Optimal algorithms for constrained 1-center problems[⋆]

Luis Barba[1,2], Prosenjit Bose[1], and Stefan Langerman[2⋆⋆]

[1] Carleton University, Ottawa, Canada, jit@scs.carleton.ca
[2] Université Libre de Bruxelles, Brussels, Belgium, {lbarbafl,slanger}@ulb.ac.be

Abstract. We address the following problem: Given two subsets Γ and Φ of the plane, find the minimum enclosing circle of Γ whose center is constrained to lie on Φ. We first study the case when Γ is a set of n points and Φ is either a set of points, a set of segments (lines) or a simple polygon. We propose several algorithms, the first solves the problem when Φ is a set of m segments (or m points) in expected $\Theta((n+m)\log\omega)$ time, where $\omega = \min\{n, m\}$. Surprisingly, when Φ is a simple m-gon, we can improve the expected running time to $\Theta(m + n\log n)$. Moreover, if Γ is the set of vertices of a convex n-gon and Φ is a simple m-gon, we can solve the problem in expected $\Theta(n + m)$ time. We provide matching lower bounds in the algebraic computation tree model for all the algorithms presented in this paper. While proving these results, we obtained a $\Omega(n\log m)$ lower bound for the following problem: Given two sets A and B in $\mathbb{R}$ of sizes m and n, respectively, decide if A is a subset of B.

Keywords: minimum enclosing circle, facility location problems

1 Introduction

Let P be a set of n points in the plane. The minimum enclosing circle problem, originally posed by Sylvester in 1857 [17], asks to identify the center and radius of the minimum enclosing circle of P. For ease of notation we say that every circle containing P is a P-circle. Several independent solutions were proposed to solve the problem in $O(n\log n)$ time [10,15,16]. Megiddo [14] settled the complexity of this problem and presented a $\Theta(n)$-time algorithm using prune and search.

Finding the minimum P-circle is also known as the 1-center facility location problem: Given the position of a set of clients (represented by P), compute the optimal location for a facility such that the maximum distance between a client and its closest facility is minimized. The aforementioned algorithms provide solutions to this problem. However, in most situations the location of the facility is constrained by external factors such as the geography and features of the terrain. Therefore, the study of constrained versions of the 1-center problem is of importance and has received great attention from the research community [3,4,5,8].

Megiddo [14] proposed a linear time algorithm to find the minimum P-circle whose center is constrained to lie on a given line. Extending these ideas, Hurtado

[⋆] Research supported in part by NSERC.

[⋆⋆] Directeur de recherches du F.R.S.-FNRS.

et al. [8] presented a $\Theta(n+m)$-time algorithm to find the minimum P-circle whose center is constrained to lie inside a convex m-gon. Bose and Toussaint [4] generalized this result by restricting the center of the P-circle to lie inside a simple m-gon Q. They proposed an $O((n+m)\log(n+m)+k)$-time algorithm, where k is the number of intersections of Q with the farthest-point Voronoi diagram of P. The dependency on k was later removed from the running time [5].

Bose *et al.* [3] addressed the query version of the problem and proposed an $O(n\log n)$-time preprocessing on P, that allows them to find the minimum P-circle with center on a given query line in $O(\log n)$ time. Using this result, they showed how to compute the minimum P-circle, whose center is constrained to lie on a set of m segments, in $O((n+m)\log n)$ time. However, when $m = O(1)$, the problem can be solved in $O(n)$ time by using Megiddo's algorithm [14] a constant number of times. Moreover, when $n = O(1)$, the problem can be solved in $O(m)$ time by finding the farthest point of P from every given segment. Therefore, one would expect an algorithm that behaves like the algorithm presented in [3] when $m = O(n)$ but that converges to a linear running time as the difference between n and m increases (either to $O(n)$ or to $O(m)$). In this paper we show that such an algorithm exists and prove its optimality. When constraining the center on a simple m-gon however, the order of the vertices along its boundary allows us to further speed up our algorithm, provided that m is larger than n.

Let M be a set of m points, let S be a set of m segments and let Q be a simple polygon on m vertices. We say that a P-circle C has its center on M, on S or on Q if the center of C is either a point of M, lies on a segment of S or belongs to Q, respectively. The (P,M)*-optimization problem* asks to find the minimum P-circle with center on M. Given a radius r, the $(P,M)_r$*-decision problem* asks if there is a P-circle of radius r with center on M. Analogous problems exist for S and Q. In Section 2, we show a $\Theta((n+m)\log\omega)$-time algorithm for the $(P,S)_r$-decision problem where $\omega = \min\{n,m\}$. In Section 3 we transform it to solve the (P,S)-optimization problem with the same running time. In Section 4, we show a matching lower bound in the algebraic computation tree model provided that $n \leq m$ and the restriction is on a set of points, lines, segments or even on a simple polygon. When $m > n$ however, we only prove a matching lower bound when the center is restricted to be on a set of points, segments or lines, yet the lower bound breaks down when the restriction is on a simple polygon. Indeed, given a simple m-gon Q, we show an $\Theta(m + n\log n)$-time algorithm for the (P,Q)-optimization problem. To put this in perspective, note that whenever $m = \Omega(n\log n)$, the algorithm runs in $\Theta(m)$ time. Since the bottleneck of this algorithm is the computation of the farthest-point Voronoi diagram, if we assume that P is the set of vertices of a given convex n-gon we can reduce the running time to $\Theta(m+n)$ [1,11]. Finally, we show a matching lower bound for these algorithms, thereby solving the problem for all ranges of n and m and all possible restrictions on points, lines, segments and simple polygons.

As a side note, while proving these lower bounds, we stumbled upon the following problem: Given two sets A and B of $\mathbb{R}$ of sizes m and n, respectively, is $A \subseteq B$? While a lower bound of $\Omega(n\log n)$ is known in the case where $n = m$

[2], no lower bounds were known when m and n differ. Using a method of Yao [18] and the topology of affine subspace families, we were able to prove an $\Omega(n \log m)$ lower bound, even when A is restricted to be a sorted set of m real numbers.

Although the main techniques used in this paper have been around for a while [7,12], they are put together in a different way in this paper, showing the potential of several tools that were not specifically designed for this purpose. Furthermore, these results provide significant improvements over previous algorithms when n and m differ widely as it is the case in most applications. Due to space constraints, in this extended abstract we provide only proof sketches. The full version of this paper is included as an appendix.

Preliminaries. Given a subset X of the plane, the interior and convex hull of X are denoted by $\text{INT}(X)$ and $\text{CH}(X)$, respectively. A point x is *enclosed* by a circle C if $x \in \text{CH}(C)$; otherwise we say that x is *excluded* by C. An X-*circle* is a circle that encloses every point of X.

Given a point $x \in \mathbb{R}^2$, let $\bigcirc_r(x)$ be the circle with radius r and center on x. Let P be a set of n points in $\mathbb{R}^2$. Given $W \subseteq P$, let $\Lambda_r(W) = \bigcap_{p \in W} \text{CH}(\bigcirc_r(p))$, i.e., the intersection of every disk of radius r with center on a point of W. Notice that $\Lambda_r(W)$ is a convex set whose boundary is composed of circular arcs each with the same curvature. A point $p \in W$ *contributes* to $\Lambda_r(W)$ if there is an arc of the circle $\bigcirc_r(p)$ on the boundary of $\Lambda_r(W)$. We refer to this arc as the *contribution of* p to $\Lambda_r(W)$. As the curvature of all circles is the same, a point contributes with at most one arc to the boundary of $\Lambda_r(W)$.

Given two subsets X and Y of the plane, let $B_X(Y)$ be the minimum X-circle with center on Y and let $b_X(Y)$ denote its center. If $X = P$, we let $\rho(Y)$ denote the radius of $B_P(Y)$, i.e., $\rho(Y)$ is the radius of the minimum P-circle with center on Y. Let C_P be the minimum P-circle, c_P be its center and let r_P be its radius.

Observation 1 *Given a point $x \in \mathbb{R}^2$ and a real number $r \geq r_P$, $\rho(x) \leq r$ if and only if $x \in \Lambda_r(P)$.*

2 Solving the decision problem on a set of segments

Let S be a set of m segments and let $r > r_P$. In this section we present an $O((n+m)\log \omega)$ time algorithm to solve the $(P,S)_r$-decision problem for the given radius r, where $\omega = \min\{n, m\}$. Notice that by Observation 1, if we could compute $\Lambda_r(P)$, we could decide if there is a P-circle of radius r with center on S by checking if there is a segment of S that intersects $\Lambda_r(P)$. However, we cannot compute $\Lambda_r(P)$ explicitly as this requires $\Omega(n \log n)$ time. Thus, we approximate it using ε-nets and use it to split both S and P into a constant number of subsets each representing a subproblem of smaller size. Using divide and conquer we determine if there is an intersection between S and $\Lambda_r(P)$ by solving the decision problem recursively for each of the subproblems. The algorithm runs in $O(\min\{\log n, \log m\})$ phases and on each of them we spend $O(n+m)$ time.

The algorithm. Initially, compute the minimum P-circle C_P, its center c_P and its radius r_P in $O(n)$ time [14]. In $O(m)$ time we can verify if c_P lies on a segment of S. If it does, then C_P is the minimum P-circle with center on S. Otherwise, as we assume from now on, the radius of $B_P(S)$ is greater than r_P.

Consider a family of convex sets $\mathcal{G}$ defined as follows. A set $G \in \mathcal{G}$ is the intersection of $A_1 \cap \ldots \cap A_6$, where each A_i is either the interior of a circle or an open halfplane supported by a straight line (A_i may be equal to A_j for some $i \neq j$). Given a family $\mathscr{Y}$ of geometric objects in the plane (segments, lines or points), we define a set of ranges on $\mathscr{Y}$ as follows. For each $G \in \mathcal{G}$, let $G_{\mathscr{Y}} = \{y \in \mathscr{Y} : G \cap y \neq \emptyset\}$ and let $\mathcal{G}_{\mathscr{Y}} = \{G_{\mathscr{Y}} : G \in \mathcal{G}\}$ be the family of subsets of $\mathscr{Y}$ induced by $\mathcal{G}$.

Fix a constant $0 < \varepsilon \leq 1$ and consider the range space defined by S and $\mathcal{G}_S$. As the VC-dimension of this range space is finite, we can compute an ε-net N_S of $(S, \mathcal{G}_S)$ of size $O(1)$ in $O(m)$ time [13], i.e., any convex set $G \in \mathcal{G}$ that intersects more than εm segments of S must intersect at least one segment of N_S.

For each segment s of N_S, compute $B_P(s)$ in $O(n)$ time [14] and mark three points of P that uniquely define this circle by lying on its boundary. Let r_{min} be the radius of the minimum circle among the computed P-circles. If $r_{min} \leq r$, then there is a positive answer to the $(P,S)_r$-decision problem and the decision algorithm finishes. Otherwise, let $P^0 \subset P$ be the set of marked points and note that $|P^0| \leq 3|N_S| = O(1)$. By the minimality of r_{min}, any point in the interior of $\Lambda_{r_{min}}(P^0)$ is at distance at least r_{min} from the segments of N_S. That is, the interior of $\Lambda_{r_{min}}(P^0)$ intersects no segment of N_S. As $r_{min} > r$, we know that $\Lambda_r(P^0) \subset \Lambda_{r_{min}}(P^0)$ and hence $\Lambda_r(P^0)$ intersects no segment of N_S.

We refine this intersection using another ε-net. Let $\mathcal{C} = \{\bigcirc_r(p) : p \in P\}$ be the set of circles of radius r centered at the points of P. Compute an ε-net N_P of the range space $(\mathcal{C}, \mathcal{G}_{\mathcal{C}})$ in $O(n)$ time [13]. That is, if a convex set $G \in \mathcal{G}$ intersects more than εn circles of $\mathcal{C}$, then G intersects at least one circle of N_P.

Let $P^1 = \{p \in P : \bigcirc_r(p) \in N_P\}$ i.e., P^1 is the subset of P defining N_P where $|P^1| = O(1)$. Notice that for every $p \in P^1$, $\Lambda_r(P^1)$ is enclosed by $\bigcirc_r(p)$, i.e. the circle $\bigcirc_r(p)$ does not intersect the open set $\Lambda_r(P^1)$. Let $P^+ = P^0 \cup P^1$, as $\Lambda_r(P^+)$ is contained in both $\Lambda_r(P^0)$ and $\Lambda_r(P^1)$, we observe the following.

Lemma 1. *No segment of N_S and no circle of N_P intersects $int(\Lambda_r(P^+))$.*

We assume the existence of a set of points $\Pi \supset P^+$ of constant size such that Π inherits the properties of P^+. This set and its properties will be described later.

Because $r > r_P$, c_P is enclosed by $\bigcirc_r(p)$ for every $p \in P$ and hence, c_P lies in the interior of $\Lambda_r(\Pi)$. Therefore, we can consider $k = O(1)$ rays with apex at c_P that pass through some of the vertices along the boundary of $\Lambda_r(\Pi)$. These rays split the plane into k cones $\mathcal{D} = \{\Delta_1, \ldots, \Delta_k\}$.

For every $1 \leq i \leq k$, let $R_i = \Delta_i \cap \Lambda_r(\Pi)$ be a "slice" of $\Lambda_r(\Pi)$; see Fig. 1. By constructing

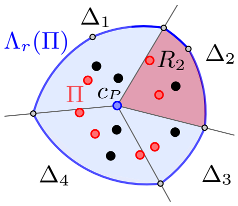

Fig. 1. The set $\Pi \subset P$ is shown in red. The vertex set of $\Lambda_r(\Pi)$ is used to split the plane into cones $\Delta_1, \ldots, \Delta_4$ by shooting rays from c_P. The "slice" R_2 is the portion of $\Lambda_r(\Pi)$ inside Δ_2.

these cones such that each contains at most four vertices of $\Lambda_r(\Pi)$, we guarantee that each element in $\mathcal{R} = \{R_1, \ldots, R_k\}$ is a convex region of the family $\mathcal{G}$ used to define the ε-nets. Because $P^+ \subset \Pi$, we know that $\Lambda_r(\Pi) \subset \Lambda_r(P^+)$. Therefore, by Lemma 1 the interior of each region R_i intersects no segment of N_S and no circle of N_P. Because N_S and N_P are both ε-nets, we obtain the following.

Lemma 2. *For each $1 \leq i \leq k$, at most εm segments of S intersect the region R_i and at most εn circles of $\mathcal{C}$ intersect R_i.*

Due to space constraints, we omit a full description of Π. However, we provide a summary of its properties and a sketch of its construction.

Lemma 3. *We can construct $\Pi \supset P^+$ and the partition cones $\mathcal{D} = \{\Delta_1, \ldots, \Delta_k\}$ in $O(n)$ time such that:* (1) *for every $s \in S$, if s intersects $\Lambda_r(\Pi)$, then $s \cap \Lambda_r(\Pi)$ is contained in exactly one cone of $\mathcal{D}$, and* (2) *for any point $p \in P$, if p contributes to $\Lambda_r(P)$, then its contribution is contained in exactly one cone of $\mathcal{D}$.*

Proof sketch. For a given direction, we can shoot a ray from c_P in that direction and compute the first circle from $\mathcal{C}$ that this ray intersects. Thus, this circle defines an actual arc of the boundary of $\Lambda_r(P)$ in the direction of the ray. Moreover, we can compute its neighboring arc along this boundary, i.e., we can find an actual vertex of $\Lambda_r(P)$ and the points of P that define it. By doing this for a constant number of directions, given by the vertices of $\Lambda_r(P^+)$, we obtain a constant size subset Y of the vertices of $\Lambda_r(P)$. Using this vertices to shoot the rays from c_P, we construct $\mathcal{D}$ and ensure that a point $p \in P$ will contribute to $\Lambda_r(P)$ inside only one of these cones. Moreover, the segments of S cannot cross the segment connecting a vertex of Y with c_P. Therefore, using convexity arguments we show that if a segment $s \in S$ intersects $\Lambda_r(\Pi)$, then it can do it in at most one cone of $\mathcal{D}$. □

The idea is to use divide and conquer using Lemma 2. That is, we split both P and S into k subsets according to their intersection with the elements of $\mathcal{R}$, where each pair represents a subproblem. Finally, we prove that the $(P,S)_r$-decision problem has a positive answer if and only if some subproblem has a positive answer.

Lemma 4. *We can compute sets $S_1, \ldots, S_k \subset S$ in $O(m)$ time such that $|S_i| < \varepsilon m$ and $\sum_{i=1}^{k} |S_i| \leq m$. Moreover, S_i contains all segments of S that intersect $\Lambda_r(\Pi)$ inside R_i.*

Proof sketch. Let S_i be the set of segments of S that intersect R_i. The construction of $S_1, \ldots, S_k$ can be performed in $O(m)$ time since the size of R_i is constant. By Lemma 3, a segment of S belongs to at most one set of the partition and hence, $\sum_{i=1}^{k} m_i \leq m$. Moreover, for any $1 \leq i \leq k$ at most εm segments of S intersect R_i by Lemma 2. Consequently, $|S_i| < \varepsilon m$. □

Lemma 5. *We can compute sets $P_1, \ldots P_k \subset P$ in $O(n)$ time such that $|P_i| < \varepsilon n$, $\sum_{i=1}^{k} |P_i| \leq n$ and $\Lambda_r(P) = \Lambda_r(P_1 \cup \ldots \cup P_k)$. Moreover, if a point $p \in P_i$ contributes to $\Lambda_r(P)$, then this contribution intersects R_i.*

Proof sketch. Let $P_i = \{p \in P : \bigcirc_r(p)$ intersects $R_i\}$. This partition of P can be computed in $O(n)$ time as the size of R_i is constant. For each point $p \in P$ that contributes to $\Lambda_r(P)$, $\bigcirc_r(p)$ has to intersect $\Lambda_r(\Pi)$. Hence, the contribution of p to $\Lambda_r(P)$ intersects at least one region of $\mathcal{R}$. By Lemma 3, a point of P contributes to $\Lambda_r(\Pi)$ inside only one cone of $\mathcal{D}$, i.e., a point of P belongs to at most one of the computed sets. By Lemma 2, at most εn circles of $\mathcal{C}$ intersect R_i, i.e., $|P_i| < \varepsilon n$. □

Theorem 2. *The $(P, S)_r$-decision problem has positive answer if and only if there is a P_i-circle of radius r with center on S_i for some $1 \leq i \leq k$.*

Proof sketch. Let C be a P-circle of radius r with center c lying on a segment $s \in S$. Since the cones of $\mathcal{D}$ partition the plane, c belongs to some cone Δ_i for some $1 \leq i \leq k$, i.e., s intersects the cone Δ_i. By Observation 1, s intersects $\Lambda_r(P) \subset \Lambda_r(\Pi)$. Consequently, by Lemma 4, s belongs to S_i. Assume that c lies on the arc being the contribution of some point $p \in P$. Because c is in Δ_i and on the boundary of $\Lambda_r(P) \subset \Lambda_r(\Pi)$, $\bigcirc_r(p)$ intersects $\Lambda_r(\Pi) \cap \Delta_i = R_i$. Thus, by Lemma 5 p belongs to P_i, i.e., C is a P_i-circle of radius r with center on S_i. The other implication is similar and can be found in the full version of the paper. □

By Lemmas 4 and 5, in $O(n + m)$ time we can either give a positive answer to the decision algorithm, or compute sets $P_1, \ldots, P_k$ and $S_1, \ldots, S_k$ in order to define k decision subproblems each stated as follows: Decide if there is a P_i-circle of radius r with center on S_i. Because Theorem 2 allows us to solve each subproblem independently, we proceed until we find a positive answer on some branch of the recursion, or until either P_i or S_i reaches $O(1)$ size and can be solved in linear time. Since $|S_i| < \varepsilon m$ and $|P_i| < \varepsilon n$ by Lemmas 4 and 5, the number of recursion steps needed is $O(\min\{\log n, \log m\})$. Furthermore, by Lemmas 4 and 5, the size of all subproblems at the i-th level of the recursion is bounded above by $n + m$.

Lemma 6. *Given sets P of n points and S of m segments and $r > 0$, the $(P, S)_r$-decision problem can be solved in $O((n + m) \log \omega)$ time, where $\omega = \min\{n, m\}$.*

3 Converting decision to optimization

In the previous section, we showed an algorithm for the $(P, S)_r$-decision problem. However, our main objective is to solve its optimization version. To do that, we use the technique presented by Chan [6]. This technique requires an efficient algorithm to partition the problem into smaller subproblems, where the global solution is the minimum among the subproblems solutions. By presenting an $O(n + m)$-time partition algorithm, we obtain a randomized algorithm for the (P, S)-optimization problem having an expected running time of $O((n+m) \log \omega)$, where $\omega = \min\{n, m\}$. As the partition of the plane into cones used in the previous section has no correlation with the structure of $\Lambda_r(P)$ as r changes, the partition of P used in this section requires a different approach. However, the partition of S is very similar.

Lemma 7. *We can compute sets $P'_1, \dots, P'_h \subset P$ and $S'_1, \dots, S'_h \subset S$ in $O(n+m)$ time such that $|P'_i| < \varepsilon n$, $|S'_i| < \varepsilon m$ and $B_P(S)$ is the circle of minimum radius among the elements in the set $\{B_{P'_1}(S'_1), \dots, B_{P'_h}(S'_h)\}$.*

Proof sketch. Given any subset of $\mathbb{R}^2$, it can be *embedded* into $\mathbb{R}^3$ by identifying $\mathbb{R}^2$ with the plane $Z_0 = \{(x, y, z) \in \mathbb{R}^3 : z = 0\}$. As a first step embed P into $\mathbb{R}^3$. Given a point $p \in P$, let γ_p be the boundary of the 3-dimensional cone, lying above p, with apex on p and $45\,^\circ$ slope with respect to the plane Z_0.

Consider an $O(1)$ size sample P^+ of P whose properties will be specified later and let $\mathit{\Gamma} = \{\gamma_p : p \in P^+\}$. Construct the farthest-point Voronoi diagram of P^+ and triangulate it. Then, compute $\mathit{\Lambda}_r(P^+)$ and pseudo-triangulate it by joining c_P with every vertex on its boundary. By choosing P^+ carefully, we can guarantee that at most εm segments of S intersect each of the pseudo-triangles.

Let $\mathcal{T}$ be the geometric graph obtained as the union of the triangulation of the farthest-point Voronoi diagram of P^+ and the pseudo-triangulation of $\mathit{\Lambda}_r(P^+)$. Then, we embed $\mathcal{T}$ in the plane Z_0. Since $|P^+| = O(1)$, the size of $\mathcal{T}$ is also constant. Recall that the furthest-point Voronoi diagram of P^+ is the upper envelope $\mathcal{U}$ of $\mathit{\Gamma}$ when projected onto the plane Z_0. That is, a point $x \in Z_0$ is farther from p if and only if γ_p is the last cone intersected by a ray shooting upwards, orthogonally to the plane Z_0, from x. Let $\mathcal{U}^+$ be the set of points lying strictly above $\mathcal{U}$.

Consider the vertical lifting of $\mathcal{T}$, which is simply the union of the vertical lines passing through the points on every edge of this triangulation. This vertical lifting partitions $\mathbb{R}^3$ into $O(1)$ solid prisms each defined by the intersection of at most three vertical halfspaces or cylinders. Finally, intersect each of these prisms with $\mathcal{U}^+$ to obtain a family of convex regions $\mathcal{Y} = \{Y_1, \dots, Y_h\}$ for some $h \in O(1)$. Since $\mathcal{U}^+$ intersects no cone of $\mathit{\Gamma}$, no region of $\mathcal{Y}$ intersects the boundary of a cone in $\mathit{\Gamma}$. By choosing P^+ carefully, we can guarantee that for each $1 \le i \le h$, at most εn cones of $\mathit{\Gamma}$ intersect Y_i. Moreover, we can also guarantee that the vertical lifting of at most εm segments of S intersect each Y_i. For every $1 \le i \le h$, let $P'_i = \{p \in P : \gamma_p \cap Y_i \neq \emptyset\}$ and note that P'_i can be computed in $O(n)$ time. Let $S'_i = \{s \in S : \text{the vertical lifting of } s \text{ intersects } Y_i\}$ and note that S'_i can be computed in $O(m)$ time. Moreover, we have that $|P'_i| < \varepsilon n$ and $|S_i| < \varepsilon m$.

Recall that $b_P(S)$ is the center of the minimum P-circle $B_P(S)$ with center on S. Let $s^* \in S$ be the segment where $b_P(S)$ lies and let $p \in P$ be a point on the boundary of $B_P(S)$. We claim that s^* and p belong to the same subproblem, i.e., belong to S'_j and P'_j, respectively, for some $1 \le j \le k$. Notice that if this claim is true, then all the points of P through which $B_P(S)$ passes belong to P'_j. That is, $B_{P'_j}(S'_j)$ and $B_P(S)$ are defined as the circles with center on s^* passing through the same set of points, i.e., $B_{P'_j}(S'_j) = B_P(S)$. Thus, by computing $B_{P'_i}(S'_i)$ for each $1 \le i \le k$, the minimum P-circle with center on S can be obtained by choosing the minimum among $B_{P'_1}(S'_1), \dots, B_{P'_h}(S'_j)$.

We proceed to prove that s^* and p belong to the same subproblem. Since $B_P(S)$ contains every point in P^+, $b_P(S)$ lies in $\mathit{\Lambda}_r(P^+)$. Therefore, $b_P(S)$ lies inside the projection of Y_j for some $1 \le j \le h$. i.e, $s^* \in S'_j$. Consider the ray σ shooting upwards (perpendicular to Z_0) from $b_P(S)$. Since p lies on the boundary of

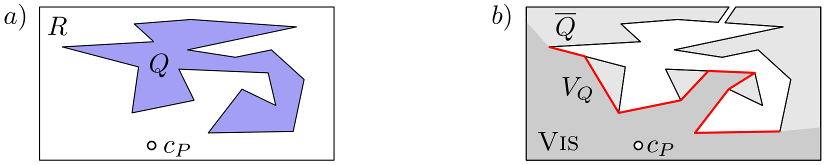

Fig. 2. *a*) A simple polygon Q and a rectangle R enclosing both c_P and Q. By removing Q from R we obtain a polygon with one hole. *b*) The simple polygon $\overline{Q}$ obtained by connecting the hole with the boundary of R. In red, the visible chain V_Q of $\overline{Q}$ from c_P.

$B_P(S)$, p is farther away from $b_P(S)$ than any other point of P. That is, γ_p is the last cone intersected by σ. Therefore, γ_p intersects Y_j and consequently $p \in P'_j$. □

By Lemmas 6 and 7, we can use Chan's technique [6] to obtain the following.

Theorem 3. *Given a set P of n points and a set S of m segments in the plane, the (P, S)-optimization problem can be solved in expected $O((n + m) \log \omega)$ time where $\omega = \min\{n, m\}$.*

When constraining to a simple m-gon, the sequence of points along its boundary allows us to improve upon Theorem 3 provided that $m \geq n$.

Theorem 4. *Given a set P of n points and a simple polygon Q of m vertices, the (P, Q)-optimization problem can be solved in expected $O(m + n \log n)$ time.*

Proof sketch. If c_P lie inside Q, then C_P is the solution to our problem. Therefore, we assume that c_P lies outside Q. In this case, we allow ourselves to compute the farthest-point Voronoi diagram of P explicitly in $O(n \log n)$ time. Using this structure, we can compute $\Lambda_r(P)$ in $O(n)$ time for any given value of $r > r_P$ which is key to the speed up of the $(P, Q)_r$-decision algorithm.

In $O(m)$ time, compute a rectangle R sufficiently large to enclose Q and c_P in its interior. Let $\overline{Q} = R - \text{INT}(Q)$ which is a polygon with one hole. This polygon can be turned into a simple polygon by adding a thin corridor connecting the hole with the exterior in such a way that no point on this corridor is visible from c_P. In this way, c_P lies in the interior of $\overline{Q}$; see Fig. 3 for an illustration.

Compute the visibility polygon VIS of $\overline{Q}$ from c_P in $O(m)$ time using the algorithm from Joe and Simpson [9]. Finally, let V_Q be the polygonal chain obtained by removing the edges of the boundary of VIS that have an endpoint lying on the boundary of the rectangle R (there may be none). Because the boundary of $\Lambda_r(P)$ is a Jordan curve, $\Lambda_r(P)$ intersects Q if and only if $\Lambda_r(P)$ intersects V_Q.

A polygonal chain is *star-shaped* if there exists a set of points called its *kernel* such that every point on this chain is visible from every point in its kernel. Note that V_Q is a star-shaped polygonal chain with c_P in its kernel. Thus, since $\Lambda_r(P)$ can be computed in $O(n)$ time from the Voronoi diagram of P, we can decide if V_Q intersect $\Lambda_r(P)$ in $O(n + m)$ time. Hence, we can solve the $(P, Q)_r$-decision problem in linear time. By considering the set of segments along the boundary of Q, we can use Lemma 7 to construct $O(1)$ subproblems such that the solution to the (P, Q)-optimization problem is the minimum among the subproblems solutions. Consequently, we can use Chan's technique [6] to obtain our result. □

Because the bottleneck of this algorithm is the construction of the farthest-point Voronoi diagram, whenever P is the set of vertices of a convex polygon, we can compute its farthest-point Voronoi diagram in linear time [11,1].

Corollary 1. *Let N be a convex n-gon and let Q be a simple m-gon. The minimum enclosing circle of N, whose center is constrained to lie on Q, can be found in expected $\Theta(m+n)$ time.*

4 Lower bounds

We prove lower bounds for the decision problems: We show inputs where the decision problem is equivalent to answering a membership query in a set with "many" disjoint components. We then use Ben-Or's Theorem [2] to obtain lower bounds for any decision algorithm that solves this membership problem.

Lemma 8. *Let P be a set of n points and a let M be a set of m points (m segments or m lines). Given a radius r, the $(P, M)_r$-decision problem has a lower bound of $\Omega(m \log n)$ in the algebraic computation tree model.*

Proof sketch. We construct a set of points P such that for any point set M, with certain constraints, the $(P, M)_r$-decision problem has a lower bound of $\Omega(m \log n)$.

Let $r > 0$ and let r' be a number such that $0 < r' < r$. Let P be the set of vertices of a regular n-gon circumscribed on a circle of radius r'. Because $r > r'$, $\Lambda_r(P)$ is a non-empty convex region whose boundary is composed of circular arcs. Notice that by Observation 1, the decision algorithm has an affirmative answer if and only if there is a point of M lying in $\Lambda_r(P)$. Let C be the circumcircle of the vertices of $\Lambda_r(P)$. Partition this circle into $\varphi_1 = C \cap \Lambda_r(P)$ and $\varphi_0 = C - \varphi_1$. Because φ_1 consists of exactly n points being the vertices of $\Lambda_r(P)$, φ_0 consists of n disconnected open arcs all lying outside of $\Lambda_r(P)$; see Fig. 3(a). Moreover, a point on C supports a P-circle of radius r if and only if it lies on φ_1.

Consider the restriction of the decision problem where M is constrained to lie on C. Notice that any lower bound for this restricted problem is also a lower bound for the general decision problem. Because an input on m points for this restricted problem can be seen as a point in $\mathbb{R}^{2m}$, its input space defines a subspace $C^m = C \times \ldots \times C$ of $\mathbb{R}^{2m}$. Moreover, this set of points can be split into two regions, the "yes" and the "no" region (with respect to the decision problem), where the "no" region is equal to φ_0^m. That is, a point $(x_1, y_1, x_2, y_2, \ldots, x_m, y_m)$ lies in the "no" region φ_0^m if for every index $1 \leq j \leq m$, the point (x_j, y_j) lies inside $\varphi_0 \subset C$.

Because φ_0 contains n disjoint components, φ_0^m contains $O(n^m)$ disjoint components being the product-space of m copies of φ_0. Recall that the $(P, M)_r$-decision problem is equivalent to answering if the input, seen as a point in $\mathbb{R}^{2m}$, lies in the "no" region. Therefore, by Ben-Or's Theorem [2] we obtain a lower bound of $\Omega(m \log n)$ for every decision algorithm in the algebraic computation tree model. □

Lemma 9. *Let P be a set of n points and a let Q be a simple polygon on m vertices such that $m \geq n$. Given a radius r, the $(P, Q)_r$-decision problem has a lower bound of $\Omega(m + n \log n)$ in the algebraic computation tree model.*

Proof sketch. In this proof, we construct a simple m-gon Q such that for any input P on n points, the $(P, Q)_r$-decision problem has a lower bound of $\Omega(m+n\log n)$.

Let $r > 0$ and let $N = \{p_1, \ldots, p_n\}$ be the set of vertices of a regular n-gon whose circumcircle C has radius smaller than r and center on c. Let $\varepsilon > 0$ and let $r_\varepsilon = r + \varepsilon$. Because r_ε is greater than the radius of C, $\Lambda_{r_\varepsilon}(N)$ is non-empty. Consider the middle points of every arc along $\Lambda_{r_\varepsilon}(N)$ and label them so that m_i is the middle point on the arc opposite to p_i. Let C' be any circle with center on c and radius greater than r_ε. For every $1 \leq i \leq n$, let q_i be the intersection point of C' with the ray shooting from c that passes through p_i. Let Q' be a star-shaped polygon with vertex set $\{m_1, \ldots, m_n\} \cup \{q_1, \ldots, q_n\}$ where edges connect consecutive vertices in the radial order around c; see Fig. 3(b).

Let R be a sufficiently large rectangle to enclose Q' and let $Q = R \setminus \textsc{int}(Q')$. Remove the star-shaped hole of Q by connecting the boundary of R with an edge of Q using a small corridor; see Fig. 3(c) for an illustration.

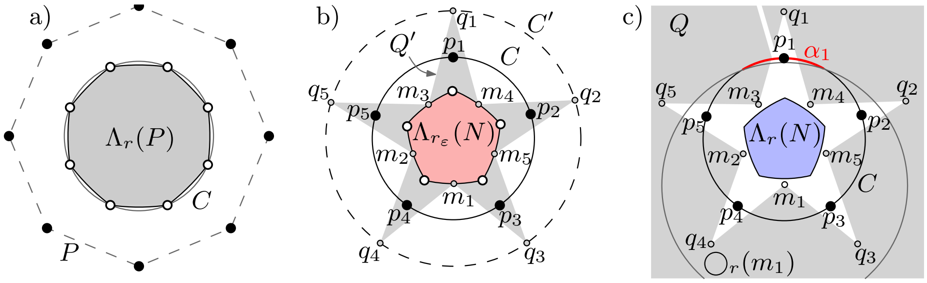

Fig. 3. a) The construction presented in Lemma 8. b) The construction of the star-shaped polygon Q' used in Lemma 9. c) The polygon Q constructed in Lemma 9 being disjoint from $\Lambda_r(P)$. The arc α_1 (in red) is the arc of C excluded by $\bigcirc_{r_\varepsilon}(m_1)$.

Consider the restriction of the decision problem where every point of P is constrained to lie on C. Note that any lower bound for this restricted problem is also a lower bound for the general problem. Recall that every input on n points constrained to lie on C can be indistinctly seen as a point in $C^n \subset \mathbb{R}^{2n}$ and vice versa. Let γ_0 be a subset of C^n such that $(x_1, y_1, \ldots, x_n, y_n) \in \gamma_0$ if and only if the decision problem on Q with input $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and radius r has a negative answer. Let $\gamma_1 = C^n - \gamma_0$. Because $r < r_\varepsilon$, $\Lambda_r(N)$ doesn't intersect Q, i.e., $N \in \gamma_0$. Note that every point of N lies inside $\bigcirc_r(m_i)$ except for p_i, i.e., there is a portion of C excluded by $\bigcirc_r(m_i)$. For every $1 \leq i \leq n$, let α_i be the arc of C excluded by the circle $\bigcirc_r(m_i)$ where p_i lies on α_i.

By letting ε sufficiently small, α_i is disjoint from α_j for any $i \neq j$. Moreover, every point lying on $C \setminus \alpha_i$ is enclosed by $\bigcirc_r(m_i)$. Therefore, if an input P on n points has no point lying on α_i for some $1 \leq i \leq n$, then $\bigcirc_r(m_i)$ is a P-circle of radius r, i.e., P belongs to γ_1. Notice that every permutation of the same input of n points induces a different point in $\mathbb{R}^{2n}$. That is, a set of n points in the plane can be represented by $n!$ different points in $\mathbb{R}^{2n}$. Recall that each arc α_i has exactly one point of N (say p_i) on it. Because Q is disjoint from $\Lambda_r(N)$, every one of the $n!$ points representing N in $\mathbb{R}^{2n}$ lies in γ_0. In the full version of the paper, we show that if P_0 and P_1 are two points in C^n representing different permutations of P, then they are in disjoint connected components of γ_0. The idea is that any continuous transformation from P_0 to P_1 will reach a state in which an arc α_i is empty of input points, meaning that the it belongs to γ_1. Thus, as there are $n!$ permutations of N and each of them belongs to a different connected component in γ_0, γ_0 contains at least $n!$ disjoint connected regions. By Ben-Or's Theorem [2] we obtain a lower bound of $\Omega(n \log n)$ for the restricted $(P, M)_r$-decision problem in the algebraic computation tree model. To obtain this lower bound, the m-gon Q needs to have at least n vertices, i.e., $m \geq n$.

Finally, notice that any decision algorithm has also a lower bound of $\Omega(m)$ since every vertex of Q has to be considered. Otherwise, an adversary could perturb a vertex so that the solution switches to from a negative to a positive answer without affecting the execution of the algorithm. □

4.1 Another lower bound when constraining to sets of points.

Let A and B be two sets of m and n numbers in $[0, 1]$ such that $m \leq n$ and A is sorted in increasing order. The *A-B-subset problem* asks if A is a subset of B.

In the extended version of this paper, we show that any A-B-subset problem can be reduced in linear time to a $(P, Q)_r$-decision problem for some simple polygon Q. Hence, any lower bound for the A-B-subset problem is a lower bound for the $(P, Q)_r$-decision problem. In fact, the lower bound for the A-B-subset problem considers arbitrary sets of real numbers. Furthermore, the lower bound holds when A is given as a fixed sorted set prior to the design of the algorithm. Note that a set of n numbers can be represented by a point in $\mathbb{R}^n$ and vice versa.

Lemma 10. *Let n, m be two integers such that $n \geq m$. For any $A \in \mathbb{R}^m$ such that A is given in sorted order, there is a lower bound of $\Omega(n \log m)$ in the algebraic computation tree model for the A-B-subset problem given any $B \in \mathbb{R}^n$.*

Proof sketch. Let $A = \{a_1, \ldots, a_m\}$ be a sorted set of m real numbers and think of it as a point in $\mathbb{R}^m$. Let γ_1 be the subspace of $\mathbb{R}^n$ containing all points representing a set B such that $A \subseteq B$, i.e., the "yes" region.

An A-constraint is an equation of the form $(x_i = a_j)$ for some $1 \leq i \leq n$, $1 \leq j \leq m$. Two A-constraints $(x_i = a_j)$ and $(x_h = a_k)$ are compatible if $i \neq h$ (j may be equal to k). A point $X \in \mathbb{R}^n$ *satisfies* an A-constraint $(x_i = a_j)$ if its i-th coordinate is equal to a_j. A set φ of pairwise compatible A-constraints is *complete* if it contains exactly one A-constraint of the form $(x_j = a_i)$ for every $a_i \in A$, i.e., it contains exactly m A-constraints, one for each element of A. Given a complete set φ of A-constraints, let $K_\varphi = \{X \in \mathbb{R}^n : X \text{ satisfies every } A\text{-constraint in } \varphi\}$. Notice that $dim(K_\varphi) = n - m$ and $codim(K_\varphi) = m$. Moreover, if a point B belongs to K_φ, then $A \subseteq B$, i.e., every point in K_φ belongs to γ_1. Additionally, if $A \subseteq B$, then B belongs to some $K_{\varphi'}$ for some complete set φ' of A-constraints. Therefore, if we let $\mathscr{A} = \{K_\varphi : \varphi \text{ is a complete set of } A\text{-constraints}\}$, then $\gamma_1 = \cup\mathscr{A}$. In the full version of this paper, we study the topological structure of $\cup\mathscr{A}$ and obtain an $\Omega(n \log m)$ lower bound for the membership problem in γ_1.

We consider the poset induced by the intersection semilattice of $\mathscr{A}$ ordered by the reverse inclusion. We then consider the Möebius function on the elements of this poset and use result (16) of [18] to obtain our lower bound. □

Corollary 2. *Given a set P of n points and a simple polygon Q on m vertices (or a set of m segments or a set of m points), the $(P,Q)_r$-decision problem has a lower bound of $\Omega(n \log m)$ in the algebraic computation tree model.*

References

1. A. Aggarwal, L. Guibas, J. Saxe, and P. Shor. A linear time algorithm for computing the Voronoi diagram of a convex polygon. In *Proceedings of STOC*, pages 39–45, New York, NY, USA, 1987. ACM.
2. M. Ben-Or. Lower bounds for algebraic computation trees. In *Proceedings of STOC*, pages 80–86, New York, NY, USA, 1983. ACM.
3. P. Bose, S. Langerman, and S. Roy. Smallest enclosing circle centered on a query line segment. In *Proceedings of CCCG*, pages 167–170, 2008.
4. P. Bose and G. Toussaint. Computing the constrained Euclidean, geodesic and link centre of a simple polygon with applications. In *Proceedings of CGI*, pages 102 –111, 1996.
5. P. Bose and Q. Wang. Facility location constrained to a polygonal domain. In S. Rajsbaum, editor, *LATIN 2002*, volume 2286 of *Lecture Notes in Computer Science*, pages 153–164. Springer Berlin Heidelberg, 2002.
6. T. M. Chan. Geometric applications of a randomized optimization technique. *Discrete and Computational Geometry*, 22:547–567, 1999.
7. B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair, and parametric searching. *DCG*, 10:183–196, 1993.
8. F. Hurtado, V. Sacristan, and G. Toussaint. Some constrained minimax and maximim location problems. *Studies in Locational Analysis*, 15:17–35, 2000.
9. B. Joe and R. B. Simpson. Corrections to Lee's visibility polygon algorithm. *BIT Numerical Mathematics*, 27:458–473, 1987.
10. D. Lee. Farthest neighbor Voronoi diagrams and applications. *Report 80-11-FC-04, Dept. Elect. Engrg. Comput. Sci.*, 1980.
11. D. T. Lee. On finding the convex hull of a simple polygon. *International Journal of Parallel Programming*, 12(2):87–98, 1983.
12. J. Matousek. Computing the center of planar point sets. *Discrete and Computational Geometry*, 6:221, 1991.
13. J. Matoušek. Construction of epsilon nets. In *Proceedings of SCG*, pages 1–10, New York, 1989. ACM.
14. N. Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983.
15. F. Preparata. *Minimum spanning circle, in F.P. Preparata (Ed.), Steps in Computational Geometry.* University of Illinois, 1977.
16. M. Shamos and D. Hoey. Closest-point problems. In *Proceedings of FOCS*, pages 151–162, Washington, DC, USA, 1975. IEEE Computer Society.
17. J. J. Sylvester. A Question in the Geometry of Situation. *Quarterly Journal of Pure and Applied Mathematics*, 1, 1857.
18. A. C.-C. Yao. Decision tree complexity and Betti numbers. In *Proceedings of STOC*, pages 615–624, New York, NY, USA, 1994. ACM.