Minimum Enclosing Circle Problem with Base Point

Binay Bhattacharya* Lily Li†

Abstract

This article presents a linear time algorithm to solve a variant of the minimum enclosing circle (MEC) problem. The inputs are a point set S of size n, and a point b in the plane called the *free point*. Our goal is to locate a circle center o^* such that the maximum distance of all points in S to o^* divided by the distance from o^* to b is minimized. The original investigation by Qiu et al. [5] found an $O(n \log n)$ algorithm using the furthest point Voronoi diagram of the point set S. This problem can be formulated as a generalized linear programming problem when the domain for the optimal solution is restricted and therefore, can be solved in linear expected time [3]. We describe here a simple deterministic linear time algorithm based on Meggido's prune-and-search solution to the standard problem [4]. We extend our technique to solve similar variants of the MEC problem where the free point is replaced with other geometric objects such as a free line, a free line segment, and a set of free points.

1 Introduction

The classical minimum enclosing circle (MEC) problem takes a point set S of size n and seeks to find a covering circle of smallest radius. Since this enclosing circle is uniquely defined by its center, the problem is equivalent to finding a point o^* to minimize its maximum distance to the points of S. This problem, proposed as early as 1856 by James J. Sylvester, yielded to various techniques [7]. Shamos and Hoey developed an $O(n \log n)$ algorithm [6]. Later, in 1983, Megiddo presented an $O(n)$ algorithm using the prune-and-search technique. By first constructing and solving a restricted problem, he was able to eliminate a fraction of the points at each iteration to solve the general problem optimally [4]. Simple linear time randomized algorithms have been proposed by Matousek et al. [3].

In addition to the search for an optimal algorithm, researchers studied variants of the standard problem by introducing weight to the points of S and by restricting the placement of the circle center [4, 2]. We came across the basic premise of this problem from a paper by Qiu et al. [5]. In previous MEC problems, a cost can be assigned to each point of the point set. The goal of the problems can be restated as minimizing the maximum cost. In the classical MEC problem, the cost of a point is its euclidean distance to the circle center. In the weighted MEC problem the cost of a point is this distance scaled by the weight of the point. Instead of a constant weight associated to each $p \in S$, Qiu et al. proposed a dynamic weight on the distance to a chosen point, known as the *free point*, in the plane. Their paper described an application of this work to target registration error.

Each variant of the classical problem considered here introduces a fixed geometric object to augment the distance to the circle center. The simplest geometric object to consider is a point so, stated formally, the free point variant of the problem is as follows:

Input: A point set S, $|S| = n$, and a free point b.

Goal: If $d(u, v)$ denotes the Euclidean distance between points u and v, then the cost $c_o(p, b)$ for a circle center o and point $p \in S$ with respect to the base point b is defined as:

$$c_o(p, b) = \frac{1}{d(b, o)} \cdot d(p, o).$$

Thus our objective is to determine o^* such that:

$$\max_{p \in S} \frac{d(p, o^*)}{d(b, o^*)} = \min_{o \in \mathbb{R}^2} \left(\max_{p \in S} \frac{d(p, o)}{d(b, o)} \right).$$

As we will see later, we are interested in the distance function $c_o(p, b)$ where o lies in the half-space, delineated by the bisector of p and b, containing p. As a result, the problem under consideration can be formulated by a quasiconvex program. Such programs can be solved in expected linear time by using the randomized algorithm of Matousek et al. [3]. The contribution of this paper is to give a simpler deterministic algorithm using Megiddo's prune-and-search method. Similar algorithms can be designed for other dynamic MEC problems considered here.

In Section 2 we will present a linear time algorithm to solve this problem. Section 3 will explore the MEC problem with dynamic weight with respect to a *free line* and a *free line segment*. These three variants are solved in linear time by extending the algorithm described in Section 2 and by using previously constructed building blocks. Section 4 extends the free point variant by considering a set of free points. We will present an algorithm for this problem which is linear in the number

*School of Computer Science, Simon Fraser University, binay@cs.sfu.ca

†Same as above, xyl9@sfu.ca

of input points given that a *non-trivial* solution — a concept that we will define later — exists.

2 One Free Point

We begin our investigation into the dynamic weighted MEC problem with respect to a free point by considering the range of possible costs. Observe that as the circle center o approaches infinity in any direction, $c_o(p,b) \to 1$ for all $p \in S$ since $d(p,o)$ approaches $d(b,o)$ in value. Let o at infinity be the *trivial solution*. Thus it only makes sense to optimize the placement of o when $c_o(p,b) < 1$ for all $p \in S$.

Lemma 1 *If b is inside the convex hull of S, then for any circle center o, there exists a point $p \in S$ such that $c_o(p,b) \geq 1$.*

Proof. Place o anywhere in the plane and construct the convex hull $CH(S)$ of S. If the segment $\overline{ob}$ is inside or on the convex hull then there exists a point $p \in S$ such that $|\overline{op}| \geq |\overline{ob}|$. Suppose instead that o is outside $CH(S)$. Let L be the line containing b and o. Since b is inside $CH(S)$, L intersects an edge e of $CH(S)$ between o and b and an edge f of $CH(S)$ after b. Let u and v be the end-points of f. Either $d(u,o) \geq d(b,o)$ or $d(v,o) \geq d(b,o)$. If $d(u,o) \geq d(b,o)$ then $c_o(u,b) \geq 1$. If $d(v,o) \geq d(b,o)$ then $c_o(v,b) \geq 1$. See Figure 1. □

By Lemma 1, a nontrivial solution only exists when b is strictly outside the convex hull of S. We can check that this is the case in linear time. In the following, we assume that b is outside the convex hull of S.

The basis of our linear time algorithm will be Megiddo's prune-and-search algorithm for finding the unweighted MEC [4]. Our algorithm proceeds in two steps. First, we solve a restricted version of the problem in linear time. Using this solution as a subroutine, we will address the problem in full generality. At each step of the algorithm, we will be able to prune at least kn points where k is some constant in the interval $(0,1)$ and n is the number of points remaining.

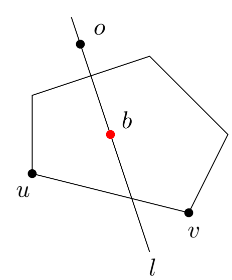


Figure 1: Configuration when base point is inside of the convex hull of point set.

2.1 Restricted Case for a Free Point

The restricted version of the problem is as follows:
Input: A point set S where $|S| = n$, a free point b, and a line L.
Goal: Find the optimum circle center o^* of the general one free point problem if it lies on L. Otherwise determine the side of L containing o^*.

2.1.1 Building Tools

To solve the restricted problem, we must understand the geometry of the optimum circle center o^*. We require that $c_{o^*}(p,b) < 1$ so, intuitively, o^* should be placed closer to the points in S than to b. We formalize our intuition in the following.

Let $L_{u,v}$ be the bisector line of points u and v. For any point p and circle center o, if o is on $L_{b,p}$, then $c_o(p,b) = 1$. Further, $L_{b,p}$ divides the plane in two halves, one side closer to b and the other closer to p. If o is placed on the side closer to p, then $c_o(p,b) < 1$. Thus, to ensure that $c_o(p,b) < 1$ for all points, we must place o on the side of $L_{b,p}$ closest to p for all $p \in S$. This is exactly the cell of b in the furthest point Voronoi diagram $S \cup b$. Since b is an extreme point of the convex hull of $S \cup b$, this cell is non-empty. Call this cell the *feasible region* of S with respect to b. Each point in the interior of the feasible region represents a circle center with cost less than one. Let the intersection of a line L with the feasible region be the *feasible interval* of L.

Lemma 2 *The feasible interval of L with respect to the point set S of size n and a free point b can be found in optimal $O(n)$ time.*

Proof omitted.

Next, focus on one point $p \in S$ and consider how moving o changes $c_o(p,b)$. In particular, we investigate all locations for o which yield the same value of $c_o(p,b)$. Let $c_o(p,b)$ be a constant α with $0 < \alpha < 1$. To simplify the calculation, locate p at the origin. Let $o = (o_x, o_y)$ and $b = (b_x, b_y)$. We determine the coordinates of o which keep the cost constant.

$$\alpha^2 = \left(\frac{d(p,o)}{d(b,o)}\right)^2 \quad (1)$$

$$= \frac{o_x^2 + o_y^2}{(o_x - b_x)^2 + (o_y - b_y)^2} \quad (2)$$

$$o_x^2 + o_y^2 = \alpha^2((o_x - b_x)^2 + (o_y - b_y)^2) \quad (3)$$

$$\alpha^2(b_y^2 + b_x^2) = (1-\alpha^2)o_x^2 + (1-\alpha^2)o_y^2 + 2\alpha^2 b_x o_x + 2\alpha^2 b_y o_y \quad (4)$$

Thus $c_o(p,b)$ remains unchanged when o is placed on the circle described by equation 4 with variable in o_x and o_y. Let this circle and its interior be denoted by

$D_\alpha(p, b)$. Observe that if o is located on the boundary (resp. inside, outside) of $D_\alpha(p, b)$, then $c_o(p, b) = \alpha$ (resp. $c_o(p, b) < \alpha$, $c_o(p, b) > \alpha$). See Figure 2. For the restricted case we need to find the value α such that $D_\alpha(p, b)$ is tangent to a line L. The interior of $D_\alpha(p, b)$, in this case, lies entirely on one side of L and this side contains the optimum circle center with respect to b.

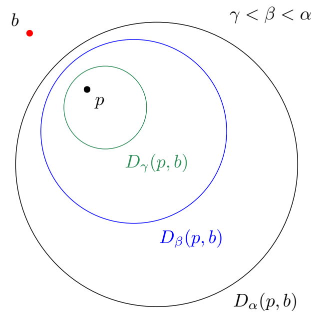


Figure 2: Circles of equal cost about point p with respect to the *free point* b.

Lemma 3 *Given a line L, a point p of the point set, and a free point b, we can find an α such that $D_\alpha(p, b)$ is tangent to L in constant time.*

2.1.2 Solving the Restricted Case for a Free Point

We will use the tools built thus far to solve the restricted problem in linear time.

Lemma 4 *Let a line L be given. If the optimum circle center o^* lies on L, then o^* can be found in linear time. Otherwise, it takes at most linear time to determine the side of L containing o^*.*

Proof. To simplify the explanation, we perform a transformation on the input so that L coincides with the x-axis.

1. Use Lemma 2 to find the feasible interval of L. If the interval is empty, return the side of L which does not contain the free point b. This side contains the optimal solution.

2. Suppose that the feasible interval of L is non-empty. Proceed through the following loop:

   (a) If the point set S contains fewer than two points, then the loop terminates. Otherwise, randomly pair up the points in S. For each pair $\{p, q\}$, find the intersection of the bisector $L_{p,q}$ with L. If $L_{p,q}$ is parallel to L, then the point in the pair closer to L is redundant and can be removed. Let the set of intersection points on L be I.

   (b) Find the median of I using a linear time algorithm [1]. Let this median value be x_m.

   (c) Suppose, without loss of generality, that x_m is to the left of the feasible interval of L. The optimum circle center x^* on L satisfies $x_m < x^*$. For each bisector $L_{p,q}$ which intersect L to the left of x_m, we can remove the rightmost point q of $L_{p,q}$ from S since q is closer to x^* than p. See Figure 3. Start again at step (a) with this modified S.

   (d) Otherwise, x_m falls strictly within the feasible interval. Let E be the subset of points in S farthest from x_m such that $d(p, x_m) = d_m$ for all $p \in E$. Then the cost $c_{x_m}(p, b) = \frac{d(p,x_m)}{d(x_m,b)} = \frac{d_m}{d(x_m,b)}$ for each $p \in E$. Let $\alpha = \frac{d_m}{d(x_m,b)}$.

      i. For every point $p \in E$ find the α-cost disk $D_\alpha(p, b)$ of p as discussed in Section 2.1.1. The boundary of each $D_\alpha(p, b)$ will intersect L at x_m and at most one other point.

      ii. Suppose, without loss of generality, that the boundary of some $D_\alpha(p, b)$ intersects L at a point to the right of x_m. Since $c_o(p, b) < \alpha$ when the circle center o is in the intersection of $D_\alpha(p, b)$ for all $p \in E$, $x_m < x^*$. Prune one point from each bisector intersecting L to the left of x_m as before. After removing the unnecessary points from S, restart from step (a).

      iii. Otherwise all α-cost disks are tangent to L at x_m. Exit the loop. See Figure 4c.

Upon termination, we can find the global optimal circle center o^* if it is on L or determine the side of L containing o^* by constructing the intersection of the $D_\alpha(p, b)$ for the remaining $p \in S$. See Figure 4.

In each iteration we do a linear amount of work and prune away at least one fourth of the remaining points. Thus the running time of the algorithm is linear. □

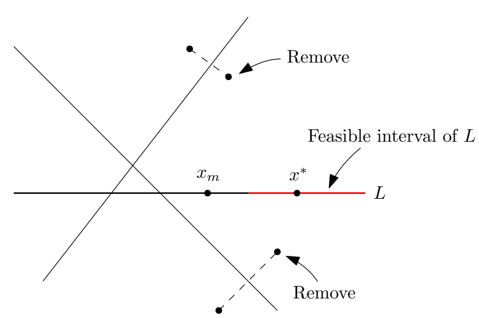
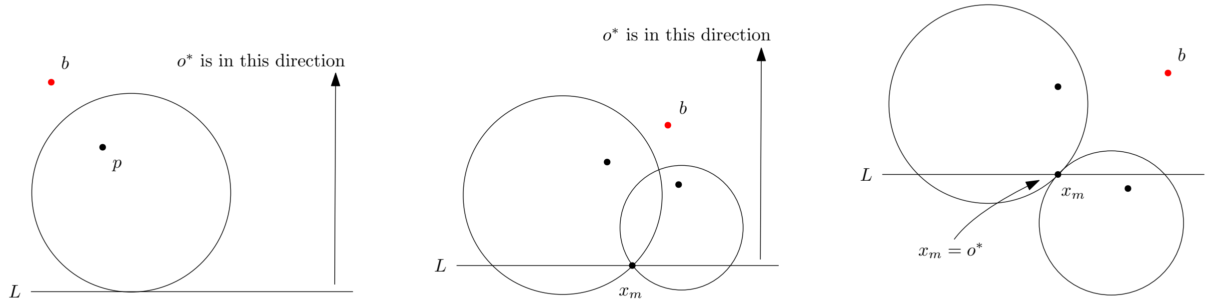


Figure 3: Points to remove if x_m falls outside the feasible interval of L.



(a) Terminate with only one point.

(b) Terminate with multiple points whose equal-cost circle all intersect.

(c) Terminate with multiple points where at least two have equal-cost circle which do not intersect.

Figure 4: Terminating conditions for the general case with one free point and global optimum circle center o^*.

2.2 General Case for a Free Point

We can design the algorithm for the general problem with respect to a free point using the linear time oracle for the restricted case. It is almost identical to Megiddo's general algorithm [4].

Theorem 5 *The optimum circle center o^* to minimize the maximum cost $c_{o^*}(p, b)$ of all points in the point set S with respect to a free point b can be found in linear time with respect to the size of S.*

3 A Free Line and a Free Line Segment

Similar to the dynamic MEC problem involving one free point, we will introduce the problem variant where the cost changes with respect to a free line.

Input: A point set S where $|S| = n$ and a free line B.

Goals: Find a circle center o such that the maximum cost $c_o(p, B)$ for the points $p \in S$ is minimized where the cost $c_o(p, B)$ of a point p is:

$$c_o(p, B) = \frac{1}{\min_{b \in B} d(b, o)} \cdot d(p, o).$$

The problem involving a free line segment is identical except that the input B is a line segment.

3.1 General Case with Free Line

We begin by considering cases where the trivial solution is optimal. In the case of the free point, the base point cannot be inside the convex hull of the point set S.

Lemma 6 *If B intersects the convex hull of S, then there exists a point $p \in S$ such that $c(p, B) \geq 1$.*

Proof. Suppose the free line (resp. free line segment) intersects edge e of the convex hull of S at some point b. Let u and v be the end points of e. Then for any circle center o, $d(o, u) \geq d(o, b)$ or $d(o, v) \geq d(o, b)$ similar to Lemma 1. Without loss of generality suppose $d(o, u) \geq d(o, b)$. If b' is the actual base point closest to o, then $d(o, b) \geq d(o, b')$ so

$$c_o(u, B) = c_o(u, b') = \frac{d(o, u)}{d(o, b')} \geq \frac{d(o, u)}{d(o, b)} \geq 1.$$

□

Thus the trivial solution, which locates the circle center at infinity orthogoal to B, is optimal when B intersects the convex hull of S. Determining if a line intersects the convex hull of a point set S takes linear time. In the forgoing, let the intersection of the free line B and the convex hull of S be empty and, without loss of generality, that all $p \in S$ are to the right of B.

Given a line L, first find the feasible interval of L with respect to B. Instead of intersecting L with the bisectors $L_{b,p}$ for $p \in S$ as is the case for the free point, we intersect L with the parabola of equal distance between p and the line B. Call this the 1-cost parabola of p and denote it by $H_1(p, B)$. Since a parabola intersects a line in at most two places, finding the intersection of L with $H_1(p, B)$ for all $p \in S$ takes linear time. Without too much ambiguity, let this intersection be the feasible interval of L.

Theorem 7 *Let B be a free line. The optimum circle center o^* to minimize the maximum cost $c_{o^*}(p, B)$ of all points in the point set S can be found in linear time.*

The crucial observation is that the intersection of the α-disks behaves as though we have a free point. Choose a point u_m on the line L as the circle center. Let b_m be the closest point on B to u_m. Suppose that the boundary of every $D_\alpha(p, b_m)$ intersect L to the left of u_m. For any u' on L right of u_m, with closest point b' on B, the cost for a point p is:

$$c_{u'}(p, b') = \frac{d(p, u')}{d(b', u')} > \frac{d(p, u')}{d(b_m, u')} \geq c_{u_m}(p, b_m).$$

The first inequality holds since $d(b_m, u') > d(b', u')$ and the second holds since u' is outside $D_\alpha(p, b_m)$.

3.2 A Free Line Segment

We observe that the linear time solution for the free line variant of the MEC problem easily adapts to a solution for a free line segment. Let K be the free line segment. Instead of the 1-cost parabolas considered in Theorem 7, we note that region in the plane equidistant between K and any point p in the point set is composed of constantly many line and parabola parts. Let these be 1-cost curves. Finding the intersection of all 1-cost curves takes linear time just like the 1-cost parabolas. Thus a slight modification to the algorithm in Theorem 7 yields an algorithm for the free line segment.

Corollary 8 *Let K be a line segment and S be a point set of order n. Finding the optimum circle center o^* to minimize the maximum cost $c_{o^*}(p, K)$ overall points $p \in S$ takes $O(n)$.*

4 A Set of Free Points

We extend the one free point problem by using a point set as the fixed geometric object. This variant of the MEC problem is stated formally as:
Input: A point set S where $|S| = n$ and a free point set B where $|B| = m$.
Goals: Find a circle center o^* such that the maximum cost $c_{o^*}(p, B)$ for the points $p \in S$ is minimized where the cost $c_o(p, B)$ of a point p with respect to center o is:

$$c_o(p, B) = \frac{1}{\min_{b \in B} d(b, o)} \cdot d(p, o).$$

Given a circle center o, this slightly altered cost calculation first finds the closest point $b^* \in B$ to o, then divides the distance to p by the distance to this closest free point b^*. More succinctly our objective is to find o^* such that:

$$\max_{p \in S} \left(\frac{d(p, o^*)}{\min_{b \in B} d(b, o^*)} \right) = \min_{o \in R^2} \max_{p \in S} \left(\frac{d(p, o)}{\min_{b \in B} d(b, o)} \right).$$

As before, we begin by considering when the trivial solution is optimal. Suppose that some $b \in B$ falls inside the convex hull, $CH(S)$, of S. For any o in the plane as the circle center, there exists a points $p \in S$ such that $c(p, b) \geq 1$ by Lemma 1. Consider the closest point $b' \in B$ to o. Observe that

$$1 \leq \frac{d(o, p)}{d(o, b)} \leq \frac{d(o, p)}{d(o, b')} = c_o(p, b') = c_o(p, B).$$

Since $d(o, b) \geq d(o, b')$, the trivial solution is optimal when any point in B falls within $CH(S)$. However, even if all points in B are outside the $CH(S)$, a non-trivial solution might still not exist. Possible cases include:

1. If S and B are linearly separable, then the optimal cost is less than one.
2. If S and B are not linearly separable, we have the following sub-cases.
   (a) If there exist a point of B inside $CH(S)$, the optimal cost is trivially one.
   (b) If all points of B lie outside $CH(S)$, but $CH(S)$ and $CH(B)$ intersect, the optimal cost could be one or less than one.

In the next section, we consider the case (1) where the optimal cost is less than one. Again we will solve this problem in two steps. First we build an oracle to solve the restricted problem on a line in linear time. Then we use the oracle to develop an algorithm to solve the general case. This second step is a modified version of the general case of the one free point variant so will only be mentioned briefly.

4.1 Restricted Case for a Set of Free Points

Input: A point set S where $|S| = n$, a free point set B where $|B| = m$ and a line L.
Goals: Find a circle center o^* which minimizes the maximum cost $c_{o^*}(p, B)$ for all $p \in S$ if o^* is on L. Otherwise determine the side of L containing o^*.

Lemma 9 *Let a line L, a point set S of order n, and a set of free points B of order m be given. Assume that B is linearly separable from S. We can find the side of L, possibly including L, containing o^* in $O(n + m)$ time.*

Proof. Perform a transformation of the input so that L coincides with the x-axis.

1. Randomly pair up the points of S. For each pair $\{p, q\}$, find the intersection of the bisector $L_{p,q}$ with L. Do the same with the free points of B. Let the set of all intersections on L be I. Note that $|I| = \lfloor \frac{n}{2} \rfloor + \lfloor \frac{m}{2} \rfloor$.
2. Find the median x-coordinate of the points of I using a linear time algorithm [1]. Let this median value be x_m.
3. Let $E = \{p \in S : d(p, x_m) = \max_{q \in S} d(q, x_m)\}$ be the points in S farthest from x_m and $F = \{b \in B : d(b, x_m) = \min_{c \in B} d(c, x_m)\}$ be the points in B closest to x_m. Further let $d(p, x_m) = u$ for all $p \in E$, $d(b, x_m) = v$ for all $b \in B$, and $u/v = \alpha$.
4. If $\alpha \geq 1$ then x_m is outside the feasible interval of L. We must decide which side of x_m contains the feasible interval of L, if exists. Randomly select one point $p \in E$ and one free point $b \in F$.

(a) Find the bisector $L_{p,b}$ of p and b. Suppose that $L_{p,b}$ intersects L at $x' \geq x_m$. Since $\alpha \geq 1$ and $u \geq v$, we must have b is to the left of $L_{p,b}$. Thus the optimum circle center x^* on L satisfies $x_m \leq x^*$. For bisectors $L_{s,q}$, with $s, q \in S$, intersecting L to the left of x_m, remove the non-dominating point associated with $L_{s,q}$. This point is closer to x^* so has lower cost. For bisectors $L_{r,t}$, with $r, t \in B$, intersecting L to the left of x_m, remove the dominating base point associated with $L_{r,t}$. This base point is farther from x^*. The case where every $D_\alpha(p, b)$ intersects L to the left of x_m can be handled similarly. Restart the loop after removing the redundant points.

5. If $\alpha < 1$ then x_m is in the feasible interval of the line L. Pick any $p \in E$ and for every $b \in F$ then calculate $D_\alpha(p, b)$. The optimum circle center o^* must be in $D_\alpha(p, b)$ for any b since the optimum cost of p is less than or equal to α.

   (a) Suppose there exists some $b \in F$ such that $D_\alpha(p, b)$ intersects L to the right of x_m. Then x^* satisfies $x_m \leq x^*$. Prune one point for every bisector which intersects L to the left of x_m as described above. The case where $D_\alpha(p, b)$ intersect L to the left of x_m can be treated similarly. Restart the loop after removing the redundant points.

   (b) Otherwise $D_\alpha(p, b)$ is tangent to L at x_m for every $b \in F$. Since a non-trivial solution exists, the half-plane of L containing any $D_\alpha(p, b)$ for any $p \in E$ and $b \in F$ contains the optimum circle center. Return this side of L.

The running time analysis is identical to that of the restricted case for one free point in Lemma 4. Here, however, the total size of the input is $|S| + |B| = n + m$. Thus the total running time is $O(n + m)$. □

4.2 Unrestricted Case for a Set of Free Points

Theorem 10 *Let B be a set of free points where $|B| = m$. The optimum circle center o^* to minimize the maximum cost $c_{o^*}(p, B)$ of all points in the point set S can be found in optimal $O(n + m)$ time.*

We can modify the algorithm for one free point to obtain an $O(n + m)$ algorithm for the general case for a free point set B.

We use the oracle presented in Lemma 9 when solving for the restricted problem. Next, since the standard algorithm already handles bisector intersections formed from points of S, we will instead consider bisector intersections formed from points of B. By determining the quadrant containing the optimal solution, a fraction of the base points can be eliminated from further considerations.

5 Conclusion

We considered a variant of the MEC problem where the cost of each point in the input point set S was dynamic with respect to a fixed geometric object such as a point, a line, a line segment, and a point set — with some restrictions. We have presented a deterministic prune-and-search based linear time algorithm to solve each of these problems by using the fact that the distance function is quasiconvex in the domain where the optimal solution could lie. For the case 2b of Section 4, it is not known whether a linear time solution exists.

Future work may consider a convex polygon as the fixed geometric object. The cost of a point p given a circle center o would be the $d(o, p)$ divided by the closest distance between o and a point on the convex polygon.

References

[1] Aho, A., Hopcroft, J., and Ullman, J. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] Hurtado, F., Sacristán, V., et al. Some constrained minimax and maximin location problems. In *Studies in Locational Analysis* (2000), Citeseer.

[3] Malousek, J., Sharir, M., and Welzl, E. A subexponential bound for linear programming. *Algorithmica 16*, 498–516.

[4] Megiddo, N. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM Journal on Computing 12*, 4 (1983), 759–776.

[5] Qiu, L., Zhang, Y., and Zhang, L. Minimum enclosing circle of a set of static points with dynamic weight from one free point. *arXiv preprint arXiv:1703.00112* (2017).

[6] Shamos, M. I., and Hoey, D. Closest-point problems. In *Foundations of Computer Science, 1975., 16th Annual Symposium on* (1975), IEEE, pp. 151–162.

[7] Sylvester, J. J. A question in the geometry of situation. *Quarterly Journal of Pure and Applied Mathematics 1* (1857).

[8] Welzl, E. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*. Springer, 1991, pp. 359–370.