Let G be such a graph (finite subdivision), with m vertices, $O(m)$

edges (one-dimensional cells) and $O(m)$ faces or regions (open two-dimensional cells). Only one of the two-dimensional cells is unbounded.

Given the subdivison G, and a finite subset α of its bounded regions we next define the planar compact polygonal set S_α as follows: S_α consists of the vertices of G, the points on the union of all edges of G and the union of the regions in α. (Note that for each α, the boundary of the compact polygon set S_α consists of all the vertices and edges of G.) The set of open two-dimensional cells that are not in α, and the isolated vertices of G, are viewed as the "holes" of S_α. A polygon with no holes is called *simple*. (Note that a simple polygon is usually defined by a simple nonintersecting closed and piecewise linear curve, i.e., the planar graph is a simple cycle. Our definition is slightly more general.)

We now consider the rectilinear version of the 2-obnoxious facility location problem, and show how to solve it for the case where the convex polygon is replaced by some general compact polygonal set S, defined by some finite subdivision G and a subset of its regions α. We assume that the number of vertices (corner points) of G is $m = O(n)$.

Extending the notation in Section 2, we set $S(L) = S \cap Q(L)$. From the above arguments we still conclude that the diameter of $S(L)$ is attained at the extreme points of $CH(S(L))$. Moreover, each extreme point of $CH(S(L))$ is either a corner point (vertex) of G, which is also in $Q(L)$, or an extreme point of the convex hull of the intersection of an edge of $\delta(Q(L))$ with S. Note that unlike the convex case, an edge of $\delta(Q(L))$ can intersect the boundary of S at more than two points. (In fact, an edge can intersect the boundary at $\Omega(n)$ points, even when S is a simple polygon.) Nevertheless, for our purposes it is sufficient to find only the at most two extreme points of the convex hull of the intersection of the edge with S. We note that for each edge the at most two extreme points can be found in $O(\log n)$ time by using data structures for point location and ray shooting, (see [36]).

Our algorithm is based on the parametric approach used in Section 2 with the following exception. Before we apply the approach based on the parallel algorithm, we first show how to identify all the vertices of G contained in $Q(L^*)$ without explicitly knowing L^*. To apply this step, we will need a serial algorithm to test the feasibility of a prescribed value of L.

4.1. A serial $O(n \log n)$ time feasibility test

First we compute $\delta(Q(L))$, the boundary of $Q(L)$, in $O(n \log n)$ time as in Section 2.

Secondly, when $\delta(Q(L))$ is given, the set of vertices of S contained in $Q(L)$ can be found in $O(n \log n)$ time by a sweep-line procedure as described in [29].

As noted above the relevant sets of pairs of points contributed by all edges of $\delta(Q(L))$ can be generated in $O(n \log n)$ time by using data structures for point location and ray shooting, (see [36]).

Thus, it takes $O(n \log n)$ time to generate a set $V(L) \subset S(L)$, with $|V(L)| = O(n)$, which contains all the extreme points of $CH(S(L))$.

Finally, the rectilinear diameter of the set $V(L)$ can be generated in $O(n)$ time, as described in Section 2.

To conclude, it takes $O(n \log n)$ serial time to test the feasibility of a prescribed value of L.

4.2. Finding the set of vertices of G contained in $Q(L^*)$

We denote the vertices of G by $\{v_1, \ldots, v_m\}$. Since $Q(L)$ is the complement of the union of n open squares with edges parallel to the axes, we conclude that there exist a point $p_i \in P$ and a vertex v_j of G, such that the set of vertices of G contained in $Q(L^*)$ is the same as the set of vertices of G contained in $Q(L_{i,j})$, where $L_{i,j} = w_i d(p_i, v_j)$. Define the set

$$X = \{L_{i,j} : L_{i,j} = w_i d(p_i, v_j),\\ i = 1, \ldots, n;\ j = 1, \ldots, m\}.$$

X is of $O(n^2)$ cardinality, and our goal now is to identify two consecutive elements in the sorted list of elements of X, say, L^1 and L^2, such that $L^1 \leqslant L^* < L^2$, and for each $L^1 \leqslant L < L^2$, the set of vertices of G contained in $Q(L)$ is the same as the set of vertices of G contained in $Q(L^*)$.

As in Section 2, for convenience, we replace the rectilinear norm L_1 by the equivalent maximum norm L_∞. We will search over a superset X', containing X, which is also of $O(n^2)$ cardinality. X' is defined as follows. For $i = 1, \ldots, n$, let $p_i = (p_i(1), p_i(2))$ denote the two coordinates of p_i. Similarly, for $j = 1, \ldots, m$, let $v_j = (v_j(1), v_j(2))$ denote the two coordinates of v_j.

Let $\{a_1, \ldots, a_m\}$ be the sorted list of the elements of the set $\{v_j(1)\}$ and let $\{b_1, \ldots, b_m\}$ be the sorted list of the elements of the set $\{v_j(2)\}$. Next, for each $i = 1, \ldots, n$, define the following four vectors (sorted columns):

$$L_i(1, +) = w_i(p_i(1) - a_1, \ldots, p_i(1) - a_m),$$

$$L_i(2, +) = w_i(p_i(2) - b_1, \ldots, p_i(2) - b_m),$$

and

$$L_i(1, -) = -L_i(1, +), L_i(2, -) = -L_i(2, +).$$

Finally, define the set X' to be the collection of the $4n$ sorted columns

$$\{L_i(1, +), L_i(1, -), L_i(2, +), L_i(2, -): i=1, \ldots, n\}.$$

It is clear that each element in the above set X appears as a component of at least one of these $4n$ columns. Hence, instead of searching over X, we can search over X'. Specifically, viewing X' as a matrix with $4n$ sorted columns, we apply the search procedures in [21,22] to find two consecutive elements in the sorted list of elements of X', say, L' and L'', such that $L' \leqslant L^* < L''$, and for each $L' \leqslant L < L''$, the set of vertices of G contained in $Q(L)$ is the same as the set of vertices of G contained in $Q(L^*)$. (Note that these search procedures do not require X' to have both sorted columns and sorted rows.) The total time needed to identify L' and L'' is $O(n \log n + T \log n)$ (serial) time, where T is the serial time to test the feasibility of a prescribed value of L. From above, we have $T = O(n \log n)$, and therefore the total time to identify the set of vertices of G contained in $Q(L^*)$ is $O(n \log^2 n)$. (Note that X' contains all the elements in X, and therefore we have $L^1 \leqslant L' \leqslant L^* < L'' \leqslant L^2$. Since, we already have identified the set of vertices of G contained in $Q(L^*)$, we do not need to find L^1 and L^2 explicitly. We can proceed with the interval $[L'; L'']$, which contains our target L^*.)

4.3. A parallel algorithm to find $V(L)$

Suppose now that L' and L'' have already been computed. In particular, $L^1 \leqslant L' \leqslant L'' < L^2$ and for each $L' \leqslant L < L''$ we already know the (fixed) set of vertices of G contained in $Q(L)$.

To find $V(L)$ it is now sufficient to find for each edge of $\delta(Q(L))$ the at most two extreme points of the convex hull of the intersection of the edge with S. Let $e(L)$ denote such an edge. As noted above, to find the at most two extreme points corresponding to $e(L)$ we can use the $O(\log n)$ time data structures for point location and ray shooting ([36,16,28] and also [10] which treats only simple polygons). The preprocessing time and space are $O(n \log n)$ and $O(n)$, respectively. (The preprocessing for the ray shooting is on the set S, which is independent of the parameter L. Hence, we do not need a parallel polylogarithmic algorithm for this phase. Also note that we need to apply the ray shooting only for rays parallel to the axes, and that can be used to simplify the preprocessing.)

To summarize, for any $L' \leqslant L < L''$, we can identify in parallel in $O(\log n)$ time, using only $O(n)$ processors, a set of points $V(L)$ in $S(L)$ that includes all the extreme points of $CH(S(L))$. Moreover, $|V(L)| = O(n)$.

We now proceed exactly as in Section 2. To conclude, we have presented an $O(n \log^3 n)$ algorithm to solve the rectilinear 2-obnoxious facility location problem even when S is a general compact planar polygonal set.

Theorem 4.1. *Let S be a planar compact polygonal set with $m = O(n)$ edges and corner points. Then, there is an $O(n \log^3 n)$ deterministic algorithm to solve the rectilinear 2-obnoxious facility problem in S.*

5. Final comments

An improvement in the complexity of the algorithm to solve the rectilinear case can possibly be achieved as follows: The serial time for testing feasibility of a prescribed value of L is $O(n \log n)$. Hence, if we can identify apriori, a "well structured" set containing L^*, we might be able to apply efficient search procedures, as in [32,13], to find L^* in $O(n \log^2 n)$ time.

For example, consider the one-dimensional version of the 2-obnoxious facility location problem, and suppose without loss of generality that S is the convex hull of P and $p_1 \leqslant p_2 \leqslant \cdots \leqslant p_n$. Then for some pair $i, j = 1, \ldots, n$, L^* is of the form $(p_i - p_j)/(w'_i + w'_j)$, where $w'_i = w_i + \frac{1}{2}$, $w'_j = w_j + \frac{1}{2}$; (L^* satisfies the bottleneck equation $L^* w_i + L^* + L^* w_j = |p_i - p_j|$), or L^* is of the form $(p_i - p_j)/(w_i + w_j)$ (L^* satisfies the bottleneck equation $L^* w_i + L^* w_j = |p_i - p_j|$).

Thus, L^* is an element in the set $X^1 \cup X^2$, where

$$X^1 = \{(p_i - p_j)/(w_i' + w_j') : i, j = 1, \ldots, n\},$$

and

$$X^2 = \{(p_i - p_j)/(w_i + w_j) : i, j = 1, \ldots, n\}.$$

In the one-dimensional case testing for feasibility can clearly be done in $O(n)$ time. Therefore, by using the search procedure in [32] with the modification in [13], L^* can be found in $O(n \log n)$ time.

An interesting topic to investigate is the extension of the above results to the case of 3 or any fixed number of new facilities. In particular, we suspect that the rectilinear weighted 3-obnoxious facility location problem is solvable in subquadratic time. As noted above, for the case of 3 new facilities, a partial answer is given in [26]. They consider the case where S is a rectangular planar region, and present an $O(n \log^2 n)$ algorithm for maximizing the minimum distance between the existing facilities and the new facilities, when there is a prescribed lower bound L' on the distances between the new facilities.

References

[1] G. Appa, I. Giannikos, Locating two obnoxious facilities in the plane under the rectilinear metric, LSE OR Working Paper LSE 04-69, London School of Economics, 2004, submitted for publication.

[2] M.J. Atallah, S.R. Kosaraju, An efficient parallel algorithm for the row minima of a totally monotone matrix, J. Algorithms 13 (1992) 394–413.

[3] B. Ben-Moshe, M.J. Katz, M. Segal, Obnoxious facility location: complete service with minimal harm, Internat. J. Comput. Geom. Appl. 10 (2000) 581–592.

[4] S. Bespamyatnikh, Packing two disks in a polygon, Comput. Geom. 23 (2002) 31–42.

[5] S. Bespamyatnikh, K. Kedem, M. Segal, A. Tamir, Optimal facility location under various distance functions, Internat. J. Comput. Geom. Appl. 10 (2000) 523–534.

[6] B. Bhattacharya, H. Elgindy, An efficient algorithm for an intersection problem and an application, Technical Report 86-25, Department of Computer and Information Sciences, University of Pennsylvania, 1986.

[7] P. Bose, J. Czyzowicz, E. Kranakis, Algorithms for packing two circles in a convex polygon, Proceedings of Japan Conference on Discrete and Computational Geometry (JCDCG'98) pp. 93–103.

[8] P. Bose, P. Morin, A. Vigneron, Packing two disks into a polygonal environment, J. Discrete Algorithms 2 (2004) 373–380.

[9] J. Brimberg, A. Mehrez, Multifacility location using a maximin criterion and rectangular distances, Location Sci. 2 (1994) 11–19.

[10] B. Chazelle, L. Guibas, Visibility and intersection problems in plane geometry, Discrete Comput. Geom. 4 (1989) 557–581.

[11] C.K. Chen, R.S. Garfinkel, The generalized diameter of a graph, Networks 12 (1982) 335–340.

[12] R.L. Church, R.S. Garfinkel, Locating an obnoxious facility on a network, Transport. Sci. 12 (1978) 107–118.

[13] R. Cole, Slowing down sorting networks to obtain faster sorting algorithms, J. Assoc. Comput. Mach. 34 (1987) 200–208.

[14] Z. Drezner, G. Wesolowsky, A maximin location problem with maximum distance constraints, IEE Trans. 12 (1980) 249–252.

[15] Z. Drezner, G. Wesolowsky, The location of an obnoxious facility with rectangular distances, J. Regional Sci. 23 (1983) 241–248.

[16] H. Edelsbrunner, L.J. Guibas, J. Stolfy, Optimal point location in a monotone subdivision, SIAM J. Comput. 15 (1986) 317–340.

[17] E. Erkut, The discrete p-dispersion problem, European J. Oper. Res. 46 (1990) 48–60.

[18] E. Erkut, S. Neuman, Analytical models for locating undesirable facilities, European J. Oper. Res. 40 (1989) 275–291.

[19] F. Follert, E. Schomer, J. Sellen, Subquadratic algorithms for the weighted maximin facility location problem, Technical Report, Universitat des Saarlands, Germany; Proceedings of the Seventh Canadian Conference on Computational Geometry, 1995, pp. 1–6.

[20] R.J. Fowler, M.S. Paterson, S.L. Tanimoto, Optimal packing and covering in the plane are NP-complete, Inform. Process. Lett. 12 (1981) 133–137.

[21] G.N. Frederickson, D.B. Johnson, The complexity of selection and ranking in $X + Y$ and matrices with sorted columns, J. Comput. System Sci. 24 (1982) 197–208.

[22] G.N. Frederickson, D.B. Johnson, Finding kth paths and p-centers by generating and searching good data structures, J. Algorithms 4 (1983) 61–80.

[23] M.R. Ghouse, M.T. Goodrich, Fast randomized parallel methods for planar convex hull construction, Comput. Geom. Theory Appl. 7 (1997) 219–235.

[24] I. Giannikos, Optimal location of single and multiple obnoxious facilities: algorithms for the MAXIMIN criterion under different norms, Ph.D. Thesis, London School of Economics and Political Sciences, 1993.

[25] M.T. Goodrich, M.R. Ghouse, J. Bright, Sweep methods for parallel computational geometry, Algorithmica 15 (1996) 126–153.

[26] M.J. Katz, K. Kedem, M. Segal, Improved algorithms for placing undesirable facilities, Comput. Oper. Res. 29 (2002) 1859–1872.

[27] K. Kedem, R. Livne, J. Pach, M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, Discrete Comput. Geom. 1 (1986) 59–71.

[28] D.G. Kirkpatrick, Optimal search in planar subdivisions, SIAM J. Comput. 12 (1983) 28–35.

[29] Y. Konforti, A. Tamir, The single facility location problem with minimum distance constraints, Location Sci. 5 (1997) 147–163.

[30] M.J. Kuby, Programming models for facility dispersion: the p-dispersion and maximum dispersion problems, Geographical Anal. 19 (1987) 315–329.

[31] N. Megiddo, Applying parallel computation algorithms in the design of serial algorithms, J. Assoc. Comput. Mach. 30 (1983) 852–865.

[32] N. Megiddo, A. Tamir, New results on the complexity of p-center problems, SIAM J. Comput. 12 (1983) 751–758.

[33] E. Melachrinoudis, J.M. Smith, An $O(mn^2)$ algorithm for the maximin problem in E^2, Oper. Res. Lett. 18 (1995) 25–30.

[34] F.P. Preparata, M.I. Shamos, Computational geometry: An Introduction, Springer, Berlin, New York, 1985.

[35] C.P. Ranagan, R. Govindan, An $O(n \log n)$ algorithm for a maximin location problem, Discrete Appl. Math. 36 (1992) 203–205.

[36] N. Sarnak, R.E. Tarjan, Planar point location using persistent search trees, Comm. Assoc. Comput. Mach. 29 (1986) 669–679.

[37] A. Tamir, Obnoxious facility location on graphs, SIAM J. Discrete Math. 4 (1991) 550–567.

[38] S.S. Ting, Obnoxious facility location problems on networks, Ph.D. Dissertation, The Johns Hopkins University, 1988.

[39] G. Toussaint, Computing largest empty circles with location constraints, Internat. J. Comput. Inform. Sci. 12 (1983) 347–357.

[40] L.S. Valiant, Parallelism in comparison problems, SIAM J. Comput. 4 (1975) 348–355.

[41] E. Zemel, On search over rationals, Oper. Res. Lett. 1 (1981) 34–38.