Constrained Geodesic Centers of a Simple Polygon[*]

Eunjin Oh[1], Wanbin Son[2], and Hee-Kap Ahn[3]

1 **Department of Computer Science and Engineering, POSTECH, Pohang, Korea**
jin9082@postech.ac.kr
2 **Center for Geometry and Its Applications (GAIA), POSTECH, Pohang, Korea**
minibiny@gmail.com
3 **Department of Computer Science and Engineering, POSTECH, Pohang, Korea**
heekap@postech.ac.kr

Abstract

For any two points in a simple polygon P, the geodesic distance between them is the length of the shortest path contained in P that connects them. A geodesic center of a set S of sites (points) with respect to P is a point in P that minimizes the geodesic distance to its farthest site. In many realistic facility location problems, however, the facilities are constrained to lie in feasible regions. In this paper, we show how to compute the geodesic centers constrained to a set of line segments or simple polygonal regions contained in P. Our results provide substantial improvements over previous algorithms.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Geodesic distance, simple polygons, constrained center, facility location, farthest-point Voronoi diagram

Digital Object Identifier 10.4230/LIPIcs.SWAT.2016.29

1 Introduction

For a simple polygon P with n vertices in the plane, the geodesic path, denoted by $\pi(x, y)$, between any two points x and y in P is the shortest path between x and y contained in P, and the geodesic distance between x and y, denoted by $d(x, y)$, is the length of $\pi(x, y)$, that is, the sum of the Euclidean lengths of each segment in $\pi(x, y)$.

Let S be a set of k sites (points) in P. For any point x in P, a *geodesic farthest site* of x, denoted by $f_S(x)$, is a site of S that is farthest from x among all sites of S with respect to the geodesic distance. A point x in P that minimizes $d(x, f_S(x))$ among all points in P is called the *geodesic center* of S with respect to P. The geodesic center is unique and can be computed in $O(n + k)$ time if S consists of points on the boundary of P [1]. For S consisting of arbitrary points lying in P, the geodesic center can be computed in $O(n + k(\log n + \log k))$ time by constructing the geodesic convex hull $\mathsf{CH}_P(S)$ of S [13] and the geodesic center of $\mathsf{CH}_P(S)$.

For a subset Q of P, a geodesic center of S *constrained to* Q with respect to P is a point $q \in Q$ that minimizes $d(q, f_S(q))$ among all points in Q. Such a set Q is called a *constraint* or *feasible region* for facilities to be located in many realistic facility location problems. If the unconstrained geodesic center c coincides with a point $q \in Q$, then the geodesic center of

[*] This work was supported by the NRF grant 2011-0030044 (SRC-GAIA) funded by the government of Korea.

15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016).
Editor: Rasmus Pagh; Article No. 29; pp. 29:1–29:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

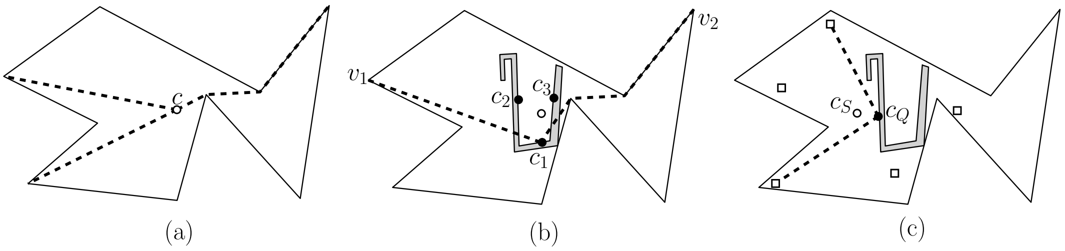

Figure 1 (a) The point c is the unconstrained geodesic center of the polygon. (b) The points c_1, c_2, and c_3 are the geodesic centers of the polygon constrained to the (gray) polygonal region. Here, $f(c_2)$ is v_2 and $f(c_3)$ is v_1, while c_1 has two farthest sites, v_1 and v_2. (c) For the sites (squares) lying in the polygon, c_S is the geodesic center of the sites and c_Q is the geodesic center constrained to the (gray) polygonal region.

S constrained to Q with respect to P is unique which is q. The geodesic center $c = q$ can be computed in $O(n + m + k(\log n + \log k))$ time, where m is the complexity of Q. If the unconstrained geodesic center c lies in $P \setminus Q$, a constrained geodesic center of S is a point q on the boundary of Q that minimizes $d(q, f_S(q))$ among all boundary points of Q. See Figure 1. Contrary to the unconstrained case, there might be more than one constrained geodesic centers in Q, but the geodesic distance from any constrained geodesic center q to its farthest point $f_S(q)$ is the same. We call the distance *the radius of the geodesic centers constrained to* Q and denote it by r_Q.

In this paper, we consider the problem of computing the geodesic centers of S with respect to a simple polygon P that are constrained to a subset Q of P consisting of line segments or simple polygonal regions.

Related works. Asano and Toussaint [3] studied the geodesic problem in which Q is the polygon P and S is the vertex set of P, and gave the first algorithm for computing the unconstrained geodesic center of P with n vertices which runs in $O(n^4 \log n)$ time. Afterwards, Pollack et al. [20] improved it to $O(n \log n)$ time. Finally, Ahn et al. [1] settled the problem by presenting a linear time algorithm for the problem.

To the extent of our knowledge, there is no known result for computing the geodesic center constrained to lie in a subset of P, except the one by Bose and Toussaint [7]. Their algorithm computes the geodesic center of P constrained to lie in a polygonal region $Q \subset P$ with m vertices in $O(n(n + \ell))$ time, where ℓ is the number of intersections of the geodesic farthest-point Voronoi diagram of the vertices of P with Q, and therefore $\ell = \Theta(nm)$ in the worst case.

The constrained center problem has been studied extensively under the Euclidean metric in the plane. Here P is the whole plane and S is a set of k points in the plane that we want to enclose. This problem is known as the constrained minimum enclosing circle problem or the constrained 1-center facility location problem. Megiddo [18] presented an $O(k)$-time algorithm for the problem in which the constraint Q is a line, and Hurtado et al. [15] presented an $O(k + m)$-time algorithm for the problem in which the constraint Q is a convex m-gon. Bose and Toussaint [7] considered the problem in which the center of the enclosing circle is constrained to lie in a simple polygon Q with m vertices and presented an $O((k+m) \log k + k \log m + \ell)$-time algorithm, where ℓ is the number of intersections of Q with the farthest-point Voronoi diagram of S. Later, Bose and Wang [8] removed the dependency on ℓ from the running time. Bose et al. [6] showed that the minimum enclosing circle whose center is constrained to lie on a query line segment can be reported in $O(\log k)$ time after

Table 1 Our results for constrained geodesic centers. $T(n,k) = O(n \log\log n + k\log(n+k))$ time [19] and $T(n,k) = \Omega(n+k)$. When S is the vertex set of P, $T(n,k) = O(n\log\log n)$ [19].

Constraints	Total running time
line segments	$O(m\log(n+k+m)) + T(n,k)$ or $O((m+k)\log(n+k+m) + mk\log n + n)$
disjoint line segments, disjoint polygonal regions	$O(m\log(n+k)) + T(n,k)$ or $O((m+k)\log(n+k) + mk\log n + n)$
geodesic convex polygon, disjoint geodesic pseudo polygons	$O(m) + T(n,k)$
disjoint polygonal regions with vertices on ∂P	$O(n+m+k\log(n+k))$

computing the farthest-point Voronoi diagram of S. Barba [4] presented an algorithm that reports the minimum enclosing circle with center on a given disk in $O(\log k)$ time after computing the farthest-point Voronoi diagram of S. Recently, Barba et al. [5] proposed algorithms that return the constrained center for constraint either a set of points or a set of segments in expected $\Theta((k+m)\log\min\{k,m\})$ time. For a constraint of a simple polygon Q, the expected running time becomes $\Theta(m+k\log k)$.

Our results. We begin with a set of (possibly crossing) line segments as the constraint Q and present an efficient algorithm that returns all constrained geodesic centers in $O(n+k+m\log(n+k+m))$ time after constructing the farthest-point geodesic Voronoi diagram FVD of S with respect to P in $T(n,k) = O(n\log\log n + k\log(n+k))$ time [19]. The algorithm also works in $O((m+k)\log(n+k+m) + mk\log n + n)$ time if we do not construct FVD of S.

Then we show that the running time can be improved slightly to $O(n+k+m\log(n+k) + T(n,k))$ if Q is a set of m disjoint line segments or disjoint polygonal regions with m vertices in total. When S is the vertex set of P, the running time becomes $O(n\log\log n + m\log n)$, which improves the $O(n(n+\ell))$-time result by Bose and Toussaint [7]. The algorithm also works in $O((m+k)\log(n+k) + mk\log n + n)$ time if we do not construct FVD of S.

Finally, we show that the running time can be further improved to linear if Q belongs to one of a few special polygon types. If Q is a geodesic convex polygon or a set of disjoint geodesic pseudo polygons (to be defined later) with m vertices in total, we can solve the problem in $O(n+m+k)$ time once FVD is computed. We can solve the problem in $O(n+m+k\log(n+k))$ time, for a set Q of disjoint polygonal regions with m vertices in total whose vertices are on the boundary of P without computing FVD. Our results are summarized in Table 1.

Recently we notice that the algorithm of Bose et al. [6] that computes the smallest enclosing circle whose center is constrained to lie on a line segment in the plane can be extended for the problem to find the geodesic centers constrained to line segments. By a similar argument of the algorithm of Bose et al., the geodesic centers of S constrained to Q with respect to P can be computed in $O(n+k+m\log(n+k))$, once FVD of S with respect to P is computed.

The algorithm of Bose et al. [6] relies on a property that the farthest-point Voronoi diagram of points in the plane is a tree, so it is hard to extend the algorithm for problems that do not satisfy this property. One representative example is the constrained weighted minimum enclosing circle problem. If we replace P with the whole plane and assign a positive weight for each point in S, the problem becomes the constrained weighted minimum enclosing circle problem in the plane. In this problem, the distance between two points $x \in P$ and

$y \in S$ is $w(y) \cdot d(x, y)$, where $w(y)$ is the weight of y. The algorithm of Bose et al. may not work since the farthest-point Voronoi diagram for weighted points is not necessarily a tree [17]. However our algorithms work for the constrained weighted minimum enclosing circle problem since most of the properties in this paper still hold for the problem.

2 Preliminaries

For a set A of points, let ∂A and $\mathrm{int}(A)$ denote the boundary and the interior of A, respectively. A subset A of a simple polygon P is *geodesically convex* if $\pi(x, y) \subseteq A$ for any two points $x, y \in A$. Let $\mathsf{hub}(r)$ be the set of points $x \in P$ such that $d(x, f_S(x))$ is at most r. Clearly, $\mathsf{hub}(r_S)$ with $r_S = \min_{p\in P} d(p, f_S(p))$ consists of a single point which is the geodesic center of S with respect to P. For any $r \geq \max_{p\in P} d(p, f_S(p))$, $\mathsf{hub}(r)$ is P itself.

▶ **Lemma 1.** *The set* $\mathsf{hub}(r)$ *is geodesically convex and monotone, that is,* $\mathsf{hub}(r') \subseteq \mathsf{hub}(r)$ *for any* $0 \leq r' \leq r$.

Proof. The monotonicity follows from the definition of $\mathsf{hub}(r)$ directly. For $0 < r < r_S$, $\mathsf{hub}(r) = \emptyset$. For $r \geq r_S$, $\mathsf{hub}(r)$ is the common intersection of the geodesic disks of radius r, each centered at a site of S. Since a geodesic disk is connected and geodesically convex, their nonempty common intersection is also geodesically convex. ◀

By the definition of constrained geodesic centers and Lemma 1, the lemma below holds.

▶ **Lemma 2.** *Every geodesic center constrained to Q lies on the boundary of* $\mathsf{hub}(r_Q)$*, where r_Q is the geodesic distance from any constrained geodesic center c to $f_S(c)$. Moreover, no point in Q lies in the interior of* $\mathsf{hub}(r_Q)$*.*

2.1 The Refined Farthest-point Geodesic Voronoi diagram

The *farthest-point geodesic Voronoi diagram* of S with respect to P is the subdivision of P such that each cell consists of the points with the common farthest site. Aronov et al. [2] showed that a farthest-point geodesic Voronoi diagram has linear complexity.

We consider a *refined* version of the farthest-point geodesic Voronoi diagram. Let C be a cell in the farthest-point geodesic Voronoi diagram FVD and let $f \in S$ be the common farthest site of the points in C. The shortest path map of f, which can be obtained by extending all edges of the shortest path tree [12], subdivides C into smaller cells, which we call *refined cells* of C (and of FVD). The *refined farthest-point geodesic Voronoi diagram* of FVD is the set $\mathrm{int}(P) \setminus \bigcup_{C\in\mathcal{C}} \mathrm{int}(C)$, where $\mathcal{C}$ is the collection of all refined cells of FVD. Here, a cell in the shortest path map of f contains at least one (hyperbolic or straight) arc of the boundary of C. This implies that the complexity of a refined farthest-point geodesic Voronoi diagram is still linear. Moreover, a refined cell has the following property, which comes directly from its definition.

▶ **Lemma 3.** *Every refined cell has exactly one boundary line segment that lies on the boundary of P.*

Given a simple polygon P with n vertices and a set S of k sites, Aronov et al. [2] gave an $O((n + k) \log(n + k))$ algorithm to compute the farthest-point geodesic Voronoi diagram of S with respect to P. Recently, Oh et al. [19] gave an $O(n \log \log n)$-time algorithm to compute the farthest-point geodesic Voronoi diagram for the special case that the sites are the vertices of P. The algorithm in [19] can be generalized to the case that S is a set of arbitrary points in P. To this end, we first compute the geodesic convex hull of S with respect to P in $O(n + k\log(n + k))$ time [13]. Then we can compute the farthest-point geodesic Voronoi diagram inside the geodesic convex hull in $O((k + n)\log\log n)$ time by [19]. For the region outside the geodesic convex hull, we can apply a technique similar to [19] and compute the diagram in $O((k + n)\log\log n)$ time. Therefore, we can compute the diagram in total $O(n\log\log n + k\log(n + k))$ time.

Both algorithms in [2] and [19] can compute also the refined farthest-point geodesic Voronoi diagram without increasing their running times. In the following, we assume that we already have the refined farthest-point geodesic Voronoi diagram of S with respect to P.

3 Overlay of FVD and Curves in Geodesic Convex Position

In this section, we consider a simple constraint, *curves in geodesic convex position*. We say that curves are in *geodesic convex position* if the curves are contained in the boundary of the geodesic convex hull of themselves. We give a combinatorial property of the overlay between the farthest-point geodesic Voronoi diagram of sites with respect to P and curves in geodesic convex position. Specifically, we will show that the overlay has complexity linear to the number of sites and the complexity of the curves.

Since Euclidean farthest-point Voronoi diagrams have straight line segments as their arcs, the overlay of a diagram and curves in convex position has complexity linear to sum of their complexities - each line segment of the diagram intersects the curves at most twice and the complexity of the Euclidean Voronoi diagram is linear to the number of sites. However, a farthest-point geodesic Voronoi diagram defined in a simple polygon P might have hyperbolic arcs which intersect a convex curve contained in P more than a constant number of times, and therefore the argument for the Euclidean case does not work for the geodesic case.

To bound the complexity for the geodesic case, we consider a polygonal subdivision of P with respect to the diagram as follows. Let S be a set of sites contained in P and C be a refined cell of the geodesic farthest-point Voronoi diagram FVD of S. The boundary of C consists of (possibly empty) line segments and (possibly empty) hyperbolic arcs. Every point $x \in C$ has the same farthest neighbor $f_S(x)$ and has the same combinatorial structure of $\pi(f_S(x), x)$. Moreover, exactly one line segment of ∂C lies on ∂P by Lemma 3.

For a point $x \in C$, we call the last vertex that the path $\pi(f_S(x), x)$ from $f_S(x)$ goes through before x the *anchor of* x and denote it by $\mathsf{anchor}(x)$. For a hyperbolic arc α bounding C with endpoints a and b, let a' be the first point of ∂P hit the ray from a in the direction opposite to $\mathsf{anchor}(a)$ with respect to the site of C. See Figure 2(a). The point b' is defined analogously. We claim that the two line segments aa' and bb' subdivide C into at most three disjoint regions. To show this claim, we need the following lemma, which can be proved by the triangle inequality.

▶ **Lemma 4** ([19]). *Let x be a point in a refined cell C of a farthest-point geodesic Voronoi diagram of P. Then the line segment connecting x and y is contained in C, where y is the point on the boundary of P hit by the ray from* $\mathsf{anchor}(x)$ *towards x.*

The above lemma implies that aa' and bb' intersect ∂C only at their endpoints unless they are completely contained in ∂C; If there is another point $x \in \partial C$ in the interior of aa', aa' touches ∂C at x and there must be a point $x' \in \partial C \setminus \{a, a', x\}$ such that the segment $x'y'$ crosses ∂C at a point in the interior of $x'y'$, where y' is the point on the boundary of P hit by the ray from $\mathsf{anchor}(x')$ towards x'. Therefore, aa' and bb' subdivide C into at most three disjoint regions.

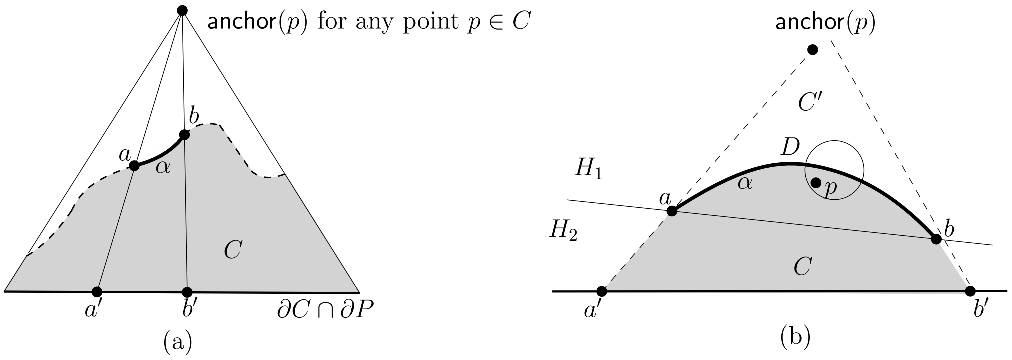

Figure 2 (a) A refined cell C is subdivided into at most three disjoint regions by aa' and bb'. (b) A hyperbolic arc α is incident to two refined cells, C and C'. If the anchor q of p lies on H_2, the ray from p in the direction opposite to q passes through α, which is a contradiction. Thus anchor(p) lies on H_1 and ab is contained in the region bounded by aa', bb' and α.

Let M be the subdivision of P with respect to FVD by introducing for each refined cell C of FVD, the line segments of ∂C and three line segments aa', bb', and ab for every hyperbolic arc α with endpoints a and b on ∂C. Each hyperbolic arc of ∂C is completely contained in a cell, and each cell of M contains at most one hyperbolic arc of ∂C as shown in the following lemma.

▶ **Lemma 5.** *Let α be a hyperbolic arc of a farthest-point geodesic Voronoi diagram* FVD*. The line segment connecting the two endpoints of α does not intersect in its interior any arc of* FVD *or ∂P.*

Proof. The arc α is incident to exactly two refined cells of FVD, say C and C'. Without loss of generality, we assume that α is locally convex with respect to C. Let a and b be the two endpoints of α. Let H_1 be the half plane that is bounded by the line through a and b and contains α, and let H_2 be the other half plane. In the following, we show that the anchor of a point x in α defined by the geodesic path from the site of C to x is contained in H_1, and the anchor of x defined by the geodesic path from the site of C' to x is contained in H_2.

There exists a disk D of sufficiently small radius such that $D \subset C \cup C'$ and α subdivides D into two pieces, one contained in the region R bounded by α and the segment ab, and the other contained in C'. See Figure 2(b). Let p be a point lying in the interior of $D \cap R$. Then, anchor(p) must be contained in H_1. If anchor(p) is in H_2, the ray from p in the direction opposite to anchor(p) intersects the interior of α, and then it intersects C', which contradicts to Lemma 4.

Then the region of C bounded by aa', bb', α, and $a'b'$ contains ab, where a' (and b') is the first point of ∂P hit by the ray from a (and from b) in the direction opposite to anchor(a) (and opposite to anchor(b)) with respect to the site of C. Thus no arc of FVD intersects ab in its interior.

To show that ∂P does not intersect ab, we use Lemma 3 that exactly one line segment of ∂C lies on ∂P. If the line segment lying on $\partial P \cap \partial C$ intersects ab, then ∂P also intersects α. However, α is contained in P, which is a contradiction.

Therefore, ab does not intersect in its interior any arc of FVD or ∂P. ◀

Clearly, the subdivision M has complexity $O(k+n)$ because it is constructed by overlaying $O(k+n)$ line segments in the plane which are pairwise-disjoint in their interiors, where k the number of sites in S.

▶ **Lemma 6.** *The subdivision M consists of $O(n+k)$ cells.*

Now, consider curves on geodesic convex position with m vertices in total. Since M consists of $O(n+k)$ edges (line segments), the overlay of the curves and M has complexity $O(n+m+k)$. Since a hyperbolic arc α is contained in exactly one cell Δ of M and a hyperbolic arc intersects a line segment at most twice, α intersects the curves at most $2m_\Delta$ times, where m_Δ is the complexity of parts of the curves lying in Δ. Since the cells of M are disjoint each other, the total complexity of parts of the curves lying in Δ over all cells of M is $O(n+m+k)$, which implies that the curves intersects the hyperbolic arcs of FVD $O(n+m+k)$ times in total. This section is summarized as follows.

▶ **Lemma 7.** *The overlay of* FVD *and curves in geodesic convex position with m vertices has complexity $O(n+m+k)$.*

4 Geodesic Centers Constrained to Line Segments

In the following, we assume that S is the vertex set of P. We will show how to handle the general case that S is a set of arbitrary points in P at the end of this section. We use $f(x)$ to denote the farthest site in the vertex set of P of x. In this section, we give an algorithm to compute the geodesic centers of P constrained to Q in the case that Q is a set of m (possibly crossing) line segments.

Once we have the overlay of Q and the farthest-point geodesic Voronoi diagram FVD of the vertices of P, we can compute the geodesic centers constrained to Q in time linear to the complexity of this overlay. Indeed, each line segment of Q is partitioned into smaller pieces in the overlay. We find the points c that minimize $d(c, f(c))$ in each smaller piece in constant time since each piece is contained in a refined cell of FVD. However, the number of these smaller pieces, that is, the complexity of the overlay might be quadratic.

Therefore, we avoid computing the overlay. Instead, we construct the cell Γ in the overlay of Q with $\mathsf{hub}(r)$ for a certain value r such that Γ contains the unconstrained geodesic center c. We will show that every geodesic center constrained to Q lies on the boundary of Γ. Once we find Γ, we can compute all geodesic centers constrained to Q in $O(n+m)$ time.

One might think that instead of computing a hub, considering the arrangement of Q inside P alone without overlaying it with $\mathsf{hub}(r)$ makes the algorithm simpler and easier. However, since Q is a set of line segments, the cell containing the unconstrained geodesic center in the arrangement of Q inside P has $O((n+m)\alpha(n+m))$ complexity [11], where $\alpha(n)$ is the inverse Ackermann function of n. Moreover, the best known algorithm for computing the cell takes $O((n+m)\alpha(n+m)\log(n+m))$ time [11]. Even worse, the cell is not necessarily convex, so the overlay of FVD and the boundary of the cell might be still quadratic.

Algorithm. Our algorithm works as follows. In the first step, we compute the farthest-point geodesic Voronoi diagram FVD of the vertices of P. Let Q_V be the set of the endpoints of the line segments in Q. For each $q \in Q_V$, we find the cell of FVD containing q. We preprocess FVD in $O(n)$ time to support an $O(\log n)$-time point-location query for a connected polygonal subdivision [9, 16]. In our case, some arcs of FVD might be hyperbolic while others are straight. To apply their point-location query structure to our case, we make use of the subdivision M of P with respect to FVD which we defined in Section 3. The subdivision M is a connected polygonal subdivision of $O(n)$ complexity. (See Lemma 6.) To find the cell of FVD containing a point q, we first find the cell of M containing q in $O(\log n)$ time. Recall that the interior of a cell of M intersects at most two cells of FVD. Thus, the cell containing

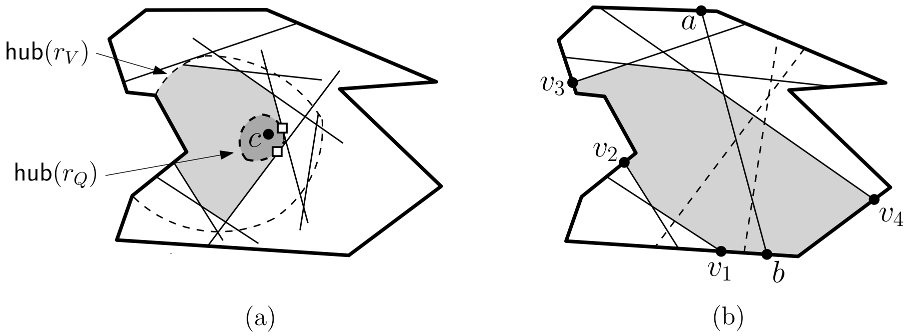

Figure 3 (a) The points with squares are the geodesic centers constrained to Q. (b) The gray region is the intersection we computed before considering ab. The last chain in the gray region connecting v_3 and v_4 is intersected by ab. Dashed line segments are the line segments lying after ab.

a point $q \in Q_V$ can be found in $O(\log n)$ time and the point-location queries can be done in $O(m \log n)$ time for all points $q \in Q_V$.

Now, we have $f(q)$ for every $q \in Q_V$. Let r_V denote the minimum distance $d(q, f(q))$ among all points q of Q_V. Note that the combinatorial structure of $\pi(p, f(p))$ is the same for any point p in the same refined cell of FVD. Thus, we can compute $d(q, f(q))$ in constant time once we have the refined cell of FVD containing q. We compute r_V in $O(m)$ time.

By Lemma 2, hub(r_V) contains no point of Q_V in its interior. But it contains some points in Q on its boundary, thus we have $r_V \geq r_Q$. Consider the case that $r_V = r_Q$. Then, the points in Q lying on the boundary of hub(r_V) are the geodesic centers of P constrained to Q by Lemma 2. If $r_V > r_Q$, there are some line segments of Q that cross hub(r_V). Moreover, the geodesic centers constrained to Q are contained in such line segments. See Figure 3(a).

We compute hub(r_V). For each refined cell of FVD, we can compute part of hub(r_V) contained in the refined cell in time linear to the complexity of the refined cell, because we already have the farthest site and the anchor of the refined cell. This can be done in $O(n)$ time for all refined cells once we construct the refined cells of FVD.

Then, for each line segment of Q, we check whether it crosses hub(r_V). If so, we additionally find the circular arcs of hub(r_V) crossed by the line segment. We do this for all line segments in $O(m \log n)$ time. The detailed procedure will be described in Section 4.1.

The line segments of Q crossing hub(r_V) subdivide hub(r_V) into $O(m^2)$ geodesic convex regions. Note that we do not need to construct the whole subdivision. We construct only the cell containing the unconstrained geodesic center c of P in the subdivision. This is because the geodesic centers constrained to Q are on the boundary of the cell containing c in this subdivision by Lemma 1 and 2, and the fact that hub(r) contains c for any $r \geq d(c, f(c))$. We find the cell Γ containing c in $O(n + m \log(n + m))$ time, which will be explained in Section 4.2.

Finally, we compute the overlay of the boundary of Γ and FVD in time linear to their total complexity. This can be done by traversing the boundary of Γ and the refined cells in FVD. Since Γ is geodesically convex, the complexity of the overlay is linear to their total complexity by Lemma 7. Then we can find the geodesic centers of P constrained to Q in the overlay in the same time.

4.1 Finding the Circular Arcs Intersecting a Line Segment

We are given hub(r) for some $r \in \mathbb{R}$ and a set Q of m line segments contained in P. Let c be the unconstrained geodesic center of P, which can be computed in $O(n)$ time [1]. In this

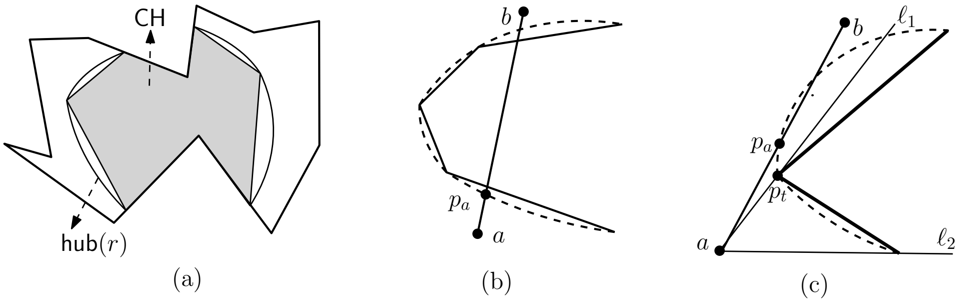

Figure 4 (a) The gray region is the geodesic convex hull of the endpoints of the circular arcs of the hub. (b) If ab crosses the convex chain, it crosses also the boundary of the hub. (c) If ab crosses the hub but does not cross the convex chain, then ab crosses the arc one of whose endpoint is p_t, where p_t is the point where a line passing through a is tangent to the convex chain.

section, we compute the intersection points of hub(r) and line segments in Q.

The boundary of hub(r) consists of (possibly empty) circular arcs and (possibly empty) polygonal chains which are from the boundary of P. Let ab be a line segment contained in P. Since hub(r) is geodesically convex by Lemma 1, ab intersects at most two circular arcs of hub(r). Moreover, one intersection point is closer to a than b, and the other one is closer to b than a. We first show how to compute the intersection point closer to a. The other intersection point can be computed analogously.

Let p_a be the intersection point closer to a. Consider the geodesic convex hull CH of the endpoints of circular arcs of hub(r). See Figure 4(a). Since we have the boundary of hub(r), we can compute CH in $O(n)$ time. We find the connected component R of $P \setminus \mathsf{CH}$ containing a in $O(\log n)$ time [16]. The connected region R contains a convex chain H of CH on its boundary. If the ray from a towards b hits H at some point in an edge e of H, then p_a is contained in the circular arc of hub(r) whose endpoints are the endpoints of e. See Figure 4(b). Thus, we can compute p_a in $O(\log n)$ time.

However, it is possible that p_a exists but the ray from a towards b does not hit H. See Figure 4(c). In this case, we consider the two lines ℓ_1 and ℓ_2 passing through a and tangent to H, which can be computed in $O(\log n)$ time. The point p_a lies in a circular arc of hub(r) one of whose endpoints is a point where ℓ_1 or ℓ_2 is tangent to H. Thus, in any case, we can compute p_a in $O(\log n)$ time.

▶ **Lemma 8.** *Given* hub(r) *with $r \in \mathbb{R}$ and a line segment ab contained in P, the circular arcs of* hub(r) *intersected by ab can be computed in $O(\log n)$ time after linear-time preprocessing for* hub(r).

4.2 Finding the Cell Containing the Geodesic Center

Let Q be a set of m line segments whose endpoints lie on the boundary of hub(r). In this section, we compute the cell Γ in the arrangement of Q inside hub(r) containing the unconstrained geodesic center c in $O(n + m \log(n + m))$ time.

For each line segment in Q, we extend the line segment in both directions until the two endpoints hit the boundary of P in $O(\log n)$ time [10]. Then a line segment ℓ in Q partitions P into two subpolygons one of which contains c. Let ℓ^+ be the subpolygon bounded by ℓ and containing c. We first compute the intersection I of all subpolygons ℓ^+ for all line segments ℓ in $O(m \log m)$ time as follows.

We sort the line segments in Q by the order of their first endpoints along the boundary of P, and then handle them one by one in order as follows. Initially we set P to I. While we handle the line segments, we update I to the intersection of ℓ^+ for all line segments ℓ which are handled so far. The intersection I is bounded by polygonal chains from ∂P and parts of line segments of Q. Moreover, parts of line segments of Q lying on ∂I form a number of convex chains. To maintain I, we store each convex chain using a binary search tree. The first line segment ℓ of Q subdivides I into two subpolygons, and we update I to the subpolygon containing c. Here, ℓ is the only one element stored in a binary search tree. As we handle more line segments, we create more binary search trees. For the next line segment ℓ', if both endpoints of ℓ' lies after the most clockwise point in the convex chain stored in the last binary search tree, we create a new binary search tree containing only one element ℓ'. See Figure 3(b). Otherwise, ℓ' may cross $\partial I \setminus \partial P$ in at most two points. To find this, it is sufficient to check the first and the last binary search trees. Thus, this takes $O(\log m)$ time.

By definition, Γ is the intersection of I and $\mathsf{hub}(r)$. So, we compute the intersection of I and $\mathsf{hub}(r)$ by traversing the boundary of I starting from a line segment on ∂I in clockwise order as follows. When we reach an endpoint of some line segment, we find the endpoint next to it. Then we connect these two endpoints by the boundary of $\mathsf{hub}(r)$. In this procedure, we traverse the boundary of I and the boundary of $\mathsf{hub}(r)$ once. Thus, we can compute Γ in $O(n + m\log(n + m))$ time.

▶ **Lemma 9.** *Given $\mathsf{hub}(r)$ and a set of m line segments crossing the hub, the cell containing the geodesic center of P in the arrangement of $\mathsf{hub}(r)$ and the line segments can be computed in $O(n + m\log(n + m))$ time.*

Until now, we assumed that S coincides with the vertex set of P. However, once the farthest-point geodesic Voronoi diagram of S is computed, the algorithm in this section works also for the case where the points of S are allowed to lie in the interior of P. The arguments in this section prove the following theorem.

▶ **Theorem 10.** *Let P be a simple n-gon and let Q be a set of m line segments the lie in P. For a set S of k sites (points) in P, the geodesic centers of S constrained to Q with respect to P can be computed in $O(n + k + m\log(n + k + m))$ time, once the farthest-point geodesic Voronoi diagram of S with respect to P is computed.*

For small m and k, the running time to compute FVD dominates the time complexity of our algorithm. The running time of our algorithm can be improved slightly for the case by avoiding to compute FVD explicitly. Recall that our algorithm uses FVD to compute r_V, $\mathsf{hub}(r_V)$ and the overlay of the boundary of Γ and FVD. We can compute them without constructing FVD of S as follows. The geodesic distance between two points can be computed in $O(\log n)$ time [13] after $O(n)$ preprocessing, so r_V can be computed in $O(mk\log n)$ time by finding $f(q)$ for all $q \in Q_V$. We compute $\mathsf{hub}(r_V)$ in $O(k)$ time, once the geodesic convex hull of S is computed in $O(n+k\log(n + k))$ time by applying a technique similar to Theorem 6 in [19] which shows how to compute FVD of points on the boundary of a simple k-gon in $O(k)$ time. The overlay of the boundary of Γ and FVD can also be computed similarly.

▶ **Theorem 11.** *Let P be a simple n-gon and let Q be a set of m line segments the lie in P. For a set S of k sites (points) in P, the geodesic centers of S constrained to Q with respect to P can be computed in $O((m + k)\log(n + k + m) + n)$ time.*

4.3 Geodesic Centers Constrained to Disjoint Line Segments

In this section, we give an algorithm to compute the geodesic centers of S with respect to P constrained to a set Q of m disjoint line segments. For ease of explanation, we assume that S is the vertex set of P.

Recall that once FVD is computed, the algorithm for the general case of crossing line segments takes $O(n + m \log n)$ time except the last step, which finds the cell Γ in the arrangement of Q and $\mathsf{hub}(r)$ containing the unconstrained geodesic center c. We show how to compute the cell Γ in $O(n + m \log n)$ time for the case of disjoint line segments, which improves the running time slightly to $O(n + m \log n)$.

We have $\mathsf{hub}(r)$ and a set of line segments crossing the hub. Moreover, we know the intersection points of the boundary of the hub and each line segment but they are not sorted. Instead of sorting the intersection points along the boundary of the hub, which takes $O(m \log m)$ time, we give an $O(n + m)$-time algorithm to compute the cell containing the geodesic center of P in the arrangement of the line segments and $\mathsf{hub}(r)$.

For a circular arc β of the hub, we have the line segments intersecting β. Without loss of generality, we assume that the two endpoints of β are on the x-axis. There are at most two line segments that contribute to the boundary of Γ among the line segments intersecting β: one is the line segment s_L that is closest geodesically to c among the line segments that are to the left of c, and the other is the line segment s_R that is closest geodesically to c among the line segments that are to the right of c. We find two line segments s_L and s_R, if they exist, for every circular arc β of the hub in $O(n + m)$ time. After doing this, we have $O(n)$ line segments which are sorted along the boundary of the hub, and we can compute the cell containing the geodesic center in $O(n)$ time.

▶ **Lemma 12.** *Given $\mathsf{hub}(r)$ and a set of m disjoint line segments, the cell containing the unconstrained geodesic center of P in the arrangement of the hub and the line segments can be computed in $O(n + m \log n)$ time.*

For the case that Q is a set of disjoint polygonal regions contained in P with m vertices in total, the geodesic centers of P constrained to Q lie on the boundary of Q unless they coincide with the unconstrained geodesic center of P. Thus we can use the algorithm in this section to compute the geodesic centers constrained to a set of disjoint polygonal regions.

▶ **Theorem 13.** *Let P be a simple n-gon and let Q be a set of m disjoint line segments or disjoint polygonal regions with m vertices in total that lie in P. For a set S of k sites (points) in P, the geodesic centers of S constrained to Q with respect to P can be computed in $O(n + k + m \log(n + k))$ time, once the farthest-point geodesic Voronoi diagram of S with respect to P is computed.*

The algorithm also works in $O((m + k) \log(n + k) + mk \log n + n)$ time without constructing FVD of S as similar to Theorem 11.

▶ **Theorem 14.** *Let P be a simple n-gon and let Q be a set of m disjoint line segments or disjoint polygonal regions with m vertices in total that lie in P. For a set S of k sites (points) in P, the geodesic centers of S constrained to Q with respect to P can be computed in $O((m + k) \log(n + k) + mk \log n + n)$ time.*

5 Geodesic Centers Constrained to a Polygon of Special Types

In this section, we consider a few special types of polygons. When Q is a geodesic convex polygon or a set of disjoint geodesic pseudo polygons, which will be defined, we can compute the constrained geodesic centers in linear time once we have the farthest-point geodesic Voronoi diagram of S with respect to P. In addition, when all the vertices of Q lie on ∂P, we can compute the constrained geodesic centers efficiently without computing a farthest point geodesic Voronoi diagram. We assume that S is the vertex set of P unless stated otherwise.

5.1 Geodesic Convex Polygons and Geodesic Pseudo Polygons

In this subsection, we assume that the farthest-point geodesic Voronoi diagram FVD of S with respect to P is already computed.

Let Q be a geodesic convex polygon. By Lemma 7, the complexity of the overlay of FVD and Q is linear to the complexity of FVD and Q. Thus, we compute the overlay of FVD and Q in linear time by traversing the cells of FVD and the edges of Q. Then, we choose the points which minimize the geodesic distance to their farthest sites in linear time.

We call a polygon contained in P a *geodesic pseudo polygon* if its boundary consists of (possibly empty) polygonal chains from ∂P and (possibly empty) concave chains lying in the interior of P. Let Q be a set of disjoint geodesic pseudo polygons contained in P. Note that the region of P lying outside of the polygons of Q may not be connected. If the unconstrained geodesic center of P is contained in Q, then it is also the unique geodesic center of P constrained to Q. Thus, we are done. Otherwise, we find the connected component R of the region of P lying outside of the polygons of Q containing the unconstrained geodesic center in linear time. Then, by Lemma 2 and the geodesic convexity of a hub, all constrained geodesic centers lie on the boundary of the concave chains of Q shared by R. Thus, we compute the overlay of FVD and the concave chains in linear time by Lemma 7 and return the answer.

▶ **Theorem 15.** *Let P be a simple n-gon and let Q be a geodesic convex polygon or a set of disjoint geodesic pseudo polygons with m vertices in total. For a set S of k sites (points) in P, the geodesic centers of S constrained to Q with respect to P can be computed in $O(n+m+k)$ time once the farthest-point geodesic Voronoi diagram of S with respect to P is computed.*

5.2 Polygons with Vertices on the boundary of P

In this subsection, we consider a set Q of disjoint polygonal regions whose vertices are on the boundary of P and show how to compute the geodesic centers constrained to Q efficiently without computing the whole FVD.

We assume that Q does not contain the unconstrained geodesic center. As we did in the previous case, we compute the connected component R of $P \setminus Q$ containing the unconstrained geodesic center in linear time. Then, we have the set Q' of the edges of the regions in Q that lie on ∂R. By Lemma 2 and the geodesic convexity of a hub, all constrained geodesic centers lie on line segments in Q'.

We compute the overlay of a line segment $\ell \in Q'$ and FVD as follows. The line segment ℓ subdivides P into two parts exactly one of which contains R. Let R' be the part of P which does not contain R. Let S_1 be the set of sites of S contained in R', and S_2 be the set of sites in S whose refined cells of FVD intersect the boundary of R' excluding ℓ. Then we consider the farthest-point geodesic Voronoi diagram of S_1 restricted to ℓ, which we denote by FVD_1, and the farthest-point geodesic Voronoi diagram of S_2 restricted to ℓ, which we denote by FVD_2. Once we have FVD_1 and FVD_2, we can compute the overlay of FVD and ℓ in time linear to the total complexity of FVD_1 and FVD_2.

If all sites are on the boundary of P, we can compute FVD_1 and FVD_2 in linear time for all line segments $\ell \in Q'$ [19]. Otherwise, we compute FVD_1 and FVD_2 in $O(n + k\log(n+k))$ time for all line segments $\ell \in Q'$ combining the results by [2, 14, 19].

▶ **Theorem 16.** *Let P be a simple n-gon and let Q be a set of disjoint polygonal regions with m vertices in total whose vertices are on the boundary of P. For a set S of k sites (points) in P, the geodesic centers of S constrained to Q with respect to P can be computed in $O(n+m+k\log(n+k))$ time.*

References

1 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. In *Proc. 31st Int'l Symposium on Computational Geometry (SoCG 2015)*, pages 209–223, 2015.

2 Boris Aronov, Steven Fortune, and Gordon Wilfong. The furthest-site geodesic Voronoi diagram. *Discrete & Computational Geometry*, 9(1):217–255, 1993.

3 Tetsuo Asano and Godfried Toussaint. Computing the geodesic center of a simple polygon. Technical Report SOCS-85.32, McGill University, 1985.

4 Luis Barba. Disk constrained 1-center queries. In *Proc. 24th Canadian Conference on Computational Geometry (CCCG 2012)*, pages 15–19, 2012.

5 Luis Barba, Prosenjit Bose, and Stefan Langerman. Optimal algorithms for constrained 1-center problems. In *Proc. 11th Latin American Theoretical Informatics Symposium (LATIN 2014)*, pages 84–95, 2014.

6 Prosenjit Bose, Stefan Langerman, and Sasanka Roy. Smallest enclosing circle centered on a query line segment. In *Proc. 20th Canadian Conference on Computational Geometry (CCCG 2008)*, pages 167–170, 2008.

7 Prosenjit Bose and Godfried Toussaint. Computing the constrained Euclidean geodesic and link center of a simple polygon with applications. In *Proc. 14th Computer Graphics International (CGI 1996)*, pages 102–110, 1996.

8 Prosenjit Bose and Qingda Wang. Facility location constrained to a polygonal domain. In *Proc. 5th Latin American Theoretical Informatics Symp. (LATIN'02)*, pages 153–164, 2002.

9 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991.

10 Bernard Chazelle, Herbert Edelsbrunner, Michelangelo Grigni, Leonidas Guibas, John Hershberger, Micha Sharir, and Jack Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12(1):54–68, 1994.

11 Bernard Chazelle, Herbert Edelsbrunner, Leonidas Guibas, Micha Sharir, and Jack Snoeyink. Computing a face in an arrangement of line segments and related problems. *SIAM Journal on Computing*, 22(6):1286–1302, 1993.

12 Leonidas Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1):209–233, 1987.

13 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.

14 John Hershberger and Subhash Suri. Matrix searching with the shortest-path metric. *SIAM Journal on Computing*, 26(6):1612–1634, 1997.

15 Ferran Hurtado, Vera Sacristán, and Godfried Toussaint. Some constrained minimax and maximin location problems. *Studies in Locational Analysis*, 15:17–35, 2000.

16 David Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12(1):28–35, 1983.

17 D.T. Lee and V.B. Wu. Multiplicative weighted farthest neighbor Voronoi diagrams in the plane. In *Proc. International Workshop on Discrete Mathematics and Algorithms*, pages 154–168, 1993.

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

19 Eunjin Oh, Luis Barba, and Hee-Kap Ahn. The farthest-point geodesic Voronoi diagram of points on the boundary of a simple polygon. To appear in Proc. 32nd International Symposium on Computational Geometry (SoCG 2016), 2016.

20 Richard Pollack, Micha Sharir, and Günter Rote. Computing the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 4(6):611–626, 1989.