# Disk constrained 1-center queries

Luis Barba *


## Abstract

We show that a set $P$ of $n$ points in the plane can be preprocessed in $O(n \log n)$-time to construct a data structure supporting $O(\log n)$-time queries of the following form: Find the minimum enclosing circle of $P$ with center on a given disk.


## 1 Introduction

Given a set $P$ of $n$ points in the plane, the minimum enclosing circle problem, originally posed by Sylvester in 1857 [14], asks to identify a point $c_P$ in the plane such that the maximum Euclidean distance from the points of $P$ to $c_P$ is minimized. Therefore, this problem can be thought as that of finding the center of the minimum enclosing circle of $P$. For ease of notation we say that every circle containing $P$ is a $P$-circle. An $O(n^2)$-time algorithm was presented by Elzinga and Hearn [5] to find the minimum $P$-circle. Later, Preparata in [11], and Shamos and Hoey in [13], independently proposed two algorithms to solve this problem in $O(n \log n)$-time. Lee presented the farthest-point Voronoi diagram, which can be also used to solve this problem in $O(n \log n)$-time [9]. Finally, Megiddo proposed an optimal $O(n)$-time algorithm to find the center of the minimum $P$-circle using a prune and search approach [10]. Furthermore, the problem of finding the minimum enclosing $d$-sphere that contains a given set of $n$ points in $\mathbb{R}^d$ can be solved in $O(n)$-time for any fixed $d$ [3][4].

Several constrained versions of the minimum $P$-circle problem have been studied lately. Hurtado, Sacristan and Toussaint presented an optimal $O(n + m)$-time algorithm to find the minimum $P$-circle whose center is constrained to satisfy $m$ linear inequalities [6]. Bose and Toussaint considered the generalized version of this problem by restricting the center of the $P$-circle to lie inside a simple polygon of size $m$. They proposed an $O((n + m) \log(n + m) + k)$-time algorithm to solve this problem, where $k$ is the number of intersections of $Q$ with the farthest-point Voronoi diagram of $P$ [2]. Megiddo studied the problem of finding the minimum $P$-circle with center on a given straight line and proposed an $O(n)$-time algorithm to solve this problem [10]. He also posed the on-line version of this problem in which a preprocessing of the point set $P$ is allowed and the objective is to answer the following query: Given a straight line $\ell$, find the minimum $P$-circle with center on $\ell$. Das, Karmakar, Nandy and Roy first addressed this problem and proposed an $O(n \log n)$-time preprocessing on $P$, which allows them to answer these queries in $O(\log^2 n)$-time [12]. They improved the query running time to $O(\log n)$ using $O(n^2)$ preprocessing time and space [7]. Finally, Bose, Langerman and Roy showed an $O(n \log n)$-time preprocessing to construct a linear space data structure that answers queries in $O(\log n)$-time [1].

In this paper, we address a generalized version of this problem in which the center of the minimum $P$-circle is constrained to lie on a query disk. This problem has a direct application in wireless communication: think of a set of locations that need to receive a certain message (represented by $P$) and think of a main moving antenna that is broadcasting a message within a certain range (represented by a query disk $Q$). Our objective is then to determine the location for a re-transmitter $C$, inside the range of $Q$, such that every location in $P$ receives the message from $C$ at the lowest cost.

We propose an $O(n \log n)$-time preprocessing on the point set $P$, to construct a linear space data structure that answers both disk and line queries, in $O(\log n)$-time.

## 2 Preliminaries

In this paper, the words *disk* and *circle* refer to the same geometric object. The former refers to the query objects while the latter to the solutions of the query. Given $S \subset \mathbb{R}^2$, $\partial S$ denotes its boundary while $int(S)$ denotes its interior.

Let $P$ be a set of $n$ points in the plane. A circle containing $P$ is called a $P$-circle. Given a disk $Q$ with center on $q$, let $p_Q$ be a point in $P$ such that $q$ lies in the farthest-point Voronoi region associated with $p_Q$. The farthest-point Voronoi diagram of $P$ can be seen as a tree with $n$ unbounded edges and is denoted in this paper by $\mathcal{V}(P)$. For any point $p$ of $P$, let $R(p)$ be the farthest-point Voronoi region associated with $p$. Let $C_P$ be the minimum enclosing circle of $P$ and let $c_P$ be its center. If $c_P$ is not a vertex of $\mathcal{V}(P)$, we insert it into $\mathcal{V}(P)$ by splitting the edge where it belongs. We consider $\mathcal{V}(P)$ as a rooted tree at $c_P$. Given a point $x$ on $\mathcal{V}(P)$, $\pi_x$ denotes the unique path joining $c_P$ with $x$

*Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium, lbarbafl@ulb.ac.be

contained in $\mathcal{V}(P)$. Given a set $S \subseteq \mathbb{R}^2$, let $C(S)$ be the minimum $P$-circle with center on $S$ and let $c(S)$ be its center. We say that $c(S)$ is the query center of $S$. Whenever $S = \{x\}$, $C(x)$ denotes the minimum $P$-circle with center on $x$ and we let $\rho(x)$ be its radius. If $x$ belongs to $R(p)$ for some vertex $p$ of $P$, then $C(x)$ passes through $p$ and hence, $\rho(x) = d(p, x)$, where $d(*, *)$ represents the Euclidean distance between two points in the plane. Given a disk $Q$ with center on $q$ and a point $x$ outside $Q$, the projection of $x$ on $Q$, denoted by $\sigma(x, Q)$, is the intersection of the segment $[x, q]$ with the circumcircle of $Q$.

## 3 The minimum $P$-circle with center on $Q$

**Proposition 1** *Let $w$ be a point on an edge of $\mathcal{V}(P)$. The function $\rho$ is monotonically increasing along the path $\pi_w$ starting at $c_P$.*

**Proof.** In [12] (Result 2), it is shown that if $x$ is an ancestor of $y$ on the rooted tree $\mathcal{V}(P)$, then $\rho(x) < \rho(y)$. Since $\rho(x)$ is a convex function [1] (Lemma 3), it is also convex when restricted to any segment of $\pi_w$. □

Let $Q$ be a disk with center on $q$. The following results characterize the position of $c(Q)$ with respect to $\mathcal{V}(P)$.

**Proposition 2** *The point $c(Q)$ lies on $\partial Q$ if and only if $c_P \notin int(Q)$.*

**Proof.** $\rightarrow$) If $c_P \in int(Q)$, then $C(Q) = C_P$ and hence $c(Q) = c_P$ lies in $int(Q)$.

$\leftarrow$) Assume that $c(Q)$ lies in the interior of $Q$ but $c_P$ does not. Let $p$ be a point of $P$ such that $c(Q)$ belongs to $R(p)$. Two cases arise:

If $c(Q) \in int(R(p))$, then there is a point $x$ in the vicinity of $c(Q)$, slightly closer to $p$, such that $x$ belongs to $Q \cap R(p)$. Therefore, $\rho(x) < \rho(c(Q))$ which is a contradiction. Otherwise, if $c(Q)$ lies on an edge of $\mathcal{V}(P)$, we can consider a point $x$ slightly closer to $c_P$ along the path $\pi_{c(Q)}$, such that $x$ still belongs to $Q$. Thus, by Proposition 1 $\rho(x) < \rho(c(Q))$ which is also a contradiction. Therefore, if $c_P$ does not belong to the interior of $Q$, then $c(Q)$ lies on the boundary of $Q$. □

From now on we assume that $c_P$ is not contained in $Q$. Otherwise, $c_P$ is trivially the query center of $Q$.

**Lemma 3** *Given a disk $Q$ with center on $q$. If $p_Q$ is a point of $P$ such that $q \in R(p_Q)$, then:*

1. *The circumcircle of $C(Q)$ passes through exactly one point $p$ of $P$, if and only if $p = p_Q$ and $\sigma(p_Q, Q) \in int(R(p_Q))$. In this case, $c(Q) = \sigma(p_Q, Q)$.*

2. *The circumcircle of $C(Q)$ passes through at least two points of $P$, if and only if $c(Q)$ lies on an edge of $\mathcal{V}(P)$.*

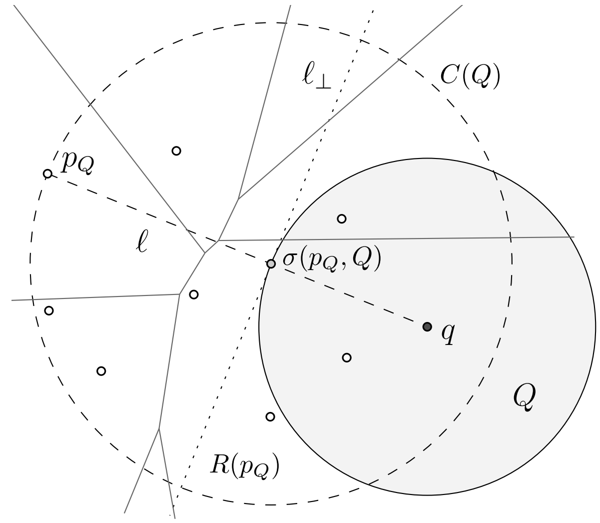


Figure 1: Case 1 of Lemma 3 where the projection of $p_Q$ on $Q$ lies inside $R(p_Q)$ and determines the center of $C(Q)$.

**Proof.** 1 $\rightarrow$) If $C(Q)$ passes through only one point $p$ of $P$, then $c(Q) \in int(R(p))$. Let $\ell$ be the line joining $p$ with $c(Q)$ and let $\ell_\perp$ be the perpendicular line to $\ell$ that passes through $c(Q)$; see Figure 1. Note that $\ell_\perp$ must leave all points of $Q$ on the halfplane defined by $\ell_\perp$ that is farther away from $p$. Otherwise, we can choose a point $x$ inside $Q \cap R(p)$ such that $x$ is closer to $p$ than $c(Q)$—a contradiction since $C(x)$ would be a $P$-circle with smaller radius than $C(Q)$. Since $\ell_\perp$ leaves all points of $Q$ in one halfplane, $\ell_\perp$ is tangent to $Q$ and hence $c(Q) = \sigma(p, Q)$. Moreover, the points $q, c(Q)$ and $p$ are collinear. Thus, the circle with center on $q$ and passing through $p$ is also a $P$-circle, which means that $q \in R(p)$, i.e. $p = p_Q$.

1 $\leftarrow$) Let $C$ be the circle with center on $\sigma(p_Q, Q)$ and radius $d(\sigma(p_Q, Q), p_Q)$ Since $\sigma(p_Q, Q)$ is the closest point of $Q$ to $p_Q$, $C$ is the smallest circle containing $p_Q$ with center on $Q$. Moreover, since $\sigma(p_Q, Q) \in int(R(p_Q))$, $C$ is a $P$-circle passing only through $p_Q$ and any other $P$-circle with center on $Q$ must contain $p_Q$. Thus, $C(Q) = C$ and it passes only through one point of $P$.

2) Follows from the definition of the farthest-point Voronoi diagram; see Figure 2. □

If case 1 of Lemma 3 holds we are done since $C(Q)$ will be the circle with center on $\sigma(p_Q, Q)$ and radius $d(\sigma(p_Q, Q), p_Q)$. Therefore, we assume from now on that $c(Q)$ is a point lying on an edge of $\mathcal{V}(P)$.

## 4 Sketch of the algorithm

The idea behind the algorithm that we will present is to shrink the disk $Q$, obtaining in this way a new disk

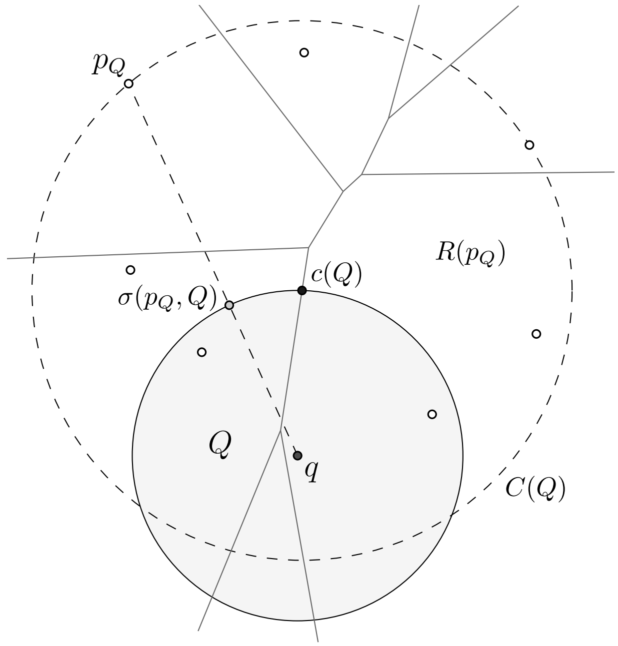


Figure 2: Case 2 of Lemma 3 where $c(Q)$ lies on an edge of $\mathcal{V}(P)$ and $C(Q)$ passes through two points of $P$. Moreover, the projection of $p_Q$ on $Q$ lies outside of $R(p_Q)$.

$Q'$ with the same center, such that case 1 of Lemma 3 holds for $Q'$. Thus, $c(Q')$ can be efficiently computed and we can scale $Q'$ back to its original size, tracking the position of $c(Q')$ during this scaling.

Let $\ell$ be the line joining $q$ with $p_Q$ and let $\omega$ be the intersection of $\ell$ with $\partial R(p_Q)$. It is well known that this intersection is unique. Let $Q'$ be the circle with center on $q$ and radius $d(q, \omega)$. Note that $Q'$ can be seen as the disk $Q$ scaled down such that $\omega$ is the projection of $p_Q$ on $Q'$ and $\omega$ lies in $R(p_Q)$. Thus, by Lemma 3, $C(Q')$ is the circle with center on $\omega$ and radius $d(\omega, p_Q)$; see Figure 3.

The idea is now to scale back $Q'$ to $Q$, without losing the position of the query center of $Q'$ along the process. In order to do that, we construct a family of disks representing this scaling as follows. Assume that $r$ and $r'$ are the radius of $Q$ and $Q'$, respectively, and let $Q(t)$ be the disk with center on $q$ and radius $r' + t(r - r')$, $t \in [0, 1]$. Note that $Q(t)$ represents a continuous scaling starting with $Q(0) = Q'$ and ending with $Q(1) = Q$. Let $\gamma(t)$ be the curve described by query center of $Q(t)$, $t \in [0, 1]$.

**Lemma 4** *The curve $\gamma(t)$ is a continuous curve such that $\gamma(0) = \omega$, $\gamma(1) = c(Q)$ and $\gamma(t)$ lies on $\pi_\omega$ for every $0 \leq t \leq 1$.*

**Proof.** The curve $\gamma(t)$ is continuous since $\rho$ is a continuous function. Thus, it only remains to prove that $\gamma(t)$ is contained in $\mathcal{V}(P)$.

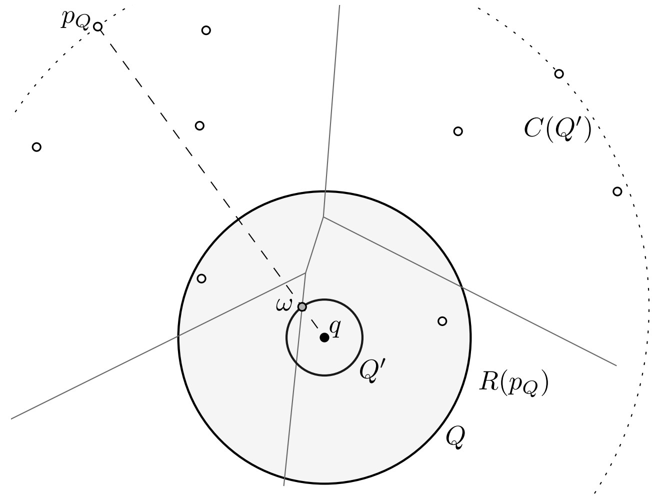


Figure 3: The disk $Q'$ as the reduction of $Q$. In this case, $c(Q') = \omega$ is the projection of $p_Q$ on $Q'$.

Since every $Q(t)$ is centered on $q$, $p_Q = p_{Q(t)}$ for every $0 \leq t \leq 1$. Furthermore, for every $0 < t \leq 1$, the projection of $p_Q$ on $Q(t)$ lies outside $R(p_Q)$; see Figure 3. Therefore, Lemma 3 implies that every $Q(t)$ has its query center lying on an edge of $\mathcal{V}(P)$. In other words, the curve $\gamma(t)$ is completely contained in $\mathcal{V}(P)$.

Since we assumed that $c_P$ lies outside $Q$ and since $Q(t) \subseteq Q(t')$ for every $0 \leq t < t' \leq 1$, the value of $\rho(\gamma(t))$ decreases monotonically as $t$ increases. Thus, because $\gamma(t)$ is contained in $\mathcal{V}(P)$, Proposition 1 implies that $\gamma(t)$ is contained on the path joining $c_P$ with $\omega$. $\square$

Our objective will be to find $c(Q)$ along the path $\pi_\omega$ using a binary search. However, the boundary of a disk may intersect a path on $\mathcal{V}(P)$ more than once. Thus, we need the following result.

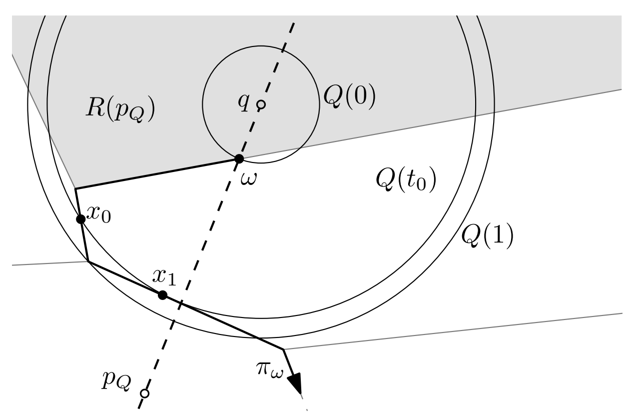


Figure 4: The point $x_0$ is an accumulation point of $\gamma(t)$ while $x_1$ represents a discontinuity of $\gamma(t)$.

**Lemma 5** *There is a unique intersection point between the path $\pi_\omega$ and the boundary of $Q$.*

**Proof.** Proceed by contradiction and assume that the boundary of $Q = Q(1)$ intersects $\pi_\omega$ in at least two points. Recall that $Q' = Q(0)$ intersects $\pi_\omega$ at a unique point $\omega = \sigma(p_Q, Q')$. Let $t_0$ be the minimum value in $[0, 1]$ such that $\partial Q(t_0)$ intersects $\pi_\omega$ in more than 2 points. Let $x_0, \dots, x_k$ be the points of intersection between $Q(t_0)$ and $\pi_\omega$, and assume that they are sorted in decreasing order with respect to their depth on the tree $\mathcal{V}(P)$. Note that, for every $0 \le t < t_0$, $Q(t)$ intersects $\pi_\omega$ in exactly one point and, by Lemma 4, this intersection defines the position of $\gamma(t)$. Therefore, $x_0$ is an accumulation point of the curve $\gamma(t)$; see Figure 4. However, Proposition 1 implies that $\rho(x_0) > \dots > \rho(x_k)$ and hence $\gamma(t_0)$ must be equal to $x_k$. This represents a discontinuity of the curve $\gamma(t)$ and hence a contradiction. $\square$

Using both lemmas presented in this section, we obtain the following result.

**Corollary 6** *The point $c(Q)$ is the unique intersection point between $\pi_\omega$ and $\partial Q$.*

## 5 The algorithm

Recall that our objective is to design a data structure on $P$ to answer the following query: Given any disk $Q$, find the minimum $P$-circle with center on $Q$.

In the previous section we presented the relation existing between the query center of $Q$ and $\mathcal{V}(P)$. In this section, we use that relation to construct a data structure on $\mathcal{V}(P)$, that allow us to perform a binary search for $c(Q)$ along the paths contained in $\mathcal{V}(P)$.

### 5.1 Preprocessing

Compute $\mathcal{V}(P)$ and $c_P$ in $O(n \log n)$-time [13]. Assume that $\mathcal{V}(P)$ is stored as a binary tree with $n$ (unbounded) leaves, so that every edge and every vertex of the tree has a set of pointers to the vertices of $P$ defining it. Every vertex $p$ of $P$ has a pointer to $R(p)$ which is stored as a convex polygon. Construct a point location data structure on top of the farthest-point Voronoi diagram in $O(n \log n)$-time [8] so that we can answer furthest-point queries in $O(\log n)$-time. If $c_P$ is not a vertex of $\mathcal{V}(P)$, we insert it to $\mathcal{V}(P)$ by splitting the edge that it belongs to.

We will use an operation on the vertices of $\mathcal{V}(P)$ called PointBetween with the following properties. Given two vertices $u, v$ in $\pi_\omega$, PointBetween$(u, v)$ returns a vertex $z$ that splits the path on $\pi_\omega$ joining $u$ and $v$ into two subpaths. Moreover, if we discard the subpath that does not contain $c(Q)$ and we proceed recursively on the other, then, after $O(\log n)$ iterations, the search interval becomes only an edge of $\pi_\omega$ containing $c(Q)$.

A data structure that supports this operation was presented in [12]. This data structure can be constructed on top of $\mathcal{V}(P)$ in $O(n)$ time and uses linear space.

### 5.2 The search for $c(Q)$

Given a query disk $Q$ with center on $q$ and radius $r$, we present an algorithm to determine the position of $c(Q)$ in $O(\log n)$-time using the data structure described in the previous section. Let $p_Q$ be a point of $P$ such that $q$ belongs to $R(p_Q)$. To find $p_Q$, an $O(\log n)$-time point-location query on the farthest-point Voronoi diagram suffices.

Let $\ell$ be the line joining $q$ with $p_Q$ and let $\omega$ be the intersection of the boundary of $R(p_Q)$ with $\ell$. Since $R(p_Q)$ is a convex polygon, this intersection can be computed in $O(\log n)$-time. Let $Q'$ be the disk with center on $q$ and radius $d(q, \omega)$. By Corollary 6, $c(Q)$ is the unique intersection between $\pi_\omega$ and $\partial Q$. Thus, we search on $\pi_\omega$ for $c(Q)$ as follows:

The procedure PointBetween$(w, c_P)$ provides a point $z$ that splits $\pi_\omega$ into two subpaths. Let $\pi$ (resp. $\pi'$) be the subpath joining $z$ with $c_P$ (resp. $\omega$ with $z$) contained in $\pi_\omega$. If $z \in Q$ (resp. $z \notin Q$), then $c(Q)$ lies on $\pi$ (resp. $\pi'$). Thus, we can discard either $\pi$ or $\pi'$ and continue the search on the subpath containing $c(Q)$. We proceed until finding two consecutive vertices on $\pi_\omega$, such that the first one lies inside $Q$ but the second one does not. The details can be found in Algorithm 1.

**Algorithm 1** Algorithm to find $c(Q)$ given the path $\pi_\omega = (\omega = u_0, \dots, u_r = c_P)$

```
1:  Define the initial search interval:
    u ← u_0, v ← u_r.
2:  if uv is an edge of π_ω then
3:      Finish and report the segment s = [u, v].
4:  end if
5:  z ← PointBetween(u, v).
6:  if z ∈ Q then
7:      Move forward, let u ← z and return to step 2.
8:  else
9:      Move backwards, let v ← z and return to step 2.
10: end if
```

When our algorithm finishes, it reports an edge $s = [u, v]$ of the path $\pi_\omega$, such that $u$ is contained in $Q$ but $v$ is not. By Corollary 6, we conclude that $c(Q)$ is the intersection point between $s$ and $\partial Q$. Since the number of steps on this binary search is $O(\log n)$ [12] and since each step requires a constant number of operations, the overall running time of the algorithm is $O(\log n)$.

**Theorem 7** *After preprocessing a set $P$ of $n$ points in $O(n \log n)$-time, the minimum $P$-circle with center on a query disk $Q$ can be found in $O(\log n)$-time.*

## References


[1] P. Bose, S. Langerman, and S. Roy. Smallest enclosing circle centered on a query line segment. In *CCCG'08*, pages 167–170, 2008.

[2] P. Bose and G. Toussaint. Computing the constrained euclidean, geodesic and link centre of a simple polygon with applications. In *Computer Graphics International, 1996. Proceedings*, pages 102 –111, jun 1996.

[3] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. In *Proceedings of SODA*, pages 281–290, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[4] M. E. Dyer. On a multidimensional search technique and its application to the Euclidean one centre problem. *SIAM J. Comput.*, 15:725–738, August 1986.

[5] J. Elzinga and D. W. Hearn. Geometrical solutions for some minimax location problems. *Transportation Science*, 6(4):379–394, 1972.

[6] F. Hurtado, V. Sacristan, and G. Toussaint. Some constrained minimax and maximim location problems. *Studies in Locational Analysis*, 15:17–35, 2000.

[7] A. Karmakar, S. Roy, and S. Das. Fast computation of smallest enclosing circle with center on a query line segment. *Information Processing Letters*, 108(6):343 – 346, 2008.

[8] D. G. Kirkpatrick. Optimal search in planar subdivisions. Technical report, University of British Columbia, Vancouver, BC, Canada, Canada, 1981.

[9] D. Lee. Farthest neighbor Voronoi diagrams and applications. *Report 80-11-FC-04, Dept. Elect. Engrg. Comput. Sci.*, 1980.

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

[11] F. Preparata. Minimum spanning circle. *Steps in Computational Geometry*, pages 3–5, 1977.

[12] S. Roy, A. Karmakar, S. Das, and S. C. Nandy. Constrained minimum enclosing circle with center on a query line segment. *Computational Geometry*, 42(6-7):632 – 638, 2009.

[13] M. Shamos and D. Hoey. Closest-point problems. In *Proceedings of FOCS*, pages 151–162, Washington, DC, USA, 1975. IEEE Computer Society.

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