



Fréchet Distance for Uncertain Curves

KEVIN BUCHIN, TU Dortmund, Germany

CHENGLIN FAN, Sorbonne Université, France

MAARTEN LÖFFLER, Utrecht University, The Netherlands

ALEKSANDR POPOV, TU Eindhoven, The Netherlands

BENJAMIN RAICHEL, University of Texas at Dallas, USA

MARCEL ROELOFFZEN, TU Eindhoven, The Netherlands

In this article, we study a wide range of variants for computing the (discrete and continuous) Fréchet distance between uncertain curves. An uncertain curve is a sequence of *uncertainty regions*, where each region is a disk, a line segment, or a set of points. A *realisation* of a curve is a polyline connecting one point from each region. Given an uncertain curve and a second (certain or uncertain) curve, we seek to compute the lower and upper bound Fréchet distance, which are the minimum and maximum Fréchet distance for any realisations of the curves.

We prove that both problems are NP-hard for the Fréchet distance in several uncertainty models, and that the upper bound problem remains hard for the discrete Fréchet distance. In contrast, the lower bound (discrete [5] and continuous) Fréchet distance can be computed in polynomial time in some models. Furthermore, we show that computing the expected (discrete and continuous) Fréchet distance is #P-hard in some models.

On the positive side, we present an FPTAS in constant dimension for the lower bound problem when Δ/δ is polynomially bounded, where δ is the Fréchet distance and Δ bounds the diameter of the regions. We also show a near-linear-time 3-approximation for the decision problem on roughly δ -separated convex regions. Finally, we study the setting with Sakoe–Chiba time bands, where we restrict the alignment between the curves, and give polynomial-time algorithms for the upper bound and expected discrete and continuous Fréchet distance for uncertainty modelled as point sets.

CCS Concepts: • **Theory of computation** → **Computational geometry**;

Additional Key Words and Phrases: Curves, uncertainty, Fréchet distance, hardness

K. Buchin, C. Fan, M. Löffler, A. Popov, B. Raichel, and M. Roeloffzen contributed equally to this research.

B. Raichel was partially supported by the National Science Foundation (NSF) CAREER Award 1750780. M. Löffler was partially supported by the Dutch Research Council (NWO) under project numbers 614.001.504 and 628.011.005. A. Popov was supported by the Dutch Research Council (NWO) under project number 612.001.801. M. Roeloffzen was supported by the Dutch Research Council (NWO) under project number 628.011.005.

Authors' addresses: K. Buchin, Department of Computer Science, TU Dortmund, Otto-Hahn-Straße 14, 44227, Dortmund, Germany; email: kevin.buchin@tu-dortmund.de; C. Fan, Computer Lab of Paris 6 (Lip6), Sorbonne Université, 4 Pl. Jussieu, Paris, 75005, France; email: fanchenglin@gmail.com; M. Löffler, Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, 3584 CC, Utrecht, The Netherlands; email: m.loffler@uu.nl; A. Popov and M. Roeloffzen, Department of Mathematics and Computer Science, TU Eindhoven, Groene Loper 5, 5612 AE, Eindhoven, The Netherlands; emails: {a.popov, m.j.m.roeloffzen}@tue.nl; B. Raichel, Department of Computer Science, University of Texas at Dallas, 800 West Campbell Road; MS EC31, Richardson, TX 75080; email: benjamin.raichel@utdallas.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).

1549-6325/2023/07-ART29

<https://doi.org/10.1145/3597640>

ACM Reference format:

Kevin Buchin, Chenglin Fan, Maarten Löffler, Aleksandr Popov, Benjamin Raichel, and Marcel Roeloffzen. 2023. Fréchet Distance for Uncertain Curves. *ACM Trans. Algor.* 19, 3, Article 29 (July 2023), 47 pages. <https://doi.org/10.1145/3597640>

1 INTRODUCTION

In this article, we investigate the well-studied topic of curve similarity in the context of the burgeoning area of geometric computing under uncertainty. Classical algorithms in computational geometry typically assume the input point locations are known exactly; however, in recent years, there has been a concentrated effort to adapt these algorithms to uncertain inputs, which can more faithfully model real-world inputs. The need to model such uncertain inputs is perhaps no more clear than for the location data of a moving object obtained from physical devices, which is inherently imprecise due to issues such as measurement error, sampling error, and network latency [49, 52]. Moreover, to ensure location privacy, one may purposely add uncertainty to the data by adding noise or reporting positions as geometric regions rather than points. (See the survey by Krumm [42] and the references therein.)

Here we consider both the continuous and discrete Fréchet distance for uncertain curves. Given the preceding applications, our uncertain input is given as a sequence of compact regions, from which a polygonal curve is *realised* by selecting one point from each region. Our goal is to find, for a given pair of uncertain curves, the upper bound, the lower bound, and the expected Fréchet distance, where the upper (respectively, lower) bound Fréchet distance is the maximum (respectively, minimum) distance over any realisation. For the expected Fréchet distance, we assume a probability distribution is provided that describes how each vertex on a curve is chosen from the compact region.

1.1 Previous Work

Geometric Computing Under Uncertainty. The two most common models of geometric uncertainty are the locational model [43] and the existential model [53, 57]. In the existential model, the location of an uncertain point is known, but the point may not be present; in the locational model, we know that each uncertain point exists, but not its exact location.

In this article, we consider the locational model. Each uncertain point is a set of potential locations. We call an uncertain point *indecisive* if the set of potential locations is finite, or *imprecise* if the set is not finite but is a convex region. A *realisation* of a set of uncertain points is a selection of one point from each uncertain point. The goal is typically to compute the realisation of a set of uncertain points that minimises or maximises some quantity (e.g., area, distance, perimeter) of some underlying geometric structure (e.g., convex hull, MST). A large number of minimisation and maximisation variants for imprecise points can be found in the thesis of Löffler [43] and other works [41, 44, 46]. For indecisive points, such problems are often called *colour-spanning problems*, as each indecisive point can be viewed as a colour and the goal is to select a point of each colour to minimise or maximise some quantity [1, 7, 23, 30]. Besides finding tight upper and lower bounds for various measures, there have also been studies on visibility [22], imprecise terrains [27, 35], and Voronoi diagrams [51] and Delaunay triangulations [15, 45, 55].

By assigning a probability distribution to uncertain points, one can also consider the expectation or distribution of various measures [2, 4, 21, 39, 48]. Finally, imprecision has also been studied from a movement perspective, with the focus on the imprecision between measurements [19] and how imprecision grows and shrinks as time passes and new location information becomes available [29].

Fréchet Distance. Computing the Fréchet distance between two precise curves can be done in near-quadratic time [3, 6, 12], and assuming the strong exponential time hypothesis, it cannot be computed or even approximated well in strongly subquadratic time [9, 18]. However, for several restricted versions, the Fréchet distance can be calculated faster, for example, for c -packed curves [26], when the edges are long [36], or when the alignment of curves is restricted [11, 47]. Many variants of the problem have been considered: Fréchet distance with shortcuts [20, 25], weak Fréchet distance [6], discrete Fréchet distance [3, 28], Fréchet gap distance [31], Fréchet distance under translations [10, 33], and more.

There are also numerous applications of different variants of Fréchet distance in common curve and trajectory analysis tasks, such as clustering [13, 14] or curve simplification [54, 56].

Fréchet Distance Under Uncertainty. There has been surprisingly little work incorporating uncertainty in curve and trajectory analysis. Buchin and Sijben [21] have studied the discrete Fréchet distance for uncertain points modelled by a probability distribution. However, their problem is quite different from our variant: they show how to compute the distance distribution for a fixed coupling between the two curves and then solve the problem of finding the optimal coupling that achieves a given Fréchet distance. We look at the problem with the different order of quantifiers: we know how to compute the Fréchet distance between two curves and want to find ‘optimal’ realisations yielding a certain distance.

Previously, Ahn et al. [5] considered the lower bound problem as we define it for the discrete Fréchet distance, giving a polynomial-time algorithm for uncertain points modelled by balls or hyperrectangles in constant dimension. The authors also gave efficient approximation algorithms for the discrete upper bound Fréchet distance for uncertain inputs, where the approximation factor depends on the spread of the region diameters or how well separated they are. Subsequently, Fan and Zhu [32] showed that the discrete upper bound Fréchet distance is NP-hard for uncertain inputs modelled as thin rectangles. To our knowledge, we are the first to consider either variant for the continuous Fréchet case, and the first to consider the expected Fréchet distance.

Subsequently to this work, Buchin et al. [16] have studied the lower and upper bound problems in 1D, as well as obtained some results for the weak Fréchet distance. They have obtained stronger results showing NP-hardness of the upper bound problem for indecisive points and points modelled with intervals (or line segments), both for the discrete and the continuous Fréchet distance. For the lower bound problem, they show a polynomial-time algorithm when uncertainty is modelled with intervals—essentially, a setting where we prove NP-hardness in 2D—thus delineating the point where the problems become difficult. Finally, Buchin et al. [17] also studied the related problem of curve simplification under uncertainty, showing how to obtain a minimal-length subsequence of an uncertain curve that has a small distance to the original curve, no matter the realisation.

1.2 Our Contributions

In this article, we present an extensive study of the Fréchet distance for uncertain curves. We provide a wide range of hardness results and present several approximations and polynomial-time solutions to restricted versions. We are the first to consider the continuous Fréchet distance in the uncertain setting, as well as the first to consider the expected Fréchet distance.

On the negative side, we present a plethora of hardness results (Table 1; details follow in Section 3). The hardness of the lower bound case is curious: although the discrete Fréchet distance on imprecise inputs [5] and, as we prove, continuous Fréchet distance on indecisive inputs both permit a simple dynamic programming solution, the continuous Fréchet distance problem on imprecise input has just enough (literal) wiggle room to show NP-hardness by reduction from SUBSETSUM. Buchin et al. [16] explore this in 1D and find a similar dichotomy for the weak Fréchet distance.

Table 1. Hardness Results for the Decision Problems in This Article (in 2D)

		Indecisive	Imprecise	
			Disks	Line Segments
Discrete Fréchet distance	LB	Polynomial [5]	Polynomial [5]	Polynomial [5]
	UB	NP-complete	NP-hard	NP-hard
	Exp	#P-hard	—	#P-hard
Fréchet distance	LB	Polynomial	—	NP-hard
	UB	NP-complete	NP-hard	NP-hard
	Exp	#P-hard	—	—

Ahn et al. [5] solve the lower bound problem for disks, but their algorithm extends to the indecisive curves as well as line segment imprecision.

We complement the lower bound hardness result by two approximation algorithms (Section 4). The first is an FPTAS for general uncertain curves in constant dimension when the ratio between the diameter of the uncertain points and the lower bound Fréchet distance is polynomially bounded. The second is a 3-approximation for separated imprecise curves but uses a simpler greedy approach that runs in near-linear time.

The NP-hardness of the upper bound by a reduction from CNF-SAT is less surprising but requires a careful setup and analysis of the geometry to then extend it to a reduction from #CNF-SAT to the expected (discrete or continuous) Fréchet distance under the uniform distribution. However, by adding the common constraint that the alignment between the curves needs to stay within a Sakoe–Chiba [50] band of constant width (see Section 5 for definition and results), we can solve these problems in polynomial time for indecisive curves. Sakoe–Chiba bands are frequently used for time-series data [8, 40, 50] and trajectories [11, 24], when the alignment should (or is expected to) not vary too much from a certain ‘natural’ alignment.

2 PRELIMINARIES

In this section, we introduce the notation relevant to the rest of this article, as well as recall the definitions of the (discrete) Fréchet distance.

2.1 Curves

Denote $[n] \equiv \{1, 2, \dots, n\}$. Consider a *sequence* of d -dimensional points $\pi = \langle p_1, p_2, \dots, p_n \rangle$. A *polygonal curve* π is defined by these points by linearly interpolating between the successive points and can be seen as a continuous function: $\pi(i + \alpha) = (1 - \alpha)p_i + \alpha p_{i+1}$ for $i \in [n - 1]$ and $\alpha \in [0, 1]$. The *length* of such a curve is the length of the sequence, $|\pi| = n$. Where we deem important to distinguish between points that are a part of the curve and other points, we denote the polygonal curve by $\pi = \langle \pi_1, \pi_2, \dots, \pi_n \rangle$. We denote the *concatenation* of two sequences π and σ by $\pi \sqcup \sigma$; this also naturally defines concatenation of polygonal curves. We denote a *subsequence* from vertex i to j of π as $\pi[i : j] \equiv \langle p_i, p_{i+1}, \dots, p_j \rangle$. Finally, $p \sqcup q$ (or simply pq) denotes the line segment between points p and q . We can generalise this notation:

$$\bigsqcup_{i \in [n]} p_i \equiv p_1 \sqcup p_2 \sqcup \dots \sqcup p_n \equiv \langle p_1, p_2, \dots, p_n \rangle \equiv \pi.$$

2.2 Metrics Definitions

Given two points $x, y \in \mathbb{R}^d$, denote their Euclidean distance by $\|x - y\|$. For two compact sets $X, Y \subset \mathbb{R}^d$, denote their distance by $\|X - Y\| = \min_{x \in X, y \in Y} \|x - y\|$. Throughout the article, we treat the dimension d as a small constant.

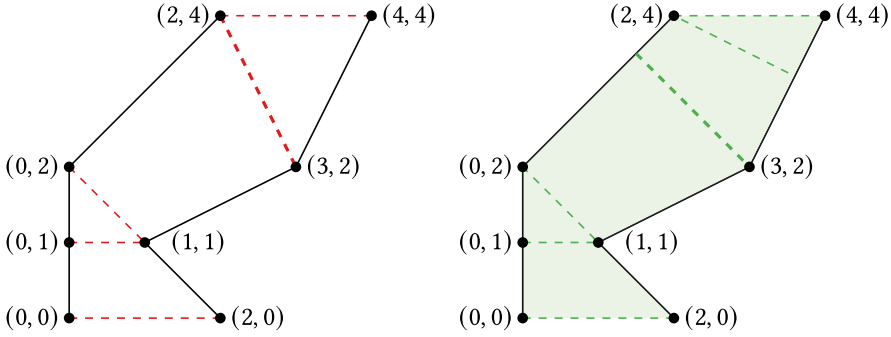


Fig. 1. Left: Discrete Fréchet distance, where an optimal coupling is shown in dashed red lines. Right: Fréchet distance, where the dashed green lines indicate specific values for an optimal alignment ϕ_1, ϕ_2 .

Let Φ_n denote the set of all *reparametrisations* of length n , defined as continuous non-decreasing functions $\phi : [0, 1] \rightarrow [1, n]$, where $\phi(0) = 1$ and $\phi(1) = n$. Given a pair of curves π and σ of lengths n and m , respectively, and corresponding reparametrisations $\phi_1 \in \Phi_n$ and $\phi_2 \in \Phi_m$, define $\text{width}_{\phi_1, \phi_2}(\pi, \sigma) = \max_{t \in [0, 1]} \|\pi(\phi_1(t)) - \sigma(\phi_2(t))\|$. We call the pair (ϕ_1, ϕ_2) an *alignment*.

The width represents the maximum distance between two points traversing the curves from start to end according to ϕ_1 and ϕ_2 (which allow varying speed, but no backtracking). The *Fréchet distance* $d_F(\pi, \sigma)$ is defined as the minimum possible width over all such traversals:

$$d_F(\pi, \sigma) = \inf_{\phi_1 \in \Phi_n, \phi_2 \in \Phi_m} \text{width}(\pi, \sigma) = \inf_{\phi_1 \in \Phi_n, \phi_2 \in \Phi_m} \max_{t \in [0, 1]} \|\pi(\phi_1(t)) - \sigma(\phi_2(t))\|.$$

The *discrete Fréchet distance* $d_{dF}(\pi, \sigma)$ is defined similarly, except that we do not traverse edges of the curves but must jump from one vertex to the next on either or both curves. We define a valid *coupling* as a sequence $c = \langle (p_1, q_1), \dots, (p_r, q_r) \rangle$ of pairs from $[n] \times [m]$, where $(p_1, q_1) = (1, 1)$, $(p_r, q_r) = (n, m)$, and, for any $i \in [r - 1]$, we have $(p_{i+1}, q_{i+1}) \in \{(p_i + 1, q_i), (p_i, q_i + 1), (p_i + 1, q_i + 1)\}$. Let C be the set of all valid couplings on curves of lengths n and m ; then

$$d_{dF}(\pi, \sigma) = \inf_{c \in C} \max_{s \in [c]} \|\pi(p_s) - \sigma(q_s)\|,$$

where $c_s = (p_s, q_s)$ for all $s \in [c]$. Both distances are illustrated in Figure 1.

Computing the Discrete Fréchet Distance. We recall the standard dynamic programming approach by Eiter and Mannila [28]. The algorithm is deduced in a standard manner from the following recursion:

$$\begin{aligned} d_{dF}(\pi[1 : i + 1], \sigma[1 : j + 1]) = & \max(\|\pi(i + 1) - \sigma(j + 1)\|, \\ & \min(d_{dF}(\pi[1 : i], \sigma[1 : j]), \\ & d_{dF}(\pi[1 : i + 1], \sigma[1 : j]), \\ & d_{dF}(\pi[1 : i], \sigma[1 : j + 1]))). \end{aligned}$$

In other words, the discrete Fréchet distance is the maximum of the distance of the newly added element in the coupling and the value that was considered best previously. Due to the coupling restrictions, there are only three possible subproblems that we need to consider, and we may choose the best of them, thus obtaining the preceding recursion. It is straightforward to turn it into a dynamic program.

Table 2 gives the distance matrix and the computation of the discrete Fréchet distance for the example of Figure 1. Each cell of the table on the right shows the value of the discrete Fréchet distance so far; the final result can be read out from the top right corner of the table, and the

Table 2. Distance Matrix and Computation of discrete Fréchet Distance for the Example of Figure 1

4	$\sqrt{10}$	$\sqrt{5}$	2
$2\sqrt{2}$	$\sqrt{2}$	3	$2\sqrt{5}$
$\sqrt{5}$	1	$\sqrt{10}$	5
2	$\sqrt{2}$	$\sqrt{13}$	$4\sqrt{2}$

4	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{5}$
$2\sqrt{2}$	2	3	$2\sqrt{5}$
$\sqrt{5}$	2	$\sqrt{10}$	5
2	2	$\sqrt{13}$	$4\sqrt{2}$

Left: Distance matrix on vertices. Right: Dynamic program for the discrete Fréchet distance, filled from the bottom left corner. Rows correspond to points from the left trajectory and columns to points from the right trajectory. The optimal path is marked in grey.

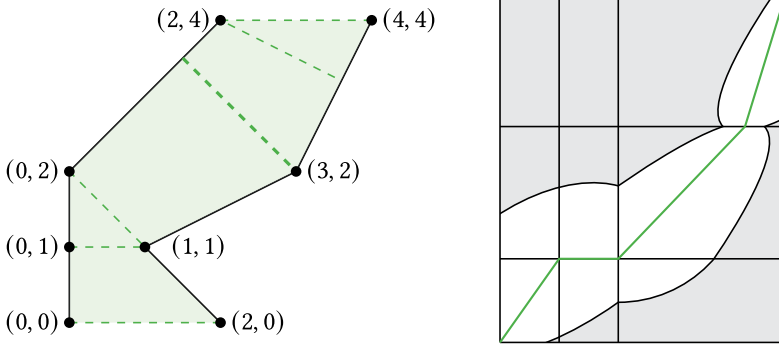


Fig. 2. Left: Visualisation of the Fréchet distance. Right: Free-space diagram for the threshold $\varepsilon = 2.15$. One can draw a monotonous **path** (in green) from the lower left corner to the upper right corner of the diagram, so the Fréchet distance between the trajectories is below the threshold.

coupling that yields this result can be read from the sequence of grey cells. Notice that the table shows the same coupling as Figure 1.

Given two trajectories of length n and m in two dimensions, this approach takes $\Theta(mn)$ time to run. More recently, Agarwal et al. [3] presented an algorithm that computes the discrete Fréchet distance in time $O(mn \log \log n / \log n)$ in two dimensions, for $m \leq n$. However, it is rather complex and does not help the intuition about the problems discussed in this article, so we will not go into further detail. The decision version of the problem can be solved in a similar fashion, but propagating Boolean values instead.

Computing the Fréchet Distance. One can use a similar approach to solve the decision version of the Fréchet distance problem, except now we have free and blocked areas within each cell of the table rather than simply having a Boolean value in each cell. The resulting table is called a *free-space diagram*. On polygonal curves, each cell becomes an intersection of an ellipse with the cell, with the inside of the ellipse being free. The answer to the problem is TRUE if and only if there is a monotone path from the bottom left corner to the top right corner of the free-space diagram. A free-space diagram for the example of the two polygonal curves of Figure 1 is shown in Figure 2.

Algorithmically, this can be checked by keeping the open intervals on the edges of the cells (i.e., the white segments on cell borders shown in Figure 2). The algorithm then runs in time $\Theta(mn)$. For further details, the reader is invited to consult the work by Alt and Godau [6] or previous work on the same topic [34].

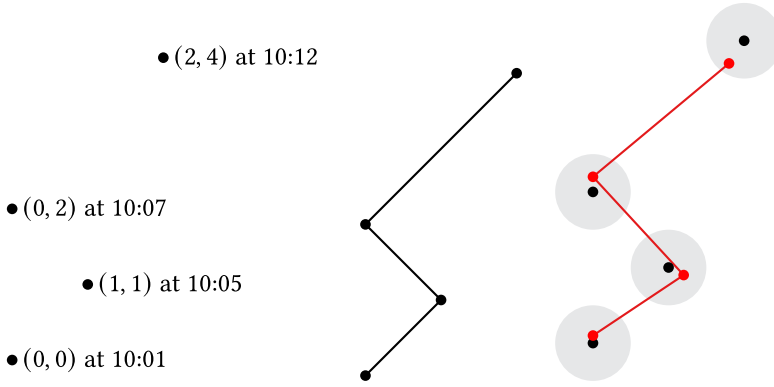


Fig. 3. Left: Trajectory data. Centre: Polygonal curve on the data. Right: Imprecise curve with disks as imprecision regions and the real curve.

2.3 Uncertainty Model

An *uncertain* point is commonly represented as a compact region $U \subset \mathbb{R}^d$. Usually, it is a finite set of points, a disk, a rectangle, or a line segment. The intuition is that only one point from this region represents the true location of the point; however, we do not know which one. A *realisation* p of such a point is one of the points from the region U . When needed, we assume the realisations are drawn from U according to a known probability distribution $\mathbb{P}$. We denote the diameter of any compact set (e.g., an uncertain point) $U \subset \mathbb{R}^d$ by $\text{diam}(U) = \max_{p, q \in U} \|p - q\|$. An *indecisive* point is a special case of an uncertain point: it is a set of points $U = \{p^1, \dots, p^k\}$, with each point $p^i \in \mathbb{R}^d$ for $i \in [k]$. Similarly, an *imprecise* point is a compact convex region $U \subset \mathbb{R}^d$. We will often use disks or line segments as such regions. Note that a precise point is a special case of an indecisive point (set of size one) and an imprecise point (disk of radius zero).

2.4 Uncertain Curves and Distances

Define an *uncertain curve* as a sequence of uncertain points $\mathcal{U} = \langle U_1, \dots, U_n \rangle$. A *realisation* $\pi \in \mathcal{U}$ of an uncertain curve is a polygonal curve $\pi = \langle p_1, \dots, p_n \rangle$, where each p_i is a realisation of the corresponding uncertain point U_i . We denote the set of all realisations of an uncertain curve $\mathcal{U}$ by $\text{Real}(\mathcal{U})$ (Figure 3). In a probabilistic setting, we write $\pi \in_{\mathbb{P}} \mathcal{U}$ to denote that each point of π gets drawn from the corresponding uncertainty region independently according to distribution $\mathbb{P}$.

For uncertain curves $\mathcal{U}, \mathcal{V}$, define the upper bound, the lower bound, and the expected discrete Fréchet distance (and extend to the continuous Fréchet distance $d_F^{\max}, d_F^{\min}, d_F^{\mathbb{E}(\mathbb{P})}$ using d_F) as follows:

$$\begin{aligned} d_{\text{dF}}^{\max}(\mathcal{U}, \mathcal{V}) &= \max_{\pi \in \mathcal{U}, \sigma \in \mathcal{V}} d_{\text{dF}}(\pi, \sigma), & d_F^{\max}(\mathcal{U}, \mathcal{V}) &= \max_{\pi \in \mathcal{U}, \sigma \in \mathcal{V}} d_F(\pi, \sigma), \\ d_{\text{dF}}^{\min}(\mathcal{U}, \mathcal{V}) &= \min_{\pi \in \mathcal{U}, \sigma \in \mathcal{V}} d_{\text{dF}}(\pi, \sigma), & d_F^{\min}(\mathcal{U}, \mathcal{V}) &= \min_{\pi \in \mathcal{U}, \sigma \in \mathcal{V}} d_F(\pi, \sigma), \\ d_{\text{dF}}^{\mathbb{E}(\mathbb{P})}(\mathcal{U}, \mathcal{V}) &= \mathbb{E}_{\pi \in_{\mathbb{P}} \mathcal{U}, \sigma \in_{\mathbb{P}} \mathcal{V}} [d_{\text{dF}}(\pi, \sigma)], & d_F^{\mathbb{E}(\mathbb{P})}(\mathcal{U}, \mathcal{V}) &= \mathbb{E}_{\pi \in_{\mathbb{P}} \mathcal{U}, \sigma \in_{\mathbb{P}} \mathcal{V}} [d_F(\pi, \sigma)]. \end{aligned}$$

If the distribution is clear from the context, we write $d_F^{\mathbb{E}}$ and $d_{\text{dF}}^{\mathbb{E}}$. The preceding definitions also apply if one of the curves is precise, as a precise curve is a special case of an uncertain curve.

3 HARDNESS RESULTS

In this section, we first discuss the hardness results for the upper bound and the expected value of the continuous and discrete Fréchet distance for indecisive and imprecise curves. We then demonstrate hardness of finding the lower bound continuous Fréchet distance on imprecise curves.

3.1 Upper Bound and Expected Fréchet Distance

We present proofs of NP-hardness and #P-hardness for the upper bound and the expected Fréchet distance for both indecisive and imprecise curves by showing polynomial-time reductions from CNF-SAT (satisfiability of a Boolean formula) and #CNF-SAT (its counting version). We consider the upper bound problem for indecisive curves and then illustrate how the construction can be used to show #P-hardness for the expected Fréchet distance (both discrete and continuous). We then illustrate how the construction can be adapted to show hardness for imprecise curves. All our constructions are in two dimensions.

3.1.1 Upper Bound Fréchet Distance: Basic Construction. Define the following problem.

PROBLEM 3.1 (UPPER BOUND DISCRETE FRÉCHET). *Given two uncertain curves $\mathcal{U}$ and $\mathcal{V}$ and a threshold $\delta \in \mathbb{R}^+$, decide if $d_{\text{df}}^{\max}(\mathcal{U}, \mathcal{V}) > \delta$.*

We can similarly define its continuous counterpart, using $d_{\text{F}}^{\max}$ instead.

PROBLEM 3.2 (UPPER BOUND CONTINUOUS FRÉCHET). *Given two uncertain curves $\mathcal{U}$ and $\mathcal{V}$ and a threshold $\delta \in \mathbb{R}^+$, decide if $d_{\text{F}}^{\max}(\mathcal{U}, \mathcal{V}) > \delta$.*

We first give some extra definitions to make the proofs clearer. Suppose we are given a CNF-SAT formula C with

$$C = \bigwedge_{i \in [n]} C_i, \quad C_i = \bigvee_{j \in J \subseteq [m]} x_j \vee \bigvee_{k \in K \subseteq [m] \setminus J} \neg x_k \quad \text{for all } i \in [n].$$

Here, n and m are the number of clauses and variables, respectively, and x_j for any $j \in [m]$ is a Boolean variable. Such a variable may be assigned ‘true’ or ‘false’; an *assignment* is a function $a : \{x_1, \dots, x_m\} \rightarrow \{\text{TRUE}, \text{FALSE}\}$ that assigns a value to each variable, $a(x_j) = \text{TRUE}$ or $a(x_j) = \text{FALSE}$ for any $j \in [m]$. We denote by $C[a]$ the result of substituting $x_j \mapsto a(x_j)$ in C for all $j \in [m]$. As an aid to the reader, the problem we reduce from is as follows.

PROBLEM 3.3 (CNF-SAT). *Given a CNF-SAT formula C , decide if there is an assignment a such that $C[a] = \text{TRUE}$.*

We pick some value $0 < \varepsilon < 0.25$.¹ Construct a *variable curve*, where each variable corresponds to an indecisive point with locations $(0, 0.5 + \varepsilon)$ and $(0, -0.5 - \varepsilon)$; the locations are interpreted as assigning the variable TRUE and FALSE. Any realisation of the curve corresponds to a variable assignment.

Intuitively, one curve encodes the variables, and the other encodes the structure of the formula. We define a variable gadget on a variable curve to encode the value of a Boolean variable, and we define assignment gadgets on the other curve to encode the literals x and $\neg x$ occurring in the formula. The gadgets interact with each other, so if a literal is true, the distance is large. The assignment gadgets have positions for ‘true’, ‘false’, and ‘do not care’ values, the latter being used to skip a variable unused in a clause. We repeat the construction for each variable on both curves with some synchronisation enforcement, constructing a variable clause gadget and an assignment

¹This range is determined by the relative distances in the construction.

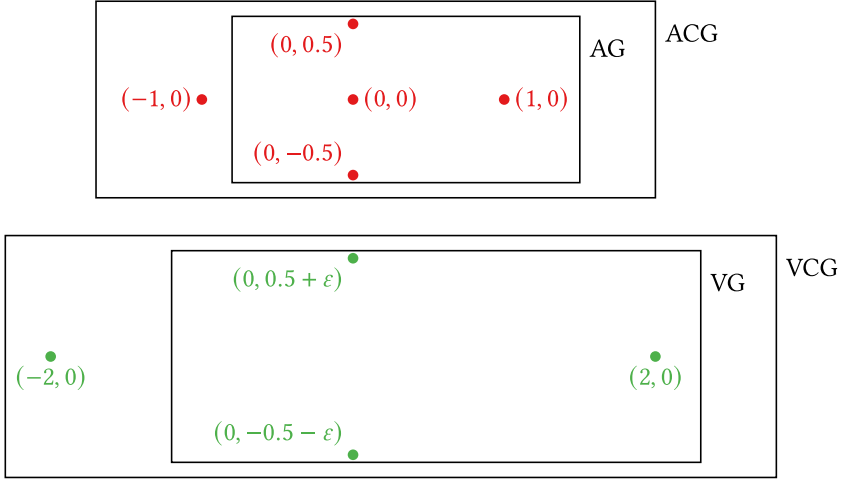


Fig. 4. Illustration of the gadgets used in the basic construction. Assignment gadgets are repeated to make up assignment clause gadgets; they are repeated to make up the clause curve. Variable gadgets are repeated to make up the variable clause gadget; it is prepended and appended by $(0, 0)$ to make up the variable curve.

clause gadget, so the distance is large if the clause is satisfied by setting the variables in a specific way. Finally, we construct the full variable curve and the clause curve. Here the goal is that we have a single copy of variables that can be assigned TRUE or FALSE, and we can choose which clause we want to align with them. The other clauses are caught by extra points on the variable curve so as to not affect the distance. Some clauses are not satisfied and will yield a small distance, whereas others are satisfied and yield a large distance; therefore, since we can choose the clause freely, we only get large distance between full curves if *all* clauses give a large distance, so all are satisfied, and so is the formula. Finding the upper bound Fréchet distance now corresponds to finding the realisation of the points that achieves the large distance, or finding the truth assignment of the variables that satisfies the formula. We show the locations used by the gadgets and their nesting in Figure 4. We show an example construction for a specific formula and a realisation in Figure 5, showing also the possible alignment options between the clause curve and the variable curve and the resulting distances. Next, we define the gadgets formally level by level and prove that the distances are correct.

Literal Level. Define a *variable gadget*, where an indecisive point corresponds to a variable and is followed by a precise point far away, to force synchronisation with the other curve:

$$VG_j = \{(0, 0.5 + \varepsilon), (0, -0.5 - \varepsilon)\} \sqcup (2, 0).$$

Consider a specific clause C_i of the formula. We define an *assignment gadget* $AG_{i,j}$ for each variable x_j and clause C_i depending on how the variable occurs in the clause:

$$AG_{i,j} = \begin{cases} (0, -0.5) \sqcup (1, 0) & \text{if } x_j \text{ is a literal of } C_i, \\ (0, 0.5) \sqcup (1, 0) & \text{if } \neg x_j \text{ is a literal of } C_i, \\ (0, 0) \sqcup (1, 0) & \text{otherwise.} \end{cases}$$

Note that if the assignment $x_j = \text{TRUE}$ makes the clause C_i true, then the first precise point of the corresponding assignment gadget appears at distance $1 + \varepsilon$ from the realisation corresponding to setting $x_j = \text{TRUE}$ of the indecisive point in VG_j .

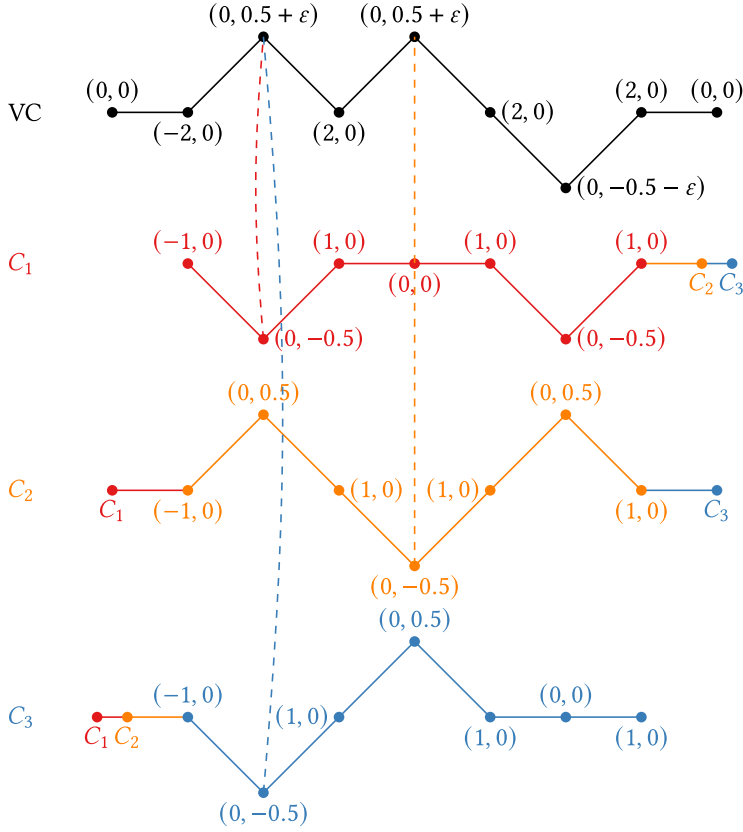


Fig. 5. Realisation of VC for the assignment $x_1 = \text{TRUE}$, $x_2 = \text{TRUE}$, $x_3 = \text{FALSE}$ and CC for the formula $C = (x_1 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2)$. We show the variable curve, and three times the clause curve, since we have three feasible options for matching the curves, corresponding to the three clauses. The other clauses are matched to $(0, 0)$ and are collapsed to a point in the figure. Note that $C = \text{TRUE}$ with the given variable assignment. Also note that we can choose any of C_1 , C_2 , C_3 to couple to VC; we always get the bottleneck distance of $1 + \varepsilon$, as all three are satisfied, so here $d_{\text{df}}(\text{VC}, \text{CC}) = 1 + \varepsilon$.

We now show the relation between the gadgets. To do so, we introduce the *one-to-one coupling* as a valid coupling $c = \langle (p_1, q_1), \dots, (p_r, q_r) \rangle$, where the coupling is restricted to $(p_{s+1}, q_{s+1}) = (p_s + 1, q_s + 1)$ for all $s \in [r - 1]$. Necessarily, such a coupling only exists for curves of equal length.

LEMMA 3.4. *Suppose we are given a clause C_i and a variable x_j that both occur in a CNF-SAT formula C , and we restrict the set of valid couplings C to only contain one-to-one couplings. We only get the discrete Fréchet distance equal to $1 + \varepsilon$ if the realisation of VG_j we pick corresponds to the assignment of x_j that ensures the clause C_i is satisfied; otherwise, the discrete Fréchet distance is 1. In other words, if we consider $\pi \in \text{VG}_j$ that corresponds to setting $a(x_j)$, then*

$$d_{\text{df}}(\pi, \text{AG}_{i,j}) = \begin{cases} 1 + \varepsilon & \text{if assigning } x_j \text{ satisfies } C_i, \\ 1 & \text{otherwise.} \end{cases}$$

PROOF. First of all, observe that as we only consider one-to-one couplings, the second points of both gadgets must be coupled; the distance between them is $\|(2, 0) - (1, 0)\| = 1$. Thus, the discrete Fréchet distance between the curves must be at least 1.

Now consider the possible realisations of VG_j . Say that we pick the realisation $(0, 0.5 + \varepsilon) \sqcup (2, 0)$, which corresponds to assigning $a(x_j) = \text{TRUE}$. If x_j is a literal of C_i , so $C_i[a] = \text{TRUE}$, then by construction we know that $AG_{i,j}$ is $(0, -0.5) \sqcup (1, 0)$. Since we consider only the one-to-one couplings, we must couple the first points together, yielding the distance $\|(0, 0.5 + \varepsilon) - (0, -0.5)\| = 1 + \varepsilon > 1$, so the discrete Fréchet distance in this case is $1 + \varepsilon$, and indeed we picked the assignment that ensures that C_i is satisfied. If instead $\neg x_j$ is a literal of C_i , so $C_i[a] = \text{FALSE}$, then we know that $AG_{i,j}$ is $(0, 0.5) \sqcup (1, 0)$, and it is easy to see that, as $\|(0, 0.5 + \varepsilon) - (0, 0.5)\| = \varepsilon < 1$, we get the discrete Fréchet distance of 1, and that we picked an assignment that does not ensure that C_i is satisfied.

A symmetric argument can be applied when we consider the realisation $(0, -0.5 - \varepsilon) \sqcup (2, 0)$ for VG_j : if $\neg x_j$ is a literal of C_i , then we get the discrete Fréchet distance of $1 + \varepsilon$ and we picked an assignment that surely satisfies C_i .

Finally, consider the case when $AG_{i,j} = (0, 0) \sqcup (1, 0)$. This implies that assigning a value to x_j has no effect on C_i (i.e., a literal involving x_j does not occur in C_i), so neither assignment (and neither realisation of VG_j) would ensure that C_i is satisfied. Also observe that $\|(0, 0.5 + \varepsilon) - (0, 0)\| = \|(0, -0.5 - \varepsilon) - (0, 0)\| = 0.5 + \varepsilon < 1$, so both realisations yield the discrete Fréchet distance of 1.

So, we can conclude that we get the distance $1 + \varepsilon$ if and only if the partial assignment of a value to x_j ensures that C_i is satisfied; otherwise, we get the distance 1. $\square$

Clause Level. We can repeat the construction, yielding a *variable clause gadget* and an *assignment clause gadget*:

$$VCG = (-2, 0) \sqcup \bigsqcup_{j \in [m]} VG_j, \quad ACG_i = (-1, 0) \sqcup \bigsqcup_{j \in [m]} AG_{i,j}.$$

Consider the Fréchet distance between the two gadgets. Observe that coupling a synchronisation point from one gadget with a non-synchronisation point in the other yields a distance larger than $1 + \varepsilon$, whereas coupling synchronisation points pairwise and non-synchronisation points pairwise will yield the distance at most $1 + \varepsilon$. So, we only consider one-to-one couplings—that is, we couple point i on one curve to point i on the other curve, for all i .

Now, if a realisation corresponds to a satisfying assignment, then for some x_j we have picked the realisation that is opposite from the coupled point on the clause curve, yielding the bottleneck distance of $1 + \varepsilon$. If the realisation corresponds to a non-satisfying assignment, then the synchronisation points establish the bottleneck, yielding the distance 1. So, we can clearly distinguish between a satisfying and a non-satisfying assignment for a clause. It is crucial now that we show the following.

LEMMA 3.5. *Given a CNF-SAT formula C containing some clause C_i and m variables $x_1, \dots, x_m$, consider curves $\alpha_1 \sqcup VCG \sqcup \alpha'_1$ and $\alpha_2 \sqcup ACG_i \sqcup \alpha'_2$ for arbitrary precise curves $\alpha_1, \alpha'_1, \alpha_2, \alpha'_2$ with $|\alpha_1| = k$ and $|\alpha_2| = l$. If an optimal coupling between $\alpha_1 \sqcup VCG \sqcup \alpha'_1$ and $\alpha_2 \sqcup ACG_i \sqcup \alpha'_2$ for any realisation of VCG has a pair $(k + 1, l + 1)$, then there is an optimal coupling that has pairs $(k + s, l + s)$ for all $s \in [2m + 1]$ —that is, there is an optimal coupling that is one-to-one for any realisation of VCG .*

PROOF. Observe that both gadgets have exactly $2m + 1$ points. Suppose the optimal coupling Opt has a pair $(k + 1, l + 1)$, so it couples the first points of VCG and ACG_i . If Opt is already one-to-one for all $s \in [2m + 1]$, there is nothing to be done. Suppose now that it is one-to-one until some $1 \leq r < 2m + 1$, so it has pairs $(k + s, l + s)$ for all $s \in [r]$, but it does not have a pair $(k + (r + 1), l + (r + 1))$. Then one of the following cases occurs:

- $r = 2q + 2$ is even; then we know that the point $(2, 0)$ in VG_{q+1} is not coupled to the point $(1, 0)$ in $\text{AG}_{i,q+1}$, but the preceding indecisive point is coupled to the assignment point. Then either $(2, 0)$ is coupled to an assignment point, with the distance at least 2, or $(1, 0)$ is coupled to an indecisive point, yielding the distance of $\sqrt{1 + (0.5 + \varepsilon)^2} > 1$. If we eliminate that pair and instead couple $(2, 0)$ to $(1, 0)$, we will still have a valid coupling and obtain the distance of 1 on this pair; thus, the new coupling is not worse than the original one, and so it is also an optimal coupling that is one-to-one for all $s \in [r + 1]$.
- $r = 2q + 1$ is odd; then we know that the indecisive point in VG_{q+1} is not coupled to the assignment point in $\text{AG}_{i,q+1}$, but the preceding $(2, 0)$ and $(1, 0)$ (or $(-2, 0)$ and $(-1, 0)$) are coupled. Then either Opt has a pair of the indecisive point and $(1, 0)$, or it has a pair of the assignment point and $(2, 0)$. (The cases for $(-1, 0)$ and $(-2, 0)$ are symmetrical.) In either case, we want to eliminate that pair from the coupling and instead add the pair of the indecisive point and the assignment point, yielding a valid coupling that is one-to-one for all $s \in [r + 1]$. To complete the proof for this case, we need to show that such a coupling is optimal.

Consider the first possible coupling. The distance between the indecisive point and $(1, 0)$ is $\sqrt{1 + (0.5 + \varepsilon)^2}$, whereas the distance between the indecisive and the assignment point is ε , $0.5 + \varepsilon$, or $1 + \varepsilon$. As $\varepsilon < 0.25$, note that

$$\begin{aligned} 0.25 + \varepsilon &> 2\varepsilon \\ 1 + 0.25 + \varepsilon + \varepsilon^2 &> 1 + 2\varepsilon + \varepsilon^2 \\ 1 + (0.5 + \varepsilon)^2 &> (1 + \varepsilon)^2 \\ \sqrt{1 + (0.5 + \varepsilon)^2} &> 1 + \varepsilon, \end{aligned}$$

so our change to the optimal coupling will replace a pair with a pair of lower distance, so the new coupling is at least as good as the original one, and thus optimal.

Now consider the second coupling. The distance between the assignment point and $(2, 0)$ is at least 2, and $2 > 1 + \varepsilon > 0.5 + \varepsilon > \varepsilon$, so again our change yields an optimal coupling.

By induction on r , we conclude that the statement of the lemma holds. $\square$

We can now use the two previous results to show the following.

LEMMA 3.6. *Given a CNF-SAT formula C containing some clause C_i and m variables $x_1, \dots, x_m$, construct curves $\alpha_1 \sqcup \text{VCG} \sqcup \alpha'_1$ and $\alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2$ for arbitrary precise curves $\alpha_1, \alpha'_1, \alpha_2, \alpha'_2$ with $|\alpha_1| = k$ and $|\alpha_2| = l$. If some optimal coupling between $\alpha_1 \sqcup \text{VCG} \sqcup \alpha'_1$ and $\alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2$ for any realisation of VCG has a pair $(k + 1, l + 1)$ and $d_{\text{dF}}(\alpha_1, \alpha_2) \leq 1$ and $d_{\text{dF}}(\alpha'_1, \alpha'_2) \leq 1$, then the discrete Fréchet distance between the curves is $1 + \varepsilon$ for realisations of VCG that correspond to satisfying assignments for C_i , and 1 for realisations that do not. In other words, if $\pi \in \text{VCG}$ corresponds to an assignment a and we only consider the restricted couplings, then*

$$d_{\text{dF}}(\alpha_1 \sqcup \pi \sqcup \alpha'_1, \alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2) = \begin{cases} 1 + \varepsilon & \text{if } C_i[a] = \text{TRUE}, \\ 1 & \text{otherwise.} \end{cases}$$

PROOF. First of all, since some optimal coupling between $\alpha_1 \sqcup \text{VCG} \sqcup \alpha'_1$ and $\alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2$ for any realisation of VCG has a pair $(k + 1, l + 1)$, we can use Lemma 3.5 to find an optimal coupling Opt that is one-to-one on the subcurves corresponding to the gadgets. That means that we can,

essentially, split the curves, if we consider only such restricted couplings:

$$\begin{aligned} d_{\text{dF}}(\alpha_1 \sqcup \pi \sqcup \alpha'_1, \alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2) &= \max(d_{\text{dF}}(\alpha_1, \alpha_2), d_{\text{dF}}(\pi, \text{ACG}_i), d_{\text{dF}}(\alpha'_1, \alpha'_2)) \\ &= \max(1, d_{\text{dF}}(\pi, \text{ACG}_i)), \end{aligned}$$

where the last equality follows from the fact that $d_{\text{dF}}(\pi, \text{ACG}_i) \geq 1$, since the first points are in a coupling and have the distance 1, and from the assumption that $d_{\text{dF}}(\alpha_1, \alpha_2) \leq 1$ and $d_{\text{dF}}(\alpha'_1, \alpha'_2) \leq 1$. Note that here we do not restrict the coupling on α_1, α_2 and α'_1, α'_2 .

To obtain the end result, we need to consider the distance between π and ACG_i under a one-to-one coupling. Using Lemma 3.4, it is easy to see that if we have $a(x_j) = \text{TRUE}$ for some variable x_j and x_j is a literal in C_i , then $C_i[a] = \text{TRUE}$, and $d_{\text{dF}}(\pi, \text{ACG}_i) = 1 + \varepsilon$; similarly, if $a(x_j) = \text{FALSE}$ for some variable x_j and $\neg x_j$ is a literal in C_i , then $C_i[a] = \text{TRUE}$, and $d_{\text{dF}}(\pi, \text{ACG}_i) = 1 + \varepsilon$. If there is no such x_j , then $C_i[a] = \text{FALSE}$ and $d_{\text{dF}}(\pi, \text{ACG}_i) = 1$. We conclude that the lemma holds. $\square$

Formula Level. Next, we define the *variable curve* and the *clause curve* as follows:

$$\text{VC} = (0, 0) \sqcup \text{VCG} \sqcup (0, 0), \quad \text{CC} = \bigsqcup_{i \in [n]} \text{ACG}_i.$$

Observe that the synchronisation points at $(-2, 0)$ and $(-1, 0)$ ensure that for any optimal coupling we match up VCG with some ACG_i as described before. Also note that all the points on CC are within distance 1 from $(0, 0)$. Therefore, we can always pick any one of n clauses to couple to VCG, and couple the remaining points to $(0, 0)$; the bottleneck distance will then be determined by the distance between VCG and the chosen ACG_i .

Now consider a realisation of VCG. If the corresponding assignment does not satisfy C , then we can synchronise VCG with a clause that is false to obtain the distance of 1. If the assignment corresponding to the realisation satisfies all the clauses, we must synchronise VCG with a satisfied clause, which yields a distance of $1 + \varepsilon$. We show the following important property of our construction.

LEMMA 3.7. *Given a CNF-SAT formula C with n clauses and m variables, construct the curves VC and CC as defined earlier and consider a realisation $(0, 0) \sqcup \pi \sqcup (0, 0)$ of curve VC, corresponding to some assignment a . Then, under no restrictions on the couplings except those imposed by the definition,*

$$d_{\text{dF}}((0, 0) \sqcup \pi \sqcup (0, 0), \text{CC}) = \begin{cases} 1 + \varepsilon & \text{if } C[a] = \text{TRUE}, \\ 1 & \text{if } C[a] = \text{FALSE}. \end{cases}$$

In other words, the discrete Fréchet distance is $1 + \varepsilon$ if the realisation corresponds to a satisfying assignment, and is 1 otherwise.

PROOF. We can show this by proving that the premises of Lemma 3.6 are satisfied.

First of all, note that all the points of CC are within distance 1 from $(0, 0)$. Furthermore, note that we can always give a coupling with the distance at most $1 + \varepsilon$: couple $(0, 0)$ to $(-1, 0)$ from ACG_1 , then walk along realisation of VCG and ACG_1 in a one-to-one coupling, and then couple the remaining points in CC to $(0, 0)$. As all the points of CC are within distance 1 from $(0, 0)$ and as this is otherwise the construction of Lemma 3.6, this coupling yields the discrete Fréchet distance of at most $1 + \varepsilon$ for any realisation of VC. Therefore, any coupling that has pairs further away than $1 + \varepsilon$ cannot be optimal. Observe that the only point within that distance from $(-2, 0)$ is $(-1, 0)$. Therefore, we only need to consider couplings that couple the first point of realisation of VCG to the first point of some ACG_i as possibly optimal. Thus, for each of the n couplings we get, we can apply Lemma 3.6. There are two cases to consider:

- There is some gadget ACG_i with the distance 1 to π under the one-to-one coupling. Then we can choose that gadget to couple to π and couple all the other points in CC to $(0, 0)$ at the beginning or at the end of VC as suitable. As all the points of CC are within distance 1 from $(0, 0)$, this coupling will yield distance 1; as lower distance is impossible, this coupling is optimal, so then $d_{df}((0, 0) \sqcup \pi \sqcup (0, 0), CC) = 1$. Observe that by our construction, this situation corresponds to the case when $C_i[a] = \text{FALSE}$, by Lemma 3.6, and so indeed $C[a] = \text{FALSE}$.
- The distance between any gadget ACG_i and π under the one-to-one coupling is $1 + \varepsilon$. Then, no matter which gadget we choose to couple to π , we will get the distance of $1 + \varepsilon$, so in this case $d_{df}((0, 0) \sqcup \pi \sqcup (0, 0), CC) = 1 + \varepsilon$. Note that, by our construction, this means that $C_i[a] = \text{TRUE}$ for all $i \in [n]$; therefore, indeed $C[a] = \text{TRUE}$.

As we have covered all the possible cases, we conclude that the lemma holds. $\square$

We illustrate the gadgets of the construction in Figure 4. We also show an example of the correspondence between a Boolean formula and our construction in Figure 5.

3.1.2 Upper Bound Discrete Fréchet Distance on Indecisive Points.

THEOREM 3.8. *The problem UPPER BOUND DISCRETE FRÉCHET for indecisive curves is NP-complete.*

PROOF. First of all, observe that if two realisations of lengths n and m are given as a certificate for a ‘Yes’-instance of the problem, then one can verify the solution by computing the discrete Fréchet distance between the realisations and checking that it is indeed larger than the threshold δ . The computation can be done in time $\Theta(mn)$, using the algorithm proposed by Eiter and Mannila [28]. Therefore, the problem is in NP.

Now suppose we are given an instance of CNF-SAT—that is, a CNF-SAT formula C with n clauses and m variables. We construct the curves VC and CC , as described previously, and get an instance of UPPER BOUND DISCRETE FRÉCHET on curves VC and CC with the threshold $\delta = 1$. If the answer is ‘Yes’, then we also output ‘Yes’ as an answer to CNF-SAT; otherwise, we output ‘No’.

Using Lemma 3.7, we see that if there is some assignment a such that $C[a] = \text{TRUE}$, then for the corresponding realisation the discrete Fréchet distance is $1 + \varepsilon$, and the other way around, if for some realisation we get the distance $1 + \varepsilon$, then by our construction all the clauses are satisfied and $C[a] = \text{TRUE}$; thus, $d_{df}^{\max}(VC, CC) = 1 + \varepsilon$. However, if there is no such assignment a , then for any assignment a there is some C_i with $C_i[a] = \text{FALSE}$, yielding $C[a] = \text{FALSE}$, and also for any realisation of VC there is some gadget ACG_i that yields the discrete Fréchet distance of 1; thus, $d_{df}^{\max}(VC, CC) = 1$. Therefore, the formula C is satisfiable if and only if $d_{df}^{\max}(VC, CC) > 1$, and so our answer is correct.

Furthermore, observe that the curves have $2m + 2$ and $2mn + n$ points, respectively, and so the instance of UPPER BOUND DISCRETE FRÉCHET that gives the answer to CNF-SAT can be constructed in polynomial time. Thus, we conclude that UPPER BOUND DISCRETE FRÉCHET for indecisive curves is NP-hard; combining it with the first part of the proof shows that it is NP-complete. $\square$

3.1.3 Upper Bound Fréchet Distance on Indecisive Points. We use the same construction as for the discrete Fréchet distance. To do the same proof, we need to present arguments for the continuous case that lead up to an alternative to Lemma 3.7. For the arguments to work, we need to further restrict the range of ε to be $[0.12, 0.25]$.

Consider the construction drawn in Figure 6. The key points here are that $(0, 0.5 + \varepsilon)$ is far from any point on the clause curve, and that $(2, 0)$ is only close enough to $(1, 0)$. We can present a lemma similar to Lemma 3.4.

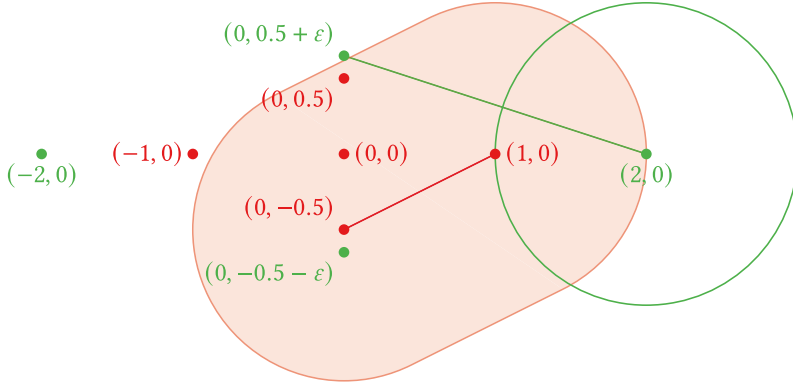


Fig. 6. Construction for $\varepsilon = 0.15$. The shaded red area shows the points within distance 1 from the segment $(0, -0.5) \sqcup (1, 0)$. Observe that $(0, 0.5 + \varepsilon)$ is outside that region, and that $(1, 0)$ is the only red point within distance 1 from $(2, 0)$.

LEMMA 3.9. *Given a clause C_i and a variable x_j that both occur in the CNF-SAT formula C , we only get the Fréchet distance equal to $(1 + \varepsilon) \cdot 2/\sqrt{5}$ if the realisation of VG_j we pick corresponds to the assignment of x_j that ensures the clause C_i is satisfied; otherwise, the Fréchet distance is 1. In other words, if we consider $\pi \in \text{VG}_j$ that corresponds to setting $a(x_j)$, then*

$$d_F(\pi, \text{AG}_{i,j}) = \begin{cases} (1 + \varepsilon) \cdot \frac{2}{\sqrt{5}} & \text{if assigning } x_j \text{ satisfies } C_i, \\ 1 & \text{otherwise.} \end{cases}$$

PROOF. Consider the possible realisations of VG_j . Suppose we pick the realisation $(0, 0.5 + \varepsilon) \sqcup (2, 0)$, which corresponds to assigning $a(x_j) = \text{TRUE}$. If x_j is a literal in C_i , so $C_i = \text{TRUE}$, then by construction we know that $\text{AG}_{i,j}$ is $(0, -0.5) \sqcup (1, 0)$. As noted in Figure 6, the distance between $(0, 0.5 + \varepsilon)$ and any point on $(0, -0.5) \sqcup (1, 0)$ is larger than 1. To be more specific, the distance between the point (x, y) and the line defined by $(x_1, y_1) \sqcup (x_2, y_2)$ can be determined using a standard formula as

$$d = \frac{|x(y_2 - y_1) - y(x_2 - x_1) + x_2y_1 - x_1y_2|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}.$$

In our case, we get

$$d = \frac{|0 - (0.5 + \varepsilon) \cdot (1 - 0) - 1 \cdot 0.5 - 0|}{\sqrt{(1 - 0)^2 + (0 + 0.5)^2}} = \frac{2 \cdot (1 + \varepsilon)}{\sqrt{5}}.$$

As the point $(0, 0.5 + \varepsilon)$ must be aligned with some point on $\text{AG}_{i,j}$, the Fréchet distance we get in this case cannot be smaller than d . Furthermore, it is easy to see that the point $(0, 0.5 + \varepsilon)$ is the furthest point from $\text{AG}_{i,j}$; thus, we get that Fréchet distance is exactly d .

However, if $\neg x_j$ is a literal in C_i , then by construction we know that $\text{AG}_{i,j}$ is $(0, 0.5) \sqcup (1, 0)$. As noted in Figure 6, the distance between $(2, 0)$ and any point on $(0, 0.5) \sqcup (1, 0)$ is at least 1, with the smallest distance achieved at $(1, 0)$. It is clear that this is the furthest pair of points on the two gadgets in this case; thus, we get the Fréchet distance of 1.

A symmetric argument can be applied when we consider the realisation $(0, -0.5 - \varepsilon) \sqcup (2, 0)$ for VG_j : if $\neg x_j$ is a literal in C_i , then we get the Fréchet distance of d and we picked an assignment that satisfies C_i ; in the other case, we get that C_i is not necessarily satisfied and the Fréchet distance is 1.

Finally, consider the case when $\text{AG}_{i,j} = (0, 0) \sqcup (1, 0)$. Again, this implies that assigning a value to x_j has no effect on C_i , so neither assignment (and neither realisation of VG_j) would ensure

that C_i is satisfied. Also observe that both realisations give rise to curves that are entirely within distance 1 of $(0, 0) \sqcup (1, 0)$, yielding the Fréchet distance of 1. $\square$

We can now naturally get a lemma similar to Lemma 3.6.

LEMMA 3.10. *Given a CNF-SAT formula C containing some clause C_i and m variables $x_1, \dots, x_m$, construct curves $\alpha_1 \sqcup \text{VCG} \sqcup \alpha'_1$ and $\alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2$ for arbitrary precise curves $\alpha_1, \alpha'_1, \alpha_2, \alpha'_2$ with $|\alpha_1| = k$ and $|\alpha_2| = l$. If some optimal alignment ϕ_1, ϕ_2 between $\alpha_1 \sqcup \text{VCG} \sqcup \alpha'_1$ and $\alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2$ for any realisation of VCG has some value t such that $\phi_1(t) = k + 1$ and $\phi_2(t) = l + 1$ and $d_F(\alpha_1, \alpha_2) \leq 1$ and $d_F(\alpha'_1, \alpha'_2) \leq 1$, then the Fréchet distance between the curves is $(1 + \varepsilon) \cdot 2/\sqrt{5}$ for realisations of VCG that correspond to satisfying assignments for C_i , and 1 for other realisations. In other words, if $\pi \in \text{VCG}$ corresponds to assignment a and we only consider the restricted alignments, then*

$$d_F(\alpha_1 \sqcup \pi \sqcup \alpha'_1, \alpha_2 \sqcup \text{ACG}_i \sqcup \alpha'_2) = \begin{cases} (1 + \varepsilon) \cdot \frac{2}{\sqrt{5}} & \text{if } C_i[a] = \text{TRUE}, \\ 1 & \text{otherwise.} \end{cases}$$

PROOF. First of all, observe that as we traverse VCG, we need to align $(2, 0)$ with $(1, 0)$ to obtain an optimal alignment. Therefore, essentially, the traversal can be split into m parts, each of which corresponds to traversing VG_j and $\text{AG}_{i,j}$ at the same time for all $j \in [m]$. We can use Lemma 3.9 to note that if some variable x_j is assigned a value that makes the clause C_i satisfied, then the Fréchet distance becomes $(1 + \varepsilon) \cdot 2/\sqrt{5}$; if that is not the case for any variables, then we can traverse the entire curve, as well as α_1 and α'_1 by linearly interpolating our position between the vertices of the curves and otherwise using the alignment derived from the coupling of the discrete case, while staying within distance 1 of the other curve, yielding the Fréchet distance of 1. The distance also cannot be smaller than 1 due to aligning $(2, 0)$ and $(1, 0)$. $\square$

Although this proof is a bit less formal than that of Lemma 3.6, its validity should be sufficiently clear from the geometric considerations described earlier in this section.

Now we can provide a lemma that mirrors Lemma 3.7.

LEMMA 3.11. *Given a CNF-SAT formula C with n clauses and m variables, construct the curves VC and CC as defined earlier and consider a realisation $(0, 0) \sqcup \pi \sqcup (0, 0)$ of curve VC, corresponding to some assignment a . Then*

$$d_F((0, 0) \sqcup \pi \sqcup (0, 0), \text{CC}) = \begin{cases} (1 + \varepsilon) \cdot \frac{2}{\sqrt{5}} & \text{if } C[a] = \text{TRUE}, \\ 1 & \text{if } C[a] = \text{FALSE}. \end{cases}$$

In other words, the Fréchet distance is $(1 + \varepsilon) \cdot 2/\sqrt{5}$ if the realisation π corresponds to a satisfying assignment, and is 1 otherwise.

PROOF. First of all, observe that any point of CC is within distance 1 of $(0, 0)$; furthermore, when starting to traverse π , we must match $(-2, 0)$ with $(-1, 0)$ in an optimal alignment. Thus, the premise of Lemma 3.10 is satisfied, and, using reasoning similar to that of Lemma 3.7, we observe that an optimal alignment chooses one of the clauses to traverse in parallel with the variable curve, and so if there is a clause that is not satisfied, then we get the Fréchet distance of 1, and if all of them are satisfied, then all of them yield the Fréchet distance of $(1 + \varepsilon) \cdot 2/\sqrt{5}$. $\square$

Finally, we can show the main result.

THEOREM 3.12. *The problem UPPER BOUND CONTINUOUS FRÉCHET for indecisive curves is NP-complete.*

PROOF. First of all, observe that if two realisations of lengths n and m are given as a certificate for a ‘Yes’-instance of the problem, then one can verify the solution by checking that the Fréchet

distance between the realisations is larger than the threshold δ . The computation can be done in time $\Theta(mn)$, using the algorithm proposed by Alt and Godau [6, 34]; so the problem is in NP.

Now suppose we are given an instance of CNF-SAT—that is, a CNF-SAT formula C with n clauses and m variables. We construct the curves VC and CC, as described previously, and get an instance of UPPER BOUND CONTINUOUS FRÉCHET on curves VC and CC with the threshold $\delta = 1$. If the answer is ‘Yes’, then we also output ‘Yes’ as an answer to CNF-SAT; otherwise, we output ‘No’.

Using Lemma 3.11, we can see that if there is an assignment a such that $C[a] = \text{TRUE}$, then for the corresponding realisation the Fréchet distance is $(1 + \epsilon) \cdot 2/\sqrt{5}$, and the other way around, if for some realisation we get the distance $(1 + \epsilon) \cdot 2/\sqrt{5}$, then by our construction all the clauses are satisfied and $C[a] = \text{TRUE}$; thus, $d_F^{\max}(\text{VC}, \text{CC}) = (1 + \epsilon) \cdot 2/\sqrt{5}$. However, if there is no such assignment a , then for any assignment a there is some C_i with $C_i[a] = \text{FALSE}$, yielding $C[a] = \text{FALSE}$, and also for any realisation of VC there is some gadget ACG_i that yields the Fréchet distance of 1; thus, $d_F^{\max}(\text{VC}, \text{CC}) = 1$. Therefore, the formula C is satisfiable if and only if $d_F^{\max}(\text{VC}, \text{CC}) > 1$, and so our answer to the CNF-SAT instance is correct.

Furthermore, as before, the instance of UPPER BOUND DISCRETE FRÉCHET that gives the answer to CNF-SAT can be constructed in polynomial time. Thus, we conclude that UPPER BOUND CONTINUOUS FRÉCHET for indecisive curves is NP-hard; combining it with the first part of the proof shows that it is NP-complete. $\square$

3.1.4 Expected Fréchet Distance on Indecisive Points. We show that finding the expected discrete Fréchet distance is #P-hard under the uniform distribution by providing a polynomial-time reduction from #CNF-SAT—that is, the problem of finding the number of satisfying assignments to a CNF-SAT formula. Define the following problem and its continuous counterpart.

PROBLEM 3.13 (EXPECTED DISCRETE FRÉCHET). Find $d_{\text{df}}^{\mathbb{E}(\mathbb{U})}(\mathcal{U}, \mathcal{V})$ for uncertain curves $\mathcal{U}, \mathcal{V}$.

PROBLEM 3.14 (EXPECTED CONTINUOUS FRÉCHET). Find $d_F^{\mathbb{E}(\mathbb{U})}(\mathcal{U}, \mathcal{V})$ for uncertain curves $\mathcal{U}, \mathcal{V}$.

The main idea is to derive an expression for the number of satisfying assignments in terms of $d_{\text{df}}^{\mathbb{E}(\mathbb{U})}(\text{VC}, \text{CC})$. This works, since there is a one-to-one correspondence between Boolean variable assignment and a choice of realisation of VC, so counting the number of satisfying assignments corresponds to finding the proportion of realisations yielding large Fréchet distance. We can establish the result for EXPECTED CONTINUOUS FRÉCHET similarly.

THEOREM 3.15. *The problems EXPECTED DISCRETE FRÉCHET and EXPECTED CONTINUOUS FRÉCHET for indecisive curves are #P-hard.*

PROOF. Suppose we are given an instance of the #CNF-SAT problem—that is, a CNF-SAT formula C with n clauses and m variables. Denote the (unknown) number of satisfying assignments of C by N . We can construct the curves VC and CC in the same way as previously. We then get an instance of EXPECTED DISCRETE FRÉCHET on indecisive curves under the uniform distribution. Assuming we solve it and get $d_{\text{df}}^{\mathbb{E}(\mathbb{U})}(\text{VC}, \text{CC}) = \mu$, we can now compute N :

$$N = (\mu - 1) \cdot \frac{2^m}{\epsilon}.$$

N is then the output for the instance of #CNF-SAT that we were given. Clearly, construction of the curves can be done in polynomial time, and so can the computation of N ; hence, the reduction takes polynomial time.

We still need to show that the result we obtain is correct. For each assignment, there is exactly one realisation of the curve VC. Furthermore, as we choose the realisation of each indecisive point uniformly and independently, all the realisations of VC have equal probability of 2^{-m} . There are

N satisfying assignments, and each of the corresponding realisations yields the discrete Fréchet distance of $1 + \varepsilon$. In the remaining $2^m - N$ cases, the distance is 1. Using the definition of expected value, we can derive

$$\mu = d_{\text{dF}}^{\mathbb{E}(\mathbb{U})}(\text{VC}, \text{CC}) = N \cdot 2^{-m} \cdot (1 + \varepsilon) + (2^m - N) \cdot 2^{-m} \cdot 1 = 1 + \frac{N \cdot \varepsilon}{2^m}.$$

Then it is easy to see that indeed $N = (\mu - 1) \cdot \frac{2^m}{\varepsilon}$. So, we get the correct number of satisfying assignments, if we know the expected value under the uniform distribution. Therefore, EXPECTED DISCRETE FRÉCHET for indecisive curves is #P-hard.

One can derive a very similar formula to show that EXPECTED CONTINUOUS FRÉCHET is also #P-hard for indecisive curves. We can use almost the same reduction as for the discrete case, so given an instance of #CNF-SAT (CNF-SAT formula C with n clauses and m variables), we construct the two curves, solve EXPECTED CONTINUOUS FRÉCHET to obtain the value of μ , and compute

$$N = 2^m \cdot (\mu - 1) \cdot \frac{\sqrt{5}}{2(1 + \varepsilon) - \sqrt{5}}$$

as the output for #CNF-SAT. To show that the output is correct, note that

$$\mu = 2^{-m} \cdot N \cdot \frac{2}{\sqrt{5}} \cdot (1 + \varepsilon) + 2^{-m} \cdot (2^m - N) \cdot 1 = 1 + 2^{-m} \cdot N \cdot \left(\frac{2}{\sqrt{5}}(1 + \varepsilon) - 1 \right),$$

so we can express N as

$$N = 2^m \cdot (\mu - 1) \cdot \frac{\sqrt{5}}{2(1 + \varepsilon) - \sqrt{5}}.$$

Again, the reduction is correct and can be done in polynomial time, so EXPECTED CONTINUOUS FRÉCHET for indecisive curves is #P-hard. $\square$

We use the uniform distribution; however, we only need to compute the probability of picking a realisation that corresponds to a satisfying assignment, $N \cdot 2^{-m}$ above. If we can do so for a different distribution, then the rest of the proof does not require modifications to show #P-hardness.

3.1.5 Upper Bound Discrete Fréchet Distance on Imprecise Points. Here we consider imprecise points modelled as disks and as line segments; the results and their proofs turn out to be very similar. We denote the disk with the centre at $p \in \mathbb{R}^d$ and radius $r \geq 0$ as $D(p, r)$. We denote the line segment between points p_1 and p_2 by $S(p_1, p_2)$.

Disks. We use a construction very similar to that of the indecisive points case, except now we change the gadget containing a non-degenerate indecisive point so that it contains a non-degenerate imprecise point, for all $j \in [m]$:

$$\text{VG}_j = D((0, 0), 0.5 + \varepsilon) \sqcup (2, 0).$$

Essentially, the two original indecisive points are now located on the points realising the diameter of the disk.

We can reuse the proof leading up to Theorem 3.8, if we can show the following.

LEMMA 3.16. *Suppose $d_{\text{dF}}^{\max}(\text{VC}, \text{CC}) = \nu$. If one considers all the realisations π of VC that yield $d_{\text{dF}}(\pi, \text{CC}) = \nu$, then among them there will always be a realisation that only places the imprecise point realisations at either $(0, 0.5 + \varepsilon)$ or $(0, -0.5 - \varepsilon)$.*

PROOF. First of all, note that the points $(2, 0)$ and $(1, 0)$ are still in the curves in the same quality as before, so they must be coupled, and hence the lowest discrete Fréchet distance achievable with any realisation is 1.

Now consider a realisation of an imprecise point. Suppose that all the clause assignment points for that imprecise point are placed at $(0, -0.5)$. Then geometrically it is obvious that the distance is maximised by placing the realisation at $(0, 0.5 + \varepsilon)$; if there is a realisation that achieves the best possible value v without doing this, then we can move this point and still get v .

Suppose that some clause assignment points are at $(0, -0.5)$ and some at $(0, 0.5)$. As the realisation comes from the disk of radius $0.5 + \varepsilon$, there is no realisation that is further than 1 away from both assignment points; therefore, to maximise the distance, we have to choose one of the two locations and then the previous case applies.

So, it is clear that, from an arbitrary optimal realisation, moving to the (correct) indecisive point realisation will still yield an optimal realisation for the maximum discrete Fréchet distance; thus, the statement of the lemma holds. $\square$

Line Segments. We change the gadget to be, for all $j \in [m]$,

$$VG_j = S((0, -0.5 - \varepsilon), (0, 0.5 + \varepsilon)) \sqcup (2, 0).$$

Again, the two original indecisive points are now located on the ends of the segment; moreover, the segment is a strict subset of the disk.

We can state a similar lemma.

LEMMA 3.17. *Suppose $d_{\text{df}}^{\max}(\text{VC}, \text{CC}) = v$. If one considers all the realisations π of VC that yield $d_{\text{df}}(\pi, \text{CC}) = v$, then among them there will always be a realisation that only places the imprecise point realisations at either $(0, 0.5 + \varepsilon)$ or $(0, -0.5 - \varepsilon)$.*

PROOF. Since the line segments include these points and are subsets of the disks, the statement of Lemma 3.16 immediately yields this result. $\square$

So, now we can state the following for both models.

THEOREM 3.18. *The problem UPPER BOUND DISCRETE FRÉCHET for imprecise curves modelled as line segments or as disks is NP-hard.*

PROOF. As shown in Lemma 3.16 and Lemma 3.17, for the same CNF-SAT formula, the upper bound discrete Fréchet distance on indecisive and imprecise points is equal for our construction. So, trivially, UPPER BOUND DISCRETE FRÉCHET is NP-hard for imprecise curves. $\square$

3.1.6 Upper Bound Fréchet Distance on Imprecise Points. We use exactly the same construction as in the previous section. The argument here follows the previous ones very closely, so we can immediately state the following theorem.

THEOREM 3.19. *The problem UPPER BOUND CONTINUOUS FRÉCHET for imprecise curves modelled as line segments or as disks is NP-hard.*

PROOF. Note that we can apply exactly the same argument as the one in Lemma 3.16 and Lemma 3.17 to reduce this problem to the one on indecisive points. Then, we can apply the same argument as in the proof of Theorem 3.18 to conclude that the problem is NP-hard. $\square$

3.1.7 Expected Discrete Fréchet Distance on Imprecise Points. We can also consider the value of the expected Fréchet distance on imprecise points. We show the result only for points modelled as line segments; in principle, we believe that for disks a similar result holds, but the specifics of our reduction do not allow for clean computations.

We cannot immediately use our construction: we treat subsegments at the ends of the imprecision segments as TRUE and FALSE, but we have no interpretation for points in the centre part of a segment. So, we want to separate the realisations that pick any such invalid points. To that aim, we

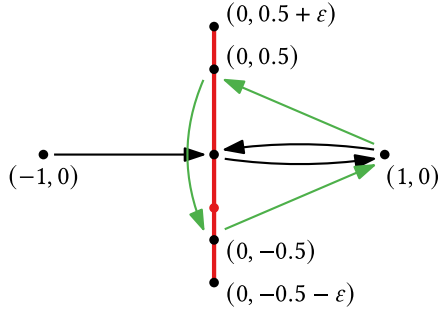


Fig. 7. The curve FG_j hops between $(0, 0)$ and $(1, 0)$ for every variable x_k (in black) except when $k = j$; in the latter case, the curve goes to $(0, 0.5)$, $(0, -0.5)$, and back to $(1, 0)$ (in green). Consider the line segment on the variable curve representing x_j (in red). As a consequence, for any realisation of the variable clause gadget such that the realisation of x_j falls within $S((0, -0.5), (0, 0.5))$, the gadget FG_j can be aligned with VCG to obtain Fréchet distance 1.

introduce extra gadgets to the clause curve that act as clauses but catch these invalid realisations, so each of them yields the distance of 1. Now we have three distinct cases: realisation is satisfying, non-satisfying, or invalid. For every $j \in [m]$, define

$$FG_j = (-1, 0) \sqcup \bigsqcup_{k \in [j-1]} \left((0, 0) \sqcup (1, 0) \right) \sqcup (0, 0.5) \sqcup (0, -0.5) \sqcup (1, 0) \sqcup \bigsqcup_{k \in [m] \setminus [j]} \left((0, 0) \sqcup (1, 0) \right).$$

So, we define a clause gadget that ignores all the variables except for x_j and then features both ‘true’ and ‘false’ for x_j . The intuition is that any realisation corresponding to the invalid state of a variable will be close to both $(0, 0.5)$ and $(0, -0.5)$, and every other variable value is close to $(0, 0)$, so aligning the gadget FG_j with the variable curve will yield a small Fréchet distance if x_j is in an invalid state. See also Figure 7. We then define the clause curve as

$$CC = \bigsqcup_{i \in [n]} ACG_i \sqcup \bigsqcup_{j \in [m]} FG_j.$$

We can now choose to couple one of the FG clauses to the variable curve. As before, due to the synchronisation points, we can never get the Fréchet distance below 1. If one of the realisations x_j of the segments falls into the interval $[(0, -0.5), (0, 0.5)]$, then it will be not further away than 1 from both the corresponding points on FG_j ; all the other points, being in the middle at $(0, 0)$, are guaranteed to be at most $0.5 + \varepsilon < 1$ away from their coupled point. So, the one-to-one coupling² will yield the discrete Fréchet distance of 1; thus, the optimal discrete Fréchet distance in this case is 1. Therefore, we only need to consider the situations when all the realisations happen to fall in either the interval $((0, 0.5), (0, 0.5 + \varepsilon)]$ or $[(0, -0.5 - \varepsilon), (0, -0.5))$. We will treat the first interval as TRUE and the second interval as FALSE. Denote the number of satisfying assignments by N . To find the expression for the expected discrete Fréchet distance, we need to consider three cases:

- At least one realisation of m variables falls within the y -interval $[-0.5, 0.5]$. Note that the realisation on each segment is uniform and independent of other segments. Under the

²Technically, it is one-to-one on all points except the realisation corresponding to x_j ; that one has to be coupled to both $(0, 0.5)$ and $(0, -0.5)$ in FG_j .

uniform distribution, we get

$$\Pr[\text{at least one realisation from } [-0.5, 0.5]] = 1 - \prod_{j \in [m]} \frac{2\varepsilon}{1 + 2\varepsilon} = 1 - \left(\frac{2\varepsilon}{1 + 2\varepsilon} \right)^m.$$

Note that in each such case, we get the discrete Fréchet distance of 1, as discussed before.

- All realisations fall outside the y -interval $[-0.5, 0.5]$, and they correspond to a non-satisfying assignment. Each specific non-satisfying assignment corresponds to picking values on the specific interval, either $((0, 0.5), (0, 0.5 + \varepsilon))$ or $((0, -0.5 - \varepsilon), (0, -0.5))$, so

$$\Pr[\text{specific assignment}] = \prod_{j \in [m]} \frac{\varepsilon}{1 + 2\varepsilon} = \left(\frac{\varepsilon}{1 + 2\varepsilon} \right)^m.$$

There are $2^m - N$ such assignments, and each of them contributes the value of 1.

- All realisations fall outside the y -interval $[-0.5, 0.5]$, and they correspond to a satisfying assignment. Again, the probability of getting a particular assignment is $(\varepsilon/(1 + 2\varepsilon))^m$, and there are N such assignments. Now they contribute values distinct from 1; still, the optimum is contributed by one of the new clauses, and then it will be defined by the realisation closest to $(0, 0)$. This is shown in the following lemma.

LEMMA 3.20. *Consider some realisation $\pi \in \text{VC}$ where each value can be interpreted either as TRUE or FALSE and the corresponding assignment satisfies the formula. Pick j such that the subcurve of π realising VG_j contains the point closest to $(0, 0)$, at location $(0, 0.5 + \varepsilon')$ or $(0, -0.5 - \varepsilon')$ for some $\varepsilon' > 0$. Then the optimal coupling establishes a coupling between π and FG_j , and the discrete Fréchet distance is $d_{\text{dF}}(\pi, \text{CC}) = 1 + \varepsilon'$.*

PROOF. First of all, note that we still have to couple the synchronisation points and we cannot have discrete Fréchet distance below 1. So, we need to consider only the couplings of π with the gadgets of CC . Note that if we couple FG_j to π , we get discrete Fréchet distance of $1 + \varepsilon'$. Recall that we consider only satisfying assignments, so, if we consider an arbitrary subcurve ACG_i , then there is some variable x_ℓ that satisfies the corresponding clause, and so the realisation of that variable is $1 + \varepsilon''$ away from the corresponding assignment point. Therefore, such a coupling will yield the discrete Fréchet distance of $1 + \varepsilon'' \geq 1 + \varepsilon'$. Finally, it is easy to see that choosing some FG_k with $k \neq j$ will also yield some distance $1 + \varepsilon'' \geq 1 + \varepsilon'$. So, the statement of the lemma holds. $\square$

So, here we need to find $\mathbb{E}[\min_{j \in [m]} (1 + \varepsilon'_j)]$ with ε'_j sampled uniformly from $(0, \varepsilon]$; we can rephrase this to $1 + \varepsilon \cdot \mathbb{E}[\min_{j \in [m]} u_j]$ with u_j sampled uniformly from $(0, 1]$. It is a standard result that the minimum now is geometrically distributed, so we get $\mathbb{E}[\min_{j \in [m]} u_j] = 1/(1 + m)$, and hence the expected contribution is $1 + \varepsilon/(1 + m)$.

We can bring the three cases together to find

$$\begin{aligned} d_{\text{dF}}^{(\mathbb{U})}(\text{VC}, \text{CC}) &= 1 \cdot \left(1 - \left(\frac{2\varepsilon}{1 + 2\varepsilon} \right)^m \right) + 1 \cdot (2^m - N) \cdot \left(\frac{\varepsilon}{1 + 2\varepsilon} \right)^m + \left(1 + \frac{\varepsilon}{1 + m} \right) \cdot N \cdot \left(\frac{\varepsilon}{1 + 2\varepsilon} \right)^m \\ &= 1 + N \cdot \frac{\varepsilon^{m+1}}{(1 + m) \cdot (1 + 2\varepsilon)^m}. \end{aligned}$$

So, if we were to compute $d_{\text{dF}}^{(\mathbb{U})}(\text{VC}, \text{CC}) = \mu$, then the number of satisfying assignments is

$$N = (\mu - 1) \cdot \frac{(1 + m) \cdot (1 + 2\varepsilon)^m}{\varepsilon^{m+1}}.$$

This is easy to compute in polynomial time, and our construction can still be done in polynomial time; hence, the result follows.

THEOREM 3.21. *The problem EXPECTED DISCRETE FRÉCHET for imprecise curves modelled as line segments under the uniform distribution is #P-hard.*

We have stated the results for the uniform distribution; however, we conjecture that this construction could work for some other distributions. The requirements are that we need to be able to compute the probabilities of falling into each region; that all realisations are equiprobable, or we have some other way to compute the probability of getting a satisfying realisation; and that we can compute $\mathbb{E}[\min_{j \in [m]} u_j]$ under the appropriate distribution of u_j .

3.2 Lower Bound Fréchet Distance

In this section, we prove that computing the lower bound continuous Fréchet distance is NP-hard for uncertainty modelled with line segments. This contrasts with the algorithm for indecisive curves, given in Section 4.1, and with the algorithm previously suggested by Ahn et al. [5] for the discrete Fréchet distance. Unlike the upper bound proofs, this reduction uses the NP-hard problem SUBSET-SUM. We consider the following problems.

PROBLEM 3.22 (LOWER BOUND CONTINUOUS FRÉCHET). *Given an uncertain curve $\mathcal{U}$ with m vertices, a polygonal curve σ with n vertices, and a threshold $\delta > 0$, decide if $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta$.*

PROBLEM 3.23 (SUBSET-SUM). *Given a set $S = \{s_1, \dots, s_n\}$ of n positive integers and a target integer τ , decide if there exists an index set I such that $\sum_{i \in I} s_i = \tau$.*

As a polygonal curve is an uncertain curve, proving Problem 3.22 is NP-hard implies the corresponding problem with two uncertain curves is also NP-hard.

3.2.1 An Intermediate Problem. We start by reducing SUBSET-SUM to a more geometric intermediate curve-based problem.

Definition 3.24. Let $\alpha > 0$ be some value, and let $\pi = \langle \pi_1, \dots, \pi_{2n+1} \rangle$ be a polygonal curve. We call π an α -regular curve if for all $i \in [2n+1]$, the x -coordinate of π_i is $i \cdot \alpha$. Let $Y = \{y_1, \dots, y_n\}$ be a set of n positive integers. Call π a Y -respecting curve if

- (1) For all $i \in [n]$, π passes through the point $((2i + 1/2)\alpha, 0)$.
- (2) For all $i \in [n]$, π either passes through the point $((2i - 1/2)\alpha, 0)$ or $((2i - 1/2)\alpha, -y_i)$.

Intuitively, Definition 3.24 requires π to pass through $((2i + 1/2)\alpha, 0)$ as it reflects the y -coordinate about the line $y = 0$ (Figure 8). Thus, if the curve also passes through $((2i - 1/2)\alpha, 0)$, the two reflections cancel each other. If it passes through $((2i - 1/2)\alpha, -y_i)$, the following lemma argues that y_i shows up in the final vertex height.

LEMMA 3.25. *Let π be a Y -respecting α -regular curve, and let I be the subset of indices such that π passes through $((2i - 1/2)\alpha, -y_i)$ for all $i \in I$. If $\pi_1 = (\alpha, 0)$, then $\pi_{2n+1} = ((2n + 1)\alpha, 2 \sum_{i \in I} y_i)$.*

PROOF. For $j \in [n]$, let $I_j = \{i \in I \mid i \leq j\}$, and let $\beta_j = \sum_{i \in I_j} y_i$, where $\beta_0 = 0$. We argue by induction that $\pi_{2j+1} = ((2j + 1)\alpha, 2\beta_j)$, thus yielding the lemma statement when $j = n$. For the base case, $j = 0$, the statement becomes $\pi_1 = (\alpha, 0)$, which is true by assumption of the lemma.

Assume that $\pi_{2j-1} = ((2j - 1)\alpha, 2\beta_{j-1})$. Suppose that $j \notin I$. In this case, since π is Y -respecting, it passes through points $((2j - 1/2)\alpha, 0)$ and $((2j + 1/2)\alpha, 0)$. This implies $\pi_{2j} = (2j\alpha, -2\beta_{j-1})$ and $\pi_{2j+1} = ((2j + 1)\alpha, 2\beta_{j-1}) = ((2j + 1)\alpha, 2\beta_j)$. Now suppose that $j \in I$. In this case,



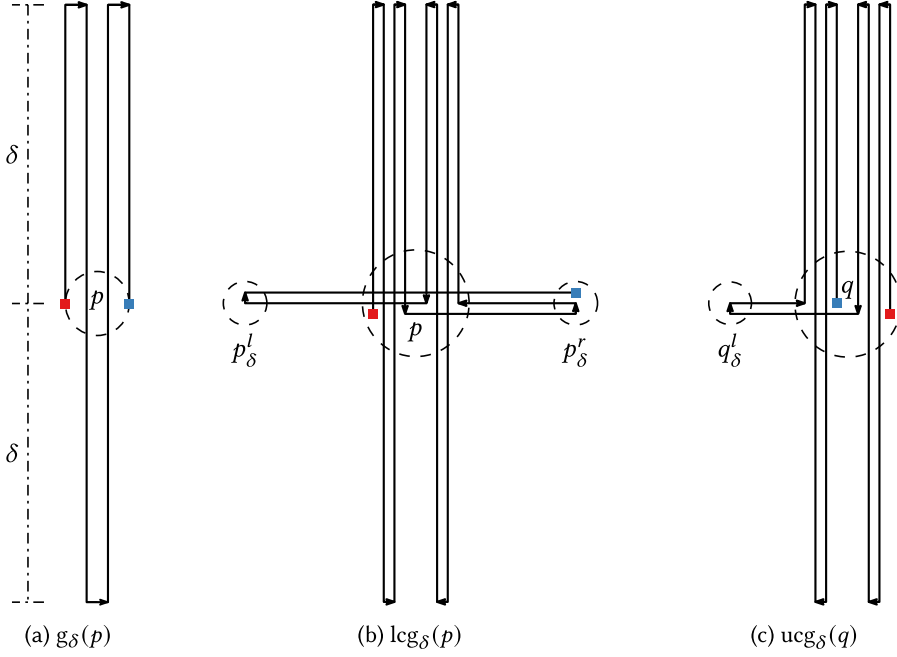


Fig. 9. Depiction of gadgets $g_\delta(p)$, $lcg_\delta(p)$, and $ucg_\delta(p)$. Dashed circles represent zero-area points; the red (blue) square represents the starting (ending) point.

PROOF. In order, $g_\delta(p)$ visits the points $(x, y + \delta)$, $(x, y - \delta)$, and $(x, y + \delta)$. Let $a = (a_x, a_y)$, $b = (b_x, b_y)$, $c = (c_x, c_y)$ be the points from S which get aligned with these respective points under an optimal Fréchet alignment. If the Fréchet distance is at most δ , then $b_y \leq y \leq a_y, c_y$. If S has a positive slope with respect to the order along S , then also $c_y \geq b_y \geq a_y$, so we have $a_y = b_y$ and so $a = b$. However, if $a = b$, then this point must be p itself, as p is the only point with distance at most δ from both $(x, y + \delta)$ and $(x, y - \delta)$. If S has a negative slope, then $c_y \leq b_y \leq a_y$, so now $b_y = c_y$ and $b = c$, and again this must be point p . Finally, if S is horizontal, then $a = b = c = p$, as this is the only point on a horizontal segment aligned with both $(x, y + \delta)$ and $(x, y - \delta)$. $\square$

For our uncertain curve to be Y -respecting, it must pass through all the points of the form $((2i + 1/2)\alpha, 0)$. This condition is satisfied by placing a δ -gadget at each such point, as follows from Lemma 3.31. The second condition for a curve to be Y -respecting is that it passes through $((2i - 1/2)\alpha, 0)$ or $((2i - 1/2)\alpha, -y_i)$. This condition is much harder to encode and requires putting several δ -gadgets together to create a composite gadget.

Definition 3.32. For any point $p = (x, y) \in \mathbb{R}^2$ and value $\delta > 0$, let $p_\delta^l = (x - \delta/2, y)$ and $p_\delta^r = (x + \delta/2, y)$. Define the δ -lower composite gadget at p , denoted $lcg_\delta(p)$, to be the curve $g_\delta(p) \sqcup p_\delta^l \sqcup p_\delta^r$. See Figure 9(b). Define the δ -upper composite gadget at q , denoted $ucg_\delta(q)$, to be the curve $g_\delta(q) \sqcup q_\delta^l \sqcup g_\delta(q)$. See Figure 9(c). Define the δ -composite gadget of p and q , denoted $cg_\delta(p, q)$, to be the curve $lcg_\delta(p) \sqcup ucg_\delta(q)$.

To use the composite gadget, we centre the lower gadget at height $-y_i$ and the upper gadget directly above it at height zero. As the two gadgets are on top of each other, ultimately, we require our uncertain curve to go back and forth once between consecutive vertical line segments; we have the following key property.

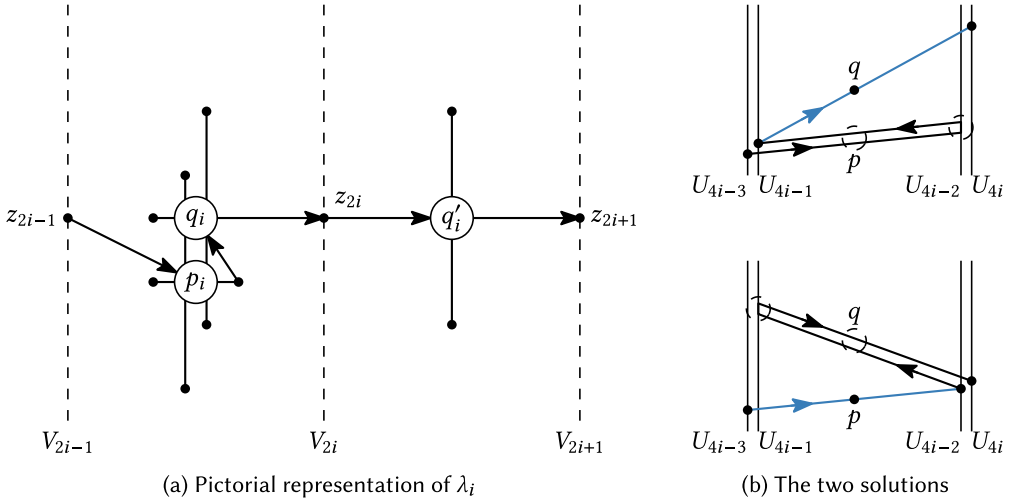


Fig. 10. On the left, λ_i . On the right, the two possible solutions with Fréchet distance at most δ . The top (respectively, bottom) corresponds to an α -regular curve passing through q (respectively, p).

LEMMA 3.33. Let $p = (p_x, -p_y)$ and $q = (p_x, 0)$ be points in $\mathbb{R}^2$. Let $\pi = \langle a, b, c, d \rangle$ be a three-segment curve such that $b_x > p_x + \delta$ and $c_x < p_x - \delta$. If $d_F(\pi, \text{cg}_\delta(p, q)) \leq \delta$, then

- (1) the segment ab must pass through p ,
- (2) the segment cd must pass through q , and
- (3) the segment bc must either pass through p or through q .

PROOF. Recall from Definition 3.32 that $\text{cg}_\delta(p, q) = g_\delta(p) \sqcup p_\delta^r \sqcup g_\delta(p) \sqcup p_\delta^l \sqcup p_\delta^r \sqcup g_\delta(q) \sqcup q_\delta^l \sqcup g_\delta(q)$, and that the gadgets $g_\delta(p)$ and $g_\delta(q)$ lie entirely on the vertical line at $p_x = q_x$. Thus, as $b_x > p_x + \delta$ and $c_x < p_x - \delta$, each occurrence of $g_\delta(p)$ or $g_\delta(q)$ in $\text{cg}_\delta(p, q)$ must map either entirely before or after b , and similarly entirely before or after c .

Moreover, as $\text{cg}_\delta(p, q)$ starts with $g_\delta(p)$ and $b_x > p_x + \delta$, this implies that a maps to p and $g_\delta(p)$ maps to a subsegment of ab , which by Lemma 3.31 implies that ab passes through p . Similarly, as $\text{cg}_\delta(p, q)$ ends with $g_\delta(q)$ and $c_x < p_x - \delta$, cd passes through q .

Finally, the portion of $\text{cg}_\delta(p, q)$ that maps to the segment bc must contain a point on the vertical line at $p_x = q_x$ (since $b_x > p_x + \delta$ and $c_x < p_x - \delta$). By the construction of $\text{cg}_\delta(p, q)$, this point must lie on one of the (middle) $g_\delta(p)$ or $g_\delta(q)$ gadgets. As we already argued, such gadgets must map entirely to one side of b or c , so Lemma 3.31 implies that bc must pass through p or q . $\square$

As bc shares an endpoint with ab and cd , the following corollary is immediate. It is used later to argue that while our uncertain curve goes back and forth between consecutive vertical lines, it defines an α -regular curve. See Figure 10 used for Theorem 3.36.

COROLLARY 3.34. If $d_F(\pi, \text{cg}_\delta(p, q)) \leq \delta$, then either ab and bc are on the same line, or cd and bc are on the same line.

The following lemma acts as a rough converse of Lemma 3.33.

LEMMA 3.35. Let $p = (p_x, -p_y)$ and $q = (p_x, 0)$ be points in $\mathbb{R}^2$, with $p_y \leq \delta/4$. Let $\pi = \langle p, b, c, q \rangle$ be a curve such that $p_x + \delta < b_x \leq p_x + 1.1\delta$, $p_x - 1.1\delta \leq c_x < p_x - \delta$, and $-\delta/2 \leq b_y, c_y \leq \delta/2$. If bc passes through either p or q , then $d_F(\pi, \text{cg}_\delta(p, q)) \leq \delta$.

PROOF. Recall that $\text{cg}_\delta(p, q) = g_\delta(p) \sqcup p_\delta^r \sqcup g_\delta(p) \sqcup p_\delta^l \sqcup p_\delta^r \sqcup g_\delta(q) \sqcup q_\delta^l \sqcup g_\delta(q)$. First, observe that all the points on the prefix $g_\delta(p) \sqcup p_\delta^r$ of $\text{cg}_\delta(p, q)$ are at most δ away from p , and thus can all be mapped to the starting point of π . Similarly, all points on the suffix $q_\delta^l \sqcup g_\delta(q)$ of $\text{cg}_\delta(p, q)$ are at most δ away from q , and thus can all be mapped to the ending point of π . Thus, it suffices to argue that $d_F(\pi, \sigma) \leq \delta$, where $\sigma = p_\delta^r \sqcup g_\delta(p) \sqcup p_\delta^l \sqcup p_\delta^r \sqcup g_\delta(q) \sqcup q_\delta^l$.

It is easiest to describe the rest of the mapping in a similar manner—that is, as an alternating sequence of moves, where we stand still at a single point on one curve while moving along a contiguous subcurve from the other curve, and then switching curves. We now describe this sequence, which differs based on whether bc passes through p or q . Ultimately, the mappings are valid, since for each move, all points on the subcurve have distance at most δ to the fixed point on the other curve. Thus, we now simply describe the moves without reiterating this property (distance at most δ) which is validating each move.

First, suppose that bc passes through p , in which case $\pi = \langle p, b, p, c, q \rangle$. In this case, we first map the prefix $\langle p, b, p \rangle$ of π to p_δ^r . Next, we map the prefix $p_\delta^r \sqcup g_\delta(p) \sqcup p_\delta^l$ of σ to p . Then we map the suffix $\langle p, c, q \rangle$ of π to p_δ^l . Finally, we map the suffix $p_\delta^l \sqcup p_\delta^r \sqcup g_\delta(q) \sqcup q_\delta^l$ of σ to q .

Now suppose that bc passes through q , in which case $\pi = \langle p, b, q, c, q \rangle$. In this case, we first map the prefix $p_\delta^r \sqcup g_\delta(p) \sqcup p_\delta^l$ of σ to p . Next, we map the prefix $\langle p, b, q \rangle$ of π to p_δ^r . Then we map the suffix $p_\delta^r \sqcup g_\delta(q) \sqcup q_\delta^l$ of σ to q . Finally, we map the suffix $\langle q, c, q \rangle$ of π to q_δ^l . $\square$

THEOREM 3.36. *LOWER BOUND CONTINUOUS FRÉCHET (Problem 3.22) is NP-hard, even when the uncertain regions are all equal-length vertical segments with the same height and the same horizontal distance (to the left or right) between adjacent uncertain regions.*

PROOF. To prove NP-hardness, we give a reduction from RR-CURVE, which is NP-hard by Theorem 3.28. Let $\alpha(Y)$, τ , $Y = \{y_1, \dots, y_n\}$ be an instance of RR-CURVE. For the reduction, we set $\delta = 4M$, where $M = \sum_{i=1}^n y_i$. Note that Theorem 3.28 allows us to choose how to set $\alpha(Y)$, and in particular we set $\alpha = 2.1\delta = 8.4M$. (More precisely, the properties we need are that $\alpha > 2\delta$ and $\delta \geq 4M$.) We now describe how to construct $\mathcal{U}$ and σ .

Let $V = \{V_1, \dots, V_{2n+1}\}$ be a set of vertical line segments where all upper (respectively, lower) endpoints of the segments have height $2M$ (respectively, $-2M$), and for all i , the x -coordinate of V_i is $i\alpha$. Let $\mathcal{U} = \langle U_1, \dots, U_{4n+1} \rangle$ be the uncertain curve such that $U_{4n+1} = V_{2n+1}$, and for all $i \in [n]$, $U_{4i-3} = V_{2i-1}$, $U_{4i-2} = V_{2i}$, $U_{4i-1} = V_{2i-1}$, and $U_{4i} = V_{2i}$. For $i \in [2n+1]$, define the points $z_i = (i\alpha, 0)$, and for $i \in [n]$, define $q_i = ((2i - 1/2)\alpha, 0)$, $q'_i = ((2i + 1/2)\alpha, 0)$, and $p_i = ((2i - 1/2)\alpha, -y_i)$. For a given value $i \in [n]$, consider the curve $\lambda_i = z_{2i-1} \sqcup \text{cg}_\delta(p_i, q_i) \sqcup z_{2i} \sqcup g_\delta(q'_i)$ (see Figure 10(a)). Let $s = (\alpha, 0)$ and $t = ((2n+1)\alpha, 2\tau)$. Then the curve σ is defined as

$$\sigma = g_\delta(s) \sqcup \lambda_1 \sqcup \lambda_2 \sqcup \dots \sqcup \lambda_n \sqcup g_\delta(t).$$

First, suppose there is a curve $\pi' = \langle \pi'_1, \dots, \pi'_{4n+1} \rangle \in \mathcal{U}$ such that $d_F(\pi', \sigma) \leq \delta$. Let $\pi = \langle \pi_1, \dots, \pi_{2n+1} \rangle$ be the curve such that $\pi_{2n+1} = \pi'_{4n+1}$, and for all $i \in [n]$, $\pi_{2i-1} = \pi'_{4i-3}$ and $\pi_{2i} = \pi'_{4i}$. We argue that π is an α -regular Y -respecting curve with $\pi_1 = s$ and $\pi_{2n+1} = t$.

Observe that π is α -regular, as by the definition of $\mathcal{U}$, π_i is a point on the vertical segment V_i . Additionally, as σ begins (respectively, ends) with $g_\delta(s)$ (respectively, $g_\delta(t)$), by Lemma 3.31, $\pi_1 = \pi'_1 = s$ (respectively, $\pi_{2n+1} = \pi'_{4n+1} = t$). Thus, it remains to argue that π is Y -respecting. To that end, consider the portion λ_i of σ for some i .

First, consider the gadget $g_\delta(q'_i)$ from λ_i lying between z_{2i} and z_{2i+1} . By our choice of α , this gadget is strictly more than δ away from both V_{2i} and V_{2i+1} , and so the portion of π' aligned with $g_\delta(q'_i)$ must lie between $\pi'_{4i} = \pi_{2i}$ and $\pi'_{4i+1} = \pi_{2i+1}$. Thus, by Lemma 3.31, π must pass through q'_i .

Now consider the gadget $\text{cg}_\delta(p_i, q_i) = \text{lcg}(p_i) \sqcup \text{ucg}(q_i)$ from λ_i lying between z_{2i-1} and z_{2i} . This gadget is strictly more than δ away from both V_{2i-1} and V_{2i} , implying both that the portion of π' aligned with $\text{cg}_\delta(p_i, q_i)$ lies between π'_{4i-3} and π'_{4i} , and that all three segments in the subcurve from π'_{4i-3} to π'_{4i} must in part map to $\text{cg}_\delta(p_i, q_i)$. Thus, by Lemma 3.33, $\pi'_{4i-3}\pi'_{4i-2}$ passes through p_i , and $\pi'_{4i-1}\pi'_{4i}$ passes through q_i . By Corollary 3.34, either $\pi'_{4i-2} = \pi'_{4i}$ or $\pi'_{4i-3} = \pi'_{4i-1}$, and thus $\pi'_{4i-3}\pi'_{4i} = \pi_{2i-1}\pi_{2i}$ passes through either p_i or q_i (see Figure 10(b)). Thus, π is Y -respecting.

Now suppose that there is an α -regular Y -respecting curve $\pi = \langle \pi_1, \dots, \pi_{2n+1} \rangle$ such that $\pi_1 = s$ and $\pi_{2n+1} = t$. Let $\text{int}(p_i)$ be the intersection with V_{2i} of the line through π_{2i-1} and p_i , and let $\text{int}(q_i)$ be the intersection with V_{2i-1} of the line through π_{2i} and q_i . Let $\pi' = \langle \pi'_1, \dots, \pi'_{4n+1} \rangle$ be the curve such that $\pi'_{4n+1} = \pi_{2n+1}$, and for all $i \in [n]$, $\pi'_{4i-3} = \pi_{2i-1}$, $\pi'_{4i-2} = \text{int}(p_i)$, $\pi'_{4i-1} = \rho$, and $\pi'_{4i} = \pi_{2i}$, where $\rho = \pi_{2i-1}$ if π passes through q_i and $\rho = \text{int}(q_i)$ if π passes through p_i . See Figure 10(b).

Let $\text{mid}(S)$ be the midpoint of a line segment S . Observe that by construction, $\text{mid}(\pi'_{4i-3}\pi'_{4i-2}) = p_i$, $\text{mid}(\pi'_{4i-1}\pi'_{4i}) = q_i$, and $\text{mid}(\pi'_{4i-2}\pi'_{4i-1}) = p_i$ (respectively, q_i) if π passed through q_i (respectively, p_i). Let $\gamma_i = \langle p_i, \pi'_{4i-2}, \pi'_{4i-1}, q_i \rangle$, which by the previous argument is a subcurve of π' .

To argue that $d_F(\pi', \sigma) \leq \delta$, we now describe how to walk along the curves π' and σ so that at all times the distance between the positions on the respective curves is at most δ . Note that γ_i satisfies the conditions of Lemma 3.35, implying that $d_F(\text{cg}_\delta(p_i, q_i), \gamma_i) \leq \delta$, and thus for all i , we can map $\text{cg}_\delta(p_i, q_i)$ to γ_i . For the other parts of the curves, first observe that with the exception of the $\text{cg}_\delta(p_i, q_i)$ gadgets, σ is x -monotone—that is, as we walk along it, the x -coordinate never decreases. Moreover, with the exception of the γ_i portions, π' is x -monotone. Finally, observe that $\text{cg}_\delta(p_i, q_i)$ and γ_i have the same starting and ending points, and π' and σ both start at s and end at t . Thus, with the exception of the $\text{cg}_\delta(p_i, q_i)$ and γ_i portions, we can map all points from σ with a given x -coordinate to the point on π' with the same x -coordinate. It is easy to verify that this maps points between the curves that are at most δ apart. First, as π' is identical to π outside of the γ_i , and since π is Y -respecting, π' passes through s , t , and q'_i for all i . Thus, the mapping stands still on π' at these respective points as σ executes the $g_\delta(s)$, $g_\delta(t)$, and $g_\delta(q'_i)$ gadgets. The vertical distance elsewhere between the curves is at most $4M$ by Corollary 3.26, and by construction $4M \leq \delta$. $\square$

4 ALGORITHMS FOR LOWER BOUND FRÉCHET DISTANCE

In the previous section, we showed that the decision problem for $d_F^{\min}$ is hard, given an uncertain curve with line-segment-based imprecision model and a polygonal curve. Interestingly, the same problem is solvable in polynomial time for indecisive curves. This result highlights a distinction between $d_F^{\min}$ and $d_F^{\max}$ and between the different uncertainty models. To tackle $d_F^{\min}$ with general uncertain curves, we develop approximation algorithms.

4.1 Exact Solution for Indecisive Curves

The key idea is that we can use a dynamic programming approach similar to that for computing the Fréchet distance [6] and only keep track of realisations of the last indecisive point considered so far. (Note that one can also reduce the problem to the Fréchet distance between paths in *DAG complexes*, studied by Har-Peled and Raichel [38], but this yields a slower running time.) We present the approach for an indecisive and a precise curve, and then generalise it to two indecisive curves.

4.1.1 Indecisive and Precise. Consider the setting with an indecisive curve $\mathcal{U} = \langle U_1, \dots, U_m \rangle$ with m points and a precise curve $\sigma = \langle q_1, \dots, q_n \rangle$ with n points; each indecisive point has k possible realisations, $U_i = \{p_i^1, \dots, p_i^k\}$. We want to solve the decision problem ‘Is the lower bound Fréchet distance between the curves below some threshold δ ?’ so $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta$?

Consider the free-space diagram for this problem; suppose $\mathcal{U}$ is positioned along the horizontal axis and σ along the vertical axis. Just as for the precise curve Fréchet distance, we are interested

in the reachable intervals on the cell boundary, since the free space in the cell interior is convex; however, now we care about the different realisations of the points, so we get a set of reachable boundaries instead of a single cell boundary. We can adapt the standard dynamic program to deal with this problem. We propagate reachability column by column. An important aspect is that we only need to make sure that a reachable point is reachable by a monotone path in the free-space diagram induced by *some* valid realisation; we do not need to remember which one, since we never return to the previous points on the indecisive curve, and we also do not care about the realisations that yield a distance higher than δ —a significant deviation from the upper bound Fréchet distance.

First of all, define $\text{Feas}(i, \ell)$ to be the *feasibility column* for the realisation p_i^ℓ of U_i . This is a set of intervals on the vertical cell boundary line in the free-space diagram, corresponding to the subintervals of one curve within distance δ from a point on the other curve. It is computed exactly the same way as for the precise Fréchet distance—it depends on the distance between a point and a line segment and gives a single interval on each vertical cell boundary. We can compute feasibility for the right boundary of all cells in a column for a given realisation, thus obtaining $\text{Feas}(i, \ell)$.

Consider the standard dynamic program for computing the Fréchet distance on precise curves. Represent it so that it operates column by column, grouping propagation of reachable intervals between vertically aligned cells. Call that procedure $\text{Prop}(R)$, where R is the *reachability column* for point i and the result is the reachability column for point $i + 1$ on one of the curves. Again, the reachability column is a set of intervals on a vertical line, indicating the points in the free-space diagram that are reachable from the lower left corner with a monotone path.

Define $\text{Reach}(i, s)$ to be the reachability column induced by p_i^s , where a point is in a reachability interval if it can be reached by a monotone path for some realisation of the previous points. Then we compute

$$\text{Reach}(i + 1, \ell) = \text{Feas}(i + 1, \ell) \cap \bigcup_{\ell' \in [k]} \text{Prop}(\text{Reach}(i, \ell')).$$

So, we iterate over all the realisations of the previous column, thus getting precise cells, and simply propagate the reachable intervals as in the precise Fréchet distance algorithm. For the column corresponding to U_1 , we set one reachable interval of a single point at the bottom for all realisations p_1^s for which $\|p_1^s - q_1\| \leq \delta$.

We now show correctness of this approach.

LEMMA 4.1. *For all $i > 1$,*

$$\text{Reach}(i, \ell) = \left\{ y \mid \exists p_1^{\ell_1}, \dots, p_{i-1}^{\ell_{i-1}} \left[d_F \left(\left(\bigsqcup_{j \in [i-1]} p_j^{\ell_j} \right) \sqcup p_i^\ell, \sigma[1 : \lfloor y \rfloor] \sqcup \sigma(y) \right) \leq \delta \right] \right\}.$$

So, for any point inside a reachability interval, there is a realisation that defines a free-space diagram and a monotone path through that diagram to this point.

PROOF. We show this by induction on i . To compute $\text{Reach}(2, \ell)$ for any fixed $\ell \in [k]$, we start from a single point in the bottom left corner of the free space for the realisations of U_1 that are close enough to q_1 , and we propagate the reachability through the resulting precise free-space column. Clearly, the statement holds in this case; if some realisation of U_1 is too far from q_1 , then the reachability column is correctly empty.

Now assume the statement holds for $\text{Reach}(i, \ell')$ for all $\ell' \in [k]$. Note that all the values that we add to $\text{Reach}(i + 1, \ell)$ for some fixed ℓ are feasible, since we explicitly take the feasibility column and intersect it with the propagated reachability. Furthermore, any point y in $\text{Reach}(i + 1, \ell)$ comes as a result of propagation from some $\text{Reach}(i, \ell')$ for some ℓ' . So, there is at least one point y' in the reachability column i for realisation $p_i^{\ell'}$ from which there is a monotone path to y . Since we know there was a realisation up to that point of the two curves that enables a monotone path from the

start of the free space diagram to y' , and since point U_{i+1} is independent from the previous points, and since we have a fixed valid realisation for points U_i and U_{i+1} that enables the continuation of the monotone path from y' to y , we conclude that the statement holds for the column $i + 1$. $\square$

Therefore, querying the upper boundary of all reachability intervals for U_m will give us the answer to the decision problem.

Now we analyse the complexity of the reachability column. A particular right cell boundary is entirely reachable if the bottom of the cell is reachable; combined with the feasibility interval, we get one reachability interval per cell. Furthermore, if a cell is only reachable from the left, since we consider monotone paths, each realisation of the previous points induces a reachable interval of $[y', 1]$ for some $0 \leq y' \leq 1$ if you assume the boundary coordinate range to be $[0, 1]$; therefore, taking a union of such intervals still gives us at most one reachability interval per cell. So, in the worst case, we store $\Theta(mk)$ intervals. To propagate, we consider all combinations of the two successive indecisive points for all cells, yielding the running time of $\Theta(mnk^2)$.

Furthermore, observe that we can also store a realisation of the previous point on the indecisive curve with the interval that corresponds to the lowest reachable point on the current interval. If we then store all the reachability columns, we can later backtrack and find a specific curve that realises the Fréchet distance below the threshold δ . This increases the storage requirements to $\Theta(mnk)$; the running time stays the same. We summarise the results next.

THEOREM 4.2. *Given an indecisive curve $\mathcal{U} = \langle U_1, \dots, U_m \rangle$, where each indecisive point has k options, $U_i = \{p_i^1, \dots, p_i^k\}$, a precise curve $\sigma = \langle q_1, \dots, q_n \rangle$, and a threshold $\delta > 0$, we can decide if $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta$ in time $\Theta(mnk^2)$ in the worst case, using $\Theta(mk)$ space. We can also report the realisation of $\mathcal{U}$ realising the Fréchet distance at most δ , using $\Theta(mnk)$ space instead. Call the algorithm that solves the problem and reports a fitting realisation $\text{DECIDER}(\delta, \mathcal{U}, \sigma)$.*

4.1.2 Indecisive and Indecisive. Now consider the setting where instead of σ we are given curve $\mathcal{V} = \langle V_1, \dots, V_n \rangle$ with k options per indecisive point, $V_i = \{q_i^1, \dots, q_i^k\}$. We can adapt the algorithm of the previous section by propagating in column-major order, but cell by cell.

A cell boundary now depends on three indecisive points, so there are k^3 options per boundary to consider. We now store the possibilities for $m - 1$ right cell boundaries, k^3 realisations per boundary, and a single horizontal boundary, with also k^3 options. So, we use $\Theta(mk^3)$ storage.

Whenever we propagate to one further cell, we need to find the reachability for the top and the right boundary of the cell based on the left and the lower boundary of the cell. We again go over all the combinations of the realisations of the points that define the cell, yielding k^4 possible precise cells to consider. We aggregate the values as before, as for both the top and the right boundary only three points matter.

Since we solve the same problem as in the previous section and never have to revisit a previously considered point, it should be clear that this approach is correct. However, now we take $\Theta(k^4)$ time per cell, so in the worst case we need $\Theta(mnk^4)$ time to complete the propagation.

THEOREM 4.3. *Given two indecisive curves $\mathcal{U} = \langle U_1, \dots, U_m \rangle$ and $\mathcal{V} = \langle V_1, \dots, V_n \rangle$, where each indecisive point has k options, $U_i = \{p_i^1, \dots, p_i^k\}$ and $V_i = \{q_i^1, \dots, q_i^k\}$, and a threshold $\delta > 0$, we can decide if $d_F^{\min}(\mathcal{U}, \mathcal{V}) \leq \delta$ in time $\Theta(mnk^4)$ in the worst case, using $\Theta(mk^3)$ space.*

4.2 Approximation by Grids

Given a general uncertain curve $\mathcal{U}$ and a polygonal curve σ , in this section we show how to find a curve $\pi \in \mathcal{U}$ such that $d_F(\pi, \sigma) \leq (1 + \epsilon)d_F^{\min}(\mathcal{U}, \sigma)$. This is accomplished by carefully discretising the regions, in effect approximately reducing the problem to the indecisive case, for which we then can use Theorem 4.2.

For simplicity, assume the uncertain regions have constant complexity. Throughout the section, we assume $d_F^{\min}(\mathcal{U}, \sigma) > 0$, justified by the following lemma.

LEMMA 4.4. *Let $\mathcal{U}$ be an uncertain curve with m vertices and σ a polygonal curve with n vertices. Then one can determine whether $d_F^{\min}(\mathcal{U}, \sigma) = 0$ in $O(mn)$ time.*

PROOF. Observe that if for some j , σ_j lies on the segment $\sigma_{j-1}\sigma_{j+1}$, then $d_F(\sigma, \sigma') = 0$, where $\sigma' = \langle \sigma_1, \dots, \sigma_{j-1}, \sigma_{j+1}, \dots, \sigma_n \rangle$. So we can assume that no vertex of σ lies on the segment between its neighbours, as otherwise we can remove that vertex and get the same result in terms of the Fréchet distance. Thus, at every vertex σ turns, implying that if there exists $\pi \in \mathcal{U}$ such that $d_F(\pi, \sigma) = 0$, then for all j , σ_j must be aligned with some π_i .

This observation leads to a simple decision procedure. Define

$$s(j) = \{i \in [m] \mid d_F(\pi[1 : i], \sigma[1 : j]) = 0\},$$

so a set of indices on σ that yield the zero Fréchet distance between the correspondent prefix curves. Then we can go through σ one vertex at a time, maintaining $s(j)$, and ultimately $d_F^{\min}(\mathcal{U}, \sigma) = 0$ if and only if $m \in s(n)$.

Initially, $s(1) = \{i \in [m] \mid \forall k \in [1] : \sigma_1 \in U_k\}$, which is easy to test and compute. For $j > 1$, $s(j)$ can be computed from $s(j-1)$ as follows. Let $\text{Stab}_j(k)$ be the set of indices $i > k$ such that there exist points $p_{k+1}, \dots, p_{i-1}$, appearing in order along $\sigma_{j-1}\sigma_j$, where $p_\ell \in U_\ell$ for all $k < \ell < i$. (Note that we always have $k+1 \in \text{Stab}_j(k)$.) So, $\text{Stab}_j(k)$ is the set of indices i of uncertainty regions, starting from $k+1$, such that all the regions between k and i are stabbed by the segment $\sigma_{j-1}\sigma_j$ in the correct order. Then we have

$$s(j) = \{i \mid \sigma_j \in U_i \wedge i \in \text{Stab}_j(k) \text{ with } k = \max_{\ell < i} \{\ell \in s(j-1)\}\}.$$

From this definition of $s(j)$, it is easy to see that it can be computed in $O(m)$ time given $s(j-1)$, and thus the total time required is $O(mn)$. In particular, if $s(j-1)$ is non-empty, then let z be the minimum value in $s(j-1)$. We now incrementally loop over values of i , where initially $i = z+1$, and add i to $s(j)$ if $\sigma_j \in U_i$ and $i \in \text{Stab}_j(z)$. Note that in constant time per iteration, we can maintain sufficient information to determine if $i \in \text{Stab}_j(z)$, as we describe further. If at any iteration $i = z' + 1$ for $z' \in s(j-1)$, we forget $\text{Stab}_j(z)$ (as we no longer need to stab those regions) and start maintaining and checking $\text{Stab}_j(z')$.

Note that the intersection of any U_ℓ with $\sigma_{j-1}\sigma_j$ is a constant number of intervals along $\sigma_{j-1}\sigma_j$. Then $\text{Stab}_j(k)$ can be computed incrementally as follows. First, let p_{k+1} be the earliest point of $\sigma_{j-1}\sigma_j \cap U_{k+1}$. For some $i > k+1$, let p_i be the earliest point of $\sigma_{j-1}\sigma_j \cap U_i$, which is at least as far along $\sigma_{j-1}\sigma_j$ as p_{i-1} (if it exists). If such p_i exists, then we know that $i \in \text{Stab}_j(k)$. Maintaining this information indeed takes constant time per iteration. $\square$

4.2.1 Decision Procedure. An algorithm is a $(1 + \varepsilon)$ -decider for Problem 3.22, if when $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta$, it returns a curve $\pi \in \mathcal{U}$ such that $d_F(\pi, \sigma) \leq (1 + \varepsilon)\delta$, and when $d_F^{\min}(\mathcal{U}, \sigma) > (1 + \varepsilon)\delta$, it returns FALSE (in between either answer is allowed). In this section, we present a $(1 + \varepsilon)$ -decider for Problem 3.22. We make use of the following standard observation.

OBSERVATION 4.5. *Given a curve $\pi = \langle \pi_1, \dots, \pi_n \rangle$, call a curve $\sigma = \langle \sigma_1, \dots, \sigma_n \rangle$ an r -perturbation of π if $\|\pi_i - \sigma_i\| \leq r$ for all $i \in [n]$. Since $\|\pi_i - \sigma_i\|, \|\pi_{i+1} - \sigma_{i+1}\| \leq r$, all points of the segment $\sigma_i\sigma_{i+1}$ are within distance r of $\pi_i\pi_{i+1}$. For segments, this implies that $d_F(\pi_i\pi_{i+1}, \sigma_i\sigma_{i+1}) \leq r$, which implies that $d_F(\pi, \sigma) \leq r$ by composing the mappings for all i .*

The high-level idea is to replace $\mathcal{U}$ with the set of grid points it intersects; however, as our uncertainty regions may avoid the grid points, we need to include a slightly larger set of points.

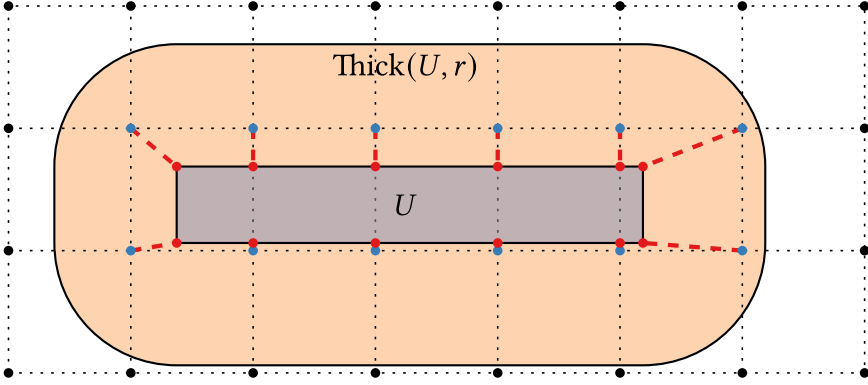


Fig. 11. An example of the sets from Definition 4.6. The region U is shown in blue, and $\text{Thick}(U, r)$ is in orange. The grid points of $GT_r(U)$ are in blue, and the corresponding set of expanded r -grid points $EG_r(U)$ are in red.

Definition 4.6. Let U be a compact subset of $\mathbb{R}^d$. We now define the set of points $EG_r(U)$ which we call the *expanded r -grid points* of U (Figure 11). Let $B(\sqrt{d}r)$ denote the ball of radius $\sqrt{d}r$, centred at the origin. Let $\text{Thick}(U, r) = U \oplus B(\sqrt{d}r)$, where $\oplus$ denotes the Minkowski sum. Let G_r denote the regular grid of side length r , and let $GT_r(U)$ denote the subset of grid vertices from G_r that fall in $\text{Thick}(U, r)$. Finally, we define

$$EG_r(U) = \{p \mid p = \arg \min_{q \in U} \|q - x\| \text{ for } x \in GT_r(U)\}.$$

In the following observation, we use the terms defined previously.

OBSERVATION 4.7. For any $x \in U$, there is a point $p \in EG_r(U)$ such that $\|p - x\| \leq 2\sqrt{d}r$.

PROOF. For any point $x \in U$, let g be its nearest grid point in G_r . Since $\|x - g\| \leq \sqrt{d}r$, we know that $g \in \text{Thick}(U, r) = U \oplus B(\sqrt{d}r)$. So let p be the point in U which is closest to g ; thus, $p \in EG_r(U)$. Therefore, $\|x - p\| \leq \|x - g\| + \|g - p\| \leq \sqrt{d}r + \sqrt{d}r = 2\sqrt{d}r$. $\square$

LEMMA 4.8. There is a $(1 + \varepsilon)$ -decider for Problem 3.22 in d dimensions with running time $O(mn \cdot (1 + (\Delta/\varepsilon\delta)^{2d}))$, for $0 < \varepsilon \leq 1$ and constant d , where $\Delta = \max_{i \in [m]} \text{diam}(U_i)$ is the maximum diameter of an uncertain region.

PROOF. It helps with the analysis if $\varepsilon\delta < \Delta$. To ensure this, we first do the following. Select an arbitrary curve $x \in \mathcal{U}$. Now using the standard $O(mn)$ time exact decider for the Fréchet distance [6], query whether $d_F(x, \sigma) \leq (1 + \varepsilon)\delta$. If the decider returns $d_F(x, \sigma) \leq (1 + \varepsilon)\delta$, then we can return x as our solution. Otherwise, $d_F(x, \sigma) > (1 + \varepsilon)\delta$, and we next query whether $d_F(x, \sigma) \leq \Delta + \delta$. By Observation 4.5 and the triangle inequality, $d_F(x, \sigma) \leq \Delta + d_F^{\min}(\mathcal{U}, \sigma)$. Thus, if the decider returns $\Delta + \delta < d_F(x, \sigma)$, then $\delta < d_F^{\min}(\mathcal{U}, \sigma)$, and so we return FALSE. Otherwise, the two decider calls tell us that $(1 + \varepsilon)\delta < d_F(x, \sigma) \leq \Delta + \delta$, implying $\varepsilon\delta < \Delta$.

Let $r = \varepsilon\delta/2\sqrt{d}$, and for any U_i of $\mathcal{U}$, let $E_i = EG_r(U_i)$ denote the expanded r -grid points of U_i , as defined in Definition 4.6. Consider the indecisive curve $\mathcal{U}' = \langle E_1, \dots, E_m \rangle$. We call the algorithm $\text{DECIDER}((1 + \varepsilon)\delta, \mathcal{U}', \sigma)$ of Theorem 4.2 and return whatever it returns—that is, if it returns a curve, then we return that curve, and if it returns that $d_F^{\min}(\mathcal{U}', \sigma) > (1 + \varepsilon)\delta$, then we return that $d_F^{\min}(\mathcal{U}, \sigma) > (1 + \varepsilon)\delta$.

First, observe that $E_i \subseteq U_i$, and thus $d_F^{\min}(\mathcal{U}, \sigma) \leq d_F^{\min}(\mathcal{U}', \sigma)$. So if $d_F^{\min}(\mathcal{U}, \sigma) > (1 + \varepsilon)\delta$, then the decider must return $d_F^{\min}(\mathcal{U}', \sigma) > (1 + \varepsilon)\delta$, as desired. Now suppose that $d_F^{\min}(\mathcal{U}, \sigma) \leq$

δ . In this case, we argue that our algorithm outputs a curve $\pi' \in \mathcal{U}$ such that $d_F(\pi', \sigma) \leq (1 + \varepsilon)\delta$. It suffices to argue that there exists some curve $\pi' \in \mathcal{U}'$ such that $d_F(\pi', \sigma) \leq (1 + \varepsilon)\delta$, as then Theorem 4.2 guarantees the decider outputs a curve (which is in $\text{Real}(\mathcal{U})$, as it is a superset of $\text{Real}(\mathcal{U}')$). So let $\pi = \langle \pi_1, \dots, \pi_m \rangle$ be the curve in $\text{Real}(\mathcal{U})$ realising the lower bound Fréchet distance to σ —that is, $d_F(\pi, \sigma) = d_F^{\min}(\mathcal{U}, \sigma)$. Let $\pi' = \langle \pi'_1, \dots, \pi'_m \rangle$ be the curve such that $\pi'_i = \min_{x \in E_i} \|x - \pi_i\|$. Note that by Observation 4.7, we have $\|\pi_i - \pi'_i\| \leq 2\sqrt{d}r$ for all i . Thus, π' is a $2\sqrt{d}r$ -perturbation of π as described in Observation 4.5, and so $d_F(\pi, \pi') \leq 2\sqrt{d}r = \varepsilon\delta$. As the Fréchet distance satisfies the triangle inequality, we therefore have $d_F(\pi', \sigma) \leq d_F(\pi, \sigma) + d_F(\pi, \pi') \leq \delta + \varepsilon\delta = (1 + \varepsilon)\delta$. Thus, as $\pi' \in \mathcal{U}'$, when our algorithm calls $\text{DECIDER}((1 + \varepsilon)\delta, \mathcal{U}', \sigma)$, it returns a curve.

For the running time, recall we first spent $O(mn)$ time to ensure $\varepsilon\delta < \Delta$, in which case we must bound the number of points in each E_i . By Definition 4.6, for all i , the number of points in E_i is bounded by the number of grid points in the region $\text{Thick}(U_i, r)$. This region is the Minkowski sum of a compact set of diameter at most Δ with a radius $\sqrt{d}r$ ball, so its diameter is at most $\Delta + 2\sqrt{d}r$. Recall that d is a constant; thus, the number of grid points and hence $|E_i|$ is

$$O\left(\left(\frac{\Delta + 2\sqrt{d}r}{r}\right)^d\right) = O\left(\left(\frac{2\sqrt{d}\Delta}{\varepsilon\delta} + 2\sqrt{d}\right)^d\right) = O\left(\left(\frac{\Delta}{\varepsilon\delta} + 1\right)^d\right) = O\left(\left(\frac{\Delta}{\varepsilon\delta}\right)^d\right).$$

Thus, by Theorem 4.2, the call to DECIDER takes time $O(mn(\Delta/\varepsilon\delta)^{2d})$, which bounds the total time of our algorithm. $\square$

4.2.2 Optimisation.

THEOREM 4.9. *Let $\mathcal{U}$ be an uncertain curve with m vertices, σ a polygonal curve with n vertices, and $\delta = d_F^{\min}(\mathcal{U}, \sigma)$. Then for any $0 < \varepsilon \leq 1$, there is an algorithm which returns a curve $\pi \in \mathcal{U}$ such that $d_F(\pi, \sigma) \leq (1 + \varepsilon)\delta$, whose running time is $O(mn(\log(mn) + (\Delta/\varepsilon\delta)^{2d}))$ for constant d , where $\Delta = \max_{i \in [m]} \text{diam}(U_i)$ is the maximum diameter of an uncertain region.*

PROOF. Fix an arbitrary curve $x \in \mathcal{U}$. First, we compute the Fréchet distance between x and σ . If $d_F(x, \sigma) \geq \Delta + \Delta/\varepsilon$, then we return x as our solution. Intuitively, this means that the Fréchet distance is large when compared to the diameter of the uncertain regions, and so any realisation we can pick works as a $(1 + \varepsilon)$ -approximation. To see why this is valid, let $\hat{\pi} \in \mathcal{U}$ be an optimal solution—that is, $d_F(\hat{\pi}, \sigma) = d_F^{\min}(\mathcal{U}, \sigma)$. Note that x is a Δ -perturbation of $\hat{\pi}$, and thus by the triangle inequality and Observation 4.5,

$$d_F(x, \sigma) \leq d_F(x, \hat{\pi}) + d_F(\hat{\pi}, \sigma) \leq \Delta + d_F(\hat{\pi}, \sigma).$$

If $\Delta + \Delta/\varepsilon \leq d_F(x, \sigma)$, then plugging in the preceding inequality implies that $\Delta \leq \varepsilon \cdot d_F(\hat{\pi}, \sigma)$, which in turn implies that

$$d_F(x, \sigma) \leq \Delta + d_F(\hat{\pi}, \sigma) \leq (1 + \varepsilon) \cdot d_F(\hat{\pi}, \sigma).$$

So suppose that $d_F(x, \sigma) < (1 + 1/\varepsilon)\Delta$, in which case

$$d_F^{\min}(\mathcal{U}, \sigma) = d_F(\hat{\pi}, \sigma) \leq d_F(x, \sigma) + d_F(\hat{\pi}, x) < \left(1 + \frac{1}{\varepsilon}\right)\Delta + \Delta = \left(2 + \frac{1}{\varepsilon}\right)\Delta = \gamma.$$

Let $\text{GRIDDECIDER}(\mathcal{U}, \sigma, \varepsilon', \delta)$ denote the $(1 + \varepsilon')$ -decider of Lemma 4.8, which correctly returns either FALSE (which implies $d_F^{\min}(\mathcal{U}, \sigma) > \delta$) or a curve in $\text{Real}(\mathcal{U})$ with the Fréchet distance at most $(1 + \varepsilon')\delta$ to σ . We perform a decreasing exponential search using GRIDDECIDER . Specifically, starting at $i = 0$, we call $\text{GRIDDECIDER}(\mathcal{U}, \sigma, \varepsilon/4, \gamma/(1 + \varepsilon/4)^i)$. If GRIDDECIDER returns a curve (i.e., TRUE), we increment i by 1 and repeat, and otherwise if GRIDDECIDER outputs FALSE, we return

the curve from iteration $i - 1$. (Note that GRIDDECIDER cannot return FALSE when $i = 0$, as this would imply that $d_F^{\min}(\mathcal{U}, \sigma) > \gamma$.)

Let j denote the index when the algorithm stops. So we know that GRIDDECIDER($\mathcal{U}, \sigma, \varepsilon/4, \gamma/(1 + \varepsilon/4)^j$) returned FALSE, and GRIDDECIDER($\mathcal{U}, \sigma, \varepsilon/4, \gamma/(1 + \varepsilon/4)^{j-1}$) returned a curve $\pi \subseteq \mathcal{U}$ such that $d_F(\pi, \sigma) \leq (1 + \varepsilon/4) \cdot \gamma/(1 + \varepsilon/4)^{j-1}$. Therefore,

$$\frac{\gamma}{(1 + \varepsilon/4)^j} < d_F^{\min}(\mathcal{U}, \sigma) \leq d_F(\pi, \sigma) \leq (1 + \varepsilon/4) \frac{\gamma}{(1 + \varepsilon/4)^{j-1}} = \frac{\gamma}{(1 + \varepsilon/4)^{j-2}},$$

which implies that

$$d_F(\pi, \sigma) \leq \left(1 + \frac{\varepsilon}{4}\right)^2 d_F^{\min}(\mathcal{U}, \sigma) = \left(1 + \frac{\varepsilon}{2} + \frac{\varepsilon^2}{16}\right) \cdot d_F^{\min}(\mathcal{U}, \sigma) < (1 + \varepsilon) \cdot d_F^{\min}(\mathcal{U}, \sigma).$$

As for the running time, by Lemma 4.8, the time for the i -th call to GRIDDECIDER is

$$O\left(mn \left(\frac{(1 + \varepsilon/4)^i \Delta}{\varepsilon \gamma}\right)^{2d}\right) = O\left(mn \left(\frac{(1 + \varepsilon/4)^i \Delta}{\varepsilon(2 + 1/\varepsilon)\Delta}\right)^{2d}\right) = O\left(mn \left(1 + \frac{\varepsilon}{4}\right)^{2di}\right).$$

Recall that $\delta = d_F^{\min}(\mathcal{U}, \sigma)$ and j is the index the last time GRIDDECIDER is called. By the preceding argument, $\delta \leq \gamma/(1 + \varepsilon/4)^{j-2}$, which implies that $j - 2 \leq \log_{1+\varepsilon/4}(\gamma/\delta)$. Recall that d is a constant; as GRIDDECIDER is called $j + 1$ times, and the running times for the calls to GRIDDECIDER form an increasing geometric series, the total time for all calls to GRIDDECIDER is

$$\begin{aligned} & O\left(mn \left(1 + \frac{\varepsilon}{4}\right)^{2d \cdot \left(3 + \log_{1+\frac{\varepsilon}{4}}(\gamma/\delta)\right)}\right) = O\left(mn \left(1 + \frac{\varepsilon}{4}\right)^{6d} \left(1 + \frac{\varepsilon}{4}\right)^{2d \cdot \log_{1+\frac{\varepsilon}{4}}(\gamma/\delta)}\right) \\ &= O\left(mn \left(1 + \frac{\varepsilon}{4}\right)^{2d \cdot \log_{1+\frac{\varepsilon}{4}}(\gamma/\delta)}\right) = O\left(mn \left(\frac{\gamma}{\delta}\right)^{2d \cdot \log_{1+\frac{\varepsilon}{4}}(1+\varepsilon/4)}\right) \\ &= O\left(mn \left(\frac{\gamma}{\delta}\right)^{2d}\right) = O\left(mn \left(\frac{(2 + 1/\varepsilon)\Delta}{\delta}\right)^{2d}\right) = O\left(mn \left(\frac{\Delta}{\varepsilon \delta}\right)^{2d}\right). \end{aligned}$$

As it takes $O(mn \log(mn))$ time to initially compute $d_F(x, \sigma)$ using the algorithm of Alt and Godau [6], the total running time is $O(mn(\log(mn) + (\Delta/\varepsilon \delta)^{2d}))$. $\square$

If the polygonal curve σ is replaced with an uncertain curve $\mathcal{V}$, it is easy to see that by discretising both $\mathcal{U}$ and $\mathcal{V}$, the same analysis gives an algorithm to compute $d_F^{\min}(\mathcal{U}, \mathcal{V})$. The only difference now is that we must use Theorem 4.3 instead of Theorem 4.2, yielding the following.

COROLLARY 4.10. *Let $\mathcal{U}$ and $\mathcal{V}$ be uncertain curves with m and n vertices, respectively, and $\delta = d_F^{\min}(\mathcal{U}, \mathcal{V})$. Then for any $0 < \varepsilon \leq 1$, there is an algorithm returning curves $\pi \subseteq \mathcal{U}$ and $\sigma \subseteq \mathcal{V}$ such that $d_F(\pi, \sigma) \leq (1 + \varepsilon)\delta$, whose running time is $O(mn(\log(mn) + (\Delta/\varepsilon \delta)^{4d}))$ for constant d , where Δ is the maximum diameter of an uncertain region.*

4.3 Greedy Algorithm

Here we argue that there is a simple 3-decider for Problem 3.22, running in near-linear time in the plane. Roughly speaking, the idea is to greedily and iteratively pick $\pi_i \in U_i$ so as to allow us to get as far as possible along σ . Without any assumptions on $\mathcal{U}$, this greedy procedure may walk too far ahead and get stuck. Thus, in this section, we assume that consecutive U_i are separated, so as to ensure optimal solutions do not lag too far behind. Here we also assume that U_i are convex (i.e., imprecise) and have constant complexity, as it simplifies certain definitions. Throughout this section, let $\mathcal{U} = \langle U_1, \dots, U_m \rangle$ be an uncertain curve and let $\sigma = \langle \sigma_1, \dots, \sigma_n \rangle$ be a polygonal curve.

Definition 4.11. Call $\mathcal{U}$ γ -separated if for all $i \in [m - 1]$, $\|U_i - U_{i+1}\| > \gamma$ and each U_i is convex. Define an r -visit of U_i to be any maximal-length contiguous portion of $\sigma \cap (U_i \oplus B(2r))$ which

intersects $U_i \oplus B(r)$, where $\oplus$ denotes the Minkowski sum. If $\mathcal{U}$ is γ -separated for $\gamma \geq 4r$, then any r -visit of U_i is disjoint from any r -visit of U_j for $i \neq j$, in which case define the *true r -visit* of U_i to be the first r -visit of U_i which occurs after the true r -visit of U_{i-1} . (For U_1 , it is the first r -visit.)

LEMMA 4.12. *If $\mathcal{U}$ is γ -separated for $\gamma \geq 4r$, then for any curve $\pi \in \mathcal{U}$ and any reparametrisations f and g such that $\text{width}_{f,g}(\pi, \sigma) \leq r$, π_i must map to a point on the true r -visit of U_i for all i .*

PROOF. First, note that since $\text{width}_{f,g}(\pi, \sigma) \leq r$, π_i must map to a point in an r -visit of U_i , and thus we only need to prove it is the true r -visit.

We prove the claim by induction on i . For $i = 1$, the claim holds, as π_1 must map to σ_1 , and σ_1 is in the first r -visit of U_1 , which is its true r -visit.

Now suppose the claim holds for $i - 1$. π_i must map to a point on an r -visit of U_i , and by the induction hypothesis, this visit must happen after the true r -visit of U_{i-1} on σ . Moreover, as $\mathcal{U}$ is $4r$ -separated, the first point in $U_i \oplus B(r)$ of the first r -visit of U_i that occurs after the true r -visit of U_{i-1} (i.e., true r -visit of U_i) must map to a point x on $\pi_{i-1}\pi_i$. Note, however, that as both x and π_i map to points in $U_i \oplus B(r)$, the portion of σ that the segment $x\pi_i$ maps to must lie within $U_i \oplus B(2r)$ (i.e., the same r -visit). Therefore, all of $x\pi_i$ is mapped to the true r -visit of U_i , completing the proof. $\square$

For two points α and β on σ , let $\alpha \leq \beta$ denote that α occurs before β , and for any points $\alpha \leq \beta$, let $\sigma(\alpha, \beta)$ denote the subcurve between α and β .

Definition 4.13. The δ -greedy sequence of σ with respect to $\mathcal{U}$, denoted $\text{gs}(\mathcal{U}, \sigma, \delta)$, is the longest possible sequence $\alpha = \langle \alpha_1, \dots, \alpha_k \rangle$ of points on σ , where $\alpha_1 = \sigma_1$, and for any $i > 1$, α_i is the point furthest along σ such that $\|\alpha_i - U_i\| \leq \delta$ and $d_F(\alpha_{i-1}\alpha_i, \sigma(\alpha_{i-1}, \alpha_i)) \leq 2\delta$.

OBSERVATION 4.14. *For any $i \leq k$, let $\alpha^i = \langle \alpha_1, \dots, \alpha_i \rangle$ be the i -th prefix of $\text{gs}(\mathcal{U}, \sigma, \delta)$. Then $d_F(\alpha^i, \sigma(\alpha_1, \alpha_i)) \leq 2\delta$, and $\alpha^i \in \mathcal{U}^i \oplus B(\delta)$, where $\mathcal{U}^i \oplus B(\delta) = \langle U_1 \oplus B(\delta), \dots, U_i \oplus B(\delta) \rangle$.*

LEMMA 4.15. *If $\mathcal{U}$ is 10δ -separated and $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta$, then $\text{gs}(\mathcal{U}, \sigma, \delta)$ has length m and $\alpha_m = \sigma_n$.*

PROOF. Let $\text{gs}(\mathcal{U}, \sigma, \delta) = \alpha = \langle \alpha_1, \dots, \alpha_k \rangle$. Let $\text{opt} = \langle \text{opt}_1, \dots, \text{opt}_m \rangle$ be any curve in $\text{Real}(\mathcal{U})$ such that $d_F(\text{opt}, \sigma) = d_F^{\min}(\mathcal{U}, \sigma)$. Throughout this proof, we fix a mapping realising $d_F(\text{opt}, \sigma)$ and let β_i be the point on σ which opt_i maps to under this mapping. For the curve α , we fix the mapping which is the composition of the maps realising $d_F(\alpha_{i-1}\alpha_i, \sigma(\alpha_{i-1}, \alpha_i)) \leq 2\delta$, and, in particular, α_i on α maps to α_i on σ .

We prove by induction that for $i \leq m$, α_i exists and $\beta_i \leq \alpha_i$. For $i = 1$, we have $\alpha_1 = \beta_1 = \sigma_1$. So assume that α_{i-1} exists. By Observation 4.14, $\alpha^{i-1} \in \mathcal{U}^{i-1} \oplus B(\delta)$, and, moreover, $d_F(\sigma(\alpha_1, \alpha_{i-1}), \alpha^{i-1}) \leq 2\delta$. Since $\mathcal{U}$ is 10δ -separated, $\mathcal{U}^{i-1} \oplus B(\delta)$ is 8δ -separated, and thus by Lemma 4.12, α_{i-1} is on the true 2δ -visit of $U_{i-1} \oplus B(\delta)$ by the prefix curve $\sigma(\alpha_1, \alpha_{i-1})$. Observe that the true 2δ -visit of $U_{i-1} \oplus B(\delta)$ by the prefix curve $\sigma(\alpha_1, \alpha_{i-1})$ is a subset of the true 2δ -visit of $U_{i-1} \oplus B(\delta)$ by σ , and thus α_{i-1} is on the true 2δ -visit of $U_{i-1} \oplus B(\delta)$ by σ . We also have that $\text{opt} \in \mathcal{U} \oplus B(\delta)$, as $U_j \subset U_j \oplus B(\delta)$ for all j , so by Lemma 4.12, β_{i-1} and β_i are on the true 2δ -visit of $U_{i-1} \oplus B(\delta)$ and $U_i \oplus B(\delta)$. In particular, this implies that $\beta_{i-1} \leq \alpha_{i-1} \leq \beta_i$, as the true 2δ -visits of $U_{i-1} \oplus B(\delta)$ and $U_i \oplus B(\delta)$ are disjoint. Thus, some point x on the segment $\text{opt}_{i-1}\text{opt}_i$ must map to α_{i-1} . Note that $d_F(x\text{opt}_i, \sigma(\alpha_{i-1}, \beta_i)) \leq \delta$. As $\|x - \alpha_{i-1}\| \leq \delta$, $d_F(x\text{opt}_i, \alpha_{i-1}\text{opt}_i) \leq \delta$, and so by the triangle inequality for the Fréchet distance, $d_F(\alpha_{i-1}\text{opt}_i, \sigma(\alpha_{i-1}, \beta_i)) \leq 2\delta$. Since $\|\beta_i - \text{opt}_i\| \leq \delta$, β_i is a possible choice for α_i , and thus α_i exists and $\beta_i \leq \alpha_i$. Finally, since α_i exists for all $i \leq m$, $\alpha = \text{gs}(\mathcal{U}, \sigma, \delta)$ has length m , and moreover, since $\beta_m \leq \alpha_m$ and $\beta_m = \sigma_n$, we conclude that $\alpha_m = \sigma_n$. $\square$

The following lemma is the only place where we require the points to be in $\mathbb{R}^2$. The proof uses a result from Guibas et al. [37].

LEMMA 4.16. *For $\mathcal{U}$ and σ in $\mathbb{R}^2$, where $\mathcal{U}$ is 10δ -separated, $\text{gs}(\mathcal{U}, \sigma, \delta)$ is computable in time $O(m + n \log n)$.*

PROOF. Given α_i from $\text{gs}(\mathcal{U}, \sigma, \delta)$, we describe how to compute α_{i+1} , if it exists. Let σ_j be the smallest-index vertex such that $\alpha_i < \sigma_j$. Let $\langle D_j, \dots, D_n \rangle$ be the sequence of 2δ -radius disks, where D_l is centred at σ_l . Observe that for α_{i+1} to be able to lie on $\sigma_z \sigma_{z+1}$, for any $z \geq j$, we first require that $d_F(\alpha_i \alpha_{i+1}, \sigma(\alpha_i, \alpha_{i+1})) \leq 2\delta$, which occurs if and only if there exist points $p_j, \dots, p_z$ that appear in order along $\alpha_i \alpha_{i+1}$ such that $p_l \in D_l$. Clearly, such points are necessary, but they are also sufficient, as $d_F(p_l p_{l+1}, \sigma_l \sigma_{l+1}) \leq 2\delta$. (As α_i and α_{i+1} lie on σ , the same holds for $\alpha_i \sigma_j$ and $\sigma_z \alpha_{i+1}$.) $\text{gs}(\mathcal{U}, \sigma, \delta)$ also requires that α_{i+1} lie within distance δ of U_{i+1} . This is equivalent to requiring that $\sigma_z \sigma_{z+1}$ intersects $U_{i+1} \oplus B(\delta)$. As both $\sigma_z \sigma_{z+1}$ and $U_{i+1} \oplus B(\delta)$ are convex regions, their intersection is convex—that is, a single subsegment of $\sigma_z \sigma_{z+1}$. Let $S_{i+1}(z)$ denote this segment, which we can compute in constant time, as U_{i+1} is a constant-complexity convex region. Note that α_{i+1} may lie on the same segment of σ as α_i (i.e., $z = j - 1$), which is an easier case, as no disks need to be intersected and $d_F(\alpha_i \alpha_{i+1}, \sigma(\alpha_i, \alpha_{i+1})) \leq 2\delta$ holds.

Given a sequence of k equal-radius disks $\langle D_1, \dots, D_k \rangle$, say that a line ℓ stabs the disks if for all $j \leq k$, there exists a point $p_j \in \ell \cap D_j$ such that the p_j appear in order along ℓ . Guibas et al. [37] give an $O(k \log k)$ -time algorithm that determines the set of all stabbing lines. As follows from the description of our problem, their algorithm can be used to determine α_{i+1} given α_i by restricting the stabbing line to first pass through α_i and requiring it to intersect $S_{i+1}(k)$ at the end.

We now sketch the necessary changes. Their algorithm inserts the disks in order, maintaining three objects—the support hull, the limiting lines, and the line stabbing wedge. The support hull consists of a pair of upper and lower concave chains that all stabbers must pass between, and the limiting lines represent the largest and the smallest slope stabbers. The wedge is the set of all points p such that there is a stabber that passes through p after passing through the required points from the disks. To modify their approach for our setting, we require the stabber to initially pass through α_i . This actually simplifies the problem by joining and collapsing the chains of the support hull,³ and thus we can focus on the wedge. After j insertions, the wedge boundary consists of $O(j)$ pieces from the disks, flanked by the limiting lines. These ordered boundary pieces are stored in a binary tree to facilitate logarithmic time updates when a new disk is inserted, and we can simply reuse this structure to determine the intersection of the wedge with $S_{i+1}(j)$.

By Definition 4.13, the line segment $\sigma_z \sigma_{z+1}$ that α_{i+1} lies on must have z be as large as possible. Thus, we run the preceding incremental procedure, where in the j -th round we check for intersection with $S_{i+1}(j)$. If no such intersection is found before we reach the end of σ or the wedge becomes empty, then α_{i+1} does not exist. Otherwise, α_{i+1} is defined. However, the rounds which have intersection with $S_{i+1}(j)$ need not be contiguous; thus, care is needed to determine the last such intersection efficiently.

Let k be the largest index such that α_k is defined. By Observation 4.14, for any $i \leq k$, we have $d_F(\alpha^i, \sigma(\alpha_1, \alpha_i)) \leq 2\delta$ and $\alpha^i \in \mathcal{U}^i \oplus B(\delta)$. Since $\mathcal{U}$ is 10δ -separated, $\mathcal{U}^i \oplus B(\delta)$ is 8δ -separated, and so by Lemma 4.12, α_i must be in the true 2δ -visit of $U_i \oplus B(\delta)$ by $\sigma(\alpha_1, \alpha_k)$. Thus, when computing α_i , we only need to consider vertices from σ which occur after α_{i-1} and before the end of the true 2δ -visit of $U_i \oplus B(\delta)$. If n_i is the number of such vertices, it therefore takes $O(1 + n_i \log n_i)$ time to compute α_i with the preceding algorithm. Moreover, as the true 2δ -visits for $U_i \oplus B(\delta)$ and

³Alternatively, one can enforce the condition by defining an initial zero-radius disk D_0 at α_i , and indeed the referenced work [37] considers stabbers for more general collections of convex objects.

$U_j \oplus B(\delta)$ for $i \neq j \leq k$ are disjoint, any vertex of σ contributes to at most two counts n_i , as we have $\alpha_j \in U_j \oplus B(\delta)$, and we may process vertices from α_j to the end of $U_j \oplus B(\delta)$ twice; thus, $\sum_i n_i \leq 2n$. Therefore, the total running time is $O(m + n \log n) + \sum_{i=1}^k O(1 + n_i \log n_i) = O(m + n \log n)$, where the leading $O(m + n \log n)$ term accounts for the time to determine if α_{k+1} does not exist for $k < m$. $\square$

THEOREM 4.17. *Let $\mathcal{U}$ be $10r$ -separated for some $r > 0$. There is a 3-decider for Problem 3.22 in the plane with the running time $O(m + n \log n)$ that works for any query value $0 < \delta \leq r$.*

PROOF. Compute $\text{gs}(\mathcal{U}, \sigma, \delta)$. If it has length m , then let $\pi = \langle \pi_1, \dots, \pi_m \rangle$ be any curve in $\text{Real}(\mathcal{U})$ such that $\|\pi_i - \alpha_i\| \leq \delta$ for all i . If this occurs and if $\alpha_m = \sigma_n$, we output π as our solution, and otherwise we output FALSE. Thus, the running time follows from Lemma 4.16.

Observe that if we output a curve π , then $d_F(\pi, \sigma) \leq 3\delta$, using the triangle inequality:

$$d_F(\pi, \sigma) \leq d_F(\pi, \alpha) + d_F(\alpha, \sigma) \leq \delta + 2\delta = 3\delta.$$

Thus, we only need to argue that when $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta$, a curve is produced, which is immediate from Lemma 4.15. $\square$

It is also possible to turn this procedure into a 9-approximation algorithm for $d_F^{\min}$. Suppose we are given a $10r$ -separated uncertain curve. We can use decreasing exponential search with a factor of 3, starting with $\delta = r$. Suppose that for $\delta = r$, we get TRUE; eventually, we switch to FALSE. Let the last TRUE value be x ; then $3x$ must be TRUE, and $x/3$ and $x/9$ must be FALSE. Note that at most one value of δ can fall into the interval with the uncertain answer of the 3-decider. Then we know that $d_F^{\min}(\mathcal{U}, \sigma) \leq 3x$ and $d_F^{\min}(\mathcal{U}, \sigma) > 3 \cdot x/9 = x/3$. Let $\delta' = 3x$ be the returned distance, then $d_F^{\min}(\mathcal{U}, \sigma) \leq \delta' < 9d_F^{\min}(\mathcal{U}, \sigma)$, so δ' is a 9-approximation to the lower bound Fréchet distance.

5 ALGORITHMS FOR UPPER BOUND AND EXPECTED FRÉCHET DISTANCE

As shown in Section 3.1, finding the upper bound and the expected discrete and continuous Fréchet distance is hard even for simple uncertainty models. However, restricting the possible couplings or alignments between the curves makes the problem solvable in polynomial time. In this section, we use *indecisive* curves. Define a Sakoe–Chiba time band [50] in terms of reparametrisations of the curves: for a band of width w and all $t \in [0, 1]$, if $\phi_1(t) = x$, then $\phi_2(t) \in [x - w, x + w]$. In the discrete case, we can only couple point i on one curve to points $i \pm w$ on the other curve.

5.1 Upper Bound Discrete Fréchet Distance: Precise and Indecisive

First of all, let us discuss a simple setting. Suppose we are given a curve $\sigma = \langle q_1, \dots, q_n \rangle$ of n precise points and $\mathcal{U} = \langle U_1, \dots, U_n \rangle$ of n indecisive points, each of them having ℓ options, so for all $i \in [n]$ we have $U_i = \{p_i^1, \dots, p_i^\ell\}$. We would like to answer the following decision problem: ‘If we restrict the couplings to a Sakoe–Chiba band of width w , is it true that $d_{\text{df}}^{\max}(\mathcal{U}, \sigma) \leq \delta$ for some given threshold $\delta > 0$?’ So, we want to solve the decision problem for the upper bound discrete Fréchet distance between a precise and an indecisive curve.

In a fully precise setting, the discrete Fréchet distance can be computed using dynamic programming [28]. We create a table where the rows correspond to vertices of one curve, say σ , and columns correspond to vertices of the other curve, say π . Each table entry (i, j) then contains a TRUE or FALSE value indicating if there is a coupling between $\pi[1 : i]$ and $\sigma[1 : j]$ with maximum distance at most δ . We use a similar approach.

Suppose we position $\mathcal{U}$ to go horizontally along the table and σ to go vertically. Consider an arbitrary column in the table, and suppose that we fix the realisation of $\mathcal{U}$ up to the previous column. Then we can simply consider the new column ℓ times, each time picking a different realisation for the new point on $\mathcal{U}$, and compute the resulting reachability. As we do this for the entire

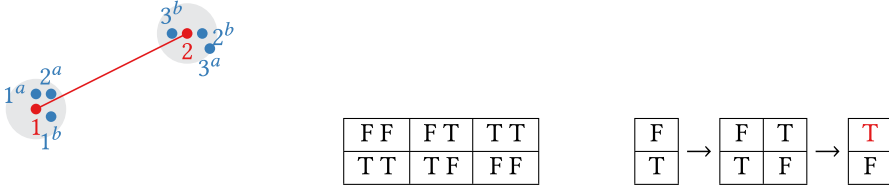


Fig. 12. Left: An indecisive and a precise curve. Middle: Distance matrix. ‘T T’ in the bottom left cell means $\|1 - 1^a\| \leq \delta$ and $\|1 - 1^b\| \leq \delta$. Right: Computing the reachability matrix, column by column. Note two reachability vectors for the second column.

column at once, we can ensure consistency of our choice of realisation. This procedure will give us a set of binary reachability vectors for the new column, each vector corresponding to a realisation. The *reachability vector* is a Boolean vector that, for the cell (i, j) of the table, states whether for a particular realisation π of $\mathcal{U}[1 : i]$ the discrete Fréchet distance between π and $\sigma[1 : j]$ is below some threshold δ .

An important observation is that we do not need to distinguish between the realisations that give the same reachability vector: once we start filling out the next column, all we care about is the existence of some realisation leading to that particular reachability vector. So, we can keep a *set* of binary vectors corresponding to reachability in the column.

This procedure was suggested for a specific realisation. However, we can also repeat this for each previous reachability vector, only keeping the unique results. As all the realisation choices happen along $\mathcal{U}$, by treating the table column-by-column we ensure that we do not have issues with inconsistent choices. Therefore, repeating this procedure n times, we fill out the last column of the table. At that point, if any vector from the last column has FALSE in the top cell, then there is some realisation $\pi \in \mathcal{U}$ such that $d_{df}(\pi, \sigma) > \delta$, and hence $d_{df}^{\max}(\mathcal{U}, \sigma) > \delta$.

In more detail, we use two tables: the distance matrix D , where cell (i, k, j) is TRUE if and only if $\|p_i^k - q_j\| \leq \delta$, and the dynamic program, referred to as the reachability matrix R . First of all, we initialise the distance matrix D and the reachability of the first column for all possible locations of U_1 . Then we fill out R column-by-column. We take the reachability of the previous column and note that any cell can be reached either with a horizontal step or with a diagonal step. We need to consider various extensions of the curve $\mathcal{U}$ with one of the ℓ realisations of the current point; the distance matrix should allow the specific coupling. Assume we find that a certain cell is reachable; if allowed by the distance matrix, we can then go upwards, marking the cells above the current cell reachable, even if they are not directly reachable with a horizontal or a diagonal step. Then we just remember the newly computed vector; we make sure to only add distinct vectors. The computation is illustrated in Figure 12; the pseudocode is given in Algorithm 1.

Correctness. We use the following loop invariant to show correctness.

LEMMA 5.1. *Consider column i . Every reachability vector of this column corresponds to at least one realisation of $\mathcal{U}[1 : i]$ and the discrete Fréchet distance between that realisation and $\sigma[1 : \min(n, i + w)]$, and every realisation corresponds to some reachability vector.*

PROOF. The statement is trivial for the first column: we consider all ℓ possible realisations of U_1 and compute reachability of cells $(1, 1)$ to $(1, 1 + w)$ in a straightforward way.

Now suppose the statement holds for column i . As follows from the recurrence establishing the discrete Fréchet distance, the reachability of column $i + 1$ only depends on the distance matrix for column $i + 1$ and the reachability of column i . We consider every possible extension of $\mathcal{U}[1 : i]$ to $\mathcal{U}[1 : i + 1]$, as for every reachability vector of column i , we consider all ℓ options from the

ALGORITHM 1: Finding the time-banded upper bound discrete Fréchet distance on an indecisive and a precise curve.

```

1  function TBDFDINDPr( $\mathcal{U}, \sigma, w, \delta$ )
2     $\triangleright$  Input constraint:  $|\mathcal{U}| = |\sigma| = n$  and  $0 \leq w < n$ 
3    Initialise matrix  $D$  of size  $n \times \ell \times (2w + 1)$ 
4    for all  $i \in [n]$  do
5      for all  $k \in [\ell]$  do
6        for all  $j \in \{\max(1, i - w), \dots, \min(n, i + w)\}$  do
7           $D_{i,k,j} \leftarrow [d(p_i^k, q_j) \leq \delta?]$ 
8    Initialise matrix  $R$  of size  $n \times 2^{2w+1} \times 2w + 1$ 
9     $R_0 \leftarrow \langle r_1 = \text{TRUE}, r_2 = \text{FALSE}, r_3 = \text{FALSE}, \dots, r_{w+1} = \text{FALSE} \rangle$ 
10   for all  $k \in [\ell]$  do
11      $R_{1,k} \leftarrow \text{PROPAGATE}(R_0, D_{1,k}, 1, w, n)$ 
12   for all  $i \in [n] \setminus \{1\}$  do
13     for all  $A \in R_{i-1}$  do  $\triangleright$  For each reachability vector
14        $B \leftarrow A \vee (A << 1)$   $\triangleright$  Horizontal or diagonal step
15       for all  $k \in [\ell]$  do
16          $C \leftarrow \text{PROPAGATE}(B, D_{i,k}, i, w, n)$ 
17         Add  $C$  to set  $R_i$ 
18    $r \leftarrow \text{TRUE}$ 
19   for all  $A \in R_n$  do
20      $r \leftarrow r \wedge A_n$ 
21   return  $r$ 
22
23 function PROPAGATE( $A, B, i, w, n$ )
24    $\triangleright$  Propagate the reachability upwards in a column
25    $C \leftarrow A \wedge B$   $\triangleright$  Step and distance matrix
26    $r \leftarrow \text{FALSE}$ 
27   for all  $j \in \{\max(1, i - w), \dots, \min(n, i + w)\}$  do
28     if  $B_j \wedge C_j$  then  $r \leftarrow \text{TRUE}$   $\triangleright$  Current cell already reachable
29     else if  $B_j \wedge \neg C_j \wedge r$  then  $C_j \leftarrow \text{TRUE}$   $\triangleright$  Vertical step
30     else if  $\neg B_j$  then  $r \leftarrow \text{FALSE}$ 
31   return  $C$ 

```

distance matrix for column $i + 1$. Thus, we only consider valid realisations for column $i + 1$, and we consider all of them from the point of view of reachability. $\square$

Running Time. First of all, populating the distance matrix takes time $\Theta(\ell n w)$. A call to PROPAGATE takes $\Theta(w)$ time, so initialisation of the first column of the reachability matrix takes $\Theta(\ell w)$ time. Note that, at any further point, we may have at most 2^{2w+1} distinct reachability vectors; for each of them, we make ℓ calls to PROPAGATE, taking $\Theta(4^w \ell w)$ time per column, so over all the columns we need $\Theta(4^w \ell w n)$ time. If we assume that adding an element to the set takes amortised constant time, then the previous value dominates. Finally, the check at the end takes $\Theta(4^w)$ time. So, overall the algorithm runs in time $\Theta(4^w \ell n w)$. This agrees with our hardness result: for a small fixed-width time band, we get the running time of $\Theta(\ell n)$, whereas if we set $w = n - 1$ to compute the unrestricted distance, the algorithm runs in exponential time— $\Theta(4^n \ell n^2)$. We can also only store vectors that dominate in terms of FALSE values, as we are interested in the worst case. This improvement reduces the running time by a factor of $\sqrt{w}$.

THEOREM 5.2. *Problem UPPER BOUND DISCRETE FRÉCHET restricted to a Sakoe–Chiba time band of width w on a precise curve and an uncertain curve comprised of indecisive points with ℓ options, both of length n , can be decided in time $\Theta(4^w \ell n \sqrt{w})$ in the worst case.*

5.2 Upper Bound Discrete Fréchet Distance: Indecisive

Now we extend our previous result to the setting where both curves are indecisive, so instead of σ we have $\mathcal{V} = \langle V_1, \dots, V_n \rangle$, with, for each $j \in [n]$, $V_j = \{q_j^1, \dots, q_j^\ell\}$. Suppose we pick a realisation for curve $\mathcal{V}$ —then we can apply the algorithm we just described. We cannot run it separately for every realisation; instead, note that the part of the realisation that matters for column i is the points from $i - w$ to $i + w$, since any previous or further points are outside the time band. So, we can fix these $2w + 1$ points and compute the column. We do so for each possible combination of these $2w + 1$ points.

LEMMA 5.3. *Any reachability vector we store in column i corresponds to some realisation of the subcurves $\mathcal{U}[1 : i]$ and $\mathcal{V}[1 : \min(i + w, n)]$, and every such realisation has the resulting reachability vector stored in column i .*

PROOF. First of all, consider the statement for column 1. Clearly, we consider all possible realisations of both subcurves, so the statement holds.

Now, as we move from column i to column $i + 1$, we fix the realisation of points $i + 1 - w$ to $i + 1 + w$ on curve $\mathcal{V}$ and consider all the vectors stemming from the possible values of point $i - w$; as in Lemma 5.1, we cover all realisations of curve $\mathcal{U}$.

As for curve $\mathcal{V}$, note that we, again, only need the reachability from the previous column and the distance matrix from the current column, so the points before $i + 1 - w$ do not play a role for the consistency between the two, and thus they can be ignored.

So, we only get reachability vectors corresponding to valid realisations, and we do not miss any, as required. $\square$

The running time is now $\Theta(4^w \ell^{2w+1} n w)$, as we consider all combinations of the $2w + 1$ relevant points on $\mathcal{V}$ with ℓ options per point. For small constants w and ℓ , we get $\Theta(n)$; for $w = n - 1$, we get $\Theta(4^n n^2 \ell^{2n-1})$ —exponential time in n . As in the previous algorithm, we can store the Boolean vectors more efficiently, reducing the running time by a factor of $\sqrt{w}$.

THEOREM 5.4. *Suppose we are given two indecisive curves of length n with ℓ options per indecisive point. Then we can decide whether the upper bound discrete Fréchet distance restricted to a Sakoe–Chiba band of width w is below the threshold in time $\Theta(4^w \ell^{2w+1} n \sqrt{w})$.*

5.3 Expected Discrete Fréchet Distance

To compute the expected discrete Fréchet distance with time bands, we need two observations:

- (1) For any two precise curves, there is a single threshold δ where the answer to the decision problem changes from TRUE to FALSE—a *critical value*. That threshold corresponds to the distance between some two points on the curves.
- (2) We can modify our algorithm to store associated counts with each reachability vector, obtaining the fraction of realisations that yield the answer TRUE for a given threshold δ .

We can execute our algorithm for each critical value and get the cumulative distribution function $\mathbb{P}(d_{\text{dF}}(\pi, \sigma) > \delta)$ for $\pi, \sigma \in \mathcal{U}, \mathcal{V}$. As explained in the rest of this section, using the fact that the cumulative distribution function is a step function, we compute $d_{\text{dF}}^{\mathbb{E}}$.

Consider first the setting of one precise and one indecisive curve. Previously, we stored the reachability vectors in a set; instead, we can store a counter with each reachability vector so that

ALGORITHM 2: Finding the time-banded upper bound discrete Fréchet distance on two indecisive curves.

```

1  function TBDFDINDIND( $\mathcal{U}, \mathcal{V}, w, \delta$ )
2     $\triangleright$  Input constraint:  $|\mathcal{U}| = |\mathcal{V}| = n$  and  $0 \leq w < n$ 
3    Initialise matrix  $D$  of size  $n \times \ell \times (2w + 1) \times \ell$ 
4    for all  $i \in [n]$  do
5      for all  $k \in [\ell]$  do
6        for all  $j \in \{\max(1, i - w), \dots, \min(n, i + w)\}$  do
7          for all  $s \in [\ell]$  do
8             $D_{i,k,j,s} \leftarrow [d(p_i^k, q_j^s) \leq \delta?]$ 
9    Initialise matrix  $R$  of size  $n \times \ell^{2w+1} \times 2^{2w+1} \times 2w + 1$ 
10    $R0 \leftarrow \langle r_1 = \text{TRUE}, r_2 = \text{FALSE}, r_3 = \text{FALSE}, \dots, r_{w+1} = \text{FALSE} \rangle$ 
11   for all  $s \in [\ell^{w+1}]$  do
12     for all  $k \in [\ell]$  do
13        $R_{1,s,k} \leftarrow \text{PROPAGATE}(R0, D_{1,k}[s], 1, w, n)$ 
14   for all  $i \in [n] \setminus \{1\}$  do
15     for all  $s \in [\ell^{2w+1}]$  do  $\triangleright$  Or fewer in edge cases
16       for all  $A \in R_{i-1}[s]$  do  $\triangleright$  For each reachability vector with fixed realisation
17          $B \leftarrow A \vee (A << 1)$ 
18         for all  $k \in [\ell]$  do
19            $C \leftarrow \text{PROPAGATE}(B, D_{i,k}[s], i, w, n)$ 
20           Add  $C$  to set  $R_i[s]$ 
21    $r \leftarrow \text{TRUE}$ 
22   for all  $A \in R_n$  do
23     for all  $s \in [\ell^{2w+1}]$  do
24        $r \leftarrow r \wedge A_n[s]$ 
25   return  $r$ 
26
27 function PROPAGATE( $A, B, i, w, n$ )
28    $\triangleright$  Propagate the reachability upwards in a column
29    $C \leftarrow A \wedge B$ 
30    $r \leftarrow \text{FALSE}$ 
31   for all  $j \in \{\max(1, i - w), \dots, \min(n, i + w)\}$  do
32     if  $B_j \wedge C_j$  then  $r \leftarrow \text{TRUE}$ 
33     else if  $B_j \wedge \neg C_j \wedge r$  then  $C_j \leftarrow \text{TRUE}$ 
34     else if  $\neg B_j$  then  $r \leftarrow \text{FALSE}$ 
35   return  $C$ 

```

every time we get an element that is already stored, we increment the counter. We cannot use the improvement that would allow us to discard some vectors, as that would eschew the count, and we are not interested in the worst possible result now. We can implement a similar mechanism in the setting of two indecisive curves. Moreover, we can propagate the count through the algorithm and in the end find the counts associated with answers TRUE and FALSE to the decision problem.

So, if we store the count of realisations that give us a certain reachability vector, we essentially obtain, for some value of δ ,

$$\mathbb{P}(d_{\text{dF}}(\pi, \sigma) > \delta) \quad \text{when } \pi, \sigma \in_{\mathcal{U}} \mathcal{U}, \mathcal{V}.$$

For any realisation, there is a specific value of δ —a *critical value*—that acts as a threshold between the answers TRUE and FALSE for that realisation, since if we fix the realisation, we just compute the regular discrete Fréchet distance. Note that that threshold must be a distance between some two

points on different curves. In the case of a precise and an indecisive curve, there are $\ell n(2w + 1)$ such distances with the time band of width w ; in the case of two indecisive curves, there are $\ell^2 n(2w + 1)$ such distances. Therefore, if we run our algorithm for each of these critical values and record the counts of TRUE and FALSE for each threshold, we obtain the complete cumulative distribution function $\mathbb{P}(d_{\text{dF}}(\pi, \sigma) > \delta)$ for $\pi, \sigma \in \mathcal{U}, \mathcal{V}$.

Then we can simply find, under the time band restriction,

$$d_{\text{dF}}^{\mathbb{E}(\mathcal{U})}(\mathcal{U}, \mathcal{V}) = \int_0^\infty \mathbb{P}_{\pi, \sigma \in \mathcal{U}, \mathcal{V}}(d_{\text{dF}}(\pi, \sigma) > \delta) d\delta.$$

For any realisation the answer may change from TRUE to FALSE only at one of the critical values. So, the distribution of TRUE and FALSE only changes at a finite set of critical values and is constant between them; therefore, $\mathbb{P}(d_{\text{dF}}(\pi, \sigma) > \delta)$ is a step function. Hence, finding the integral of interest amounts to multiplying the value of $\mathbb{P}(d_{\text{dF}}(\pi, \sigma) > \delta)$ by the distance between two successive values of δ that match, and summing all the results—that is, to finding the area under the step function by summing up the areas of the rectangles that make it up.

So, clearly, under the time band restriction, we can run one of our algorithms either $\ell n(2w + 1)$ or $\ell^2 n(2w + 1)$ times to obtain the expected discrete Fréchet distance. We show the details in Algorithm 3 for the two settings. We summarise this result as follows.

THEOREM 5.5. *Suppose we are given an indecisive curve $\mathcal{U}$ and a precise curve σ of length n with ℓ options per indecisive point and we want to compute the expected discrete Fréchet distance constrained to a Sakoe–Chiba band of width w . Then we can run $\text{EXPTBDFDINDPR}(\mathcal{U}, \sigma, w)$ to obtain the result in time $\Theta(4^w \ell^2 n^2 w^2)$ in the worst case.*

PROOF. First of all, note that from the preceding discussion it immediately follows that the algorithm is correct. In the worst case, every δ that we have to add to E will be distinct, so we have $\ell n(2w + 1)$ insertions, taking in total $\Theta(\ell n w \log \ell n w)$ time. Then we run CNTBDFDINDPR once per value in E , and its running time is the same as that of TBDFDINDPR , so here we take time $\Theta(\ell n w \cdot 4^w \ell n w)$ in the worst case, as claimed. $\square$

We can formalise the result similarly for the other setting.

THEOREM 5.6. *Suppose we are given two indecisive curves $\mathcal{U}$ and $\mathcal{V}$ of length n with ℓ options per indecisive point and want to find the expected discrete Fréchet distance when constrained to a Sakoe–Chiba band of width w . Then we can run $\text{EXPTBDFDINDIND}(\mathcal{U}, \mathcal{V}, w)$ to obtain the result in time $\Theta(4^w \ell^{2w+3} n^2 w^2)$ in the worst case.*

PROOF. Again, note that from the preceding discussion, it immediately follows that the algorithm is correct. In the worst case, we have $\ell^2 n w$ insertions, taking in total $\Theta(\ell^2 n w \log \ell n w)$ time. Then we run CNTBDFDINDIND once per value in E , and its running time is the same as that of TBDFDINDIND , so here we take time $\Theta(\ell^2 n w \cdot 4^w \ell^{2w+1} n w)$ in the worst case, as claimed. $\square$

5.4 Upper Bound Continuous Fréchet Distance

We can adapt our time band algorithms to handle the continuous Fréchet distance. Instead of the Boolean reachability vectors, we use vectors of *free space* cells, introduced by Alt and Godau [6, 34]. We now need to store reachability intervals on cell borders. The number of these intervals is limited: for any cell, the upper value of the interval is determined by the distance matrix, yielding at most ℓ^2 values; the lower value of the interval is determined by the distance matrix or by one of the cells from the same row, yielding exponential dependency on w . However, the algorithm is still polynomial time in n .

ALGORITHM 3: Finding the time-banded expected discrete Fréchet distance on an indecisive and a precise curve and two indecisive curves.

```

1  function EXP_TBDFD_IND_PR( $\mathcal{U}, \sigma, w$ )
2    ▷ Input constraint:  $|\mathcal{U}| = |\sigma| = n$  and  $0 \leq w < n$ 
3    Initialise sorted set  $E$ 
4    for all  $i \in [n]$  do
5      for all  $k \in [\ell]$  do
6        for all  $j \in \{\max(1, i - w), \dots, \min(n, i + w)\}$  do
7          Add  $d(p_i^k, q_j)$  to sorted set  $E$ 
8     $s \leftarrow E[1]$ 
9    for  $i \leftarrow 1$  to  $l(E) - 1$  do
10      $\delta \leftarrow E[i], \delta' \leftarrow E[i + 1]$ 
11      $p \leftarrow \text{CNT\_TBDFD\_IND\_PR}(\mathcal{U}, \sigma, w, \delta)$ 
12      $s \leftarrow s + (1 - p) \cdot (\delta' - \delta)$ 
13    return  $s$ 
14
15 function EXP_TBDFD_IND_IND( $\mathcal{U}, \mathcal{V}, w$ )
16   ▷ Input constraint:  $|\mathcal{U}| = |\mathcal{V}| = n$  and  $0 \leq w < n$ 
17   Initialise sorted set  $E$ 
18   for all  $i \in [n]$  do
19     for all  $k \in [\ell]$  do
20       for all  $j \in \{\max(1, i - w), \dots, \min(n, i + w)\}$  do
21         for all  $s \in [\ell]$  do
22           Add  $d(p_i^k, q_j^s)$  to sorted set  $E$ 
23    $s \leftarrow E[1]$ 
24   for  $i \leftarrow 1$  to  $l(E) - 1$  do
25      $\delta \leftarrow E[i], \delta' \leftarrow E[i + 1]$ 
26      $p \leftarrow \text{CNT\_TBDFD\_IND\_IND}(\mathcal{U}, \mathcal{V}, w, \delta)$ 
27      $s \leftarrow s + (1 - p) \cdot (\delta' - \delta)$ 
28   return  $s$ 
29
30 function CNT_TBDFD_IND_PR( $\mathcal{U}, \sigma, w, \delta$ )
31   ▷ Like TBDFD_IND_PR, but returns the fraction of count of TRUE over FALSE for the final cell.
32
33 function CNT_TBDFD_IND_IND( $\mathcal{U}, \mathcal{V}, w, \delta$ )
34   ▷ Like TBDFD_IND_IND, but returns the fraction of count of TRUE over FALSE for the final cell.

```

In more detail, one could adapt the algorithms for the upper bound discrete Fréchet distance to the case when either both curves are indecisive or one is precise and one is indecisive, and we are interested in the decision problem for the Fréchet distance and not the discrete Fréchet distance. Since we are going column-by-column, we would need to store the reachability intervals on the vertical border of each cell.

It is simpler to see how this would work in the setting of a precise and an indecisive curve: each column now is a column of a free-space diagram, and we only need to store the intervals on the right side of the column. As we progress to the next column, we need to consider all the options from the previous column, so we need to run the same algorithm, except we store and process vectors of free-space intervals instead of TRUE and FALSE. One other distinction is that we do not consider diagonal steps—for the Fréchet distance, doing so would not be meaningful, as the path

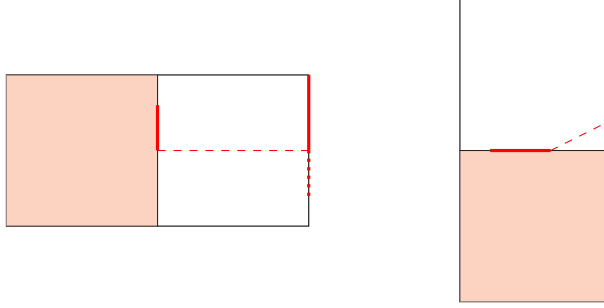


Fig. 13. Reachability adjustments. Left: Although the dotted interval is free according to the distance matrix, only the solid interval is reachable from the cell on the left with a monotone path, assuming the entire cell on the left is free. Right: The entire interval that is marked as free according to the distance matrix is reachable with a monotone path from the cell below, assuming the cell below is free.

is continuous, and the diagonal step is not distinguishable from a horizontal step followed by a vertical step, if such situation occurs.

In particular, we now take the intervals stored in the distance matrix and compute reachability based on the previous column: if a cell can be reached horizontally from the previous cell, then the lower bound of the interval in this cell may need to go up, since we can only use monotone paths. PROPAGATE will now take the intervals that correspond to the distance matrix and the precomputed reachability and make the following adjustment: if a cell is reachable from below, then the entire interval on the right is actually reachable. Figure 13 presents an example of both cases.

Other than that, the algorithm is exactly the same; clearly, we can make the same adjustments to the algorithm handling two indecisive curves.

Notice that we now do not have at most 2^{2w+1} vectors per column, since we store intervals instead of Boolean values, and they can be more varied. However, the number of values is still limited: for any cell, the upper value of the interval is determined by the distance matrix, so there can be at most ℓ or ℓ^2 values for the two settings. The lower value of the interval is determined by the distance matrix or by one of the cells from the same row; these may have at most ℓ or ℓ^2 values each, and there are at most $2w$ of them, so per cell we can have at most $\Theta(\ell w)$ or $\Theta(\ell^2 w)$ lower interval values and $\Theta(\ell)$ or $\Theta(\ell^2)$ upper interval values, instead of just two possible values in the discrete case. Note that for an interval, we only pick one of the possible lower bound values, and a lower bound value ultimately comes from the distance between some pair of points; additionally, we pick one upper bound value, giving us $\Theta(\ell^2 w)$ and $\Theta(\ell^4 w)$ possible unique intervals. We also need to modify the set operations, for example, by enumerating the possible boundaries and storing intervals as pairs of indices; adding a vector to a set would then take $O(w + \log \ell w)$ time. The running time changes accordingly, replacing 4^w with $(\ell^2 w)^{2w+1}$ and replacing $4^w \ell^{2w+1}$ with $(\ell^4 w)^{2w+1}$, but, importantly, we still have linear dependency on n , so the running time is polynomial for fixed w and ℓ .

5.5 Expected Continuous Fréchet Distance

We can, of course, again store the associated counts with the vectors of intervals in the algorithm. As we look at the final cell, we can sum up the counts associated with the cases where the upper right corner of this cell is reachable, and so we can find the proportion of TRUE to FALSE for a particular threshold δ .

We can find the critical values; now they follow in line with those discussed by Alt and Godau [6, 34]. The number of the critical values is different: case 1, where we look at the start and end

points, now yields $\Theta(\ell^2)$ events; case 2, where we look at two neighbouring cells, so at the distance between a segment and a point, yields $\Theta(\ell^3nw)$ events; and case 3, where we look at the distance between a segment and two points, yields $\Theta(\ell^4nw^2)$ events.

Otherwise, we can run Algorithm 3 on the new critical values, calling instead the counting version for the continuous Fréchet distance. This way, we can compute the expected Fréchet distance restricted to a Sakoe–Chiba band in time polynomial in n for fixed w and ℓ .

THEOREM 5.7. *Suppose we are given two indecisive curves of length n with ℓ options per indecisive point. Then we can decide the upper bound Fréchet distance and compute the expected Fréchet distance restricted to a Sakoe–Chiba band of fixed width w in time polynomial in n .*

6 CONCLUSION

In this article, we studied the upper bound, the lower bound, and the expected Fréchet distance under various uncertainty models. We conclude that it is NP-hard to decide if the upper bound is below a given threshold in all the models we consider; as the follow-up work [16] shows, also in 1D. This seems to translate to #P-hardness for computing the expected Fréchet distance under the uniform distribution. We do not have reason to believe that the variants of the expected Fréchet distance not covered in Table 1 are easier. The lower bound problem presents an interesting tradeoff, though: although the problem of deciding whether the lower bound is below a given threshold is still NP-hard for the continuous Fréchet distance for uncertain points modelled as line segments, the problem becomes tractable when either the uncertainty regions or the distance measure (or both) are discrete. We conjecture that the continuous Fréchet distance for uncertain points modelled as disks (or other continuous regions) is no easier than for line segments.

In future work on the topic, it would be helpful to understand where exactly the divide lies (i.e., what kind of uncertainty models make the problem simpler); we can also ask whether the problem is fixed-parameter tractable when parametrised by the number of allowed movement directions (two in 1D, anywhere in 2D). One could also generalise the expected Fréchet distance to other distributions and uncertainty models, ideally formulating simple conditions on the input to achieve a result. Finally, it would be interesting to see any approximation algorithms for the upper bound Fréchet distance.

REFERENCES

- [1] Manuel Abellanas, Ferran Hurtado, Christian Icking, Rolf Klein, Elmar Langetepe, Lihong Ma, Belén Palop, and Vera Sacristán. 2001. Smallest color-spanning objects. In *Algorithms—ESA 2001*. Lecture Notes in Computer Science, Vol. 2161. Springer, 278–289. DOI: https://doi.org/10.1007/3-540-44676-1_23
- [2] Pankaj K. Agarwal, Boris Aronov, Sarel Har-Peled, Jeff M. Phillips, Ke Yi, and Wuzhou Zhang. 2016. Nearest-neighbor searching under uncertainty II. *ACM Transactions on Algorithms* 13, 1 (2016), Article 3, 25 pages. DOI: <https://doi.org/10.1145/2955098>
- [3] Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. 2014. Computing the discrete Fréchet distance in subquadratic time. *SIAM Journal on Computing* 43, 2 (2014), 429–449. DOI: <https://doi.org/10.1137/130920526>
- [4] Pankaj K. Agarwal, Alon Efrat, Swaminathan Sankararaman, and Wuzhou Zhang. 2017. Nearest-neighbor searching under uncertainty I. *Discrete & Computational Geometry* 58, 3 (2017), 705–745. DOI: <https://doi.org/10.1007/s00454-017-9903-x>
- [5] Hee-Kap Ahn, Christian Knauer, Marc Scherfenberg, Lena Schlipf, and Antoine Vigneron. 2012. Computing the discrete Fréchet distance with imprecise input. *International Journal of Computational Geometry & Applications* 22, 1 (2012), 27–44. DOI: <https://doi.org/10.1142/S0218195912600023>
- [6] Helmut Alt and Michael Godau. 1995. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications* 5, 1 (1995), 75–91. DOI: <https://doi.org/10.1142/S0218195995000064>
- [7] Esther M. Arkin, Aritra Banik, Paz Carmi, Gui Citovsky, Matthew J. Katz, Joseph S. B. Mitchell, and Marina Simakov. 2018. Selecting and covering colored points. *Discrete Applied Mathematics* 250 (2018), 75–86. DOI: <https://doi.org/10.1016/j.dam.2018.05.011>

- [8] Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (AAAIWS'94)*. 359–370. <https://aaai.org/conference/kdd/kdd97/>.
- [9] Karl Bringmann. 2014. Why walking the dog takes time. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS'14)*. IEEE, Los Alamitos, CA, 661–670. DOI : <https://doi.org/10.1109/FOCS.2014.76>
- [10] Karl Bringmann, Marvin Künnemann, and André Nusser. 2019. Fréchet distance under translation. In *Proceedings of the 30th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'19)*. 2902–2921. DOI : <https://doi.org/10.5555/3310435.3310615>
- [11] Kevin Buchin, Maike Buchin, and Joachim Gudmundsson. 2010. Constrained free space diagrams. *International Journal of Geographical Information Science* 24, 7 (2010), 1101–1125. DOI : <https://doi.org/10.1080/13658810903569598>
- [12] Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. 2017. Four Soviets walk the dog. *Discrete & Computational Geometry* 58, 1 (2017), 180–216. DOI : <https://doi.org/10.1007/s00454-017-9878-7>
- [13] Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. 2019. Approximating (k, ℓ) -center clustering for curves. In *Proceedings of the 30th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'19)*. 2922–2938. DOI : <https://doi.org/10.1137/1.9781611975482.181>
- [14] Kevin Buchin, Anne Driemel, Natasja van de L'Isle, and André Nusser. 2019. klcluster. In *Proceedings of the 27th International Conference on Advances in Geographic Information Systems (SIGSPATIAL'19)*. ACM, New York, NY, 496–499. DOI : <https://doi.org/10.1145/3347146.3359111>
- [15] Kevin Buchin, Maarten Löffler, Pat Morin, and Wolfgang Mulzer. 2011. Preprocessing imprecise points for Delaunay triangulation. *Algorithmica* 61, 3 (2011), 674–693. DOI : <https://doi.org/10.1007/s00453-010-9430-0>
- [16] Kevin Buchin, Maarten Löffler, Tim Ophelders, Aleksandr Popov, Jérôme Urhausen, and Kevin Verbeek. 2023. Computing the Fréchet distance between uncertain curves in one dimension. *Computational Geometry* 109 (2023), 101923. DOI : <https://doi.org/10.1016/j.comgeo.2022.101923>
- [17] Kevin Buchin, Maarten Löffler, Aleksandr Popov, and Marcel Roeloffzen. 2021. Uncertain curve simplification. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS'21) (Leibniz International Proceedings in Informatics)*, Filippo Bonchi and Simon J. Puglisi (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Germany, Article 26. DOI : <https://doi.org/10.4230/LIPICs.MFCS.2021.26>
- [18] Kevin Buchin, Tim Ophelders, and Bettina Speckmann. 2019. SETH says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proceedings of the 30th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'19)*. 2887–2901. DOI : <https://doi.org/10.1137/1.9781611975482.179>
- [19] Kevin Buchin, Stef Sijben, T. Jean Marie Arseneau, and Erik P. Willems. 2012. Detecting movement patterns using Brownian bridges. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL'12)*. ACM, New York, NY, 119–128. DOI : <https://doi.org/10.1145/2424321.2424338>
- [20] Maike Buchin, Anne Driemel, and Bettina Speckmann. 2014. Computing the Fréchet distance with shortcuts is NP-hard. In *Proceedings of the 30th Annual Symposium on Computational Geometry (SoCG'14)*. ACM, New York, NY, 367–376. DOI : <https://doi.org/10.1145/2582112.2582144>
- [21] Maike Buchin and Stef Sijben. 2016. Discrete Fréchet distance for uncertain points. In *Proceedings of the European Workshop on Computational Geometry (EuroCG'16)*. http://www.eurocg2016.usi.ch/sites/default/files/paper_72.pdf.
- [22] Leizhen Cai and Mark Keil. 1997. Computing visibility information in an inaccurate simple polygon. *International Journal of Computational Geometry & Applications* 7, 6 (1997), 515–538. DOI : <https://doi.org/10.1142/S0218195997000326>
- [23] Sandip Das, Partha P. Goswami, and Subhas C. Nandy. 2009. Smallest color-spanning object revisited. *International Journal of Computational Geometry & Applications* 19, 5 (2009), 457–478. DOI : <https://doi.org/10.1142/S0218195909003076>
- [24] Thomas Devogele, Laurent Etienne, Maxence Esnault, and Florian Lardy. 2017. Optimized discrete Fréchet distance between trajectories. In *Proceedings of the 6th ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data (BigSpatial'17)*. ACM, New York, NY, 11–19. DOI : <https://doi.org/10.1145/3150919.3150924>
- [25] Anne Driemel and Sariel Har-Peled. 2018. Jaywalking your dog. *SIAM Journal on Computing* 42, 5 (2018), 1830–1866. DOI : <https://doi.org/10.1137/120865112>
- [26] Anne Driemel, Sariel Har-Peled, and Carola Wenk. 2012. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete & Computational Geometry* 48, 1 (2012), 94–127. DOI : <https://doi.org/10.1007/s00454-012-9402-z>
- [27] Anne Driemel, Herman Haverkort, Maarten Löffler, and Rodrigo I. Silveira. 2013. Flow computations on imprecise terrains. *Journal of Computational Geometry* 4, 1 (2013), 38–78. DOI : <https://doi.org/10.20382/jocg.v4i1a3>
- [28] Thomas Eiter and Heikki Mannila. 1994. *Computing Discrete Fréchet Distance*. Technical Report CD-TR 94/64. Technische Universität Wien. <http://www.kr.tuwien.ac.at/staff/eiter/et-archive/cdr9464.pdf>.

- [29] William Evans, David Kirkpatrick, Maarten Löffler, and Frank Staals. 2013. Competitive query strategies for minimising the ply of the potential locations of moving points. In *Proceedings of the 29th Annual Symposium on Computational Geometry (SoCG'13)*. ACM, New York, NY, 155–164. DOI: <https://doi.org/10.1145/2462356.2462395>
- [30] Chenglin Fan, Jun Luo, and Binhai Zhu. 2013. Tight approximation bounds for connectivity with a color-spanning set. In *Algorithms and Computation*. Lecture Notes in Computer Science, Vol. 8283. Springer, 590–600. DOI: https://doi.org/10.1007/978-3-642-45030-3_55
- [31] Chenglin Fan and Benjamin Raichel. 2017. Computing the Fréchet gap distance. In *Proceedings of the 33rd International Symposium on Computational Geometry (SoCG'17) (Leibniz International Proceedings in Informatics)*, Boris Aronov and Matthew J. Katz (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, Article 42, 16 pages. DOI: <https://doi.org/10.4230/LIPIcs.SocG.2017.42>
- [32] Chenglin Fan and Binhai Zhu. 2018. Complexity and algorithms for the discrete Fréchet distance upper bound with imprecise input. *arXiv:cs.CG/1509.02576v2* (2018).
- [33] Omrit Filtser and Matthew J. Katz. 2018. Algorithms for the discrete Fréchet distance under translation. In *Proceedings of the 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT'18) (Leibniz International Proceedings in Informatics)*, David Eppstein (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, Article 20, 14 pages. DOI: <https://doi.org/10.4230/LIPIcs.SWAT.2018.20>
- [34] Michael Godau. 1991. A natural metric for curves—Computing the distance for polygonal chains and approximation algorithms. In *STACS 91*. Lecture Notes in Computer Science, Vol. 480. Springer, 127–136. DOI: <https://doi.org/10.1007/BFb0020793>
- [35] Chris Gray, Frank Kammer, Maarten Löffler, and Rodrigo I. Silveira. 2012. Removing local extrema from imprecise terrains. *Computational Geometry* 45, 7 (2012), 334–349. DOI: <https://doi.org/10.1016/j.comgeo.2012.02.002>
- [36] Joachim Gudmundsson, Majid Mirzanezhad, Ali Mohades, and Carola Wenk. 2019. Fast Fréchet distance between curves with long edges. *International Journal of Computational Geometry & Applications* 29, 2 (2019), 161–187. DOI: <https://doi.org/10.1142/S0218195919500043>
- [37] Leonidas J. Guibas, John E. Hersherberger, Joseph S. B. Mitchell, and Jack Scott Snoeyink. 1993. Approximating polygons and subdivisions with minimum-link paths. *International Journal of Computational Geometry & Applications* 3, 4 (1993), 383–415. DOI: <https://doi.org/10.1142/S0218195993000257>
- [38] Sarel Har-Peled and Benjamin Raichel. 2014. The Fréchet distance revisited and extended. *ACM Transactions on Algorithms* 10, 1 (2014), Article 3, 22 pages. DOI: <https://doi.org/10.1145/2532646>
- [39] Allan Jørgensen, Jeff M. Phillips, and Maarten Löffler. 2011. Geometric computations on indecisive points. In *Algorithms and Data Structures*. Lecture Notes in Computer Science, Vol. 6844. Springer, 536–547. DOI: https://doi.org/10.1007/978-3-642-22300-6_45
- [40] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7, 3 (2005), 358–386. DOI: <https://doi.org/10.1007/s10115-004-0154-9>
- [41] Christian Knauer, Maarten Löffler, Marc Scherfenberg, and Thomas Wolle. 2011. The directed Hausdorff distance between imprecise point sets. *Theoretical Computer Science* 412, 32 (2011), 4173–4186. DOI: <https://doi.org/10.1016/j.tcs.2011.01.039>
- [42] John Krumm. 2009. A survey of computational location privacy. *Personal and Ubiquitous Computing* 13, 6 (2009), 391–399. DOI: <https://doi.org/10.1007/s00779-008-0212-5>
- [43] Maarten Löffler. 2009. *Data Imprecision in Computational Geometry*. Ph.D. Dissertation. Universiteit Utrecht. <https://dspace.library.uu.nl/bitstream/handle/1874/36022/loffler.pdf>.
- [44] Maarten Löffler and Wolfgang Mulzer. 2014. Unions of onions. *Journal of Computational Geometry* 5, 1 (2014), 1–13. DOI: <https://doi.org/10.20382/jocg.v5i1a1>
- [45] Maarten Löffler and Jack Scott Snoeyink. 2010. Delaunay triangulations of imprecise points in linear time after pre-processing. *Computational Geometry* 43, 3 (2010), 234–242. DOI: <https://doi.org/10.1016/j.comgeo.2008.12.007>
- [46] Maarten Löffler and Marc van Kreveld. 2006. Largest and smallest tours and convex hulls for imprecise points. In *Algorithm Theory—SWAT 2006*. Lecture Notes in Computer Science, Vol. 4059. Springer, 375–387. DOI: https://doi.org/10.1007/11785293_35
- [47] Anil Maheshwari, Jörg-Rüdiger Sack, Kaveh Shahbaz, and Hamid Zarrabi-Zadeh. 2011. Fréchet distance with speed limits. *Computational Geometry: Theory & Applications* 44, 2 (2011), 110–120. DOI: <https://doi.org/10.1016/j.comgeo.2010.09.008>
- [48] Jian Pei, Bin Jiang, Xuemin Lin, and Yidong Yuan. 2007. Probabilistic skylines on uncertain data. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB'07)*. 15–26.
- [49] Dieter Pfoser and Christian S. Jensen. 1999. Capturing the uncertainty of moving-object representations. In *Advances in Spatial Databases*. Lecture Notes in Computer Science, Vol. 1651. Springer, 111–131. DOI: https://doi.org/10.1007/3-540-48482-5_9

- [50] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (1978), 43–49. DOI : <https://doi.org/10.1109/TASSP.1978.1163055>
- [51] Jeff Sember and William Evans. 2008. Guaranteed Voronoi diagrams of uncertain sites. In *Proceedings of the 20th Canadian Conference on Computational Geometry (CCC'09)*.
- [52] Aravinda Prasad Sistla, Ouri Wolfson, Sam Chamberlain, and Son Dao. 1998. Querying the uncertain position of moving objects. In *Temporal Databases*. Lecture Notes in Computer Science, Vol. 1399. Springer, 310–337. DOI : <https://doi.org/10.1007/BFb0053708>
- [53] Subhash Suri, Kevin Verbeek, and Hakan Yildiz. 2013. On the most likely convex hull of uncertain points. In *Algorithms—ESA 2013*. Lecture Notes in Computer Science, Vol. 8125. Springer, 791–802. DOI : https://doi.org/10.1007/978-3-642-40450-4_67
- [54] Mees van de Kerkhof, Irina Kostitsyna, Maarten Löffler, Majid Mirzanezhad, and Carola Wenk. 2019. Global curve simplification. In *Proceedings of the 27th Annual European Symposium on Algorithms (ESA'19) (Leibniz International Proceedings in Informatics)*, Michael A. Bender, Ola Svensson, and Grzegorz Herman (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, Article 67, 14 pages. DOI : <https://doi.org/10.4230/LIPIcs.ESA.2019.67>
- [55] Marc van Kreveld, Maarten Löffler, and Joseph S. B. Mitchell. 2010. Preprocessing imprecise points and splitting triangulations. *SIAM Journal on Computing* 39, 7 (2010), 2990–3000. DOI : <https://doi.org/10.1137/090753620>
- [56] Marc van Kreveld, Maarten Löffler, and Lionov Wiratma. 2018. On optimal polyline simplification using the Hausdorff and Fréchet distance. In *Proceedings of the 34th International Symposium on Computational Geometry (SoCG'18) (Leibniz International Proceedings in Informatics)*, Bettina Speckmann and Csaba D. Tóth (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, Article 56, 14 pages. DOI : <https://doi.org/10.4230/LIPIcs.SoCG.2018.56>
- [57] Man Lung Yiu, Nikos Mamoulis, Xiangyuan Dai, Yufei Tao, and Michail Vaitis. 2009. Efficient evaluation of probabilistic advanced spatial queries on existentially uncertain data. *IEEE Transactions on Knowledge and Data Engineering* 21, 1 (2009), 108–122. DOI : <https://doi.org/10.1109/TKDE.2008.135>

Received 2 November 2021; revised 9 January 2023; accepted 4 April 2023