A near-linear time exact algorithm for the L_1-geodesic Fréchet distance between two curves on the boundary of a simple polygon*

Thijs van der Horst
Department of Information and Computing Sciences, Utrecht University, the Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands

Marc van Kreveld
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Tim Ophelders
Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands

Bettina Speckmann
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

Let P be a polygon with k vertices. Let R and B be two simple, interior disjoint curves on the boundary of P, with n and m vertices. We show how to compute the Fréchet distance between R and B using the geodesic L_1-distance in P in $\mathcal{O}(k \log nm + (n+m)(\log^2 nm \log k + \log^4 nm))$ time.

Keywords and phrases Fréchet distance, rectilinear geodesics, simple polygon

Digital Object Identifier 10.57717/cgt.v5i1.114

Funding *Tim Ophelders*: partially supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.

1 Introduction

In a seminal paper from 1992, Alt and Godau [1] introduced the *Fréchet distance* to the algorithms community as a better similarity measure for two curves than, for example, the Hausdorff distance. For two polygonal curves with n and m vertices, they addressed the algorithmic problem of computing the Fréchet distance and presented an $\mathcal{O}(nm \log nm)$ time algorithm [2], which is near-quadratic in the input size.

Since then, research efforts have developed globally in two directions: examining variations or generalizations that can still be solved in near-quadratic time, and restrictions or approximations to beat the quadratic time upper bound. In 2014, Bringmann [4] showed that the Fréchet distance cannot be computed to within a factor 1.001 in strongly subquadratic time, unless the strong exponential time hypothesis fails. This result was improved in 2019 by Buchin, Ophelders and Speckmann [6], who showed that this conditional lower bound applies even for curves in 1-dimensional space and for approximations better than a factor 3. On the other hand, Driemel, Har-Peled and Wenk [9] showed that a near-linear time $(1+\varepsilon)$-approximation exists for curves in the plane that satisfy a sparseness condition. A full overview of the results is beyond the scope of this paper.

One of the variations that has been examined concerns computing the Fréchet distance between two curves in a polygonal environment. Intuitively, the Fréchet distance is based on matching points on the two curves in a continuous and forward manner, and considering

* A preliminary version of this paper [15] has appeared in Proceedings of the 19th International Symposium on Algorithms and Data Structures (WADS 2025).

the Euclidean distance between matched points. In a polygonal environment, the Euclidean distance is replaced by the Euclidean *geodesic* distance, the length of a shortest path in the environment. In the setting of two curves inside a simple polygon, Cook and Wenk [12] showed that the Fréchet distance can still be computed in near-quadratic time. Recently, it was shown that if the two curves lie on the boundary of a simple polygon, then a $(1+\varepsilon)$-approximation exists that uses near-linear time [14]. In this paper we show that in the same setting, but using the L_1-distance to measure lengths of geodesics rather than the Euclidean distance, an exact algorithm can be given that requires near-linear time.

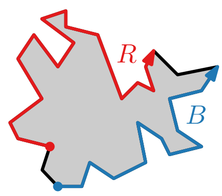
Preliminaries. Let P be a simple polygon. A *curve* on the boundary of P is a piecewise linear function $F\colon [0,1] \to \partial P$ connecting a sequence of *vertices*. Consecutive vertices are connected by a straight-line *edge*. The curve is *simple* if the preimage of a point on ∂P is at most an interval. We denote by $F[x, x']$ the subcurve of F over the domain $[x, x']$. We write $|F|$ to denote the number of vertices of F.

A *reparameterization* of $[0,1]$ is a non-decreasing surjection $f\colon [0,1] \to [0,1]$. Two reparameterizations f and g describe a *matching* (f, g) between two curves R (red) and B (blue), where for every $t \in [0,1]$ the point $R(f(t))$ is matched to $B(g(t))$. The matching (f, g) is said to have *cost* $\max_t \ d(R(f(t)), B(g(t)))$, where in this work, $d(p, q)$ measures the length of a shortest path in P from p to q under the L_1-norm. We call such a path an L_1-*geodesic*. There are potentially uncountably many L_1-geodesics from p to q. On the other hand, under the L_2-norm, P contains a unique shortest path from p to q, which we denote by $\pi(p, q)$. It turns out that $\pi(p, q)$ is also an L_1-geodesic, so we use it as a representative.

A matching with cost at most δ is called a δ-*matching*. The (continuous) L_1-*geodesic Fréchet distance* $d_{\mathrm{F}}(R, B)$ between R and B is the minimum cost over all matchings. The corresponding matching is a *Fréchet matching*.

The *parameter space* of R and B is the axis-aligned rectangle $[0,1]^2$. Any point (x, y) in the parameter space corresponds to the pair of points $R(x)$ and $B(y)$ on the two curves. A point (x, y) in the parameter space is δ-*close* for some $\delta \geq 0$ if $d(R(x), B(y)) \leq \delta$. The δ-*free space* $\mathcal{F}_\delta(R, B)$ of R and B is the subset of $[0,1]^2$ containing all δ-close points. A point $q = (x', y') \in \mathcal{F}_\delta(R, B)$ is δ-*reachable* from a point $p = (x, y)$ if $x \leq x'$ and $y \leq y'$, and there exists a bimonotone path in $\mathcal{F}_\delta(R, B)$ from p to q. Alt and Godau [2] observe that there is a one-to-one correspondence between δ-matchings between $R[x, x']$ and $B[y, y']$, and bimonotone paths from p to q through $\mathcal{F}_\delta(R, B)$. We abuse terminology slightly and refer to such paths as δ-matchings.

Problem statement and results. Our input consists of a simple polygon P with k vertices and interior-disjoint, simple curves $R, B\colon [0,1] \to \partial P$ with n and m vertices respectively, on the boundary of P. We restrict ourselves to curves R and B that are oppositely-oriented, meaning one is oriented clockwise around the boundary of P and the other counter-clockwise. See the figure on the right. We present an algorithm for computing the Fréchet distance $d_{\mathrm{F}}(R, B)$ between R and B, when distances between points are measured as the minimum length of a path in P, under the L_1-norm. For our algorithm, we make two assumptions. Firstly, we assume that R is oriented clockwise with respect to P, and B counter-clockwise. Secondly, we assume general position: no edge of P has slope ± 1, and no two vertices of P have the same y-coordinate. We compute the Fréchet distance between R and B under the geodesic L_1-distance in P in $\mathcal{O}(k \log nm + (n+m)(\log^2 nm \log k + \log^4 nm))$ time.



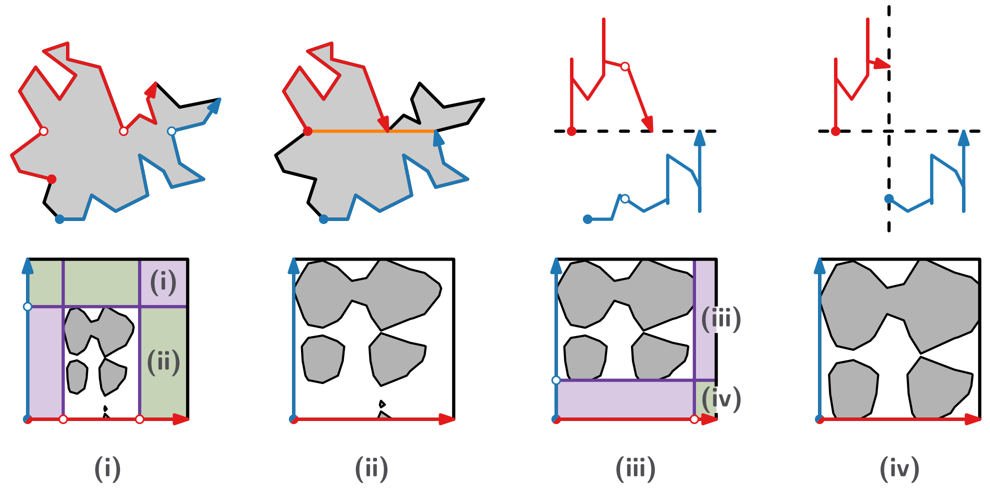


Figure 1 The four types of subproblems that arise in our divide-and-conquer scheme. The bottom row shows the parameter spaces corresponding to the top row. Between problem types, the correspondence between some subcurves and parts of the δ-free space is illustrated.

Algorithmic outline. As a subroutine in our algorithm, we make use of a *decision algorithm*. For an additional parameter $\delta \geq 0$, a decision algorithm reports whether $d_F(R, B) \leq \delta$. That is, whether a δ-matching between R and B exists. We apply our decision algorithm to $O(\log nm)$ values of δ to compute the exact value of $d_F(R, B)$. Recall that a δ-matching between R and B corresponds to a bimonotone path in $\mathcal{F}_\delta(R, B)$ from $(0, 0)$ to $(1, 1)$. Our decision algorithm determines whether such a path exists.

Our algorithm uses divide-and-conquer to reduce the general decision problem into subproblems with additional structure. We consider four types of subproblems in this work, which we label types **(i)**–**(iv)**. The types are illustrated in Figure 1. The first type, type **(i)**, is the main focus of this work. Formally, it takes as input curves R and B of the following type:

(i) R and B are oppositely oriented, interior-disjoint, simple curves on the boundary of a simple polygon P.

In Section 3.3 we find a horizontal chord e^* of P that intersects both R and B in at most a constant number of points. We split R and B at these points, resulting in subcurves R_i of R and B_j of B. Each pair (R_i, B_j) corresponds to a simpler subproblem. Such a subproblem is either of type **(i)** where R_i and B_j have at most a constant fraction of the vertices of R and B, or of a new type with extra structure:

(ii) R_i and B_j are separated by a horizontal chord of P, namely e^*.

In Figure 1 **(i)**, the subproblems partition the parameter space into six regions. The green regions correspond to subcurves of type **(ii)**, and the purple regions to subcurves of type **(i)**.

We handle the subproblems of type **(i)** recursively. We handle subproblems of type **(ii)** in Section 3.2. Suppose that R and B are separated by a horizontal chord e^* of P, such as the subproblems of type **(ii)**. For a point $p \in P$, denote by $NN(p)$ the point on e^* closest to p under the metric function d. We leverage the property that for all $r \in R$ and $b \in B$ there exists an L_1-geodesic from r to b that goes via $NN(r)$ and $NN(b)$. Using this property, we transform the problem (R, B) of type **(ii)** into an equivalent problem of a new type that is no longer constrained by the polygon:

(iii) A pair of x-monotone[1] curves $(\bar{R}, \bar{B})$ that are separated by a horizontal line.

We handle subproblems of type **(iii)** in Section 3.1. Suppose that R and B are x-monotone curves separated by a horizontal line, such as the subproblems of type **(iii)**. We again split the problem into simpler subproblems. We find a vertical line ℓ that has a constant fraction of the vertices of $R \cup B$ on either side. We split R and B at points on ℓ, resulting in subcurves R_1 and R_2 of R, and B_1 and B_2 of B. Each pair (R_i, B_j) corresponds to a simpler subproblem. Such a subproblem is either of type **(iii)** and handled recursively, or of a new type:

(iv) R_i and B_j are x-monotone curves separated by both a horizontal and a vertical line.

In Figure 1 **(iii)**, the subproblems partition the parameter space into four regions. The green regions correspond to subcurves of type **(iv)**, and the purple regions to subcurves of type **(iii)**.

The above scheme splits our original problem into several subproblems of type **(iv)**, each of which corresponds to an axis-aligned rectangle in the parameter space of the two input curves. Together, these rectangles partition the parameter space into interior-disjoint regions. Recall that we want to decide whether there exists a bimonotone path from $(0,0)$ to $(1,1)$ in the parameter space that lies completely in the δ-free space. To do so, we maintain a set of points in the δ-free space that can be reached by a bimonotone path that starts at $(0,0)$. If for a subproblem of type **(iv)** we know for its corresponding rectangle a set of reachable "entry" points on the bottom and left sides, then we can compute a sufficient set of "exit" points on the top and right sides that are reachable via at least one entry point. See Section 2 for details. By processing the subproblems of type **(iv)** in the correct order, we compute whether a bimonotone path from $(0,0)$ to $(1,1)$ through the δ-free space exists.

In Section 4.2, we use our decision algorithm to compute $d_{\mathrm{F}}(R,B)$ exactly, by binary searching over a set of $\tilde{\mathcal{O}}((n+m)^2)$ candidate values and querying the decision algorithm at every step. Computing all candidate values explicitly would take too much time. Instead, we represent them implicitly as the Cartesian sum of two sorted sets X and Y of only $\mathcal{O}((n+m)\log^2 nm)$ values. Using algorithms for linear-time selection in such implicitly represented Cartesian sums [10, 13], the time taken to compute a candidate for δ is dominated by the time taken to perform the decision algorithm. This allows us to compute $d_{\mathrm{F}}(R,B)$ with only logarithmic overhead compared to the decision algorithm.

We present our decision algorithm bottom-up, starting with the most specific problem **(iv)** that arises, and ending with the general problem **(i)**, in the following sections.

2 Doubly-separated x-monotone curves

In this section, we consider subproblems of type **(iv)**, where R and B are two x-monotone curves in $\mathbb{R}^2$ separated by both a horizontal line and a vertical line. Let R have n vertices and B have m vertices. We present an algorithm for *propagating reachability information* through the δ-free space of R and B. We are given a parameter $\delta \geq 0$, a set of $\mathcal{O}(n+m)$ points S on the bottom and left sides of the parameter space, and a set of $\mathcal{O}(n+m)$ points T on the top and right sides of the parameter space. The goal is to report the points in T that are δ-reachable from at least one point in S. See Figure 2. In this section, we measure distances under the L_1-norm $\|\cdot\|_1$, in which shortest paths are not restricted by any polygon.

Lemma 1 shows that the distances between R and B can be captured by two curves in one dimension that are separated by the origin, which can be constructed in linear time.

[1] Throughout this work, we consider monotonicity in the weak sense. That is, a curve is x-monotone if every vertical line intersects the curve in at most one subcurve.

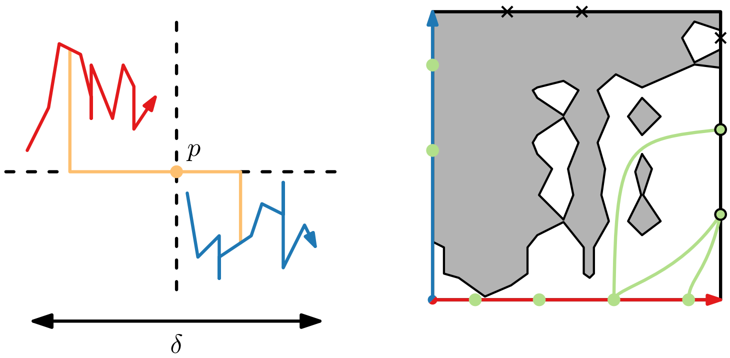


Figure 2 (left) A pair of doubly-separated x-monotone curves. We may consider the curves as lying on the union of two (degenerate) "histogram" polygons, in which case d is equal to the L_1-norm. Between any pair of red and blue points, there exists an L_1-geodesic through the point p. (right) Their δ-free space diagram. For propagating reachability, given are a set of points S (disks) on the bottom and left sides of the parameter space, and a set T (circles and crosses) of points on the top and right sides. The filled circles are δ-reachable, the crosses are not.

▶ **Lemma 1.** *In $\mathcal{O}(n+m)$ time, we can construct two curves $\bar{R}$ and $\bar{B}$ in $\mathbb{R}$ that are separated by the origin, such that $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ for all δ. The curves $\bar{R}$ and $\bar{B}$ have n and m vertices, respectively.*

Proof. Let p be a point such that the horizontal and vertical lines through p both separate R from B. Between each pair of points $R(x)$ and $B(y)$, there exists a shortest path (under the L_1-norm) through p, so $\|R(x) - B(y)\|_1 = \|R(x) - p\|_1 + \|p - B(y)\|_1$. Let $\bar{R}, \bar{B} \colon [0, 1] \to \mathbb{R}$ where $\bar{R}(x) = -\|R(x) - p\|_1$ and $\bar{B}(y) = \|B(y) - p\|_1$ (note the difference in sign). This definition lets p correspond to 0 in one-dimensional space, and places the curve $\bar{R}$ on one side of p and $\bar{B}$ on the other. For any x, y, we have $\|R(x) - B(y)\|_1 = |\bar{B}(y) - \bar{R}(x)|$, so $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ for all δ.

We can compute a horizontal and vertical line separating R and B in $\mathcal{O}(n + m)$ time by determining the orthogonal bounding boxes of R and B. From here we obtain p, and constructing $\bar{R}$ and $\bar{B}$ then takes a further $\mathcal{O}(n + m)$ time. ◀

Bringmann and Künnemann [5] and the authors [14] gave near-linear time algorithms for propagating reachability information through the free space of two separated curves in one dimension. Combining this with Lemma 1, we obtain:

▶ **Lemma 2.** *Given a parameter $\delta \geq 0$ and sets S and T of $\mathcal{O}(n + m)$ points on the bottom and left, respectively top and right, sides of the parameter space of R and B, we can report the points in T that are δ-reachable from at least one point in S in $\mathcal{O}((n + m) \log nm)$ time.*

3 Partitioning the parameter space

In this section, we consider the main problem, of type **(i)**, where R and B are simple, interior-disjoint curves on the boundary of a simple polygon P. We divide this problem into subproblems of type **(iv)** by partitioning the parameter space of R and B into axis-aligned rectangles where either the corresponding subcurves are trivial (i.e., line segments), or form a type **(iv)** instance, in that the subcurves are equivalent to a pair of doubly-separated x-monotone curves.

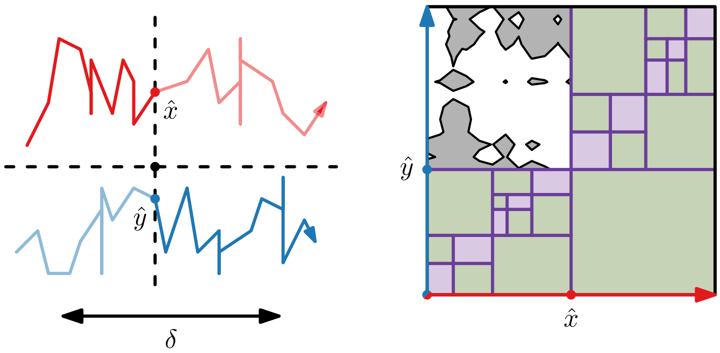


Figure 3 (left) A pair of x-monotone curves separated by a horizontal line, together with a vertical line with roughly half of all vertices on either side. (right) This line represents a partition of the parameter space into four regions, of which the top-left and bottom-right regions correspond to pairs of doubly-separated subcurves. The bottom-left and top-right regions are partitioned further. Green regions signify subproblems of type **(iv)**.

An axis-aligned rectangle $[x, x'] \times [y, y']$ in the parameter space of R and B is *doubly-separated* if there exists a pair of doubly-separated x-monotone curves $\bar{R}$ and $\bar{B}$ in $\mathbb{R}^2$, such that $\mathcal{F}_\delta(R[x, x'], B[y, y']) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ for all δ. A *doubly-separated partition* of the parameter space is a partition into axis-aligned rectangles $[x, x'] \times [y, y']$, where either $[x, x'] \times [y, y']$ is doubly-separated, or both $R[x, x']$ and $B[y, y']$ are line segments. In the latter case, we refer to such a region as *trivial*. A doubly-separated partition always exists, since we can take its regions to be only trivial regions, corresponding to the pairs of individual edges of the curves. Such a partition has high complexity however, having nm regions. Our goal is instead to compute a doubly-separated partition with few trivial regions, and where the doubly-separated regions correspond to subcurves with relatively few vertices in total, so that the total input complexity when applying the result of Lemma 2 is kept low.

To measure the "quality" of a doubly-separated partition, we define the *cost* $\mathrm{Cost}(\mathcal{R})$ of a region $\mathcal{R}$ as the total complexity of its corresponding subcurves. The cost of a trivial region is then $\mathcal{O}(1)$, and the cost of a doubly-separated region $\mathcal{R} = [x, x'] \times [y, y']$ is $\big|R[x, x']\big| + \big|B[y, y']\big|$. The cost of a doubly-separated partition $\mathcal{P}$ is then defined as $\mathrm{Cost}(\mathcal{P}) = \sum_{\mathcal{R} \in \mathcal{P}} \mathrm{Cost}(\mathcal{R})$.

In this section we construct a doubly-separated partition of only near-linear cost, namely $\mathcal{O}((n + m) \log^2 nm)$. We construct this partition in three phases. In Section 3.1 we construct a partition for subproblems of type **(iii)**, where the curves R and B are x-monotone and separated by a horizontal line. Then, in Section 3.2, we handle subproblems of type **(ii)**, where R and B are simple curves on the boundary of a simple polygon P, separated by a horizontal line segment in P. Finally, in Section 3.3, we handle subproblems of type **(i)**.

3.1 Vertically-separated x-monotone curves

In this section, we consider the setting where R and B are x-monotone curves in $\mathbb{R}^2$, separated by a horizontal line. We give a simple partition algorithm for the parameter space of R and B that yields a doubly-separated partition of near-linear cost:

▶ **Lemma 3.** *We can compute a doubly-separated partition of the parameter space of R and B of cost $\mathcal{O}((n + m) \log nm)$ in $\mathcal{O}((n + m) \log nm)$ time.*

Proof. We recursively partition the parameter space as follows. If $|R| \leq 2$ and $|B| \leq 2$ then the parameter space itself is a doubly-separated partition with only a trivial region and cost

$\mathcal{O}(1)$, so we stop partitioning further. Also, if R and B are already separated by a vertical line, then the parameter space is a doubly-separated region with cost $\mathcal{O}(n+m)$, and so we stop partitioning further as well. Otherwise, we pick a vertical line ℓ and points $R(\hat{x})$ and $B(\hat{y})$ on ℓ, such that $R[0,\hat{x}]$ and $B[0,\hat{y}]$ together have at most $(n+m)/2+1$ vertices (and $R[\hat{x},1]$ and $B[\hat{y},1]$ together have at most $(n+m)/2+1$ vertices as well). The line ℓ can be seen as partitioning the parameter space into four quadrants that meet at $(\hat{x},\hat{y})$. See Figure 3.

Let $\mathcal{P}$ be the partition defined by the four quadrants. The top-left region in $\mathcal{P}$ is the parameter space of $R[0,\hat{x}]$ and $B[\hat{y},1]$. Since these subcurves are separated by ℓ, as well as by the horizontal line ℓ^* separating R and B, they are doubly-separated. Thus, the top-left region of $\mathcal{P}$ is doubly-separated. Symmetrically, the bottom-right region of $\mathcal{P}$ is doubly-separated. The bottom-left and top-right quadrants are not doubly-separated however. We therefore recursively partition $\mathcal{P}$ in these regions.

The cost of $\mathcal{P}$ is the sum of costs of the partitions in the four quadrants. We analyze the cost of $\mathcal{P}$ through a recurrence. Let $C(n,m)$ denote the cost of the resulting doubly-separated partition when R and B have n and m vertices, respectively. That is, $\mathrm{Cost}(\mathcal{P}) = C(n,m)$. The top-left and bottom-right quadrants are not recursively partitioned and have costs $\big|R[0,\hat{x}]\big|+\big|B[\hat{y},1]\big|$ and $\big|R[\hat{x},1]\big|+\big|B[0,\hat{y}]\big|$, respectively, totaling at most $n+m+4$. The bottom-left and top-right quadrants are partitioned further. This gives the following recurrence:

$$C(n,m) \leq \begin{cases} n+m & \text{if } n \leq 2 \text{ and } m \leq 2, \\ C(n_1,m_1)+C(n_2,m_2)+n+m+4 & \text{otherwise,} \end{cases}$$

where $n_1+m_1 \leq (n+m)/2+2$ and $n_2+m_2 \leq (n+m)/2+2$. This recurrence solves to $C(n,m) = \mathcal{O}((n+m)\log nm)$.

The line ℓ used to split R and B can be found in $\mathcal{O}(n+m)$ time through a linear scan of the curves, since they are x-monotone and their vertices are hence sorted by x-coordinate. Let $T(n,m)$ denote the time taken to compute the resulting doubly-separated partition when R and B have n and m vertices, respectively. This quantity follows the following recurrence:

$$T(n,m) \leq \begin{cases} \mathcal{O}(1) & \text{if } n \leq 2 \text{ and } m \leq 2, \\ T(n_1,m_1)+T(n_2,m_2)+\mathcal{O}(n+m) & \text{otherwise,} \end{cases}$$

where again $n_1+m_1 \leq (n+m)/2+2$ and $n_2+m_2 \leq (n+m)/2+2$. This recurrence solves to $T(n,m) = \mathcal{O}((n+m)\log nm)$. ◀

3.2 Separated curves on a simple polygon

Next we consider the setting where R and B are simple curves on the boundary of a simple polygon P, where R and B are separated by a horizontal line segment in P. That is, there exists a horizontal line segment e^* in P, such that splitting P at e^* creates two subpolygons where R and B lie in different subpolygons.[2] We assume R to be oriented clockwise with respect to P, and B counter-clockwise. We reduce this setting to that of Section 3.1, by constructing a pair of vertically-separated x-monotone curves whose δ-free space (under the L_1-norm) is identical to that of R and B, for all δ.

Every L_1-geodesic between points $r \in R$ and $b \in B$ intersects e^* in exactly one (possibly degenerate) line segment. Moreover, there exists an L_1-geodesic that is composed of the

[2] We consider e^* to be part of both subpolygons.

following three parts: an L_1-geodesic between r and its closest point u on e^*, an L_1-geodesic between b and its closest point v on e^*, and the line segment $\overline{uv}$. For two vertically-separated x-monotone curves (as in Section 3.1), there exists a similar shortest path between two points, namely one that is composed of two vertical segments and a horizontal segment on the separator. We use this connection for our reduction, which we specify next.

We refer to the *projection* of a point $p \in P$ onto e^* as the point on e^* closest to p, and we denote this projection by $NN(p)$. Note that the projection of a point onto e^* is unique, as e^* has slope other than ± 1. As mentioned above, between any two points $r \in R$ and $b \in B$ there exists an L_1-geodesic that goes through the projections $NN(r)$ and $NN(b)$. We introduce two real-valued functions, which we use to define the coordinates of the x-monotone curves we reduce the problem to.

We assume e^* is parameterized by arclength, i.e., uniformly over $[0, \|e^*\|]$. The first function, $\varphi\colon P \to [0, \|e^*\|]$, indexes the projections on e^* of the points in P. That is, $e^*(\varphi(p)) = NN(p)$. The second function, $\psi\colon P \to \mathbb{R}$, represents the distance function from points to their projections, so $\psi(p) = d(p, NN(p))$.

We consider the curves $\bar{R}$ and $\bar{B}$ in the plane, defined as

$$\bar{R}\colon [0,1] \to \mathbb{R}^2, \qquad \bar{R}(x) = (\varphi(R(x)), \quad \psi(R(x))), \text{ and}$$
$$\bar{B}\colon [0,1] \to \mathbb{R}^2, \qquad \bar{B}(y) = (\varphi(B(y)), -\psi(B(y))).$$

Note that $\bar{d}(R(x), B(y)) = \|\bar{R}(x) - \bar{B}(y)\|_1$ for all $(x, y) \in [0,1]^2$, and so $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ for all δ. The curves are separated by the horizontal line $(-\infty, \infty) \times \{0\}$. Next we prove that the curves are x-monotone, thus making the algorithm of Section 3.1 applicable to them.

▶ **Lemma 4.** *The curves $\bar{R}$ and $\bar{B}$ are x-monotone.*

Proof. We prove the statement for $\bar{R}$; it follows by symmetry that $\bar{B}$ is x-monotone. Let $0 \le x \le x' \le 1$ be two values. We show that $\bar{R}(x)$ lies to the left of or on $\bar{R}(x')$.

Let $r = R(x)$ and $r' = R(x')$. It suffices to show that $NN(r)$ comes before $NN(r')$ along e^*. Suppose for a contradiction that $NN(r)$ comes strictly after $NN(r')$ along e^*, for some points $r, r' \in R$ with r before r'. Due to the orientation of R relative to e^*, every path from r to $NN(r)$ intersects every path from r' to $NN(r')$. In particular, any pair of rectilinear shortest paths from r and r' to respectively $NN(r)$ and $NN(r')$, intersect in some point p. Naturally, $NN(p)$ must be equal to both $NN(r)$ and $NN(r')$. However, $NN(r) \neq NN(r')$ by assumption, which is a contradiction. ◀

We have shown that the algorithm developed in Section 3.1 can be applied to the curves $\bar{R}$ and $\bar{B}$. Next we show that $\bar{R}$ and $\bar{B}$ have low complexity. Let R have n vertices and B have m vertices. We show that $\bar{R}$ has $\mathcal{O}(n)$ vertices and $\bar{B}$ has $\mathcal{O}(m)$ vertices. To do so, we consider the edges of R and B individually, proving that $\varphi|_e$ and $\psi|_e$ have only $\mathcal{O}(1)$ complexity for any edge e. For this, we consider the L_1-analogue of the "hourglass functions" introduced by Cook and Wenk [12] for computing the (L_2-)geodesic Fréchet distance.

Two interior-disjoint[3] segments $e, e' \subseteq P$ specify the following L_1*-hourglass function* $H_{e,e'}$:

$$H_{e,e'}\colon e \to \mathbb{R}, \qquad H_{e,e'}(p) = \min_{q \in e'} d(p, q).$$

The L_1-hourglass function generalizes ψ, since $H_{e,e^*} = \psi|_e$ for any edge e of P. We use this more general definition in Section 4.1.

[3] Hourglass functions are defined for any pair of segments in P. We only require them for interior-disjoint segments however, and limit L_1-hourglass functions to them for simplicity.

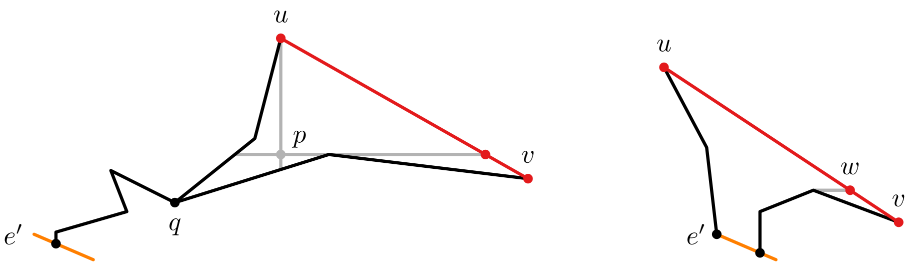


Figure 4 An illustration of the two settings discussed in Lemma 5. On the left is the case where $u^* = v^*$. On the right is the case where $u^* \neq v^*$. In this case, all points on the segment $\overline{uw}$ can reach their closest point on e' by a bimonotone path.

The hourglass functions introduced by Cook and Wenk may have $\Theta(k)$ complexity [12]. Perhaps surprisingly however, the L_1-analogues of these hourglass functions are much less complex. Namely, we show in Lemma 5 that they are piecewise linear with only $\mathcal{O}(1)$ pieces. This implies that the function ψ is piecewise linear with $\mathcal{O}(1)$ pieces when restricted to an edge of P.

▶ **Lemma 5.** *The L_1-geodesic hourglass function $H_{e,e'}$ between two interior-disjoint segments e and e' is piecewise linear with $\mathcal{O}(1)$ pieces.*

Proof. Let $e = \overline{uv}$ and let u^* and v^* be the points on e' closest to u and v, respectively. We first prove the claimed complexity bound when $u^* = v^*$. In this case, $H_{e,e'}(w) = d(w, u^*)$ for all $w \in e$, since for all points on e, the closest point on e' is u^*. We prove the existence of a point $p \in P$ such that $H_{e,e'}(w) = \|w - p\|_1 + d(p, u^*)$ for all $w \in e$. Since $d(p, u^*)$ is independent of w, this point implies that $H_{e,e'}$ behaves like the regular L_1-norm.

The representative L_1-geodesics $\pi(u, u^*)$ and $\pi(v, v^*)$ share a common suffix $\pi(q, u^*) = \pi(q, v^*)$, but are otherwise disjoint [11]. The paths $\pi(u, q)$ and $\pi(v, q)$ are convex [11], and make a total angle of strictly less than π due to them being disjoint apart from one point. Hence there is exactly one maximal horizontal segment e_H and one maximal vertical segment e_V in P incident to e and one of $\pi(u, q)$ and $\pi(v, q)$, while being tangent to the other. Furthermore, e_H and e_V must intersect in a point, which we set p to. See Figure 4 (left).

Observe that p as chosen meets our requirements. Indeed, for every point $w \in e$, there exists an L_1-geodesic to u^* that goes through p. Additionally, it is bimonotone up to p, i.e., it is monotone in both x- and y-coordinate, so $d(w, p) = \|w - p\|_1$. It follows that $H_{e,e'}(w) = \|w - p\|_1 + d(p, u^*)$. By properties of the L_1-norm, $H_{e,e'}$ is piecewise linear with at most three pieces.

Next we prove the claimed complexity bound when $u^* \neq v^*$. Here, the representative L_1-geodesics $\pi(u, u^*)$ and $\pi(v, v^*)$ are disjoint. Thus, there exists a point $w \in e$ for which the path $\pi(w, w^*)$ to its closest point w^* on e' is a line segment. In particular, it is bimonotone, i.e., monotone in both x- and y-coordinate. In Figure 4 (right) for example, all points on $\overline{uw}$ have this property. Let $\hat{e}$ be the maximal subsegment of e, such that any L_1-geodesic between a point $w \in \hat{e}$ and its closest point $w^* \in e'$ is bimonotone. Note that for such points, $d(w, w^*) = \|w - w^*\|_1$. Hence, $H_{\hat{e},e'}$ is equivalent to the L_1-normed distance function from points on $\hat{e}$ to their closest point on e'. By properties of the L_1-norm, $H_{\hat{e},e'}$ is therefore piecewise linear with $\mathcal{O}(1)$ pieces.

Let e_u and e_v be the maximal subsegments of e that are interior-disjoint from $\hat{e}$, and additionally incident to u, respectively v. All points on e_u have u^* as the point on e' closest to them. Thus, $H_{e_u,e'} = H_{e_u,\{u^*\}}$. We already proved that $H_{e_u,\{u^*\}}$ is piecewise linear with $\mathcal{O}(1)$ pieces (since trivially, the endpoints of e_u have the same closest point on $\{u^*\}$). Symmetrically, $H_{e_v,e'} = H_{e_v,\{v^*\}}$ is piecewise linear with $\mathcal{O}(1)$ pieces. Thus, $H_{e,e'}$ has the claimed complexity. ◀

Next, we give a data structure that efficiently computes the functions $\varphi|_e$ and ψ_e for any given edge e of P:

▶ **Lemma 6.** *Let e be an edge of P. After preprocessing P in $\mathcal{O}(k)$ time, $\varphi|_e$ and $\psi|_e$ can be computed in $\mathcal{O}(\log k)$ time.*

Proof. Let $e = \overline{uv}$ and let u^* and v^* be the points on e^* closest to u and v, respectively. We first compute u^* and v^*. Since e^* is a horizontal line segment, u^* and v^* are not only closest to u and v under L_1-geodesic distance, but also under regular (L_2-)geodesic distance. Thus, we can compute u^* and v^* in $\mathcal{O}(\log k)$ time, after $\mathcal{O}(k)$ preprocessing, using a data structure by Cook and Wenk [12].

To compute $\varphi|_e$ and $\psi|_e$, we first consider the case where $u^* = v^*$. In this case, $\varphi|_e$ is a constant function, equaling u^* everywhere. Additionally, $\psi|_e(w) = H_{e,\{u^*\}}(w) = d(w, u^*)$ for all $w \in e$. We use that $\pi(u, u^*)$ and $\pi(v, v^*)$ share a common suffix $\pi(q, u^*) = \pi(q, v^*)$, but are otherwise disjoint [11]. Recall from the proof of Lemma 5 that there is exactly one maximal horizontal segment e_H and one maximal vertical segment e_V in P that are incident to e and one of $\pi(u, q)$ and $\pi(v, q)$, while being tangent to the other. In that proof, we showed that $d(w, NN(w)) = \|w - p\|_1 + d(p, u^*)$ for all $w \in e$, where p is the intersection point between e_H and e_V. We compute e_H and e_V, to obtain p.

We first compute the point q. To do so, we represent $\pi(u, u^*)$ and $\pi(v, v^*)$ with balanced binary search trees, obtained in $\mathcal{O}(\log k)$ time (after $\mathcal{O}(k)$ preprocessing) from the data structure of Guibas and Hershberger [11]. Note that the vertices q' on the shared suffix $\pi(q, u^*)$ are exactly the vertices of $\pi(u, u^*)$ with $d(u, q') = d(v, q')$. We may therefore find q by performing a binary search over $\pi(u, u^*)$, and testing whether $d(u, q') = d(v, q')$ at each vertex q' encountered. This test takes constant time (after $\mathcal{O}(k)$ preprocessing) with the data structure of Bae and Wang [3], since u and v are vertices of P. Thus, we can find q in $\mathcal{O}(\log k)$ time.

Next we compute e_H and e_V, by performing another binary search over each path. This yields the two segments, and thus p, in $\mathcal{O}(\log k)$ additional time. Then, we can compute $H_{e,\{u^*\}}$, and thus $\psi|_e$, in $\mathcal{O}(1)$ time once we have computed $d(p, u^*)$, which we can do in $\mathcal{O}(\log k)$ time [3].

Next we consider the case where $u^* \neq v^*$. Here, the representative L_1-geodesics $\pi(u, u^*)$ and $\pi(v, v^*)$ are disjoint. Additionally, the vertical lines through u^* and v^* are tangent to $\pi(u, u^*)$ and $\pi(v, v^*)$, respectively. Thus, there are points $u', v' \in e$ for which $\overline{u'u^*}$ and $\overline{v'v^*}$ are vertical segments inside P. We compute u' and v' in $\mathcal{O}(1)$ time.

The segments $\overline{u'u^*}$ and $\overline{v'v^*}$ make a right angle with e^*, and thus $NN(u') = u^*$ and $NN(v') = v^*$. The points u' and v' split e into three parts: a prefix $\overline{uu'}$ and suffix $\overline{v'v}$ where every point has the same projection onto e^* (namely, u' or v', respectively), and a middle part $\overline{u'v'}$ where $\varphi|_{\overline{u'v'}}$ and $\psi|_{\overline{u'v'}}$ behave as if the L_1-geodesic distance function is simply the L_1-norm. We can compute $\varphi|_{\overline{u'v'}}$ and $\psi|_{\overline{u'v'}}$ in constant time, and so it follows from the first case, where $u^* = v^*$, that we can compute $\varphi|_e$ and $\psi|_e$ in $\mathcal{O}(\log k)$ time, after $\mathcal{O}(k)$ preprocessing time. ◀

By querying the data structure of Lemma 6 with every edge of R and B individually and combining the results, we obtain the functions φ and ψ over R and B in $\mathcal{O}((n + m) \log k)$ time, after $\mathcal{O}(k)$-time preprocessing. From this, we extract $\bar{R}$ and $\bar{B}$ in $\mathcal{O}(n + m)$ additional time. The complexities of $\bar{R}$ and $\bar{B}$ are $\mathcal{O}(n)$ and $\mathcal{O}(m)$, respectively. Thus we obtain:

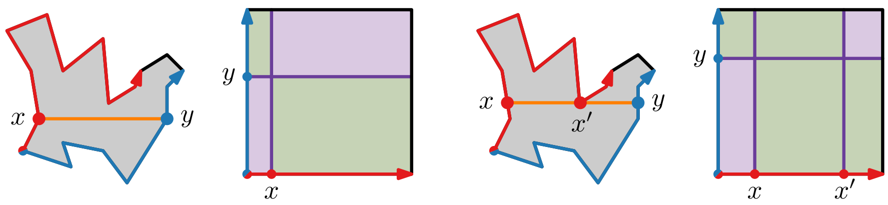


Figure 5 Two types of bichromatic chords and the partitions of the parameter space that they induce. All green regions correspond to separated subcurves; the purple regions are split recursively.

▶ **Lemma 7.** *After preprocessing P in $\mathcal{O}(k)$ time, we can construct a pair of vertically-separated x-monotone curves $\bar{R}$ and $\bar{B}$ with $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ in $\mathcal{O}((n+m)\log k)$ time. The curves $\bar{R}$ and $\bar{B}$ have $\mathcal{O}(n)$ and $\mathcal{O}(m)$ vertices, respectively.*

▶ **Corollary 8.** *After preprocessing P in $\mathcal{O}(k)$ time, we can compute a doubly-separated partition of the parameter space of R and B of cost $\mathcal{O}((n+m)\log nm)$ in $\mathcal{O}((n+m)\log nm)$ time.*

3.3 The general case

In this section, we consider the general setting, where R and B are simple, interior-disjoint curves on the boundary of a simple polygon P, and construct a doubly-separated partition of the parameter space of R and B.

We follow the approach of Section 3.1, in that we partition the parameter space recursively, though this time based on horizontal *chords* of P. A chord is a maximal segment that does not go outside P. We assume a kind of general position on the vertices of P, namely that no two vertices lie on the same horizontal line. This implies that a horizontal chord can have at most three points in common with the boundary of P.

If possible, we will use bichromatic horizontal chords, which have a point in common with R and B. Each such chord e^* splits each curve into at most three subcurves: If e^* intersects R only at endpoints of R, the chord "trivially" splits R into the curve R itself. Otherwise, R is split into the maximal prefix $R[0, x]$ whose intersection with e^* is only the point $R(x)$, the maximal suffix $R[x', 1]$ whose intersection with e^* is only the point $R(x')$, and the maximal subcurve $R[x, x']$ bounded by e^*. See Figure 5. The split of B induced by e^* is defined analogously. Note that if R is split into three subcurves, then B is split into at most two subcurves (and vice versa). A chord corresponds to a partition of the parameter space into the axis-aligned rectangles whose corresponding subcurves are induced by the splits of R and B. Thus, a chord corresponds to a partition into at most six regions.

If a bichromatic chord does not exist, then R and B can be separated trivially by a horizontal chord. Our partition algorithm works recursively, like the algorithm of Lemma 3. To bound the recursive depth, we use a chord that splits R and B into at most two and three subcurves each, where the first and last subcurves of each have a total of at most $(n+m)/c$ vertices at either side of the chord, for some constant $c > 1$.

We prove the existence of a horizontal chord with the desired properties, and give a construction algorithm. The result follows from a relaxed version of the "polygon-cutting theorem" of Chazelle [7].

▶ **Lemma 9.** *If R and B are not already separated by a horizontal chord, there exists a horizontal chord that intersects both R and B. If additionally $n + m \geq 3$, such a chord exists that splits R and B into at most three subcurves each, where for each pair (R_i, B_j) of* *subcurves not separated by the chord,* $|R_i| + |B_j| \leq 2(n+m)/3$*. Such a chord can be found in* $\mathcal{O}(k)$ *time.*

Proof. If R and B are not already separated by a horizontal chord, then naturally there exists a horizontal chord that intersects both R and B. For the remainder, assume that R and B are not separated by a horizontal chord.

For the existence of the particular chord we are after, we assume for simplicity that the endpoints of R and B are vertices of P, and that these are distinct. We make use of the relaxed polygon-cutting theorem of Chazelle [7]. To use this theorem, we assign the vertices of P that are also vertices of $R \cup B$ a weight of 1, and all other vertices a weight of 0. Let $C(P)$ denote the total weight of the vertices of P, and note that $C(P) = n + m \geq 3$. From Chazelle's relaxed polygon-cutting theorem we obtain the existence of a horizontal chord that splits P into subpolygons P_1 and P_2 with $C(P_1) \leq C(P_2) \leq 2C(P)/3$. Furthermore, we can compute such a chord e in $\mathcal{O}(k)$ time: compute the horizontal decomposition of P with the algorithm of Chazelle [8] and consider the faces in topological order.

With our general position assumption on the vertices of P, the chord intersects one curve in at most two points and the other in at most one point. Suppose without loss of generality that e intersects R in two points, splitting it into the subcurves $R_1 = R[0, x]$, $R_2 = R[x, x']$ and $R_3 = R[x', 1]$. If e intersects B in a point $B(y)$, then it splits B into the subcurves $B_1 = B[0, y]$ and $B_2 = B[y, 1]$. The prefixes R_1 and B_1 are not separated by e, but they lie in the same subpolygon of P. Thus $|R_1| + |B_1| \leq \max\{C(P_1), C(P_2)\} \leq 2(n+m)/3$. Symmetrically, the suffixes R_3 and B_2 are not separated by e, but satisfy $|R_3| + |B_2| \leq 2(n+m)/3$. The other four pairs of subcurves are separated by e.

If e does not intersect B, then the endpoints of e are $R(x)$ and $R(x')$. We compute a suitable bichromatic chord by sweeping the chord e in the direction away from $R[x, x']$, stretching and shrinking the chord as necessary to remain a chord of P. We do so until e hits a point on B, either with an endpoint of e or internal point. Let e^* be this chord.

The chord e^* splits R into subcurves where the prefix and suffix are strict subcurves of R_1 and R_3, respectively. Because the original chord e does not intersect B, we have that e splits B in a trivial way. Thus, B lies completely on one side of e, and therefore also on one side of e^*. The number of vertices of prefixes and suffixes induced by the split of e^*, on either side of e^*, are therefore bounded by those for e, and thus by $2(n+m)/3$.

To compute e^*, during the sweep we ensure that the chord always connects a point on R_1 and a point on R_3. This gives a clear choice for how to advance the chord when it reaches a polygon vertex with its interior. With the horizontal decomposition of P, constructable in $\mathcal{O}(k)$ time [8], this sweepline procedure computes e^* in $\mathcal{O}(k)$ time. ◀

With the existence proof and construction algorithm for good chords at hand, we make an initial partition of the parameter space. The partition is not yet a doubly-separated partition, but we refine it afterwards into one that is. We recursively partition the parameter space as follows. If $|R| \leq 2$ and $|B| \leq 2$ then we stop partitioning further. Otherwise, we compute a horizontal chord e^* of P with the algorithm of Lemma 9, taking $\mathcal{O}(k)$ time.

The chord intersects one curve in at most two points and the other in at most one point. Suppose that e^* intersects R in two points, splitting it into the subcurves $R_1 = R[0, x]$, $R_2 = R[x, x']$ and $R_3 = R[x', 1]$, and that e^* intersects B in one point, splitting it into the subcurves $B_1 = B[0, y]$ and $B_2 = B[y, 1]$. These five subcurves together define six axis-parallel rectangles in the parameter space, which is the partition corresponding to e^*.

Let $\mathcal{P}$ be this partition. The subcurve R_2 is bounded by e^*, and thus is separated from all of B by e^*. Further, the subcurve R_1 is separated from B_2, and the subcurve R_3 from B_1.

Thus the only regions in the partition whose corresponding subcurves are not separated by a horizontal chord are the bottom-left region $[0, x] \times [0, y]$ and the top-right region $[x', 1] \times [y, 1]$; we recursively partition $\mathcal{P}$ in these regions.

We analyze the partition in the same manner as we do for doubly-separated partitions. That is, we refer to the cost of an axis-aligned rectangle $\mathcal{R} = [x, x'] \times [y, y']$ as $\mathrm{Cost}(\mathcal{R}) = \left|R[x, x']\right| + \left|B[y, y']\right|$, the same as for doubly-separated regions, and refer to the cost of the partition $\mathcal{P}$ as $\mathrm{Cost}(\mathcal{P}) = \sum_{\mathcal{R} \in \mathcal{P}} \mathrm{Cost}(\mathcal{R})$.

▶ **Lemma 10.** *The partition $\mathcal{P}$ has cost $\mathcal{O}((n+m) \log nm)$ and can be computed in $\mathcal{O}(k \log nm)$ time.*

Proof. Let $C(n, m)$ denote the cost of $\mathcal{P}$ when R and B have n and m vertices, respectively. That is, $\mathrm{Cost}(\mathcal{P}) = C(n, m)$. This quantity follows the following recurrence:

$$C(n, m) \leq \begin{cases} n + m & \text{if } n \leq 2 \text{ and } m \leq 2, \\ C(n_1, m_1) + C(n_2, m_2) + \mathcal{O}(n + m) & \text{otherwise,} \end{cases}$$

where $n_1 + m_1 \leq 2(n + m)/3 + 2$, $n_2 + m_2 \leq 2(n + m)/3 + 2$, $n_1 + n_2 \leq n + 2$, and $m_1 + m_2 \leq m + 2$. This recurrence solves to $\mathrm{Cost}(\mathcal{P}) = C(n, m) = \mathcal{O}((n + m) \log nm)$.

The chord e^* is computed in $\mathcal{O}(k)$ time (Lemma 9). Given e^*, the corresponding partition can be computed in $\mathcal{O}(n + m)$ time by scanning over the curves. This gives a running time of $\mathcal{O}(k + n + m)$ per recursive step. Naively, since the resulting partition $\mathcal{P}$ has $\mathcal{O}(n + m)$ regions, the total time taken to compute $\mathcal{P}$ is therefore $\mathcal{O}((n + m) \cdot (k + n + m))$. Note, however, that when recursively partitioning the region $[0, x] \times [0, y]$, the curves $R_1 = R[0, x]$ and $B_1 = B[0, y]$ lie on the same side of e^*. That is, e^* splits P into two subpolygons P_1 and P_2, and R_1 and B_1 lie in the same subpolygon, say P_1. Thus, we may restrict ourselves to P_1 when computing chords for the further partitioning. The same goes for partitioning the region $[x', 1] \times [y', 1]$, though using P_2 instead. We therefore analyze the time taken to compute $\mathcal{P}$ through a recurrence.

Let $T(k, n, m)$ denote the time taken to compute an doubly-separated partition with the above procedure, when P has k vertices and R and B have n and m vertices, respectively. That is, the time taken to compute $\mathcal{P}$ is $T(k, n, m)$. This quantity follows the following recurrence:

$$T(k, n, m) \leq \begin{cases} \mathcal{O}(1) & \text{if } n \leq 2 \text{ and } m \leq 2, \\ T(k_1, n_1, m_1) + T(k_2, n_2, m_2) + \mathcal{O}(k + n + m) & \text{otherwise,} \end{cases}$$

where $k_1 + k_2 \leq k + 4$, $n_1 + m_1 \leq 2(n+m)/3 + 2$, $n_2 + m_2 \leq 2(n+m)/3 + 2$, $n_1 + n_2 \leq n + 2$, and $m_1 + m_2 \leq m + 2$. This recurrence solves to $T(k, n, m) = \mathcal{O}((k + n + m) \log nm) = \mathcal{O}(k \log nm)$. ◀

We refine $\mathcal{P}$ into a doubly-separated partition. Some regions in $\mathcal{P}$ already correspond to line segments on R and B, and so are trivial. The other regions correspond to subcurves of R and B that are separated by a horizontal chord of P. We partition these regions further using the result of Corollary 8.

▶ **Lemma 11.** *We can compute a doubly-separated partition of the parameter space of R and B that has cost $\mathcal{O}((n + m) \log^2 nm)$ in $\mathcal{O}(k \log nm + (n + m) \log^2 nm)$ time. Additionally, in the same time bound, we can compute, for each region $[x, x'] \times [y, y']$ in the partition, a pair of doubly-separated x-monotone curves $\bar{R}$ and $\bar{B}$ with $\mathcal{F}_\delta(R[x, x'], B[y, y']) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ for all δ. The curves $\bar{R}$ and $\bar{B}$ have $\mathcal{O}(\left|R[x, x']\right|)$ and $\mathcal{O}(\left|B[y, y']\right|)$ vertices, respectively.*

Proof. Let $\mathcal{R} = [x, x'] \times [y, y'] \in \mathcal{P}$ be a region where $R[x, x']$ and $B[y, y']$ are separated by a horizontal chord. We partition this region further through the connection of $R[x, x']$ and $B[y, y']$ with vertically-separated x-monotone curves. For this, we compute a pair of vertically-separated x-monotone curves $\bar{R}$ and $\bar{B}$ with $\mathcal{F}_\delta(R[x, x'], B[y, y']) = \mathcal{F}_\delta(\bar{R}, \bar{B})$. These curves have $\mathcal{O}(|R[x, x']|)$ and $\mathcal{O}(|B[y, y']|)$ vertices, respectively, and can be computed in $\mathcal{O}(\text{Cost}(\mathcal{R}) \log k)$ time, after preprocessing P in $\mathcal{O}(k)$ time (Lemma 7).

We compute a doubly-separated partition $\bar{\mathcal{P}}$ of the parameter space of $\bar{R}$ and $\bar{B}$ that has cost $\mathcal{O}(\text{Cost}(\mathcal{R}) \log \text{Cost}(\mathcal{R}))$. This partition can be computed in $\mathcal{O}(\text{Cost}(\mathcal{R}) \log \text{Cost}(\mathcal{R}))$ time (Lemma 3). The partition $\bar{\mathcal{P}}$ induces a doubly-separated partition of $\mathcal{R}$ with the same asymptotic cost.

By partitioning all regions $\mathcal{R} = [x, x'] \times [y, y'] \in \mathcal{P}$ where $R[x, x']$ and $B[y, y']$ are separated by a horizontal chord in the above manner, we obtain the desired partition of the parameter space of R and B. Let $\mathcal{P}^*$ be the resulting refined partition. Its cost is

$$\text{Cost}(\mathcal{P}^*) = \sum_{\mathcal{R} \in \mathcal{P}} \mathcal{O}(\text{Cost}(\mathcal{R}) \log \text{Cost}(\mathcal{R})) = \mathcal{O}(\text{Cost}(\mathcal{P}) \log nm) = \mathcal{O}((n + m) \log^2 nm).$$

The time taken to compute the initial partition $\mathcal{P}$ is $\mathcal{O}(k \log nm)$ (Lemma 10). The time taken to compute its refinement $\mathcal{P}^*$, together with the doubly-separated x-monotone curves associated with the respective regions, is therefore

$$\mathcal{O}(k \log nm) + \sum_{\mathcal{R} \in \mathcal{P}} \mathcal{O}(\text{Cost}(\mathcal{R}) \log nm) = \mathcal{O}(k \log nm + (n + m) \log^2 nm). \quad \blacktriangleleft$$

4 Computing the Fréchet distance

In this section, we discuss the main problem setting of this work. Let P be a simple polygon and let R and B be two simple, interior-disjoint curves on the boundary of P. We assume R is oriented clockwise with respect to P, and B counter-clockwise. We present a near-linear time algorithm for computing $d_F(R, B)$.

Our algorithm makes use of a *decision algorithm* that, given a parameter $\delta \geq 0$, reports whether $d_F(R, B) \leq \delta$ or $d_F(R, B) > \delta$. We make use of the connection between L_1-geodesic Fréchet distance and $\mathcal{F}_\delta(R, B)$. Specifically, the fact that $d_F(R, B) \leq \delta$ if and only if there exists a bimonotone path in $\mathcal{F}_\delta(R, B)$ from $(0, 0)$ to $(1, 1)$. We develop an algorithm that decides whether such a path exists.

In Section 4.1 we use the doubly-separated partition constructed in Section 3 to efficiently propagate reachability information through the various regions. Recall that there are two types of regions in a doubly-separated partition: trivial regions and doubly-separated regions. For trivial regions, we give a simple algorithm for propagating reachability information through them, using that these regions correspond to pairs of segments on R and B. For the other regions, we use the result of Section 2 instead, to also propagate reachability information through them. This leads to our decision algorithm. In Section 4.2, we turn our decision algorithm into an optimization algorithm.

4.1 The decision algorithm

We present an algorithm for deciding whether $d_F(R, B) \leq \delta$ for a given parameter $\delta \geq 0$. Our algorithm decides whether a bimonotone path in $\mathcal{F}_\delta(R, B)$ exists from $(0, 0)$ to $(1, 1)$. Let $\mathcal{P}^*$ be a doubly-separated partition of the parameter space of R and B computed with the algorithm of Lemma 11. We propagate reachability information through each region in $\mathcal{P}^*$ separately.

Before we do so, we need to address one issue. For the regions $\mathcal{R} \in \mathcal{P}^*$ where we have a pair of doubly-separated x-monotone curves $\bar{R}$ and $\bar{B}$ whose free space is identical to that inside $\mathcal{R}$, we wish to apply the result of Lemma 2. However, this algorithm is for propagating reachability from a given discrete set of points to a given discrete set of points. Hence, we need to first determine and construct sets $S_\mathcal{R}$ and $T_\mathcal{R}$ to which we apply the result of Lemma 2.

To define $S_\mathcal{R}$ and $T_\mathcal{R}$, we say that a point $B(x)$ is *locally closest* to $R(y)$ if an infinitesimal perturbation of $B(x)$ while staying on B increases its distance to $R(y)$. We call points on R locally closest to points on B if a symmetric condition holds. We define $T_\mathcal{R}$ as the set of all points (x^*, y^*) on the top and right sides of $\mathcal{R}$ for which at least one of $R(x^*)$ and $B(y^*)$ is a vertex of R or B, or locally closest to the other point. We define $S_\mathcal{R}$ as consisting of $\{(0,0)\} \cap \mathcal{R}$, as well as all points on the bottom and left sides of $\mathcal{R}$ that are in a set $T_{\mathcal{R}'}$ (for an adjacent region $\mathcal{R}'$) and are δ-reachable from $S_{\mathcal{R}'}$.

We proceed to show that there exists an optimal matching that enters and leaves each region $\mathcal{R}$ through points in $S_\mathcal{R}$ and $T_\mathcal{R}$, respectively. To do so, we construct a Fréchet matching (f', g') between R and B where for every pair (r, b) of matched points, at least one of r and b is a vertex or locally closest to the other point. We construct (f', g') by using an arbitrary Fréchet matching (f, g).

We first state (a subset of) the points that we let (f', g') match the vertices of R and B to. For a vertex r of R, let $B[y_1, y_2]$ be the maximal subcurve of B matched to it by (f, g). Let $B[\bar{y}_1, \bar{y}_2]$ be the minimal subcurve of B that contains $B[y_1, y_2]$, where $B(\bar{y}_1)$ and $B(\bar{y}_2)$ are vertices. We let (f', g') match r to a point on $B[\bar{y}_1, \bar{y}_2]$ closest to it (under L_1-geodesic distance). This point has distance at most δ to r, and is either locally closest to r, or a vertex of B. We let (f', g') match each vertex of B to a point on R defined in a symmetric manner. We prove that these matches satisfy the monotonicity of a matching between R and B:

▶ **Lemma 12.** *For any two pairs of matched points (r, b) and (r', b') currently matched in (f', g'), if r comes strictly before r' on R, then b comes before b' on B.*

Proof. Let (r, b) and (r', b') be matches in (f', g'), where r comes strictly before r' on R. Suppose r is a vertex of R; the other cases, where either r' is a vertex of R, or b is a vertex of B, are symmetric. We distinguish between two cases, namely the case where r' is a vertex of R, and the case where b' is a vertex of R.

Suppose first that r' is a vertex of R. Let $B[y_1, y_2]$ and $B[y'_1, y'_2]$ be the maximal subcurves of B matched to r and r', respectively, by (f, g). Since r comes before r', we have $y_2 \leq y'_1$. The minimal subcurves $B[\bar{y}_1, \bar{y}_2]$ and $B[\bar{y}'_1, \bar{y}'_2]$ containing $B[y_1, y_2]$ and $B[y'_1, y'_2]$, respectively, whose endpoints are vertices, overlap in at most one edge. That is, if $\bar{y}_2 > \bar{y}'_1$, the subcurve $B[\bar{y}'_1, \bar{y}_2]$ is an edge of B. If one of b and b' does not lie on this edge, it trivially holds that b comes before b'. Otherwise, if b were to come after b', then the representative geodesics $\pi(r, b)$ and $\pi(r', b')$ cross. Among the points on $B[\bar{y}'_1, \bar{y}_2]$, b and b' are closest to r and r', respectively. Since we assumed no edge of R or B to have slope ± 1, this is only possible if $b = b'$. Thus, b comes before b' in this case as well.

Next suppose that b' is a vertex of B, and assume for a contradiction that b' comes strictly before b. Let $B[y_1, y_2]$ be the maximal subcurve of B matched to r by (f, g), and let $B(\bar{y})$ be the last vertex of B before $B(y_1)$. If b' is a point on the interior of $B[y_1, y_2]$, then (f, g) matches b' to only r, from which it follows that $r' = r$. This contradicts our choice of r and r'. If b' is not a point on the interior of $B[y_1, y_2]$, then it must be a vertex of $B[0, \bar{y}]$. However, then (f, g) matches b' to a subcurve $R[x'_1, x'_2]$ where $R(x'_2)$ comes before r. Thus, r' must come before r, again contradicting our choice of r and r'. ◀

Next we state how we let (f', g') match up the remainder of the points. Consider two maximal (but possibly degenerate) segments $\overline{rr'} = R[x, x']$ and $\overline{bb'} = B[y, y']$ where currently, r is matched to b and r' is matched to b'. Let $r^* \in \overline{rr'}$ and $b^* \in \overline{bb'}$ minimize the L_1-geodesic distance $d(r^*, b^*)$ between them. Recall that $d(r, b) \leq d_{\mathrm{F}}(R, B)$, and thus $d(r^*, b^*) \leq d_{\mathrm{F}}(R, B)$. Additionally, both r^* and b^* are either vertices or locally closest to the other point. We let (f', g') match r^* to b^*.

We state how we let (f', g') match $\overline{rr^*}$ to $\overline{bb^*}$; we let (f', g') match $\overline{r^*r'}$ to $\overline{b^*b'}$ in a symmetric manner. One of r and b is a vertex; we assume without loss of generality that this is r. Let $\hat{b} \in \overline{bb^*}$ be the point closest to r. We let (f', g') match r to $\overline{b\hat{b}}$, and further match each point on $\overline{rr^*}$ to its closest point on $\overline{\hat{b}b^*}$. This is a proper matching: the function mapping each point on $\overline{rr^*}$ to its closest point on $\overline{\hat{b}b^*}$ is a monotone surjection.

We show that the cost of matching $\overline{rr^*}$ to $\overline{bb^*}$ in the above manner is at most $d_{\mathrm{F}}(R, B)$. To do so, we make use of the following structural lemma:

▶ **Lemma 13.** *The L_1-geodesic hourglass function $H_{e,e'}$ between two interior-disjoint segments e and e' has a maximum at an endpoint of e.*

Proof. Suppose for a contradiction that the maximum is not attained at a vertex of e. Let $e = \overline{uv}$, and let $w \in e$ be an interior point at which the maximum is attained. Consider a point $w^* \in e'$ closest to w. Assume first that $\pi(w, w^*)$ is bimonotone, i.e., monotone in both x- and y-coordinate. Then we have $H_{e,e'}(w) = d(w, w^*) = \|w - w^*\|_1$, whereas in general, $d(p, q) \geq \|p - q\|_1$. Since under the L_1-norm, the maximum distance between a point on e and its closest point on e' is attained at a vertex of e, this implies that either

$$H_{e,e'}(u) = \min_{u' \in e'} d(u, u') \geq \min_{u' \in e'} \|u - u'\|_1 \geq \|w - w^*\|_1 = H_{e,e'}(w)$$

or $H_{e,e'}(v) \geq H_{e,e'}(w)$, giving a contradiction.

In the remainder we assume that $\pi(w, w^*)$ is not bimonotone. Observe that in this case, either u or v has w^* as a point on e' closest to it. Indeed, if $\pi(w, w^*)$ is not bimonotone, then it is not a line segment, and thus makes a turn at some reflex vertex p of P. The subpath $\pi(w, p)$ then separates one of u and v from the entire segment e'. Since any path from the separated point to a point on e' goes through $\pi(w, p)$, it follows that this separated point has w^* as a closest point on e'.

We distinguish between two cases, based on the angle that the first segment of $\pi(w, w^*)$ makes with $\overline{uw}$. First, suppose this angle is at least $\pi/2$. Then there are points $p, q \in \pi(u, w^*)$ for which $\overline{pw}$ is a horizontal segment and $\overline{qw}$ is a vertical segment, both in P. The path obtained by replacing the subpath of $\pi(u, w^*)$ between p and q with the path over the two line segments (in the appropriate order) is an L_1-geodesic from u to w^*. Thus, $d(u, w^*) \geq d(w, w^*)$. It directly follows that $H_{e,e'}(u) \geq H_{e,e'}(w)$, which is a contradiction.

Next, suppose the angle between the first segment of $\pi(w, w^*)$ and $\overline{uw}$ is strictly smaller than $\pi/2$. As $\pi(w, w^*)$ is not bimonotone, it has at least a horizontal and a vertical line tangent to it. Let p be the first point on $\pi(w, w^*)$ through which there exists such a tangent line, and let q be the intersection point between e and this tangent line. Because $\pi(w, w^*)$ has only reflex vertices of P as interior vertices, we obtain that $\overline{uq} \supseteq \overline{uw}$. Moreover, the angle between $\overline{pq}$ and $\overline{uq}$ is strictly smaller than the angle between the first segment of $\pi(w, w^*)$ and $\overline{uw}$. Under the L_1-norm, the orthogonality of $\overline{pq}$ then implies that $\|p - q\|_1 > \|p - w\|_1$. Thus, $d(q, w^*) = \|q - p\|_1 + d(p, w^*) > \|w - p\|_1 + d(p, w^*) = d(w, w^*)$. It directly follows that $H_{e,e'}(q) > H_{e,e'}(w)$, which is a contradiction. ◀

The maximum distance from r to $\overline{b\hat{b}}$ is attained by b or $\hat{b}$ (Lemma 13), and thus the cost of matching r to $\overline{b\hat{b}}$ is $\max\{d(r,b), d(r,\hat{b})\} \leq d_{\mathrm{F}}(R,B)$. Similarly, the maximum distance between a point on $\overline{rr^*}$ and its closest point on $\overline{\hat{b}b^*}$ is $\max\{d(r,\hat{b}), d(r^*,b^*)\} = d(r,\hat{b})$ (Lemma 13). Thus, the cost of matching $\overline{rr^*}$ to $\overline{\hat{b}b^*}$ in the manner specified above is $d(r,\hat{b}) \leq d_{\mathrm{F}}(R,B)$. Hence the matching (f', g') has cost at most $d_{\mathrm{F}}(R,B)$, and is therefore a Fréchet matching.

▶ **Lemma 14.** *There exists a Fréchet matching between R and B that enters and leaves each region $\mathcal{R}$ through points in $S_{\mathcal{R}}$ and $T_{\mathcal{R}}$, respectively.*

Naturally, for our decision algorithm, we need to explicitly construct the sets $S_{\mathcal{R}}$ and $T_{\mathcal{R}}$ for all regions $\mathcal{R} \in \mathcal{P}^*$. We therefore give an efficient, near-linear time algorithm to do so:

▶ **Lemma 15.** *In $\mathcal{O}(k + (n+m)\log^2 nm \log k)$ total time, we can compute the families of sets $S_{\mathcal{R}}$ and $T_{\mathcal{R}}$, where $\mathcal{R} \in \mathcal{P}^*$.*

Proof. We focus on constructing the sets $T_{\mathcal{R}}$. It follows from the definition of $S_{\mathcal{R}}$ that the sets $S_{\mathcal{R}}$ can be computed from the sets $T_{\mathcal{R}}$ in $\mathcal{O}\big(\sum_{\mathcal{R}\in P^*} |T_{\mathcal{R}}|\big)$ additional time.

By Lemma 5, the L_1-hourglass function $H_{e,e'}$ is piecewise linear with $\mathcal{O}(1)$ pieces, for any pair of interior-disjoint segments e and e'. If e' is horizontal, we can compute $H_{e,e'}$ in $\mathcal{O}(\log k)$ time, after $\mathcal{O}(k)$ preprocessing (Lemma 6). Thus, we can compute the point on an edge of R (resp. B) that is locally closest to a point $b \in B$ (resp. $r \in R$) in $\mathcal{O}(\log k)$ time, by viewing the point as a horizontal line segment. The set $T_{\mathcal{R}}$ for a region $\mathcal{R} = [x,x'] \times [y,y'] \in \mathcal{P}^*$ can therefore be computed in $\mathcal{O}((\big|R[x,x']\big| + \big|B[y,y']\big|) \log k) = \mathcal{O}(\mathrm{Cost}(\mathcal{R}) \log k)$ time. For all regions in $\mathcal{P}^*$ together, this takes $\mathcal{O}(\mathrm{Cost}(\mathcal{P}^*) \log k) = \mathcal{O}((n+m)\log^2 nm \log k)$ time. ◀

We are now ready to give our decision algorithm, which decides whether a bimonotone path in $\mathcal{F}_\delta(R,B)$ from $(0,0)$ to $(1,1)$ exists, for a given $\delta \geq 0$. First, we compute the doubly-separated partition $\mathcal{P}^*$ of Lemma 11, taking $\mathcal{O}(k \log nm + (n+m)\log^2 nm)$ time. The partition has cost $\mathcal{O}((n+m)\log^2 nm)$, and for each region $[x,x'] \times [y,y']$ in this partition, either $R[x,x']$ and $B[y,y']$ are line segments, or we have additionally computed a pair of doubly-separated x-monotone curves $\bar{R}$ and $\bar{B}$ with $\mathcal{O}(\big|R[x,x']\big|)$ and $\mathcal{O}(\big|B[y,y']\big|)$ vertices, respectively. We compute the sets $T_{\mathcal{R}}$ for all regions $\mathcal{R} \in \mathcal{P}^*$, taking $\mathcal{O}(k + (n+m)\log^2 nm \log k)$ time. The goal is then to compute, for each set $T_{\mathcal{R}}$, the subset of points that are δ-reachable from $(0,0)$.

Consider a region $\mathcal{R} = [x,x'] \times [y,y'] \in \mathcal{P}^*$ and suppose that for each region $\mathcal{R}'$ incident to the bottom or left side of $\mathcal{R}$ we have already computed the subset of $T_{\mathcal{R}'}$ of points that are δ-reachable from $(0,0)$. The union of these subsets forms the set $S_{\mathcal{R}}$ that serves as the set of "entrances" for $\mathcal{R}$; a point in $T_{\mathcal{R}}$ is δ-reachable from $(0,0)$ if and only if it is δ-reachable from a point in $S_{\mathcal{R}}$.

If $R[x,x']$ or $B[y,y']$ has more than one edge, we have available a pair of doubly-separated x-monotone curves $\bar{R}$ and $\bar{B}$ with $\mathcal{F}_\delta(R[x,x'], B[y,y']) = \mathcal{F}_\delta(\bar{R},\bar{B})$. We apply the algorithm of Lemma 2 to these curves, using the set $S_{\mathcal{R}}$ for the "entrances" and the set $T_{\mathcal{R}}$ for the "potential exits." This algorithm reports all points in $T_{\mathcal{R}}$ that are δ-reachable from at least one point in $S_{\mathcal{R}}$ in $\mathcal{O}((|\bar{R}| + |\bar{B}|) \log |\bar{R}||\bar{B}|) = \mathcal{O}(\mathrm{Cost}(\mathcal{R}) \log \mathrm{Cost}(\mathcal{R}))$ time.

If both $R[x,x']$ and $B[y,y']$ are line segments, then we show that we can check in constant time whether a given point in $T_{\mathcal{R}}$ is reachable from a given point in $S_{\mathcal{R}}$. Thus, we can report the set of points in $T_{\mathcal{R}}$ that are δ-reachable from at least one point in $S_{\mathcal{R}}$ in $\mathcal{O}(|S_{\mathcal{R}}| \cdot |T_{\mathcal{R}}|) = \mathcal{O}(1)$ time.

▶ **Lemma 16.** *If both $R[x,x']$ and $B[y,y']$ are line segments, then a point $t \in T_{\mathcal{R}}$ is δ-reachable from a point $s \in S_{\mathcal{R}}$ if and only if $t \in \mathcal{F}_\delta(R[x,x'],\ B[y,y'])$ and t lies above and to the right of s.*

Proof. Let $s = (x_s, y_s) \in S_{\mathcal{R}}$ and $t = (x_t, y_t) \in T_{\mathcal{R}}$ with $x_t \geq x_s$ and $y_t \geq y_s$. We construct a δ-matching between $R[x_s, x_t]$ and $B[y_s, y_t]$. To do so, let $r^* \in R[x_s, x_t]$ and $b^* \in B[y_s, y_t]$ minimize $d(r^*, b^*)$. Naturally, $d(r^*, b^*) \leq d(R(x_s), B(y_s)) \leq \delta$. We construct a δ-matching between $\overline{R(x_s)r^*}$ and $\overline{B(y_s)b^*}$. We construct a δ-matching between $\overline{r^*R(x_t)}$ and $\overline{b^*B(y_t)}$ in a symmetric manner.

Let $\hat{b} \in \overline{R(x_s)r^*}$ be the point closest to $R(x_s)$. The cost of matching $R(x_s)$ to $\overline{B(y_s)\hat{b}}$ is equal to $d(R(x_s), B(y_s)) \leq \delta$. From here, we match every point on $\overline{R(x_s)r^*}$ to their closest point on $\overline{B(y_s)b^*}$. This is a proper matching, as the closest point varies continuously from $B(y_s)$ to b^* as we vary a point from $R(x_s)$ to r^*. The cost of matching these pairs of points decreases continuously, thus this part of the matching has cost $d(R(x_s)\hat{b}) \leq \delta$. The complete matching is therefore a δ-matching. ◀

We now have an algorithm that, given the sets $S_{\mathcal{R}}$ and $T_{\mathcal{R}}$, reports the points in $T_{\mathcal{R}}$ that are δ-reachable in $\mathcal{O}(\mathrm{Cost}(\mathcal{R}) \log \mathrm{Cost}(\mathcal{R}))$ time. Applying this algorithm to all regions in $\mathcal{P}^*$, in some topological order, we eventually decide whether $(1, 1)$ is δ-reachable from $(0, 0)$, since it is included in a set $T_{\mathcal{R}}$. This takes

$$\sum_{\mathcal{R} \in \mathcal{P}^*} \mathcal{O}(\mathrm{Cost}(\mathcal{R}) \log \mathrm{Cost}(\mathcal{R})) = \mathcal{O}(\mathrm{Cost}(\mathcal{P}^*) \log nm) = \mathcal{O}((n + m) \log^3 nm)$$

time. The computations of $\mathcal{P}^*$ and the sets $T_{\mathcal{R}}$, which took $\mathcal{O}(k \log nm + (n+m) \log^2 nm \log k)$ time in total, dominate the running time. However, these computations are independent of the decision parameter δ, and so we view these computations as preprocessing. This gives the following result:

▶ **Lemma 17.** *After preprocessing P, R and B in $\mathcal{O}(k \log nm + (n + m) \log^2 nm \log k)$ time, we can decide, for any $\delta \geq 0$, whether $d_{\mathrm{F}}(R, B) \leq \delta$ in $\mathcal{O}((n + m) \log^3 nm)$ time.*

4.2 Obtaining an optimization algorithm

We conclude this section by turning the decision algorithm into an algorithm that computes $d_{\mathrm{F}}(R, B)$, at the cost of a logarithmic factor in the running time. For this, we derive a polynomial number of candidate distances, one of which is the actual Fréchet distance $d_{\mathrm{F}}(R, B)$. Specifically, for two sets of real numbers X and Y, their *Cartesian sum*, denoted $X \oplus Y$, is the multiset $X \oplus Y = \{\{x + y \mid x \in X, y \in Y\}\}$. We will show that we can compute two sets X and Y of $\mathcal{O}((n + m) \log^2 nm)$ real numbers each, such that $d_{\mathrm{F}}(R, B) \in X \oplus Y$. After computing X and Y in near-linear time, we can use existing techniques [10, 13] for selection in Cartesian sums to binary search over the Cartesian sum $X \oplus Y$ without explicitly computing its $|X| \cdot |Y|$ many elements.

We again make use of the doubly-separated partition $\mathcal{P}^*$ computed with the algorithm of Lemma 11. We use this partition to identify pairs of separated one-dimensional curves that together describe (almost) all pairwise distances between points on R and points on B. The sets X and Y will be the set of values of the vertices of these curves.

First, recall the sets $T_{\mathcal{R}}$, for regions $\mathcal{R} \in \mathcal{P}^*$, and recall that there exists a Fréchet matching between R and B that corresponds to a bimonotone path in the parameter space that, for each region $\mathcal{R} \in \mathcal{P}^*$ that it intersects, goes through a point in $T_{\mathcal{R}}$. Thus, there exists a pair of points $s \in T_{\mathcal{R}'}$ and $t \in T_{\mathcal{R}}$, for a region $\mathcal{R}'$ and region $\mathcal{R}$ incident to the top or right side of $\mathcal{R}'$, such that the minimum parameter δ for which there exists a δ-matching from s to t is the Fréchet distance.

Take regions $\mathcal{R}'$ and $\mathcal{R}$, with $\mathcal{R}$ incident to the top or right side of $\mathcal{R}'$. Let $s \in T_{\mathcal{R}'}$ and let $t \in T_{\mathcal{R}}$ be above and to the right of s. Let δ^* be the minimum parameter for which there exists a δ^*-matching from s to t. We construct sets $X_{\mathcal{R}}$ and $Y_{\mathcal{R}}$ such that $\delta^* \in X_{\mathcal{R}} \oplus Y_{\mathcal{R}}$.

If the subcurves corresponding to $\mathcal{R}$ both are line segments, then from Lemma 16 we obtain that t is δ-reachable from s, for some $\delta \geq 0$, if and only if both points lie in $\mathcal{F}_\delta(R, B)$. Let $\delta(t)$ be the minimum value for which a point $t \in T_{\mathcal{R}}$ lies in $\mathcal{F}_{\delta(t)}(R, B)$. We set $X_{\mathcal{R}} = \{\delta(t) \mid t \in T_{\mathcal{R}}\}$. The set $Y_{\mathcal{R}}$ is simply set to $\{0\}$. Taken over all regions in $\mathcal{P}^*$, the sets have a total cardinality of $\mathcal{O}(\mathrm{Cost}(\mathcal{P}^*)) = \mathcal{O}((n + m)\log^2 nm)$.

If one of the subcurves corresponding to $\mathcal{R}$ has more than one edge, then we have access to a pair of doubly-separated x-monotone curves whose δ-free space is identical to the δ-free space of R and B inside $\mathcal{R}$. Moreover, by Lemma 1, we can transform these curves into a pair of curves in $\mathbb{R}$ that are separated by the point 0, such that the δ-free space remains the same. Let $\bar{R}$ and $\bar{B}$ be these curves. Alt and Godau [2] observe that δ^* must be one of the following values:

- the minimum value for which s and t lie in $\mathcal{F}_{\delta^*}(\bar{R}, \bar{B})$, or
- the distance between a vertex of $\bar{R}$ (resp. $\bar{B}$) and an edge of $\bar{B}$ (resp. $\bar{R}$), or
- the distance between two vertices of $\bar{R}$ (resp. $\bar{B}$) to an edge of $\bar{B}$ (resp. $\bar{R}$), if the vertices lie at equal distance to the edge.

Since $\bar{R}$ and $\bar{B}$ are curves in $\mathbb{R}$ that are separated by a point, the candidates for δ^* of second and third type coincide. Moreover, the distance from a vertex p of one curve to an edge e of the other curve is also the distance from p to the endpoint of e closest to p, which must be a vertex. Thus, the latter two types of candidates for δ^* can be summarized as all pairwise distances between vertices of $\bar{R}$ and vertices of $\bar{B}$. Exploiting the separation of $\bar{R}$ and $\bar{B}$ further, the pairwise distances between vertices of $\bar{R}$ and vertices of $\bar{B}$ can be represented by the Cartesian sum of two sets $\bar{X}$ and $\bar{Y}$, containing the absolute values of the vertex values of $\bar{R}$ and $\bar{B}$, respectively. We set $X_{\mathcal{R}} \leftarrow \bar{X}$ and $Y_{\mathcal{R}} \leftarrow \bar{Y}$. Taken over all regions in $\mathcal{P}^*$, the sets have a total cardinality of $\mathcal{O}(\mathrm{Cost}(\mathcal{P}^*)) = \mathcal{O}((n + m)\log^2 nm)$.

Let $X = \bigcup_{\mathcal{R} \in \mathcal{P}^*} X_{\mathcal{R}}$ and $Y = \bigcup_{\mathcal{R} \in \mathcal{P}^*} Y_{\mathcal{R}}$. These sets have a total cardinality of $\mathcal{O}((n + m)\log^2 nm)$, and contain values $x^* \in X$ and $y^* \in Y$ such that $d_{\mathrm{F}}(R, B) = x^* + y^*$. Next we search over $X \oplus Y$ to find the exact value of $d_{\mathrm{F}}(R, B)$.

After sorting X and Y, we can compute the i^{th} smallest value in $X \oplus Y$, for any given i, in $\mathcal{O}(|X| + |Y|) = \mathcal{O}((n+m)\log^2 nm)$ time [10, 13]. There are $|X| \cdot |Y|$ values in $X \oplus Y$. We binary search over the integers $1, \dots, |X| \cdot |Y|$, and at each considered integer i, we compute the i^{th} smallest value δ in $X \oplus Y$. Then we use our decision algorithm to decide whether $d_{\mathrm{F}}(R, B) \leq \delta$ and to guide the search to the value $\delta^* \in X \oplus Y$ with $\delta^* = d_{\mathrm{F}}(R, B)$.

In the above search procedure, each step takes $\mathcal{O}((n + m)\log^3 nm)$ time after preprocessing P and the curves R and B for decision queries, which takes $\mathcal{O}(k \log nm + (n + m)\log^2 nm \log k)$ time (Lemma 17). We perform $\mathcal{O}(\log(|X| \cdot |Y|)) = \mathcal{O}(\log nm)$ steps. Thus we obtain our main result:

▶ **Theorem 18.** *Let P be a polygon with k vertices. Let R and B be two simple, interior disjoint curves on the boundary of P, with n and m vertices. We can compute $d_{\mathrm{F}}(R, B)$ in $\mathcal{O}(k \log nm + (n + m)(\log^2 nm \log k + \log^4 nm))$ time.*

References

1 Helmut Alt and Michael Godau. Measuring the resemblance of polygonal curves. In *Proc. Eighth Annual Symposium on Computational Geometry (SoCG)*, pages 102–109. ACM, 1992. doi:10.1145/142675.142699.

2 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995. doi:10.1142/S0218195995000064.

3 Sang Won Bae and Haitao Wang. L_1 shortest path queries in simple polygons. *Theoretical Computer Science*, 790:105–116, 2019. doi:10.1016/J.TCS.2019.04.023.

4 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 661–670, 2014. doi:10.1109/FOCS.2014.76.

5 Karl Bringmann and Marvin Künnemann. Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds. *International Journal of Computational Geometry & Applications*, 27(1-2):85–120, 2017. doi:10.1142/S0218195917600056.

6 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. SETH says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2887–2901, 2019. doi:10.1137/1.9781611975482.179.

7 Bernard Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 339–349. IEEE Computer Society, 1982. doi:10.1109/SFCS.1982.58.

8 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6:485–524, 1991. doi:10.1007/BF02574703.

9 Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete & Computational Geometry*, 48(1):94–127, 2012. doi:10.1007/s00454-012-9402-z.

10 Greg N. Frederickson and Donald B. Johnson. Generalized selection and ranking: Sorted matrices. *SIAM Journal of Computing*, 13(1):14–30, 1984. doi:10.1137/0213002.

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 Atlas F. Cook IV and Carola Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms*, 7(1):9:1–9:19, 2010. doi:10.1145/1868237.1868247.

13 Andranik Mirzaian and Eshrat Arjomandi. Selection in X+Y and matrices with sorted rows and columns. *Information Processing Letters*, 20(1):13–17, 1985. doi:10.1016/0020-0190(85)90123-1.

14 Thijs van der Horst, Marc J. van Kreveld, Tim Ophelders, and Bettina Speckmann. The geodesic Fréchet distance between two curves bounding a simple polygon. In *Proc. 33rd Annual European Symposium on Algorithms (ESA)*, pages 35:1–35:16, 2025. doi:10.4230/LIPICS.ESA.2025.35.

15 Thijs van der Horst, Marc J. van Kreveld, Tim Ophelders, and Bettina Speckmann. A near-linear time exact algorithm for the L_1-geodesic Fréchet distance between two curves on the boundary of a simple polygon. In *Proc. 19th International Symposium on Algorithms and Data Structures (WADS)*, pages 37:1–37:13, 2025. doi:10.4230/LIPICS.WADS.2025.37.