Clustering in Polygonal Domains

Mark de Berg
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Leyla Biabani
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Morteza Monemizadeh
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Leonidas Theocharous
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

We study various clustering problems for a set D of n points in a polygonal domain P under the geodesic distance. We start by studying the discrete k-median problem for D in P. We develop an exact algorithm which runs in time $\text{poly}(n, m) + n^{O(\sqrt{k})}$, where m is the complexity of the domain. Subsequently, we show that our approach can also be applied to solve the k-center problem with z outliers in the same running time. Next, we turn our attention to approximation algorithms. In particular, we study the k-center problem in a simple polygon and show how to obtain a $(1+\varepsilon)$-approximation algorithm which runs in time $2^{O(k \log k/\varepsilon)}(n \log m + m)$. To obtain this, we demonstrate that a previous approach by Bădoiu *et al.* [5, 4] that works in $\mathbb{R}^d$, carries over to the setting of simple polygons. Finally, we study the 1-center problem in a simple polygon in the presence of z outliers. We show that a coreset C of size $O(z)$ exists, such that the 1-center of C is a 3-approximation of the 1-center of D, when z outliers are allowed. This result is actually more general and carries over to any metric space, which to the best of our knowledge was not known so far. By extending this approach, we show that for the 1-center problem under the Euclidean metric in $\mathbb{R}^2$, there exists an ε-coreset of size $O(z/\varepsilon)$.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases clustering, geodesic distance, coreset, outliers

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.23

Funding MdB and LT are supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

1 Introduction

Given a set D of n points in $\mathbb{R}^d$, the 1-center problem asks to find a point $p \in \mathbb{R}^d$ that minimizes the maximum distance from p to the points in D. The problem dates back to 1857, when Sylvester posed this question for the Euclidean plane [20]. A linear-time algorithm for the problem was first proposed by Megiddo [15], thus refuting an earlier conjecture of Shamos and Hoey that it cannot be solved faster than $O(n \log n)$ [19]. A natural way to generalise the 1-center problem, is to instead ask for a set $S \subset \mathbb{R}^d$ of k *centers* that minimizes the maximum distance of points in D from their closest center in S. The 1-center problem serves as a basic example of a facility location problem and is thus directly related to clustering.

In this paper, we are interested in studying this and similar clustering problems for sets of points in simple polygons and polygonal domains under the *geodesic distance*, that is, when the distance between any two points is the Euclidean length of a shortest path between them. In the literature, the term *obstructed distance* has also been used in the past to highlight that a polygonal domain can be used to model a physical environment where obstacles may obstruct or delay communication between different locations of the environment. For

example, as mentioned in [23], when a bank decides where to place ATMs, it is important to take into account the existence of highways which act as obstacles for pedestrians. Motivated by such applications, various practical clustering algorithms for realistic scenarios have been proposed [10, 21, 23, 24, 25]. Thus, the use of geodesic distance is well motivated from an application point of view.

We first study the discrete k-median and k-center problems in a polygonal domain, i.e. a polygon P with holes. The *discrete k-median problem* asks to find a set $S \subset D$ of k *center points* such that the quantity $\sum_{d \in D}\{\min_{s \in S}\{\|\pi(d,s)\|\}\}$ is minimized, where $\|\pi(p,q)\|$ denotes the length of the shortest path $\pi(p,q)$ between two points p, q in P. The *k-center problem* asks to find a set $S \subset \mathbb{R}^d$ of k center points such that the maximum distance of points in D to their nearest center in S is minimized (so here, in contrast with k-median, we consider the continuous version of the problem). An outlier can significantly increase the maximum distance to the nearest center, so we also study the *k-center problem with z outliers*, which asks to minimize the maximum distance of all but z points of D to their nearest center. Our interest here lies in developing an *exact* algorithm for these problems, whose running time is polynomial in n and m (for fixed k), and whose dependency on k is subexponential. (Note that the Euclidean problem in the plane is already NP-hard when k is part of the input [16], so an algorithm that is also polynomial in k is not possible, assuming P $\neq$ NP.) The k-center problem in the plane has been studied extensively, for general k [9, 13] and also for the special case $k = 2$ [1, 22]. Most relevant to our approach is the work by Hwang *et al.* [12], who presented algorithms with running time $n^{O(\sqrt{k})}$ for the Euclidean version of the problems in $\mathbb{R}^2$. Their approach works with the Voronoi diagram of the (unknown) optimal solution, and "guesses" a cycle separator of its dual graph. The separator splits the problem into two subproblems, which are then solved recursively. This is an idea that we also make use of. The same approach was employed more recently by Marx and Pilipczuk [14] to solve a wide range of covering and packing problems defined on planar graphs. This includes k-center, which they solve in $n^{O(\sqrt{k})}$. To the best of our understanding, their approach cannot be used to tackle k-median and also cannot directly handle outliers. Thus, in Section 2 we develop an exact algorithm for the discrete k-median problem in a polygonal domain. The running time is $\text{poly}(n,m) + n^{O(\sqrt{k})}$, where m is the complexity of the domain. With our approach, we can also solve the k-center problem with z outliers in the same running time.

Next, we develop an FPT approximation algorithm for the k-center problem in a simple polygon, that is, an algorithm whose running time is $O(f(k,\varepsilon) \cdot \text{poly}(n,m))$, for some computable function f. Towards this algorithm, we first study the 1-center problem. Exact algorithms for the 1-center problem in a simple polygon have been developed before. More specifically, Ahn *et al.* [2] studied the problem of computing the *geodesic center* of a (weakly) simple polygon P, where the task is to find the point $s \in P$ that minimizes the maximum geodesic distance from *any* other point in P. They developed a linear-time algorithm for this problem. Their algorithm can be used to compute the 1-center of a set D of n points in P because the 1-center of D coincides with the geodesic center of $\text{RCH}_P(D)$, the relative convex hull of D in P. Since the geodesic convex hull is a weakly simple polygon that can be computed in time $O(n \log n + m)$, where m is the complexity of P), the 1-center of D can be computed in the same time.

Here, however, we study this problem through the lens of coresets. In general, a coreset is a small subset of the input to a problem, such that the solution of the problem on the coreset is a good approximation of the solution on the whole input. Coresets can be categorised in *strong coresets* and *weak coresets*, depending on the kind of guarantee they provide. Roughly

speaking, a strong coreset provides error guarantees for *any* candidate solution on the coreset, while a weak coreset only guarantees that an optimal solution on the coreset is a good approximation of an optimal solution on the whole set. All coresets presented in this paper are weak coresets, so from now on we will just use the term coreset.

Most related to our work are previous approaches for constructing coresets for the 1-center problem in $\mathbb{R}^d$. Bădoiu *et al.* [5] showed the existence of a coreset $C \subset D$ of size $O(\frac{1}{\varepsilon^2})$, such that the 1-center of the coreset is a $(1+\varepsilon)$-approximation to the 1-center of a set of points $D \subset \mathbb{R}^d$. The time to construct this coreset is $O\left(\frac{dn}{\varepsilon^2} + \frac{1}{\varepsilon^{O(1)}}\right)$. They then showed that their approach can be extended to obtain a $(1+\varepsilon)$-approximation for the k-center problem, for $k > 1$ in time $2^{O(k \log k/\varepsilon^2)}dn$. By providing a better analysis of the approach in [5], Bădoiu and Clarkson [4] showed that the coreset obtained actually has size $O(1/\varepsilon)$, which is tight. We show that these approaches also work when the underlying space is a simple polygon. A priori this is not at all clear, because the geodesic metric in a simple polygon does not have bounded doubling dimension. Specifically, in Section 3 we show the existence of an ε-coreset of size $O(1/\varepsilon)$ for the 1-center problem for a set of points in a simple polygon. The time to construct this is $O\left(\frac{n \log m + m}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$, where m is the complexity of the polygon. Note that a coreset of size two or three always exists in the plane (both in the Euclidean setting as well as in a polygon), since the minimum enclosing (geodesic) ball of a set of points in P is defined by two or three of the points. Thus it can be computed in the same time it takes to compute the 1-center of D, which, as mentioned, is $O(n \log n + m)$. Hence, for constant m the construction takes $O(n \log n)$ time, whereas our coreset can be constructed in $O\left(n/\varepsilon + 1/\varepsilon^2 \log(1/\varepsilon)\right)$ time. More importantly, our coreset can be combined with the approach by Bădoiu et al. [5] to approximately solve the k-center problem for $k > 1$, in $2^{O(k \log k/\varepsilon)}(n \log m + m)$ time.

Finally, we study the 1-center problem with z outliers through the lens of coresets in Section 4. We show that in any metric space, there exists a coreset of size $2z+2$ that is a 3-approximation for the 1-center problem with z outliers. In the Euclidean plane, we can generalize our result and obtain an ε-coreset of size $O(z/\varepsilon)$. Coresets for the k-center problem with z outliers have been studied for the metrics of bounded doubling dimension [7, 6]. Particularly, De Berg *et al.* [7] present an ε-coreset of size $O(k/\varepsilon^d + z)$, where d is the doubling dimension. In the plane, their construction can give an ε-coreset of size $O(k/\varepsilon^2 + z)$. Note that the dependency on $1/\varepsilon$ in their bound is quadratic, while our coreset only has a linear dependency on $1/\varepsilon$. (On the downside, in our case this is multiplied by z, while [7] has an additive term in z.) They also show that under some natural conditions, any coreset with a constant approximation ratio is of size $\Omega(k+z)$.

2 k-Median and k-center with outliers in a polygonal domain

In this section, we study the discrete k-median problem in a polygonal domain P. We will develop a subexponential exact algorithm for this problem, which depends exponentially on k and not on the complexity of the polygonal domain. We will then show that our algorithm can also be used to solve the k-center problem with outliers in a polygonal domain. We start by introducing some notation.

Notation. We denote the outer polygon of our polygonal domain P by P_0, and we use $\mathcal{H}$ to denote the collection of holes in P. Recall that $\pi(p,q)$ denotes the shortest path between two points p, q in P. For a finite set $D \subset P$, the *geodesic Voronoi diagram* of D in P, denoted $\text{GVD}(D)$, is the partition of P into $|D|$ Voronoi cells, where the Voronoi cell $V(q)$ of a point

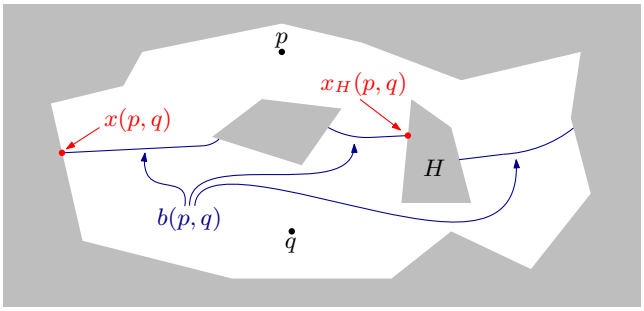

Figure 1 Illustration for the definition of $b(p,q), x(p,q)$ and $x_H(p,q)$.

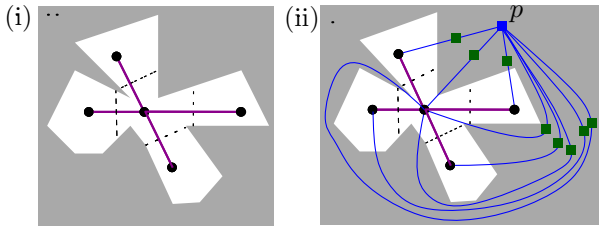

Figure 2 (i) An example where the dual of the geodesic Voronoi diagram corresponds to a tree. (ii) Adding p to the outside face of P and connecting it to cells incident to ∂P_0 via the intervals I_i.

$q \in D$ is defined as $V_D(q) := \{x \in P : \|\pi(x,q)\| \leqslant \|\pi(x,p)\| \text{ for all } p \in D\}$. When the set D is clear from the context, we may simply write $V(q)$. For two points $p, q \in P$, let $b(p,q)$ denote their geodesic bisector and let $b_D(p,q)$ denote the part of $b(p,q)$ which appears in $\text{GVD}(D)$. Let $B(p,q) = \{x \in P : \|\pi(p,x)\| \leqslant \|\pi(q,x)\|\}$. We will denote by $x(p,q)$ the first point of $b(p,q)$ that is met during a clockwise transversal of ∂P_0 which starts from a point of $\partial P_0 \cap B(p,q)$. Finally, we will denote by $x_H(p,q)$ the first point of $b(p,q)$ that is met during a clockwise transversal of hole H, starting from a point of $H \cap B(p,q)$; see Figure 1.

The main idea. The idea is to extend the approach by Hwang *et al.* [12], which worked for $\mathbb{R}^2$, to a polygonal domain. We therefore start by considering the geodesic Voronoi diagram of our (unknown) optimal solution S. In the Euclidean case, the dual of this diagram is called the Delaunay triangulation, denoted by $\text{DT}(S)$. Every inner face of $\text{DT}(S)$ is a triangle, and one can add a set I of three extra points to S sufficiently far away, such that the outside face of $\text{DT}(S \cup I)$ also becomes a triangle. This results in a maximal planar graph. Hence, by Miller's separator theorem [17] there exists a simple cycle separator C of $\text{DT}(S \cup I)$ of size $O(\sqrt{k})$ which is (2/3)-balanced with respect to $S \cup I$. (The latter means that at most 2/3 of the points in $S \cup I$ lie inside C and at most 2/3 of the points in $S \cup I$ lie outside C.) In our setting, it is not guaranteed that the dual of $\text{GVD}(S)$ is an (almost) triangulated graph. See Figure 2(i) for an example where it corresponds to a tree. Therefore we will need a few extra steps before we can apply a separator theorem.

Transforming the dual of $\text{GVD}(S)$. Let $\mathcal{G} = (V, E)$ denote the dual graph of $\text{GVD}(S)$. The goal is to transform $\mathcal{G}$ to a graph $\mathcal{G}^* = (V^*, E^*)$ such that any face of $\mathcal{G}^*$ has size at most three. The Voronoi cells of $\text{GVD}(S)$ that are incident to ∂P_0, induce a decomposition of ∂P_0 into disjoint intervals. Note that it is possible for a Voronoi cell to contribute to more than one interval. The following lemma gives a linear bound on the number of these intervals.

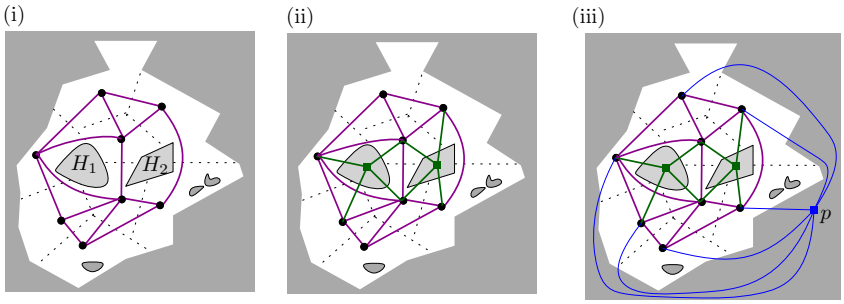

Figure 3 Holes H_1 and H_2 are essential, so we add a vertex for each of them. In (iii), observe that every face has bounded size.

▶ **Lemma 2.1.** *Let $I_1, \ldots, I_r$ denote the intervals along ∂P_0 induced by* GVD(S)*, enumerated in clockwise order. Then $r = O(k)$.*

Proof. For $1 \leqslant i \leqslant r$, let $s_i \in S$ be the center in the optimal solution S whose Voronoi cell has I_i on its boundary. Note that the s_i need not all be distinct. For $i = 1, ...r-1$, we charge I_i to $b(s_i, s_{i+1})$. Any bisector can be charged at most two times. Moreover, a bisector uniquely corresponds to an edge of $\mathcal{G}$ and we know that $\mathcal{G}$ is a planar graph. Therefore $r \leqslant 2|E| \leqslant 6k - 12$. ◀

Now let p denote an arbitrary point in the outside face of P. We connect p to each s_i via any arbitrary interior point of I_i. Let $\{e_1, \ldots, e_r\}$ denote the set of these extra edges. Then we have so far, $V^* = V \cup \{p\}$ and $E^* = E \cup \{e_i\}_{i=1}^r$. It's easy to see that we can embed these edges such that: (i) they are pairwise non-crossing and (ii) any face of the resulting graph incident to p is a triangle. See Figure 2(ii) for an example.

Handling the faces that do not contain p. Now we need to handle the faces of $\mathcal{G}^*$ that are not incident to p. By construction, the outer face of $\mathcal{G}^*$ contains p and thus is a triangle. Therefore, the only way $\mathcal{G}^*$ can contain a face of size at least four is if there exists a cycle of size four "around" a hole as in Figure 3(i).

We define an *essential hole* to be a hole $H \in \mathcal{H}$ which is incident to at least four Voronoi cells of GVD(S). Since every essential hole corresponds to a face of $\mathcal{G}$ (or $\mathcal{G}^*$), the number of essential holes is $O(k)$. Let $\mathcal{H}^*$ denote the set of essential holes and for every $H \in \mathcal{H}^*$ let p_H be an arbitrary point in H. We add the set $\{p_H\}_{H \in \mathcal{H}^*}$ to V^* and we connect p_H to the vertices of the Voronoi cells that are incident to H. If $V(q)$ is such a Voronoi cell, then, as before, we can embed the edge (p_H, q) by going through any interior point of $H \cap V(q)$. At the end of this process, $\mathcal{G}^*$ is a graph where every face is a triangle. See Figure 3(iii).

2.1 Applying the Separator Theorem to $\mathcal{G}^*$

We now want to apply Miller's Separator Theorem to $\mathcal{G}^*$. One thing that prevents us from doing so, is that $\mathcal{G}^*$ could be a multigraph, because p may be connected to the same Voronoi vertex more than once. (Recall that a Voronoi cell may contribute to more than one interval I_i). To deal with this, we can add a "dummy vertex" to each edge which has p as an endpoint; see Figure 2(ii). This way we only increase the number of vertices and edges by $O(k)$. Moreover, the faces of the resulting graph still have bounded size, which ensures that a separator theorem can still be applied (see below). Note that we want our separator to be balanced with respect to V. To ensure that, we employ the cost-balanced version of the Planar Separator Theorem, proven by Djidjev and Venkatesan [8].

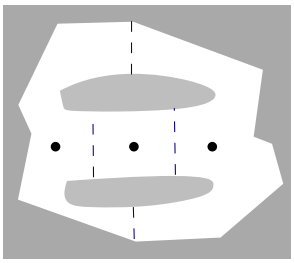

Figure 4 An example of three points and two holes, such that all three pairwise bisectors between the points intersect both holes.

> **Planar Separator Theorem.** Let $G = (\mathcal{V}, \mathcal{E})$ be a maximal planar graph with n nodes. Let each node $v \in \mathcal{V}$ have a non-negative weight, denoted weight(v), with $\sum_{v \in V}$ weight$(v) = 1$. Then $\mathcal{V}$ can be partitioned in $O(n)$ time into three sets A, B, C such that (i) C is a simple cycle of size $O(\sqrt{n})$, (ii) G has no arcs between a node in A and a node in B, and (iii) $\sum_{v \in A}$ weight$(v) \leqslant 2/3$ and $\sum_{v \in B}$ weight$(v) \leqslant 2/3$.

The theorem is stated for maximal planar graphs, but as pointed out in [8], it can be extended to graphs with faces of bounded size (as is our case). In our application, we give weight zero to the intermediate vertices as well as all vertices in $V^* \setminus V$, and weight $\frac{1}{|V|}$ to each vertex in V. Thus we obtain a simple-cycle separator C, which we can turn into a separator for $\mathcal{G}^*$ by ignoring any of the dummy vertices appearing on it. We obtain the following lemma.

▶ **Lemma 2.2.** *There exists a separator C for $\mathcal{G}^*$ with the following properties: **1.** C is a simple cycle, **2.** C has size $O(\sqrt{k})$ and **3.** C is $(2/3)$-balanced with respect to V.*

2.2 Guessing and embedding the separator

What we would like to do now, is guess the separator C of $\mathcal{G}^*$. Regarding the essential holes, note that we know that $|\mathcal{H}^*| = O(k)$, but we don't have any bound on $\mathcal{H}$ in terms of n, the size of the input point set D. This is problematic for the running time because we will need to "guess" what the essential holes are. However, we will argue that only $O(n^3)$ holes are good candidates for being essential. We start with the following lemma.

▶ **Lemma 2.3.** *Let P be a polygon and let $\mathcal{H} = \{H_1, H_2, ..., H_m\}$ be the set of holes in P. Let $T = \{p, q, r\}$ be a set of three points in P. Then there are at most two holes in $\mathcal{H}$ that are incident to all three Voronoi cells $V_T(p), V_T(q), V_T(r)$.*

Proof. Assume for contradiction that there exist three holes H_i, H_j, H_k incident to all three cells $V_T(p), V_T(q), V_T(r)$. Since Voronoi cells are connected, for each $x \in \{p, q, r\}$ and for each $H \in \{H_i, H_{,j}, H_k\}$, there exists a path connecting x to H which stays inside $V(x)$. In this way, we get a planar embedding of $K_{3,3}$, which is a contradiction. (Note that it *is* possible to have two holes bordering $V_T(p), V_T(q), V_T(r)$, see Figure 4.) ◀

Now we define the set of *candidate essential holes* $\mathcal{R}$ as follows: for every triplet of points in D we identify at most two holes which are incident to all three pairwise bisectors between the points. We then place these holes in $\mathcal{R}$. Clearly, $|\mathcal{R}| = O(n^3)$. Therefore we can afford to guess $O(\sqrt{k})$ essential holes on our separator C.

Now we give a more detailed description of how our algorithm works. Recall that each node in $\mathcal{G}^*$ (and, hence, each node on the separator we are looking for) corresponds to either a point in D, or to the extra point p we added, or to an essential hole. Thus, to find the

separator, we guess all ordered subsets of size $O(\sqrt{k})$ from the set $R \cup D \cup p$. This results in $n^{O(\sqrt{k})}$ candidate separators. We then would like to use each separator to split our problem into two independent subproblems, one for the inside and one for the outside of the separator. To do that, we have to make sure that for every demand point $d \in D$ its closest center point in the optimal solution, is located at the same side of the separator as d. For this, it suffices to embed the edges of the separator such that no edge crosses a Voronoi cell of a point which is not one of its endpoints. We have three categories of edges:

- **Edges that connect p to a Voronoi site s.** Clearly if $(p, s) \in E^*$, then there exists some $r \in S$ such that $(s, r) \in E$ and then we can embed such an edge using the shortest paths $\pi(p, x(s, r)), \pi(x(s, r), s)$. Note that we don't know r, but we can afford to guess it from D and therefore there are n options.
- **Edges that connect a Voronoi site s to a p_H for some $H \in \mathcal{R}$.** If $(s, p_H) \in E^*$, then again there exists some $r \in S$ such that $s, r \in E$ and then we can embed such an edge by going through $x_H(s, r)$, again using shortest paths. We can guess r from D and there are n options.
- **Edges that connect two Voronoi sites s and r.** Note that then one of the following holds if $(s, r) \in E$:
  1. there exists a $t \in S$, such that $V_S(s), V_S(r), V_S(t)$ meet at a point c in P or at a hole $H \in H^*$
  2. $b_S(s, r)$ intersects ∂P_0 at two points.

  Therefore we can check for all $t \in D$ whether 1. holds and if yes we embed (s, r) as $\pi(s, c) \cup \pi(c, r)$ or as $\pi(s, x_H(s, r)) \cup \pi(x_H(s, r), r)$. Otherwise, we can embed (s, r) via $x(s, r)$. Again we have at most n guesses.

Since we are using shortest paths to embed the edges, we know that they will stay inside the Voronoi cells of the sites they connect and thus we get a good embedding. Assuming our guessed separator is correct, the points on the separator have to be part of the optimal solution that we seek. Therefore in the two subproblems, these points have to be passed on as part of the input. If we assume that our separator has size i then we also need to guess how many of the remaining $k - i$ optimal centers lie in the inside and how many lie in the outside subproblem. In terms of running time, this is clearly not a problem since it can only give an extra factor of $O(k) = O(n)$. The base case of our algorithm is when $k = 1$, where we simply try all possible options.

A word on precomputing shortest paths. Our algorithm will need to make use of shortest paths between points in the set $D \cup \{p\} \cup \{p_H\}_{H \in \mathcal{R}} \cup \{x(p, q), x_H(p, q)\}_{(p,q,H) \in D \times D \times \mathcal{R}}$. Note that this set has size $\text{poly}(n)$ and thus all necessary shortest paths can be precomputed in $\text{poly}(n, m)$ time. The only information about these paths our algorithm will need during the recursion is their length and whether two paths cross or not. Indeed, this is enough to determine, for a guessed separator C, which are its two corresponding subproblems; two points p, q not on C will be on different sides of C if and only if $\pi(p, q)$ crosses C an odd number of times. Therefore, the complexity of the polygon does not appear during the recursion. Note that the same idea was used for the preprocessing step in [3].

Given the above discussion, the recursive formula of our algorithm is of the form $T(k) = n^{O(\sqrt{k})} T(2k/3)$, which solves to $T(k) = n^{O(\sqrt{k})}$. The following theorem summarises our result, where the $\text{poly}(n, m)$ term comes from precomputing shortest paths.

▶ **Theorem 2.4.** *Let D be a set of n points inside a polygonal domain P with m vertices and let k be a given positive integer. Then the discrete k-median problem for D in P can be solved in time* $\text{poly}(n, m) + n^{O(\sqrt{k})}$.

2.3 The k-Center problem with outliers

To solve the k-center problem with z outliers, we first show that the same approach as our k-median algorithm works to solve the so-called (k, r)-coverage problem, and then we show how to reduce the k-center problem with z outliers to the (k, r)-coverage problem. Let P be a polygonal domain, D denote a set of n demand points in P, and k, r be two parameters. We define a (k, r)-*coverage of* D as a set of k balls of radius at most r, such that the number of outliers (that is, points in D not covered by the balls) is minimized.

The divide-and-conquer algorithm we presented earlier for k-median clustering has a base case of $k = 1$. Our algorithm relies on the n candidates for the optimal centers in k-median, and has a running time of $n^{O(\sqrt{k})}$. However, we only use the properties of k-median to solve for the base case and determine the candidate centers when guessing the separator in an optimal solution. Note that above we solved the *discrete* k-median problem, where the set of candidate centers is given (namely, it is the same as the set D of demand points, although our algorithm would also work if a different discrete set of candidate centers is given). For the k-center problem, we wish to solve the *continuous* version, where the set of candidate centers is not given. It is well known, however, that in the continuous k-center problem, we can still restrict our attention to a discrete set of candidates, namely the centers of the smallest enclosing (geodesic) balls of every triple and pair of points in D. Thus there are $O(n^3)$ candidate centers. We denote the set of candidate centers by C^*. Therefore, the same approach can be applied to solve the (k, r)-coverage problem, where for the base case, we can consider all $O(n^3)$ balls of radius r centered at a point in C^* and find the one that covers the maximum number of points in D. This means that our algorithm can compute an optimal (k, r)-coverage in $n^{O(\sqrt{k})}$ time.

It remains to reduce the k-center problem with z outliers to the (k, r)-coverage problem. Observe that if r is at least the optimal radius for k-center clustering with z outliers, then the number of outliers for (k, r)-coverage is at most z. Moreover, there are at most $O(n^3)$ candidates for the optimal radius (namely the radii of the smallest enclosing balls of the triples and pairs of points in D). By performing a binary search over these $O(n^3)$ possible radii, we can find the minimum radius r^* such that the (k, r^*)-coverage covers all but at most z outliers. This (k, r^*)-coverage is an optimal solution for the k-center problem with z outliers, and the running time to find it is $O(n^{O(\sqrt{k})} \cdot \log(n^3)) = n^{O(\sqrt{k})}$.

▶ **Theorem 2.5.** *Let D be a set of n points inside a polygonal domain P with m vertices and let k, z be two given integers. Then the k-center problem for D with z outliers can be solved in time* $\text{poly}(n, m) + n^{O(\sqrt{k})}$.

3 A coreset for the k-center of points in a simple polygon

In this section, we turn our attention to the k-center problem in a simple polygon. As already mentioned, here we are interested in coresets for this problem. We will start by studying the 1-center. In itself, a coreset for the 1-center in a simple polygon is not so interesting, since the minimum enclosing (geodesic) ball of a set D of points inside P is always defined by two or three points, and so there exists a coreset of size two or three. However, the technique that we use (which is borrowed from Bădoiu and Clarkson [4]) forms the basis of the result for k-center. For a set S of points, let c_S denote the center of the minimum enclosing geodesic ball of S (that is, its 1-center) and let r_S be its radius. We denote the ball of radius r centered at a point c be $B(c, r)$, so $B(c_S, r_S)$ is the minimum enclosing ball of S. Note that these definitions apply in the standard Euclidean case, but also for geodesic

distances in a simple polygon. Our algorithm to compute a coreset $\mathcal{C}$ for the 1-center of a point set D inside a simple polygon P uses the approach of Bădoiu *et al.* [5, 4], which works as follows. First, we place in $\mathcal{C}$ an arbitrary point $p \in D$. Then we repeat the following procedure: we check whether there exists a point $q \in D$ such that $\|\overline{c_{\mathcal{C}}q}\| > (1+\varepsilon)r_{\mathcal{C}}$ and if yes, we add in $\mathcal{C}$ the point of D furthest from $c_{\mathcal{C}}$. Otherwise, we have our desired coreset. The analysis of the number of iterations of the above procedure relies on two key lemmas. Note that any chord in a simple polygon P (that is, any segment inside P connecting two points on ∂P) splits P into two sub-polygons, which we call *half-polygons*.

▶ **Lemma 3.1.** *Let $B(c_D, r_D)$ denote the minimum enclosing geodesic ball for a set D of points inside a simple polygon P. Then any closed half-polygon containing c_D also contains a point $p \in D$ such that $\|\pi(p, c_D)\| = r_D$.*

Proof. We proceed in the same way as in the Euclidean case. Namely, suppose there exists a chord s of P through c_D, such that one of the two defined half-polygons does not contain a point $p \in D$ such that $\|\pi(p, c_D)\| = r_D$. Let H_1 denote this half-polygon. Then we can slightly move c_D to the direction perpendicular to s and to the interior of $P \setminus H_1$. In this way, every point of D will now be fully contained in the interior of $B(c_D, r_D)$. The reason is that any shortest path $\pi(c_D, q)$, for $q \in P \setminus H_1$ is contained in $P \setminus H_1$ and thus this translation of c_D can only decrease $\|\pi(c_D, q)\|$. As a result the ball $B(c_D, r_D)$ can be slightly shrunk, contradicting its minimality. ◀

The second lemma we need is as follows.

▶ **Lemma 3.2.** *Let $B(c_D, r_D)$ denote the minimum enclosing geodesic ball for a set D of points inside a simple polygon P. For any point $q \in P$, there exists a point $p \in D$ at distance r_D from c_D such that $\|\pi(p, q)\| \geqslant \sqrt{\|\pi(p, c_D)\|^2 + \|\pi(c_D, q)\|^2}$.*

The corresponding lemma in the Euclidean case (that is when $P = \mathbb{R}^2$) follows directly from (the Euclidean version of) Lemma 3.1, by using the Pythagorean Inequality. For geodesic triangles however, we were not able to find in the literature an analog of the Pythagorean Inequality. Thus we prove now that the following property still holds for geodesic triangles in a simple polygon. Note that Lemma 3.3 together with Lemma 3.1 imply Lemma 3.2. Indeed, let s be the chord through c_D that is perpendicular to the first edge of $\pi(c_D, p)$. Then by applying Lemma 3.1 to the closed half-polygon defined by s and not containing q, we get that there exists a point $p \in D$ such that $\|\pi(p, c_D)\| = r_D$. The result follows by observing that in the geodesic triangle $\triangle_\pi p c_D q$, the angle at c_D is at least $\frac{\pi}{2}$ and thus Lemma 3.3 applies.

▶ **Lemma 3.3.** *Let p, q, r denote three points in a simple polygon P, such that in the geodesic triangle $\triangle_\pi pqr$, the angle at q is at least $\frac{\pi}{2}$. Then we have $\|\pi(p, r)\|^2 \geqslant \|\pi(p, q)\|^2 + \|\pi(q, r)\|^2$.*

Proof. For the following, refer to Figure 5. Let $\overline{qq_1}, \overline{qq_2}$ denote the first edges of the paths $\pi(q, p), \pi(q, r)$ respectively. We extend $\overline{qq_1}, \overline{qq_2}$ to the interior of $\triangle_\pi pqr$ and let q_1', q_2' denote the points where these extensions intersect $\pi(p, r)$ respectively. By the triangle inequality we have

$$\|\pi(p, q)\| \leqslant |qq_1'| + \|\pi(q_1', p)\| \text{ and } \|\pi(r, q)\| \leqslant |qq_2'| + \|\pi(q_2', r)\|.$$

Moreover, since the angle at q is at least $\pi/2$ and the Euclidean triangle $\Delta(qq_1'q_2')$ satisfies the Pythagorean Inequality, and $\|\pi(q_1', q_2') \geqslant |q_1'q_2'|$, we have

$$\|\pi(q_1', q_2')\|^2 \geqslant |qq_1'|^2 + |qq_2'|^2.$$

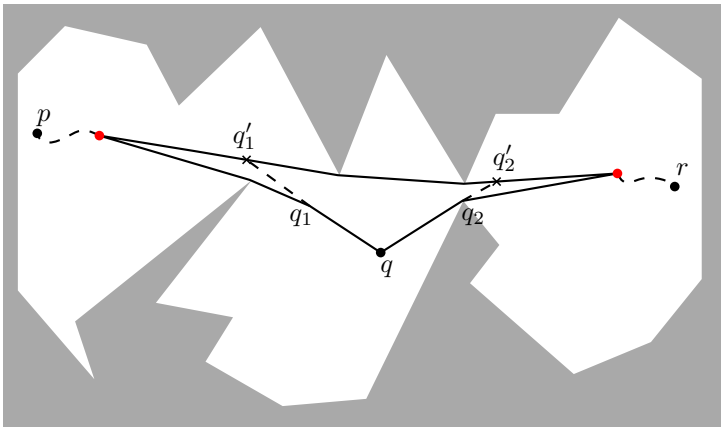

Figure 5 Illustration for the proof of Lemma 3.3. Here, the red points represent the points where the paths $\pi(p,r), \pi(p,q)$ and $\pi(r,q), \pi(r,p)$ split.

Finally, observe that for any numbers $a, b, c \geqslant 0$ we have

$$(a+b+c)^2 \geqslant a^2 + b^2 + c^2 + 2ab + 2ac.$$

Hence,

$$\begin{aligned}
\|\pi(p,q)\|^2 + \|\pi(r,q)\|^2 &\leqslant \left(|qq_1'| + \|\pi(q_1',p)\|\right)^2 + \left(|qq_2'| + \|\pi(q_2',r)\right)^2 \\
&= |qq_1'|^2 + 2|qq_1'| \cdot \|\pi(q_1',p)\| + \|\pi(q_1',p)\|^2 + |qq_2'|^2 + 2|qq_2'| \cdot \|\pi(q_2',r)\| + \|\pi(q_2',r)\|^2 \\
&= \|\pi(q_1',q_2')\|^2 + \|\pi(q_1',p)\|^2 + \|\pi(q_2',r)\|^2 + 2|qq_1'| \cdot \|\pi(q_1',p)\| + 2|qq_2'| \cdot \|\pi(q_2',r)\| \\
&\leqslant \left(\|\pi(q_1',q_2')\| + \|\pi(q_1',p)\| + \|\pi(q_2',r)\|\right)^2 \\
&= \|\pi(p,r)\|^2.
\end{aligned}$$

◀

We can now prove the following theorem.

▶ **Theorem 3.4.** *Let D be a set of n points inside a simple polygon P with m vertices. For any $\varepsilon > 0$ there is an ε-coreset $\mathcal{C} \subset D$ of size $O(1/\varepsilon)$ for the 1-center problem. The coreset can be constructed in time $O\left(\frac{n \log m + m}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$.*

Proof. The proof is similar to that of Badoiu *et al.* [4]: let $\mathcal{C}_i$ denote our coreset after i points have been added to it and let $B(r_i, c_i)$ denote its minimum enclosing ball. Observe that r_2 is a constant approximation for the optimal radius. Therefore, if one can show that $r_{i+1} \geqslant \left(1 + \frac{\varepsilon}{\alpha}\right) r_i$, for some constant α, it will follow that after $O(1/\varepsilon)$ iterations, $r_{\mathcal{C}} \geqslant r_D$. By applying Lemma 3.2 we get $r_{i+1} \geqslant \sqrt{r_i^2 + \|\pi(c_{i+1}, c_i)\|^2}$. By the triangle inequality we get $r_{i+1} > (1+\varepsilon) r_i - \|\pi(c_{i+1}, c_i)\|$. Using these two lower bounds for r_{i+1}, one can indeed get the desired relation between r_{i+1} and r_i. For further details, refer to the proof in [4].

Regarding the construction of the coreset, if we follow the procedure described, then we need to solve the 1-center problem $O\left(\frac{1}{\varepsilon}\right)$ times for a set of $O(\frac{1}{\varepsilon})$ points in a simple polygon with m vertices. This can be done in $O\left(\frac{1}{\varepsilon}\left(m + \frac{1}{\varepsilon} \log \frac{1}{\varepsilon}\right)\right)$ time [18]. In each iteration, we also need to find the point in D that is furthest from our current center. This takes $O(n \log m)$ time per iteration [11]. Hence, the total running time is $O\left(\frac{n \log m + m}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$. ◀

Extension to k-center. Theorem 3.4 combined with the methods from [5] yields a $(1+\varepsilon)$-approximate k-center for D. The algorithm starts with k, initially empty, sets $S_1, S_2, ..., S_k$. In each iteration, the point $p \in D$ furthest from $c_{S_1}, c_{S_2}, ..., c_{S_k}$ is found and added to

one of the sets. However, we do not know beforehand to which set the point p should be added, so we simply guess (that is, try all possibilities). After $O(k/\varepsilon)$ iterations, the algorithm will terminate in the branch where all guesses of where the furthest point should be added are correct. Therefore we also need to guess to which set to add p. We obtain the following theorem, whose proof is the same as the proof of the corresponding result of Bădoiu, Har-Peled, and Indyk [5].

▶ **Theorem 3.5.** *Let D be a set of n points inside a simple polygon P with m vertices. For any $1 > \varepsilon > 0$, a $(1+\varepsilon)$- approximate k-center for D can be found in time $2^{O(k \log k/\varepsilon)}(n \log m + m)$.*

4 Coresets for 1-center clustering with outliers

We now study the 1-center problem with outliers for points in the Euclidean plane (so, not inside a polygon). We will prove that there is an ε-coreset of size $O(z/\varepsilon)$ for this problem. To obtain such a coreset, our algorithm will need a constant-factor approximation as a starting point. To this end, we first show how to construct a coreset of size $2z + 2$ that gives a 3-approximation. Interestingly, this latter algorithm works in any metric space, so also for points inside a polygonal domain.

A coreset giving a 3-approximation in general metric spaces. Let D be a set of points in a metric space with distance function $d(\cdot, \cdot)$. We show in Algorithm 1 how to compute a coreset of size $2z + 2$ that 3-approximates the 1-center problem on D with z outliers.

Note that if the size of D is at most $2z + 2$, we return D in Algorithm 1 as the coreset. Therefore, without loss of generality, we can assume that the size of D is greater than $2z + 2$ for the rest of this section. We define $\text{OPT}_z(A)$ as the radius of an optimal solution for the 1-center problem on D with z outliers, where A represents any given set.

Algorithm 1 FindCoreset(D, z).

```
1: ▷ An algorithm to find a coreset for 1-center problem on D with z outliers
2: if |D| ⩽ 2z + 2 then
3:     return D
4: C_1 ← {an arbitrary point of D}
5: for i = 2 to 2z + 1 do
6:     Let B(o_{i−1}, r_{i−1}) be a minimum-radius ball containing all points of C_{i−1} but z outliers
7:     Let f_i be a point of D \ C_{i−1} that is furthest away from o_{i−1}
8:     C_i ← C_{i−1} ∪ {f_i}
9: return C_{2z+2}
```

▶ **Lemma 4.1.** *For any $1 \leqslant i \leqslant 2z + 2$, at least one of the following properties holds for the set C_i constructed by Algorithm FindCoreset(D, z).*

(i) $\text{OPT}_z(C_i) \geqslant \text{OPT}_z(D)/3$, *or*

(ii) *for all $r < \text{OPT}_z(D)/3$, any ball of radius r contains at most $z + 1$ points of C_i.*

Proof. For $i = 1$, property (ii) is trivially satisfied since we have $|C_i| = 1$ and $z + 1 \geqslant 1$. For $i > 1$, we prove the lemma by contradiction. Suppose the lemma is false, and let $t \in [2, 2z+2]$ be the smallest number such that properties (i) and(ii) do not hold for C_t. Then, there exists a ball $B(o', r')$ such that $|B(o', r') \cap C_t| > z + 1$ and $r' < \text{OPT}_z(D)/3$.

Let f_t be the point added to the coreset in line 8 of the algorithm, in the t-th iteration. We first prove that $f_t \in B(o', r')$ and $|B(o', r') \cap C_{t-1}| = z + 1$. Since $C_t \supset C_{t-1}$, we have $\text{OPT}_z(C_{t-1}) \leqslant \text{OPT}_z(C_t)$. Moreover, we assumed that property (i) does not hold for C_t,

which means $\text{OPT}_z(C_t) < \text{OPT}_z(D)/3$. Therefore, $r_{t-1} = \text{OPT}_z(C_{t-1}) < \text{OPT}_z(D)/3$, which means property (i) does not hold for C_{t-1}. As t is the smallest number such that both properties do not hold for C_t, we conclude that property (ii) holds for C_{t-1}, which implies $|B(o', r') \cap C_{t-1}| \leqslant z+1$. Adding it to $|B(o', r') \cap C_t| > z+1$ and $C_t = C_{t-1} \cup \{f_t\}$ we have $f_t \in B(o', r')$ and also $|B(o', r') \cap C_{t-1}| = z+1$.

$B(o_{t-1}, r_{t-1})$ is a solution for the 1-center problem on C_{t-1} with z outliers, and $|B(o', r') \cap C_{t-1}| = z+1$. Then, there exists a point $p^* \in B(o_{t-1}, r_{t-1}) \cap (C_{t-1} \cap B(o', r'))$. Moreover, by the triangle inequality we have $d(o_{t-1}, f_t) \leqslant d(o_{t-1}, p^*) + d(p^*, f_t)$. As $p^* \in B(o_{t-1}, r_{t-1})$, we have $d(o_{t-1}, p^*) \leqslant r_{t-1}$, and as both p^* and f_t are in $B(o', r')$ we have $d(p^*, f_t) \leqslant 2r'$. Thus, $d(o_{t-1}, f_t) \leqslant r_{t-1} + 2r'$, and since $r_{t-1} < \text{OPT}_z(D)/3$ and $r' < \text{OPT}_z(D)/3$ we have $d(o_{t-1}, f_t) < \text{OPT}_z(D)$.

On the other hand, we have $d(o_{t-1}, f_t) \geqslant d(o_{t-1}, p)$ for any $p \in D \setminus C_{t-1}$ since f_t is the furthest point in D/C_{t-1} to o_{t-1}. Furthermore, all but at most z points of C_{t-1} are in $B(o_{t-1}, r_{t-1})$. Also, $d(o_{t-1}, f_t) \geqslant r_{t-1}$, since otherwise $r_{t-1} = \text{OPT}_z(C_{t-1}) \geqslant \text{OPT}_z(D)$, which is a contradiction to $\text{OPT}_z(C_{t-1}) < \text{OPT}_z(D)/3$. Therefore, $B(o_{t-1}, d(o_{t-1}, f_t))$ is a solution for the 1-center problem on D with z outliers, which implies $\text{OPT}_z(D) \leqslant d(o_{t-1}, f_t)$. Hence, $\text{OPT}_z(D) \leqslant d(o_{t-1}, f_t) < \text{OPT}_z(D)$, which is a contradiction. ◀

▶ **Theorem 4.2.** *Let D be a set of n points in a metric space. Then there exists a coreset $\mathcal{C} \subset D$ of size at most $2z+2$ such that $\text{OPT}_z(D)/3 \leqslant \text{OPT}_z(\mathcal{C}) \leqslant \text{OPT}_z(D)$.*

Proof. Let $\mathcal{C}$ be the coreset returned by FindCoreset(D, z), and assume $|D| > 2z+2$ so that $\mathcal{C} = C_{2z+2}$. Since $C_{2z+2} \subseteq D$, then $\text{OPT}_z(C_{2z+2}) \leqslant \text{OPT}_z(D)$ trivially holds. To prove the other side of the inequality, suppose for a contradiction that $\text{OPT}_z(C) < \text{OPT}_z(D)/3$. Let $B(o, \text{OPT}_z(C_{2z+2}))$ be the optimal solution for the 1-center problem on C_{2z+2} with z outliers. Then, as $|C_{2z+2}| = 2z+2$ and at most z points are outliers, $B(o, \text{OPT}_z(C_{2z+2}))$ contains at least $z+2$ points. However, since we assume $\text{OPT}_z(C) < \text{OPT}_z(D)/3$, then $B(o, C_{2z+2})$ contains at most $z+1$ points by Lemma 4.1, which is a contradiction. ◀

A $(1+\varepsilon)$-coreset in the plane. The algorithm above works for any metric space, giving a 3-approximation. Now we explain that if D is a set of points in $\mathbb{R}^2$, we can improve the approximation ratio and obtain an ε-coreset, for any given $\varepsilon > 0$. To accomplish this, we add $O(z/\varepsilon)$ extra points to the coreset as follows. Let C_{2z+2} be the output of FindCoreset(D, z) and $B(o_{2z+2}, r_{2z+2})$ be an optimal solution for the 1-center problem on C_{2z+2} with z outliers. We partition the plane into $\ell = \lceil \frac{12\pi}{\varepsilon} \rceil$ cones $K_1, K_2, ..., K_\ell$ centered at o_{2z+2} with an opening angle of at most $\varepsilon/6$ each. Then, for each cone, we add $2z+2$ additional points to the coreset, namely the $z+1$ nearest points and the $z+1$ furthest points to o_{2k+2} from the points in D/C_{2k+2} that are located within that cone. Let A be the set of at most $(12\pi/\varepsilon) \cdot (2z+2)$ points selected in these cones, and define $\mathcal{C} := C_{2z+2} \cup A$. We will show that $\mathcal{C}$ is an ε-coreset.

▶ **Theorem 4.3.** *Let D be a set of points in $\mathbb{R}^2$. There exists an ε-coreset for the 1-center problem with z outliers for D of size $O(z/\varepsilon)$.*

Proof. Consider the set $\mathcal{C}$ defined above and let $B(\hat{o}, \hat{r})$ be an optimal solution for the 1-center problem on $\mathcal{C}$ with z outliers. It suffices to show that, for any point $q \in D \setminus \mathcal{C}$, we have that $\|\hat{o}q\| \leqslant (1+\varepsilon)\hat{r}$. Suppose for a contradiction that $\|\hat{o}q\| > (1+\varepsilon)\hat{r}$. Let $K_1, \ldots, K_\ell$ be the cones defined above, and recall that each cone has an angle of at most $\frac{\varepsilon}{6}$. As already mentioned, we place in our coreset the $z+1$ furthest and $z+1$ closest points to o_{2z+2}. Observe that $|\hat{o}o_{2z+2}| \leqslant 2\hat{r}$, since $B(o_{2z+2}, r_{2z+2})$ and $B(\hat{o}, \hat{r})$ have to intersect (otherwise there would be more than z outliers outside $B(\hat{o}, \hat{r})$). Note that this means that for any

point $p \in \partial B(\hat{o}, \hat{r})$, we have that $|o_{2z+2}p| \leqslant 3\hat{r}$. Let K_j denote the cone containing q. Since $q \notin A$, there exist $z+1$ points closer to o_{2z+2} than q and $z+1$ points further to o_{2z+2} than q. We denote the set of closer points by A_{close} and the set of further points by A_{far}. We have the following cases:

Case I: K_j does not intersect $B(\hat{o}, \hat{r})$. Then we clearly have a contradiction since $|K_j \cap A| \geqslant 2z+2$ while we are allowed at most z outliers.

Case II: K_j contains $B(\hat{o}, \hat{r})$. Then the opening angle of K_j is smallest when both of its sides are tangent to $B(\hat{o}, \hat{r})$. Let t denote one of the points of tangency. Then in the right triangle $\triangle o_{2z+2}ot$ we have $\frac{\varepsilon}{6} > \sin\left(\frac{\varepsilon}{6}\right) = \frac{\hat{r}}{|o_{2z+2}o|} \geqslant \frac{1}{2}$, which is a contradiction.

Case III: Otherwise, at least one of the sides of K_j intersects $B(\hat{o}, \hat{r})$. Let e_1 denote this side and e_2 denote the other side. Also, let e denote the half-line through $o_{2z+2}, \hat{o}$. Clearly e_1 will then also intersect $B(\hat{o}, (1+\varepsilon)\hat{r})$. We will now show the following claim.

▷ Claim 1. The side e_2 of K_j also has to intersect $B(\hat{o}, (1+\varepsilon)\hat{r})$.

Proof. Suppose for a contradiction that the claim is false. Then the opening angle of K_j is smallest when both e_1 and e_2 are tangent to $B(\hat{o}, \hat{r})$ and $B(\hat{o}, (1+\varepsilon)\hat{r})$, respectively. Let p_1, p_2 denote the points of tangency as in Figure 6(i). Let p denote the point where the line through $\hat{o}$ and p_1 intersects e_2. Then since p has to lie outside $B(\hat{o}, (1+\varepsilon)\hat{r})$, we have that $|p_1p| > \varepsilon\hat{r}$. Moreover, $|o_{2z+2}p_1| < |o_{2z+2}\hat{o}| < 2\hat{r}$ and so in the right triangle $\triangle o_{2z+2}p_1p$ we get $\tan\left(\frac{\varepsilon}{6}\right) = \frac{|p_1p|}{|o_{2z+2}p_1|} > \frac{\varepsilon\hat{r}}{2\hat{r}} = \frac{\varepsilon}{2}$, which is a contradiction, as $\tan\theta < 2\theta$ for small enough θ. ◁

Since both sides of K_j intersect $B(\hat{o}, (1+\varepsilon)\hat{r})$, we can partition K_j in two or three regions, depending on the location of o_{2z+2}. Namely, if o_{2z+2} lies inside $B(\hat{o}, (1+\varepsilon)\hat{r})$, then K_j can be partitioned in a region inside $B(\hat{o}, (1+\varepsilon)\hat{r})$ and a region outside $B(\hat{o}, (1+\varepsilon)\hat{r})$. If o_{2z+2} lies outside $B(\hat{o}, (1+\varepsilon)\hat{r})$, then K_j can be partitioned in three regions as in Figure 6(ii). Since the latter case is the most general, we will prove that one. The former case can be handled similarly. Without loss of generality, we will assume that e_2 lies above $\hat{o}$ and that the angle between e_2 and e is at least $\frac{\varepsilon}{12}$. The proof is slightly different, but essentially the same, depending on whether e is contained in K_j. To handle both at the same time, from now on we will let $e_3 \equiv e$ when e is contained in K_j and $e_3 \equiv e_1$ otherwise. Note that q has to lie either in region R_1 or R_3. We will now consider these two subcases.

Subcase I: $q \in R_1$. Observe that then the point x, where e_2 enters $B(\hat{o}, (1+\varepsilon)\hat{r})$ is the furthest q can be from o_{2z+2}. To derive a contradiction, it suffices to show that every point in A_{close} lies outside $B(\hat{o}, \hat{r})$. Since the point of $K_j \cap B(\hat{o}, \hat{r})$ closest to o_{2z+2}, is the point where e_3 enters $B(\hat{o}, \hat{r})$ (denoted by y), it suffices to show that $|o_{2z+2}x| < |o_{2z+2}y|$. We define $\phi = \angle xyo_{2z+2}$. Then, by the Law of Sines in the triangle $\triangle o_{2z+2}xy$ we get $\frac{\sin(\varepsilon/6)}{|xy|} = \frac{\sin(\phi)}{|o_{2z+2}x|}$. Therefore, $\sin\phi < \frac{\varepsilon/6}{\varepsilon r}\,|o_{2z+2}x| = \frac{|o_{2z+2}x|}{6\hat{r}}$.

Now notice that $|o_{2z+2}x| < |o_{2z+2}\hat{o}| < 2\hat{r}$. To see this, consider the tangent from o_{2z+2} to $B(\hat{o}, \hat{r})$ that lies above (o_{2z+2}, o) and let t be the point of tangency. Then x has to lie in the triangle $\triangle o_{2z+2}t\hat{o}$ and so $\angle o_{2z+2}x\hat{o} \geqslant \pi/2$. Therefore we get that $\sin\phi < \frac{1}{3}$, which gives us that $\phi < \frac{\pi}{6}$. Since we can assume $\varepsilon < 1$, we get that $\angle o_{2z+2}xy > \pi - \frac{\pi}{6} - \frac{\varepsilon}{6} > \pi/2$ and therefore we have that $|o_{2z+2}x| < |o_{2z+2}y|$.

Subcase II: $q \in R_3$. The approach is similar. The point $\hat{x}$ where e_2 exits $B(\hat{o}, (1+\varepsilon)\hat{r})$ is the closest q can be to o_{2z+2}. To derive a contradiction, it suffices to show that in that case every point in A_{far} lies outside $B(\hat{o}, \hat{r})$. Since the point of $K_j \cap B(\hat{o}, \hat{r})$ furthest from

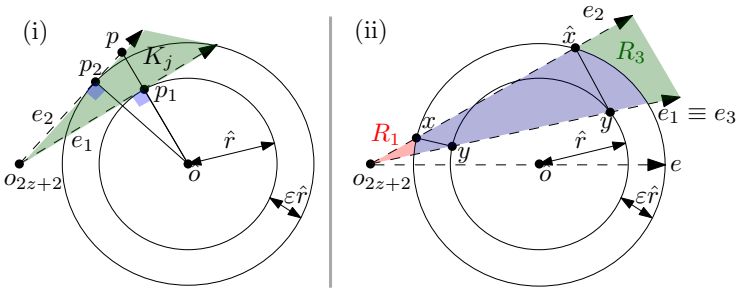

Figure 6 Illustrations for the proof of Theorem 4.3. Note that in (ii), e is not contained in the cone and therefore here we have $e_3 \equiv e_1$.

o_{2z+2}, is the point where e_3 exits $B(\hat{o}, \hat{r})$ – we denote this by $\hat{y}$ – it suffices to show that $|o_{2z+2}\hat{x}| > |o_{2z+2}\hat{y}|$. We define $\hat{\phi} = \angle\hat{y}\hat{x}o_{2z+2}$. Then, by the Law of Sines in the triangle $\triangle o_{2z+2}\hat{x}\hat{y}$ we have $\frac{\sin(\varepsilon/6)}{|\hat{x}\hat{y}|} = \frac{\sin(\hat{\phi})}{|o_{2z+2}\hat{y}|}$. Hence, $\sin\hat{\phi} < \frac{\varepsilon/6}{\varepsilon r}|o_{2z+2}\hat{y}| = \frac{|o_{2z+2}\hat{y}|}{6\hat{r}}$.

Now notice that $|o_{2z+2}\hat{y}| < 3\hat{r}$, as observed in the first paragraph of this proof. Therefore we get that $\sin\hat{\phi} < \frac{1}{2}$, which gives us that $\hat{\phi} < \frac{\pi}{6}$. Since we can assume $\varepsilon < 1$, we get that $\angle o_{2z+2}\hat{y}\hat{x} > \pi - \frac{\pi+1}{6} > \frac{\pi}{2}$ and therefore we have that $|o_{2z+2}\hat{x}| > |o_{2z+2}\hat{y}|$. This again gives a contradiction and concludes the proof. ◀

References

1 Pankaj K. Agarwal and Micha Sharir. Planar geometric location problems. *Algorithmica*, 11(2):185–195, 1994. doi:10.1007/BF01182774.

2 Hee-Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh. A linear-time algorithm for the geodesic center of a simple polygon. *Discret. Comput. Geom.*, 56(4):836–859, 2016. doi:10.1007/s00454-016-9796-0.

3 Henk Alkema, Mark de Berg, Morteza Monemizadeh, and Leonidas Theocharous. TSP in a Simple Polygon. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2022.5.

4 Mihai Bâdoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, pages 801–802, 2003.

5 Mihai Bâdoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 250–257, 2002. doi:10.1145/509907.509947.

6 Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Solving k-center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. *Proc. VLDB Endow.*, 12(7):766–778, 2019. doi:10.14778/3317315.3317319.

7 Mark de Berg, Leyla Biabani, and Morteza Monemizadeh. *k*-center clustering with outliers in the MPC and streaming model. In *Proc. 37th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2023)*, pages 853–863, 2023. doi:10.1109/IPDPS54959.2023.00090.

8 Hristo Djidjev and Shankar M. Venkatesan. Reduced constants for simple cycle graph separation. *Acta Informatica*, 34(3):231–243, 1997. doi:10.1007/s002360050082.

9 Zvi Drezner. The *p*-centre problem-heuristic and optimal algorithms. *The Journal of the Operational Research Society*, 35(8):741–748, 1984. URL: http://www.jstor.org/stable/2581980.

10 Yunjun Gao and Baihua Zheng. Continuous obstructed nearest neighbor queries in spatial databases. In *Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD 2009)*, pages 577–590, 2009. doi:10.1145/1559845.1559906.

11 Leonidas J. Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989. doi:10.1016/0022-0000(89)90041-X.

12 R. Z. Hwang, R. C. Chang, and Richard C. T. Lee. The searching over separators strategy to solve some NP-hard problems in subexponential time. *Algorithmica*, 9(4):398–423, 1993. doi:10.1007/BF01228511.

13 R. Z. Hwang, Richard C. T. Lee, and R. C. Chang. The slab dividing approach to solve the Euclidean p-center problem. *Algorithmica*, 9(1):1–22, 1993. doi:10.1007/BF01185335.

14 Dániel Marx and Michał Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. *ACM Trans. Algorithms*, 18(2), 2022. doi:10.1145/3483425.

15 Nimrod Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. In *Proc. 23rd Annual Symposium on Foundations of Computer Science (FOCS 1982)*, pages 329–338, 1982. doi:10.1109/SFCS.1982.24.

16 Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984. doi:10.1137/0213014.

17 Gary L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *Journal of Computer and System Sciences*, 32(3):265–279, 1986. doi:10.1016/0022-0000(86)90030-9.

18 Eunjin Oh, Sang Won Bae, and Hee-Kap Ahn. Computing a geodesic two-center of points in a simple polygon. *Computational Geometry*, 82:45–59, 2019. doi:10.1016/j.comgeo.2019.05.001.

19 Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proc. 16th Annual Symposium on Foundations of Computer Science (FOCS 1975)*, pages 151–162, 1975. doi:10.1109/SFCS.1975.8.

20 J. J. Sylvester. A question in the geometry of situation. *Quarterly Journal of Pure and Applied Mathematics*, 1857.

21 A.K.H. Tung, J. Hou, and Jiawei Han. Spatial clustering in the presence of obstacles. In *Proc. 17th International Conference on Data Engineering*, pages 359–367, 2001. doi:10.1109/ICDE.2001.914848.

22 Haitao Wang. On the planar two-center problem and circular hulls. *Discrete & Computational Geometry*, 68(4):1175–1226, 2022. doi:10.1007/s00454-021-00358-5.

23 Xin Wang and Howard J. HHamilton. Clustering spatial data in the presence of obstacles. *International Journal on Artificial Intelligence Tools*, 14:177–198, 2005. doi:10.1142/S0218213005002053.

24 Chenyi Xia, David Hsu, and Anthony K. H. Tung. A fast filter for obstructed nearest neighbor queries. In *Key Technologies for Data Management*, pages 203–215, 2004.

25 O.R. Zaiane and Chi-Hoon Lee. Clustering spatial data in the presence of obstacles: a density-based approach. In *Proc. International Database Engineering and Applications Symposium*, pages 214–223, 2002. doi:10.1109/IDEAS.2002.1029674.