Improved Algorithms for Placing Undesirable Facilities *

Matthew J. Katz, Klara Kedem and Michael Segal

Department of Mathematics and Computer Science
Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

July 8, 1999

Abstract

We improve a number of existing algorithms for determining the location of one or more *undesirable* facilities amidst a set P of n demand points, under various constraints and distance functions. We assume that the demand points reside within some given bounded region R. Applying concepts and techniques from Computational Geometry, we provide efficient algorithms for the following problems:

1. **Brimberg Mehrez '94:** Locate k undesirable facilities within R under the constraints that the smallest distance between each demand point and the facilities is greater than a given r, and the distance between any two facilities is greater than a given d. Under the L_∞ (L_1) norm we present efficient algorithms for any
d under the L_2 norm we can locate efficiently two such facilities. In all cases
sumed to be an axis-parallel rectangle.

er Wesolowsky '94: Given a set of weighted demand points contained in
lel rectangular region (resp. circular region) R, and given a smaller
l rectangle (resp. circle) r , locate r within R such that the sum of
the demand points in r is minimized.

on

ns deal with *undesirable* or *obnoxious* facilities [BKS1, BKS2, BM
ility is called undesirable or obnoxious if it may pose a danger to the
ving nearby, may have an adverse effect on property values, or may cause lower
life through pollution. Examples of obnoxious facilities are nuclear power plants,
rbage dump sites, mega-airports, and chemical plants.

*Work by M. Katz and K. Kedem has been supported by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities. K. Kedem has also been supported by the U.S.-Israeli Binational Science Foundation, and by the Mary Upson Award, College of Engineering, Cornell University.

In this paper we consider a number of problems, all of which are concerned with locating an undesirable facility (or a number of facilities) amidst demand points, under various distance functions and varying constraints. We survey previously known algorithms for thes
problems, and propose more efficient algorithms which are based on concepts and techni
from Computational Geometry.

Problem 1: Maxmin multi-facility location. In their paper
using a maxmin criterion and rectangular distances”, Brimber
maxmin multiple facilities location problem, under the L
as follows. Given a set $P = \{p_1, \ldots, p_n\}$ of n points i
a distance r and another distance d, locate k un
that the smallest distance between each dem
distance between any pair of facilit
bound algorithm, and their alg
in this paper run in time $O(n$
algorithms can be extended i
handle different separatio
we show that for the

Problem 2:

there exist a set of k site locations in V with pairwise distance at least d?" If there is one
we report the locations. The associated optimization problem will output the largest square
size r^* for which the answer for the decision problem is "yes". We will describe it to
the end of this section.

We solve several variants of the decision problem. Under the L_∞ norm, we fi
$O(n \log n)$-time algorithms for $k = 2, 3$, and then present a general scheme for
of k that allows us to obtain efficient algorithms that run in time $O(n^{k-}$
significantly improving the $O(n^{2k})$ (for any $k \geq 1$) solution proposed
Mehrez [BM].

We also present an algorithm that locates two obnoxious facilities,
$O(n \log n)$ time. The space requirements of our algorithms are consider
[BM]. Specifically, the algorithms in [BM] require $O(n^2)$ space, wh
of our algorithms vary between $O(n)$ and $O(n \log n)$.

$k = 2$. It is well known that the combinatorial complexity of
n squares is linear in n [PS]. In other words, the bounda
vertices and edges and can be computed in time $O(n \log n$
(see [Me]). Thus the boundary of U (and therefore
$O(n \log n)$ and space $O(n)$. The problem of locating tw
at least d, under the L_∞ norm, boils down to find
V with largest and smallest x-coordinate, an
y-coordinate. Since we can choose these
is achieved in time $O(n)$, by going o

Theorem 2.1 *The two-fa*
space.

$k = 3$. As in the previous ca

Claim 2.2 *If there*
strains above, th
of the bounda

Proof. Assu
and c is
respe

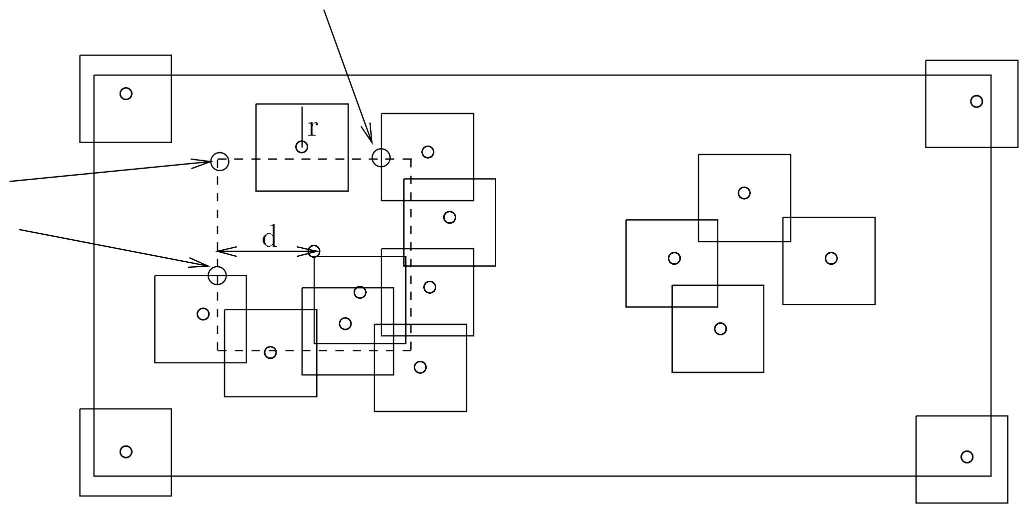

Figure 1: The extremal points of B

This approach might naively lead to a roughly quadratic runtime, since Q might intersect the boundary of V in $O(n)$ points, and we need to compute for each vertex $v \in V$ the region $V - Q$ and identify two locations in it. However, we observe that if one of the other two facilities v_1 and v_2 is located on a vertex of $B = \partial(V - Q) \cap \partial Q$, then it can be dragged to an extremal point of B, namely, the highest or lowest point in B (alternatively, leftmost or rightmost point in B), while maintaining the distances between the three facilities gr than d (see Figure 2). This is because the first chosen facility v is certainly wit d from the dragged v_1 and also the distance between v_1 and v_2 remains gre to d since we are working with L_∞ (and, thus, there must be some directic (alternatively, left or right) for which the distance between v_1 and v_2 w

The above process can be performed more efficiently by preproce the boundary of V for orthogonal range searching with the fra [BKOS, CH1, CH2]. For each boundary vertex v, perform a size $2d$ centered at v, and obtain the vertices of the boundar as a collection of $O(\log n)$ canonical sets.

In addition we need to determine for each of the vertices in B. For this, we preprocess the horizont of U together with the two horizontal (resp. logarithmic time vertical (resp. horiz and does not lie outside of R, t we perform an orthogonal ray containing e to detect the e

We proceed by solving logarithmic time. As d canonical sets whe y). The fartl

We also consider the at most 8 extreme points that were computed on the boundary of Q.

Theorem 2.3 *The three-facility location problem can be solved in $O(n \log n)$ time and $O(n \log n)$ space.*

$k \geq 4$. In this case we claim that

Lemma 2.4 *At least one of the sites is a vertex of $R-U$.*

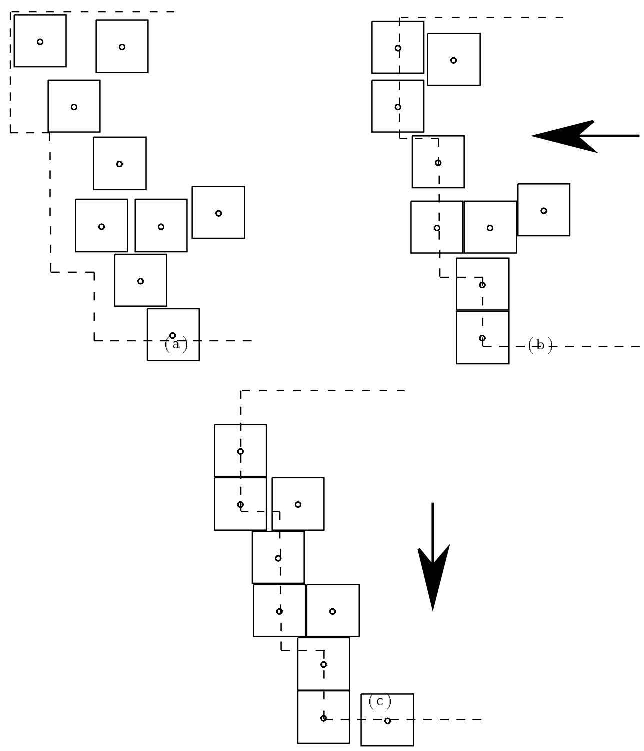

Figure 2: The movement of the squares

Proof. Let us consider the rectilinear free space $V = R - U$. Assume that there is an initial
positioning for k locations such that none of them is on a vertex of V. Our approach i
move the facilities, maintaining the $\geq d$ distance requirement, such that at least o
will be on a vertex. Denote the facilities by $F = \{f_1, \ldots, f_k\}$. About each fac
an axis-parallel square c_i with side size d (f_i being the square's center).
initial positioning the squares do not intersect. We push all the
as possible, so that they still do not intersect, and their
motion. If at some point during this stage, one of t
then we are done. At the end of this stage, t
of the boundary of V (if there are se
Next we push them as much down a

and not letting the facilities to penetrate into U nor leave R, and stopping if at some point a facility passes through a vertex of V. At the end of this stage, the bottommost facility f_j must lie on a horizontal edge of the boundary of V (and if there are several bottommost facilities f_j is the leftmost among them). See Figure 2.

Assuming we have not stopped with a facility on a vertex, then we know that since otherwise the corresponding facility lies on a vertex and we would have s now check whether we can slide the square c_j to the left, under the same l that its center f_j coincides with a vertex of V. If we can, then we are we proceed as follows. Consider the south west quarter plane defin the bottom edge of c_i and the line through the left edge of c_j. least one square that is fully contained in this quadrant square x blocking c_i from below, and there exists a if $x = y$ then this square is such a square. Othe c_j, then x is such a square. And if x do is necessarily below the bottom is fully contained in the above contained in this quadran and repeat the whol fear that th only le a

2.1 The optimization scheme

In order to find the largest value $r*$ for which there still exists a solution to the k-facility
location problem (keeping d fixed), we employ the technique of Frederickson and Johnson
[FJ], as has been done before (see, e.g., [GKS] and many others). Each pair p_i, p_j of demand
points determines eight critical values, four for each dimension. We list the critical value
the x difference d_x between p_i and p_j: (i) $d_x/2$, (ii) d_x, (iii) $(d_x - d)/2$, and (iv) $(d - d_x)$
addition, each demand point p determines four critical values; the two horizonta
between p and the boundary of R and the two vertical distances between p and th
of R.

We can represent all these distances as a constant collection of sorted m
perform a binary search on these values using the decision algorithm
was shown in [FJ], the above scheme adds a multiplicative $O(\log$
time of the decision algorithm.

3 Min sum coverage

Let P be a set of n points within an axis-parallel rectangle R. Let
rectangle which is smaller than R (both in width and in heigh
within R such that the total number of points lying in r i
Consider the optimal location of r. If the bottom
P, and does not lie on the bottom edge of R, then
these conditions does hold, without changing the
below and closed from above).

Below we show how to find the optimal locat
on the bottom edge of R or on an horizontal
to the above observation, this location
and by d its height. The solution
The segment that consisting of
of length c, centered at the p
Initially the tree is em
starting with the lowe
the upper side of
the total weight
the elementa
smaller b
the

We now start moving the slab upwards until the first of the following events occurs: either
the (i) upper or (ii) lower side of the slab hits a point of P, or the upper side of the slab
hits the top of R. At event (i) we insert into T the segment corresponding to the point of
just encountered, updating the weights in the nodes on the paths to the root. At eve
we delete the point and update the weights accordingly. Event (iii) terminate
We choose the event where the minimal weight was achieved at the root to be
determines the location of r. Clearly handling an event of type (i) or (i
time. Thus we obtain the following theorem.

3.1 A lower bound

We obtain an $\Omega(n \log n)$ lower bound for the above minsum coverage probl
1-dimensional case. Bespamyatnikh et al. [BKS2] obtained an $\Omega(n \log n$
the following problem: Given n positive real numbers and a number γ,
there exist two consecutive numbers in the sequence $a_1, \ldots, a_n$ obtained
numbers, such that their difference is greater than γ. Our reduct
be the segment $[a_1, a_n]$, and let r be a segment of length γ. Ev
a 1-dimensional point with weight 1. If we can place r within
the points lying in r is 0, then two such numbers exist.
pair. Thus we conclude that

Theorem 3.1 *Given a set P of n points within*
axis-parallel rectangle r which is smaller than
$\Theta(n \log n)$ time, such that the total number of p

References

[Bajaj] C. Bajaj, "Geometric optimization an
Tech. Report TR-84-629, Cornell Unive

[BKS1] B. Ben-Moshe, M.J. Katz and M.
service with minimal harm", *Proc.*
1999, to appear.

[BKOS] M. de Berg, M. van Kreveld, M. O
Geometry: Algorithms and Applications'

[BKS2] S. Bespamyatnikh, K. Kedem and M.
ious Distance Functions", *Proc. Work*
appear.

[BM] J. Brimberg and A. Mehrez, "Multi-F
Rectangular Obstacles", *Location Sc*

[BW] J. Brimberg and G.O. Wesolowsky,
Minimum Distance Constraints", *Loc*

[CH1] B. Chazelle, "Filtering search: A new approach to query-answering", *SIAM J. Comput.*, 15, pp. 703–724, 1986.

[CH2] B. Chazelle, "A functional approach to data structures and its use in multidimensional searching", *SIAM J. Comput.*, 17, pp. 427–462, 1988.

[DW] Z. Drezner and G.O. Wesolowsky, "Finding the Circle or Rectangle Containing the Minimum Weight of Points", *Location Science*, Vol. 2(2), pp. 83–90, 1994.

[FJ] G.N. Frederickson and D.B. Johnson, "Generalized selection and ranking: sorted matrices", *SIAM J. Comput.*, 13, pp. 14–30, 1984.

[GKS] A. Glozman, K. Kedem, G. Shpitalnik, "Efficient solution of the two-line cent problem and other geometric problems via sorted matrices", *Computational Geomet Theory and Applications*, 11, pp. 17–28, 1998.

[KT] Y. Konforty and A. Tamir, "The single facility location problem with minimum tance constraints", *Location Science*, Vol. 5(3), pp. 147–163, 1997.

[Me] K. Mehlhorn, "Multi-dimensional Searching and Computational Geometr *Structures and Algorithms*, Vol. 3, Springer-Verlag, 1984.

[PS] F.P. Preparata and M.I. Shamos, "Computational Geometry: An Intr Springer-Verlag, New York, NY, 1985.