Rectilinear 2-center problems

Sergei Bespamyatnikh and David Kirkpatrick
besp,kirk@cs.ubc.ca
Department of Computer Science
University of British Columbia
Vancouver BC, Canada V6T 1Z4

Abstract

We present efficient algorithms for two problems of facility location. In both problems we want to optimize the location of two facilities with respect to n given sites. The first problem, the continuous version, has no restrictions for facility locations but in the second one, the discrete version, facilities are chosen from a specified set of possible locations. We consider the rectilinear metric L_∞ and arbitrary dimension d and determine the locations that minimize, over all sites, the maximum distance to the closest facility. The algorithms for the continuous and discrete versions take $O(n)$ and $O(n \log^{d-2} n \log\log n + n \log n)$ running time respectively.

1 Introduction

We address the well studied 2-center problem [2, 4, 16, 14, 21, 20, 25] in facility location [7, 24]. We are given a set of n points representing customers and it is desired to locate 2 facilities in the plane to minimize the largest distance from a customer to its nearest facility. The minimum distance is called the *2-radius* and we denote it by ρ^*. For a long time the best algorithms for 2-center problem in the Euclidean plane had time bounds of the form $O(n^2 \log^c n)$ [1, 6, 15, 18]. In a recent breakthrough, Sharir [25] greatly improved time bound to $O(n \log^9 n)$. The algorithm of Sharir uses parametric searching. Eppstein found simpler algorithm with randomized expected
$O(n \log^2 n)$ time. Several papers consider the 2-center problem under rectilinear L_1, L_∞ metrics. Drezner [5] found an algorithm for the rectilinear 2-center problem in the plane with linear running time. Ko and Ching [20] gave a linear-time algorithms for a weighted version of the rectilinear 2-center problem in higher (and fixed) dimensions.

In the discrete version of the 2-center problem the possible locations of facilities are restricted to a set of points.

Discrete 2-center problem. Given a set S of n points that represent sites to be served and a set F of m points that represent potential sites of facilities. Locate 2 facilities such that maximum distance from a customer point to the closest facility is minimal. In other words the problem is to cover customer points by the union of two squares (cubes in higher dimensions) of minimum size, where the square centers are constrained to F.

The discrete 2-center problem in the Euclidean plane appears to be more difficult than the standard 2-center problem. Agarwal *et al.* [2] presents $O(n^{4/3} \log^5 n)$ algorithm for the case $S = F$. Very recently [17, 3] the discrete 2-center problem had been studied under rectilinear metric. Katz *et al.* [17] gave $O(n \log^2 n)$ algorithm for the case $S = F$. Bespamyatnikh and Segal [3] improved the running time to $O((n + m) \log(n + m))$ using an $O(n + m)$ decision algorithm.

In this paper we focus on the 2-center problems, both continuous and discrete, in higher dimensions under the rectilinear metric L_∞. We give a simple linear-time algorithm for the rectilinear 2-center problem with lower dependence on the dimension than the algorithm of Ko and Ching [20].

For the discrete rectilinear 2-center problem we present $O(N \log^{d-2} N \log\log N + N \log N)$ algorithm where $N = \max(n, m)$. We extend it to solve a general restricted problem where facility locations are restricted by axis-parallel segments or even polytopes with axis-parallel faces.

2 Notation

We assume that the dimension d is fixed througout the paper except Section 3. For a point p in d-dimensional space, i-th coordinate of p is denoted by p_i, i.e. $p = (p_1, \ldots, p_d)$. The l_∞-distance between points a and b is denoted by $d_\infty(a, b)$.

A point p *dominates* the point q if $p_i \geq q_i$ for all i. A point $p \in A$ is said to be a *maximal element* in A if p is dominated by the only point p. Similarly the *minimal element* in A dominates only itself. For a set $A \subset R^d$ the set of *maxima* (*minima*) is the set of the maximal (minimal) elements in A.

The cube of side 2ρ with center $p \in F$ is the set $[p_1 - \rho, p_1 + \rho] \times \ldots \times [p_d - \rho, p_d + \rho]$. We will refer to ρ as the size of the cube.

For a set $A \subset R^d$, the *bounding box* of A, denoted by $bb(A)$, is the smallest axis-parallel box that contains A. The bounding box of A is determined by $2d$ coordinates, two from each axis $i = 1, \ldots, d$, the leftmost coordinate $l_i(A)$ and the rightmost one $r_i(A)$. In other words

$$bb(A) = [l_1(A), r_1(A)] \times \ldots \times [l_d(A), r_d(A)].$$

We call the point $(l_1(A), \ldots, l_d(A))$ as lowest point of the bounding box $bb(A)$. The *diagonal* of $bb(A)$ is the segment whose endpoints are vertices of $bb(A)$ and whose interior lies in the interior of $bb(A)$.

3 Rectilinear 2-center Algorithm

Ko and Ching [20] considered the weighted version of the rectilinear 2-center problem where each customer point p is assigned a weight $w(p)$ and the distance to a facility is measured as $w(p)d_\infty(p, q)$. Ko and Ching gave an $O(d^2 n + d^2 \log^* d)$ algorithm using the prune and search technique [22] to solve a pseudo-2-center problem. We present a simpler algorithm for the unweighted version with $O(nd \log d)$ running time.

Lemma 1 *If two cubes of equal size cover a set S of points in R^d and no one cube is a cover by itself, then they contain two diametral vertices of the bounding box of S.*

Proof: Let $A = a_1 \times \ldots \times a_d$ and $B = b_1 \times \ldots \times b_d$ be two cubes whose union contains S. Recall that the bounding box of S is defined by $2d$ coordinates $l_1(S), \ldots, l_d(S)$ and $r_1(S), \ldots, r_d(S)$. Consider the projections of the two cubes and the bounding box into i-th coordinate axis. Clearly the union of two segments a_i and b_i contains the points $l_i(S)$ and $r_i(S)$. We can say that these points are covered by different segments. If one segment, say a_i, contains both points then other segment b_i (of the same side) contains at least one of the points, say $l_i(S)$, and we assign points $l_i(S), r_i(S)$ to the segments b_i and a_i, respectively.

Each cube is assigned d coordinates that define a vertex of the bounding box of S. These 2 vertices form a diagonal of $bb(S)$. ∎

A linear-time algorithm for the rectilinear 2-center problem can be obtained using Lemma 1. Suppose that two diametrical vertices of $bb(S)$ are chosen. We can assume that they are corresponding vertices of the cubes (they can be moved to capture the corners). Now a customer point c is contained in a cube of size ρ if and only if the distance between c and corresponding corner of the cube is at most 2ρ.

First the algorithm computes the bounding box of S. For all diagonals of $bb(S)$ we do the following. For each point of S we compute the closest distance to the endpoints of the diagonal. Take the largest distance over all points of S. The minimum distance over all diagonals is $2\rho^*$ and corresponding diagonal defines the cubes and the facility locations. The running time of the algorithm is $O(2^d dn)$ because there is $O(2^{d-1})$ diagonals. To improve exponential dependence on d we use the observation that the 2-radius is determined by the distance from a customer point to a vertex of $bb(S)$.

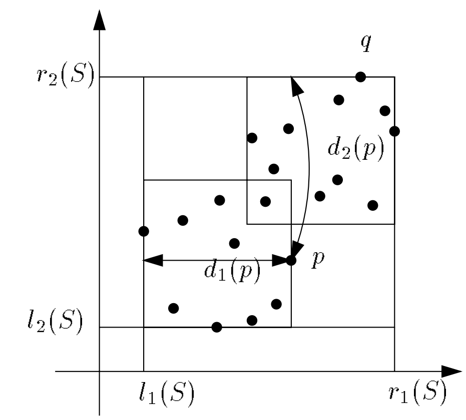

Figure 1: 2-radius is equal to $d_1(p)/2$

Theorem 2 *The rectilinear 2-center problem in $R^d, d \geq 1$ can be solved in time $O(nd \log d)$.*

Proof: It is suffice to find a diagonal D that defines the cubes of size ρ^*.

Consider the bounding box $bb(S) = [l_1(S), r_1(S)] \times \ldots \times [l_d(S), r_d(S)]$. For a point $p \in S$ we define the distance

$$d_i(p) = \max\{p_i - l_i(S), r_i(S) - p_i\}$$

and the side $s_i(p) = left$ if $d_i(p) = p_i - l_i(S)$, otherwise $s_i(p) = right$. The distances $d_i(p)$ play important role because, in the 2-dimensional case, if two cubes intersect and a point p defines the 2-radius then it is equal to $\min\{d_1(p)/2, d_2(p)/2\}$, see Fig. 1. On the other hand, the point p maximizing the objective $\min\{d_1(p)/2, d_2(p)/2\}$ defines the diagonal corresponding 2-radius. In Fig. 1 q is such a point and common point (the lowest point of $bb(S)$) of two sides of the $bb(S)$ corresponding to $d_1(q)$ and $d_2(q)$ is an endpoint of the desired diagonal D. We choose the diagonal to avoid having that the point q on the boundary of the cubes.

First the algorithm computes all distances $d_i(p)$ and, for each point p, it sorts the distances $d_i(p)$. This takes $O(nd \log d)$ time. We also keep a copy of d_1 and s_1 in separate arrays d'_1 and s'_1.

To specify the diagonal D in higher dimensions we use d (pairwise nonparallel) faces of the bounding box whose intersection is an endpoint of D. For each dimension i there is a face orthogonal to it, left $x_i = l_i(S)$ or right $x_i = r_i(S)$. We can choose the first face in the hyperplane $x_1 = l_1(S)$ because it must contain exactly one endpoint of D. The key idea is that the point with maximum $\min\{d'_1(p), \max\{d_i(p), i \neq 1\}\}$ defines the second face, i.e. it is $x_i = l_i(S)$ or $x_i = r_i(S)$ depending on the sides $s'_1(d), s_i(p)$. If they are the same $s'_1(d) = s_i(p)$ then the sides of the faces orthogonal to the 1-st and i-th coordinate axis are the same and vice versa. The computation of $d_i(p)$ can be done in linear time if the distances are presorted.

To find next face we reduce one dimension in the d-dimensional problem. We delete the distances $d_i(q)$ for all $q \in S$ and keep track of them by setting $d'_1(q) = \max\{d'_1(q), d_i(q)\}$, and if $d'_1(q) < d_i(q)$ then the side $s'_1(q)$ is changed to $s'_i(q)$. The deletion of the distances $d_i(q)$ and the update of $d'_1(q)$ and $s'_1(q)$ takes $O(n)$ time over all points q.

Now we prove the correctness of the algorithm. In k-th step we have k faces to define D. They actually define a diagonal in the k-dimensional space formed by the coordinates orthogonal to the faces. The projection of the points S gives the k-dimensional 2-center problem. Suppose the algorithm is not correct and let k be the smallest dimension such that the diagonal in k dimensions defines cubes of size greater than ρ^*. We can assume for simplicity that the i-th face is orthogonal to the i-th coordinate axis. If we put 2 cubes of size ρ^* with corresponding vertices of the diagonal then some point $q \in S$ would be outside both cubes. Hence there is an index j such that in the projection plane π formed by the k-th and j-th coordinates the point q lies outside the squares which are the projections of the cubes. Hence $d_j(q) > 2\rho^*$ and $d_k(q) > 2\rho^*$. In the k-th step $d'_1(q) > 2\rho^*$. Therefore $d_k(p)$ and $d'_1(p)$ are greater than $2\rho^*$ because the algorithm choose the point p. If we change the selection of the k-th face the point p would be at distance greater than $2\rho^*$ from each endpoint of the diagonal. It contradicts the definition of ρ^*. ∎

The algorithm of Ko and Ching reduces the weighted rectilinear 2-center problem to a maximum spanning tree computation in a complete graph with d vertices. The proof above can be interpreted as a similar maximum spanning tree computation using Prim's algorithm [23]. We avoid computing a graph because the weight of the edge (i, j) is actually the maximum of $\min\{d_i(p), d_j(p)\}$ over all points $p \in S$.

4 Discrete rectilinear 2-center algorithm

Finding the two 2-cubes in the discrete rectilinear 2-center problem can be considered as an optimization problem where the side of the two cubes is minimizing. To solve it our algorithm uses a subroutine to solve the following decision problem.

Decision problem. Given a set S of n customer points, a set F of m points in R^d, and a parameter ρ, determine whether the customer points can be covered by two cubes of side 2ρ with centers in F.

4.1 Optimization Algorithm

We apply the parametric search technique with sorted matrices [9, 11, 13]. Suppose we seek an optimum value ρ^* of a parameter ρ and we have a decision algorithm which, for any particular value ρ, decides whether ρ is equal to, smaller than, or larger than the desired value ρ^*. An $n_1 \times n_2$ matrix M is a *sorted matrix* if each row and each column of M is in nondecreasing order. The elements of M represent the possible values of the optimization problem. Let T denote the running time of the algorithm for decision problem. Frederickson and Johnson [8, 10] show that the runtime consumed by optimization algorithm is $O(T \log n_1 + n_2 \log(2n_1/n_2))$ where $n_2 \leq n_1$.

Consider the two desired cubes with centers in F. The size of the cubes cannot be decreased. Hence the boundary of the union of the cubes contains at least one customer point. In other words the size of the cube is realized by a distance between points of S and F. In metric L_∞ the distance between points is the minimum distance between corresponding coordinates. For the k-th coordinate $k = 1, \ldots, d$, we can represent the corresponding distances by $m \times n$ matrix M. Let $s_1, \ldots, s_m$ and $f_1, \ldots, f_n$ be the sorted lists of k-th coordinates of the customer points of S and the points of F. Setting $M[i, j] = f_j - s_i$ gives us monotone matrix M. Only drawback of M is that not all elements of the matrix represent distances along k-th coordinate (some elements can be negative). To fix it we split M into two matrices M^+ and M^- containing positive and negative elements of M (in the positive matrix M^+ the order of rows and columns must be changed to keep nondecreasing order of elements), i.e.

$$M^+[m - i + 1, n - j + 1] = \begin{cases} s_i - f_j & \text{if } s_i > f_j \\ 0 & \text{otherwise} \end{cases}$$

$$M^-[i, j] = \begin{cases} s_j - f_i & \text{if } f_i < s_j \\ 0 & \text{otherwise} \end{cases}$$

For each of $2d$ matrices, the optimization algorithm uses the algorithm of Frederickson and Johnson [9, 11].

Lemma 3 *The discrete rectilinear 2-center problem can be solved in $O(T \log N + M \log(2N/M))$ time where $N = \max(n, m)$, $M = \min(n, m)$ and T is the runtime of the decision algorithm.*

4.2 Decision Algorithm

In this section we generalize planar decision algorithm of Bespamyatnikh and Segal [3] to higher dimensions. Let C' and C'' be two required cubes with centers c' and c'' respectively. There are 2^d cases of relations of the coordinates of the centers c' and c''. Without loss of generality, assume that the center c'' dominates the center c'.

Consider the bounding box $bb(S) = [l_1(S), r_1(S)] \times \ldots \times [l_d(S), r_d(S)]$. It is clear that the region $S' = (-\infty, l_1(S) + \rho] \times \ldots \times (-\infty, l_d(S) + \rho]$ contains the point c'. Otherwise, for some $i = 1, \ldots, d$, $c'_i > l_i(S) + \rho$ and none of the cubes c' and c'' contain the customer point with minimum i-th coordinate. Symmetrically the region $S'' = [r_1(S) - \rho, \infty) \times \ldots \times [r_d(S) - \rho, \infty)$ contains the point c''.

We can assume that the center c' belongs to the set of maxima of $F \cap S'$ since if c' is not a maximal element of $F \cap S'$, it can be replaced by any element of $F \cap S'$ that dominates c'. Let M' denote the set of maxima of $F \cap S'$. Similarly we can assume that the center c'' belongs to the set M'' of minima of $F \cap S''$.

Consider customer points that are not covered by the cube C', i.e. $S \setminus C'$. The second cube has to cover this set. Let $l(C')$ denote the lowest point of the bounding box $bb(S \setminus C')$, i.e. $l(C') = (l_1(S \setminus C'), \ldots, l_d(S \setminus C'))$. The cube C'' contains the set $S \setminus C'$ if and only if its center $c'' \in M''$ is dominated by the point $l(c') + (\rho, \ldots, \rho)$. Let L denote the set $\{l(p) + (\rho, \ldots, \rho), p \in M'\}$. We have proved the following lemma.

Lemma 4 *The decision problem has answer "yes" if and only if there is a pair of points $p \in L$ and $q \in M''$ such that p dominates q.*

We actually reduced the decision problem to a *dominance problem.*

Dominance problem. Given sets L and U in R^d. Are there points $p \in L$ and $q \in U$ such that p dominates q?

One more subproblem concerns determinimg the set of maxima M' and minima M''.

Maxima problem. Given set $A \in R^d$ of points sorted by all coordinates. Find the set of maxima of A.

In the rest of this section we explain how to accomplish the reduction of the decision problem to the maxima problem in linear time. The dominance problem can be solved using the algorithm for maxima problem. Indeed, the dominating pair exists if and only if the set of maxima of $L \cup U$ does not include L.

It remains to show how to compute the set L in linear time. For a point $c' \in M'$, denote $U_i(c') = \{p \mid p_i > c'_i + \rho\}$. It is clear that $S \setminus C' = S \cap (\cup_{i=1}^{d} U_i(c'))$. The lowest points of $bb(S \setminus C')$ can be obtained in constant time if the lowest points of $bb(U_i(c'))$ for $i = 1, \ldots, d$ are known.

Now we show how to compute, say $U_1(c')$ for all points $c' \in M'$. We apply sweeping technique where the sweeping hyperplane has form $x_1 = const$. The set $U_1(c')$ is updated by insertions only if the sweeping hyperplane is moving down, i.e. *const* is decreasing. The processing of the inserted point is only to update the current lowest point. So we have proved the following lemma.

Lemma 5 *The decision problem can be solved in $O(n + m + T_M(n) + T_M(m))$ time where $T_M(n)$ is the running time of an algorithm for the maxima problem.*

The maxima problem is well studied [12, 19] and the best running time is

$$T_M(n) = \begin{cases} O(n) & \text{if } d = 2 \\ O(n \log^{d-3} n \log\log n) & \text{if } d > 2 \end{cases}$$

We conclude our main result of this Section.

Theorem 6 *The rectilinear discrete 2-center problem can be solved in $O(N \log^{d-2} N \log\log N + N \log N)$ time where $N = \max(n, m)$.*

5 Axis-parallel segments

In this section we consider a restricted version of the rectilinear 2-center problem where the possible locations of the facilities are restricted by a set F of m axis-parallel segments. We apply similar optimization technique as in the discrete version. The following Lemma describes all possible values of ρ^*.

Lemma 7 *In the rectilinear 2-center problem with facility locations restricted to a set of axis-parallel segments F the 2-radius ρ^* is either*

- *a distance between a customer point and a segment of F or*
- *half a distance between two customer points.*

Proof: The basic idea of the proof is to improve the facility locations if the conditions of Lemma are not satisfied. Recall that the objective of the facility locations is to minimize the maximum of the size of covering cubes. We add the second objective which is to minimize the total volume of the cubes. Consider two optimal facility locations p_1 and p_2. Two cubes C_1 and C_2 of size ρ_1 and ρ_2 with centers p_1 and p_2 cover all customer points.

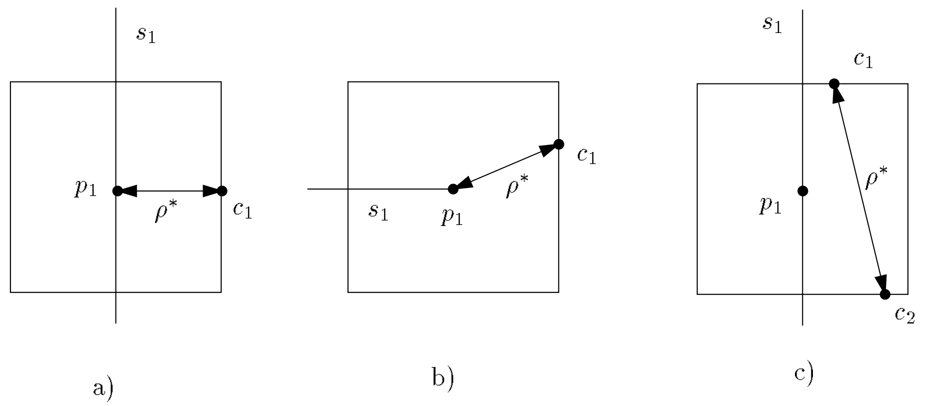

Figure 2: 2-radius configurations

It is clear that $\rho^* = \max(\rho_1, \rho_2)$. We can assume that $\rho_1 = \rho^*$. There is at least one customer c_1 at distance ρ^* from the facility p_1. The Lemma follows if c_1 is at distance ρ^* from the segment s_1 containing the facility p_1, see Fig. 2 a) and b).

Suppose that the distance between c_1 (and all customer points on the boundary of C_1) and s_1 is less than ρ_1. Hence
• c_1 lies on a face of the cube C_1 that is not parallel to s_1, and
• p_1 is not an endpoint of s_1 closest to c_1, see Fig. 2 c).
The facility p_1 cannot be moved along the segment s_1 toward the customer c_1 keeping all its customers within distance ρ^*. Therefore there is at least one customer c_2 on the opposite face of the cube C_1. Lemma follows because the distance between customer points c_1 and c_2 is $2\rho^*$. ■

To apply parametric search of Frederickson and Johnson [9, 11] we build $4d$ sorted matrices. The first $2d$ matrices correspond to the distances between the customer points and the segments of F. The remaining $2d$ matrices correspond to the distances between the customer points.

The decision algorithm builds the set of maxima M' of segments F clipped by the cube $S_1 = [l_1(S), l_1(S) + \rho] \times [l_2(S), l_2(S) + \rho]$. Clearly a segment s can participate in M' by either its endpoint or intersection point of s and the boundary of S_1. There is $O(m)$ points that could participate in M'. The algorithm finds them and compute the set of maxima M' in $O(m)$ time using presorting of the segment endpoints. The remaining part of the algorithm is the same as in the discrete version. We conclude the following theorem.

Theorem 8 *The rectilinear 2-center problem with facility locations restricted to a set of m axis-parallel segments can be solved in $O(N \log^{d-2} N \log\log N + N \log N)$ time where $N = \max(n, m)$ and n is the number of customers.*

This approach can be extended to the facility restriction to a set of polytopes with axis-parallel faces.

Corollary 9 *The rectilinear 2-center problem with facility locations restricted to a set F of polytopes with axis-parallel faces can be solved in $O(N \log^{d-2} N \log\log N + N \log N)$ time where $N = \max(n, m)$ and n is the number of customers and m is the complexity of F, i.e. the total number of the faces of all dimensions.*

6 Conclusion

In this paper we have investigated three rectilinear 2-center problems, the continuous and two constrained versions when facility locations are restricted to a set of points or a set of axis-parallel polytopes. We present a simple linear-time algorithm for the continuous version and an efficient algorithm with polylog running time for the restricted versions.

In future research we plan to improve the running time of the algorithms for the restricted versions in unbalanced cases $n << m$ and $n >> m$. Another direction is to obtain efficient algorithms for the metric L_1.

References

[1] P. K. Agarwal and M. Sharir. Planar geometric location problems. *Algorithmica*, 11:185–195, 1994.

[2] P. K. Agarwal, M. Sharir, and E. Welzl. The discrete 2-center problem. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 147–155, 1997.

[3] S. Bespamyatnikh and M. Segal. Rectilinear static and dynamic discrete 2-center problems. In *to appear in Proc. 9th Workshop Algorithms Data Struct.*, 1999.

[4] Z. Drezner. The planar two-center and two-median problem. *Transp. Sci.*, 18:351–361, 1984.

[5] Z. Drezner. On the rectangular p-center problem. *Naval research logistics*, 34:229–234, 1987.

[6] D. Eppstein. Dynamic three-dimensional linear programming. *ORSA J. Comput.*, 4(4):360–368, 1992.

[7] R. L. Francis and J. A. White. *Facility Layout and Location.* Prentice Hall, Englewood Cliffs, NJ, 1974.

[8] G. N. Frederickson. Optimal algorithms for tree partitioning. In *Proc. 2nd ACM-SIAM Symp. Discr. Algo.*, pages 168–177, 1991.

[9] G. N. Frederickson and D. B. Johnson. The complexity of selection and ranking in $X+Y$ and matrices with sorted rows and columns. *J. Comput. Syst. Sci.*, 24:197–208, 1982.

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

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

[12] H. N. Gabow, J. L. Bentley, and R. E. Tarjan. Scaling and related techniques for geometry problems. In *Proc. 16th Annu. ACM Sympos. Theory Comput.*, pages 135–143, 1984.

[13] A. Glozman, K. Kedem, and G. Shpitalnik. On some geometric selection and optimization problems via sorted matrices. *Comput. Geom. Theory Appl.*, 11:17–28, 1998.

[14] J. Hershberger. A faster algorithm for the two-center decision problem. *Inform. Process. Lett.*, 47:23–29, 1993.

[15] J. W. Jaromczyk and M. Kowaluk. An efficient algorithm for the Euclidean two-center problem. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 303–311, 1994.

[16] J. W. Jaromczyk and M. Kowaluk. A geometric proof of the combinatorial bounds for the number of optimal solutions to the 2-center Euclidean problem. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 19–24, 1995.

[17] M. J. Katz, K. Kedem, and M. Segal. Constrained square center problems. In *Proc. 6th Scand. Workshop Algorithm Theory*, volume 1432 of *Lecture Notes Comput. Sci.*, pages 95–106. Springer-Verlag, 1998.

[18] M. J. Katz and Micha Sharir. An expander-based approach to geometric optimization. *SIAM J. Comput.*, 26:1384–1408, 1997.

[19] D. G. Kirkpatrick and R. Seidel. Output-size sensitive algorithms for finding maximal vectors. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 89–96, 1985.

[20] M. T. Ko and Y. T. Ching. Linear time algorithms for the weighted tailored 2-partition problem and the weighted rectilinear 2-center problem under L_∞-distance. *Discrete Appl. Math.*, 40:397–410, 1992.

[21] M. T. Ko and R. C. T. Lee. On weighted rectilinear 2-center and 3-center problems. *Inform. Sci.*, 54:169–190, 1991.

[22] N. Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput.*, 12:759–776, 1983.

[23] R. C. Prim. Shortest connection networks and some generalizations. *Bell Syst. Tech. J.*, 36:1389–1401, 1957.

[24] J.-M. Robert and G. Toussaint. Computational geometry and facility location. In *Proc. Intern. Conf. on Operations Research and Management Science*, pages B–1–B–19, 1990.

[25] M. Sharir. A near-linear algorithm for the planar 2-center problem. *Discrete Comput. Geom.*, 18:125–134, 1997.