Computing the Fréchet Distance Between Uncertain Curves in One Dimension

Document status and date:
Published: 01/02/2023

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Download date: 30. May. 2023

Computing the Fréchet distance between uncertain curves in one dimension ☆

Kevin Buchin [a], Maarten Löffler [c,1], Tim Ophelders [b,c], Aleksandr Popov [b,*,2], Jérôme Urhausen [c,3], Kevin Verbeek [b]

[a] *TU Dortmund, Otto-Hahn-Str. 14, 44221, Dortmund, Germany*
[b] *TU Eindhoven, Postbus 513, 5600 MB, Eindhoven, the Netherlands*
[c] *Utrecht University, Postbus 80125, 3508 TC, Utrecht, the Netherlands*

ARTICLE INFO

ABSTRACT

We consider the problem of computing the Fréchet distance between two curves for which the exact locations of the vertices are unknown. Each vertex may be placed in a given *uncertainty region* for that vertex, and the objective is to place vertices so as to minimise the Fréchet distance. This problem was recently shown to be NP-hard in 2D, and it is unclear how to compute an optimal vertex placement at all.

We present the first general algorithmic framework for this problem. We prove that it results in a polynomial-time algorithm for curves in 1D with intervals as uncertainty regions. In contrast, we show that the problem is NP-hard in 1D in the case that vertices are placed to maximise the Fréchet distance.

We also study the weak Fréchet distance between uncertain curves. While finding the optimal placement of vertices seems more difficult than the regular Fréchet distance—and indeed we can easily prove that the problem is NP-hard in 2D—the optimal placement of vertices in 1D can be computed in polynomial time. Finally, we investigate the discrete weak Fréchet distance, for which, somewhat surprisingly, the problem is NP-hard already in 1D.

1. Introduction

The *Fréchet distance* is a popular distance measure for curves. Its computational complexity has drawn considerable attention in computational geometry [1–7]. The Fréchet distance between two (polygonal) curves is often illustrated using a person and a dog: imagine a person is walking along one curve, having the dog, which walks on the other curve, on a leash. The person and the dog may change their speed independently but may not walk backwards. The Fréchet distance

Table 1
Complexity results for the lower bound problems for uncertain curves.

	Fréchet distance		Weak Fréchet distance	
	discrete	continuous	discrete	continuous
1D	polynomial [24]	polynomial	NP-hard	polynomial
2D	polynomial [24]	NP-hard [25]	NP-hard	NP-hard

corresponds to the minimum leash length needed so that the person and the dog can walk from start to end on their respective curve.

The Fréchet distance and its variants have found many applications, for instance, in the context of protein alignment [8], handwriting recognition [9], map matching [10] and construction [11,12], and trajectory similarity and clustering [13,14]. In most applications, we obtain the curves by a sequence of measurements, and these are inherently imprecise. However, it is often reasonable to assume that the true location is within a certain radius of the measurement, or more generally that it stays within an *uncertainty region.*

Re-imagine the person and the dog, except now each is given a sequence of regions they have to visit. More specifically, they need to visit one location per region and move along a straight line between locations without going backwards. Then minimising the leash length corresponds to the following problem. Each curve is given as a sequence of uncertainty regions; minimise the Fréchet distance over all the possible choices of locations in the regions. This is called the *lower bound* problem for the Fréchet distance between uncertain curves.

Similar problems involving uncertainty have drawn more and more attention in the past few years in computational geometry. Most results are on uncertain point sets, where we often aim to minimise or maximise some quantity stemming from the point set, but also perform visibility queries in polygons or find Delaunay triangulations [15–23]. More recently there have also been several results on curves with uncertainty [24–27].

The earliest results for a variant of the problem we consider do not concern the Fréchet distance as such, but its variant the discrete Fréchet distance, where we restrict our attention to the vertices of the curves. Ahn et al. [24] give a polynomial-time algorithm that decides whether the lower bound discrete Fréchet distance is below a certain threshold, for two curves with uncertainty regions modelled as circles in constant dimension. The lower bound Fréchet distance with uncertainty regions modelled as point sets admits a simple dynamic program [25]. However, as recently shown, the decision problem for the continuous Fréchet distance is NP-hard already in two dimensions with vertical line segments as uncertainty regions and one precise and one uncertain curve [25]; it is not clear how to compute the lower bound at all with any uncertainty model that is not discrete. With the 2D problem being NP-hard, we turn our attention to one-dimensional curves. We present an efficient algorithm for computing the lower bound Fréchet distance with imprecision modelled as intervals. We further generalise this to a framework applicable in higher dimensions and restricted settings; it may not give polynomial-time solutions in many settings.

Next to the discrete Fréchet distance, the most common variant of the Fréchet distance is the weak Fréchet distance [2]. In the person–dog analogy, this variant allows backtracking on the paths. The weak Fréchet distance (for precise curves) has interesting properties in 1D [7,28,29]: it can be computed in linear time in 1D, while in 2D it cannot be computed significantly faster than quadratic time under the strong exponential-time hypothesis. To our knowledge, the weak Fréchet distance has not been studied in the uncertain setting before. We give a polynomial-time algorithm that solves the lower bound problem in 1D. In contrast to that, we show that the problem is NP-hard in 2D, and that discrete weak Fréchet distance is NP-hard already in 1D. We summarise these results in Table 1.

The table provides an interesting insight. First of all, it appears that for continuous distances the dimension matters, whereas for the discrete ones the results are the same both in 1D and 2D. Moreover, it may be surprising that discretising the problem has a different effect: for the Fréchet distance it makes it easier, while for the weak Fréchet distance the problem becomes harder. We discuss the polynomial-time algorithm for the Fréchet distance in 1D in Section 4. We give the algorithm for the weak Fréchet distance in 1D in Section 6.1 and show NP-hardness for the weak (discrete) Fréchet distance in Section 6.2.

Finally, we also turn our attention to the problem of maximising the Fréchet distance, or finding the upper bound. It has been shown that the problem is NP-hard in 2D for several uncertainty models, including discrete point sets, both for the discrete and the continuous Fréchet distance [25]. We strengthen that result by presenting a similar construction that already shows NP-hardness in 1D. The proof is given in Section 5.

2. Preliminaries

Denote $[n] \equiv \{1, 2, \ldots, n\}$. Consider a *sequence* of points $\pi = \langle p_1, p_2, \ldots, p_n \rangle$. We also use π to denote a *polygonal curve,* defined by the sequence by linearly interpolating between the 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 number of its vertices, $|\pi| = n$. Denote the concatenation of two sequences π and σ by $\pi \sqcup \sigma$. We can generalise this notation:

$$\pi \equiv \bigsqcup_{i \in [n]} p_i = p_1 \sqcup p_2 \sqcup \ldots \sqcup p_n.$$

Denote a subcurve from vertex i to j of π by $\pi[i:j] = \langle p_i, p_{i+1}, \ldots, p_j \rangle$. Occasionally we use the notation $\langle \pi(i) \mid i \in I \rangle_{i=1}^{n}$ to denote a curve built on a subsequence of vertices of π, where vertices are only taken if they are in set I. For example, setting $I = \{1, 3, 4\}$, $n = 5$, $\pi = \langle p_1, p_2, \ldots, p_5 \rangle$ means $\langle \pi(i) \mid i \in I \rangle_{i=1}^{n} = \langle p_1, p_3, p_4 \rangle$.

Denote the Fréchet distance between two polygonal curves π and σ by $d_{\mathrm{F}}(\pi, \sigma)$, the discrete Fréchet distance by $d_{\mathrm{dF}}(\pi, \sigma)$, and the weak Fréchet distance by $d_{\mathrm{wF}}(\pi, \sigma)$. Recall the definition of Fréchet distance for polygonal curves of lengths m and n. It is based on *parametrisations* (non-decreasing surjections) α and β with $\alpha \colon [0, 1] \to [1, m]$, $\beta \colon [0, 1] \to [1, n]$. A pair of parametrisations establish a *matching* $\mu = (\alpha, \beta)$, so that $\pi \circ \alpha(t)$ is matched to $\sigma \circ \beta(t)$ for any $t \in [0, 1]$. Denote the cost of a matching $\mu = (\alpha, \beta)$ as $\mathrm{cost}_{\mu}(\pi, \sigma) = \max_{t \in [0,1]} \|\pi \circ \alpha(t) - \sigma \circ \beta(t)\|$. Then we can define Fréchet distance and its variants as

$$d_{\mathrm{F}}(\pi, \sigma) = \inf_{\text{matching } \mu} \mathrm{cost}_{\mu}(\pi, \sigma), \quad d_{\mathrm{F}}(\pi, \sigma) = \inf_{\text{discrete matching } \mu} \mathrm{cost}_{\mu}(\pi, \sigma),$$

$$d_{\mathrm{wF}}(\pi, \sigma) = \inf_{\text{weak matching } \mu} \mathrm{cost}_{\mu}(\pi, \sigma).$$

The discrete matching is restricted to vertices, and the weak matching is not a pair of parametrisations, but a path $(\alpha, \beta) \colon [0, 1]^2 \to [1, m] \times [1, n]$, with $\alpha(0) = 1, \alpha(1) = m$ and $\beta(0) = 1, \beta(1) = n$. In the person–dog analogy for the Fréchet distance, the best choice of parametrisations means that the person and the dog choose the best speed, and the leash length is then the largest needed leash length during the walk.

To decide whether the Fréchet distance is below some threshold $\delta > 0$, Alt and Godau [2] define a *free-space diagram*. It is a two-dimensional diagram on $[1, m] \times [1, n]$, where each point (x, y) corresponds to the pair $(\pi(x), \sigma(y))$. The point (x, y) is *free* if and only if $\|\pi(x) - \sigma(y)\| \leq \delta$. The free space is the collection of all the free points. We use a non-standard variation of this technique in Sections 3 and 4, where the diagram is in $\mathbb{R}^d \times \mathbb{R}^d$ for two curves in $\mathbb{R}^d$.

An *uncertain* point in one dimension is a set of real numbers $u \subseteq \mathbb{R}$. The intuition is that only one point from this set 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 u. In this paper, we consider two special cases of uncertain points. An *indecisive* point is a finite set of numbers $u = \{x_1, \ldots, x_\ell\}$. An *imprecise* point is a closed interval $u = [x_1, x_2]$. Note that a precise point is a special case of both indecisive and imprecise points.

Define an *uncertain curve* as a sequence of uncertain points $\mathcal{U} = \langle u_1, \ldots, u_n \rangle$. A *realisation* $\pi \Subset \mathcal{U}$ of an uncertain curve is a polygonal curve $\pi = \langle p_1, \ldots, p_n \rangle$, where each p_i is a realisation of the uncertain point u_i. For uncertain curves $\mathcal{U}$ and $\mathcal{V}$, define the lower bound and upper bound Fréchet distance. The discrete and weak Fréchet distance are defined similarly.

$$d_{\mathrm{F}}^{\min}(\mathcal{U}, \mathcal{V}) = \min_{\pi \Subset \mathcal{U}, \sigma \Subset \mathcal{V}} d_{\mathrm{F}}(\pi, \sigma), \qquad d_{\mathrm{F}}^{\max}(\mathcal{U}, \mathcal{V}) = \max_{\pi \Subset \mathcal{U}, \sigma \Subset \mathcal{V}} d_{\mathrm{F}}(\pi, \sigma).$$

3. Lower bound Fréchet distance: general approach

In this section, we consider the following decision problem.

Problem 1. Given two uncertain curves $\mathcal{U} = \langle u_1, \ldots, u_m \rangle$ and $\mathcal{V} = \langle v_1, \ldots, v_n \rangle$ in $Y = \mathbb{R}^d$ for some $d, m, n \in \mathbb{N}^+$ and a threshold $\delta > 0$, decide if $d_{\mathrm{F}}^{\min}(\mathcal{U}, \mathcal{V}) \leq \delta$.

Note that this problem formulation is general both in terms of the shape of uncertainty regions and the dimension of the problem. We propose an algorithmic framework that solves this problem. As shown previously [25], the problem is NP-hard in 2D for vertical line segments as uncertainty regions, but admits a simple dynamic program for indecisive points in 2D. So, in many uncertainty models, especially in higher dimensions, the following approach will not result in an efficient algorithm. However, our approach is general in that it can be instantiated in restricted settings, e.g. in 2D assuming that the segments of the curves can only be horizontal or vertical. The inherent complexity of the problem appears to be related to the number of directions to consider, with the infinite number in 2D without restrictions and two directions in 1D. We conjecture that in this restricted setting the approach yields a polynomial-time algorithm; verifying this and making a more general statement delineating the hardness of restricted settings are both interesting open problems. Our approach shows a straightforward way to engineer an algorithm for various restricted settings in arbitrary dimension, but we cannot make any statements about its efficiency in most settings. To illustrate the approach, we instantiate it in 1D and analyse its efficiency in Section 4. The interested reader might refer to that section for a more intuitive explanation of the approach.

First we introduce some extra notation. Recall $Y = \mathbb{R}^d$. For $i \in [m]$, denote $\mathcal{U}_i = \langle u_1, \ldots, u_i \rangle$ and $\mathcal{U}_i^* = \langle u_1, \ldots, u_i, Y \rangle$. We call $\mathcal{U}_i$ and $\mathcal{U}_i^*$ the *subcurve* and the *free subcurve* of $\mathcal{U}$ at i, respectively. Intuitively, a realisation of $\mathcal{U}_i^*$ extends a realisation of $\mathcal{U}_i$ by a single edge whose final vertex position is unrestricted. Let $S := S^{d-1}$ be the unit $(d-1)$-sphere. Denote the *direction* of the i-th edge $\pi[i:i+1]$ of a realisation π by $\mathbf{d}_i(\pi) \subseteq S$. For example, in 1D there are only two options, in 2D the directions can be picked from a unit circle, in 3D from a unit sphere, etc. In the degenerate case where the edge has length 0 (or π has no i-th edge), let $\mathbf{d}_i(\pi) = S$.

We want to find realisations $\pi \Subset \mathcal{U}$ and $\sigma \Subset \mathcal{V}$ such that π and σ have Fréchet distance at most δ. Call such a pair (π, σ) a δ-*realisation* of $(\mathcal{U}, \mathcal{V})$. Intuitively, we want to create a dynamic program on the two curves that keeps track of the

(a) The sets on a cell of a regular free-space diagram.

(b) Dependencies of the dynamic program. $a \to b$ means that a depends on b.
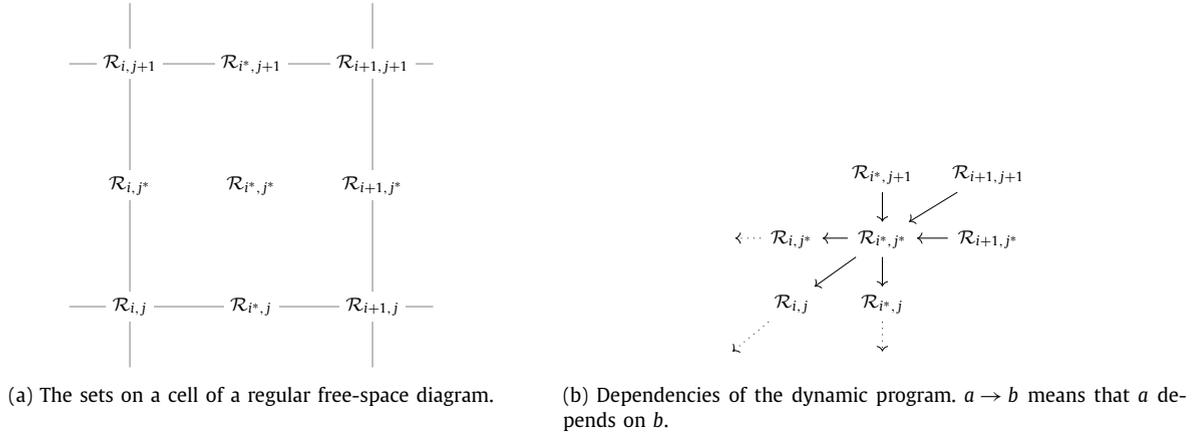
Fig. 1. Illustration for the dynamic program of Lemma 2.

possible realisations of the current uncertain points and the allowed edge directions to extend the prefix curves, so that the entire prefix curves have the Fréchet distance at most δ. Recall that two polygonal curves $\pi : [1, i] \to Y$ and $\sigma : [1, j] \to Y$ have Fréchet distance $d_F(\pi, \sigma)$ at most δ if and only if there exist *parametrisations* (non-decreasing surjections) $\alpha : [0, 1] \to [1, i]$ and $\beta : [0, 1] \to [1, j]$ such that the path $(\pi \circ \alpha, \sigma \circ \beta)$ lies in the δ-free space, defined as $\mathcal{F}_\delta = \{(p, q) \in Y \times Y \mid \|p - q\| \leq \delta\}$. For δ-close (free) subcurves of $\mathcal{U}$ at i and $\mathcal{V}$ at j, we capture their pairs of endpoints and final directions using $\mathcal{R}_{i,j}, \mathcal{R}_{i,j^*}, \mathcal{R}_{i^*,j}, \mathcal{R}_{i^*,j^*} \subseteq Y \times Y \times S \times S$:

$$
\begin{aligned}
\mathcal{R}_{i,j} &= \{(\pi(i), \sigma(j), s, t) \mid \pi \in \mathcal{U}_i, \sigma \in \mathcal{V}_j, s \in \mathbf{d}_i(\pi), t \in \mathbf{d}_j(\sigma), d_F(\pi, \sigma) \leq \delta\}, \\
\mathcal{R}_{i,j^*} &= \{(\pi(i), \sigma(j+1), s, t) \mid \pi \in \mathcal{U}_i, \sigma \in \mathcal{V}_j^*, s \in \mathbf{d}_i(\pi), t \in \mathbf{d}_j(\sigma), d_F(\pi, \sigma) \leq \delta\}, \\
\mathcal{R}_{i^*,j} &= \{(\pi(i+1), \sigma(j), s, t) \mid \pi \in \mathcal{U}_i^*, \sigma \in \mathcal{V}_j, s \in \mathbf{d}_i(\pi), t \in \mathbf{d}_j(\sigma), d_F(\pi, \sigma) \leq \delta\}, \\
\mathcal{R}_{i^*,j^*} &= \{(\pi(i+1), \sigma(j+1), s, t) \mid \pi \in \mathcal{U}_i^*, \sigma \in \mathcal{V}_j^*, s \in \mathbf{d}_i(\pi), t \in \mathbf{d}_j(\sigma), d_F(\pi, \sigma) \leq \delta\}.
\end{aligned}
$$

Note that for $\pi \in \mathcal{U}_i$, i is the final vertex, so $\mathbf{d}_i(\pi) = S$. Therefore, $\mathcal{R}_{i,j}$ captures the reachable subset of $Y \times Y$ for the realisations of the last points of the prefixes, and the two other dimensions contain all points from S to capture that we may proceed in any allowed direction. The reachable subset of $Y \times Y$ here refers to the subset of $Y \times Y$ where the realisations of the last points may be, taking into account the entire prefixes of the curves. The set $\mathcal{R}_{i^*,j}$ captures the reachable subset of $Y \times Y$ for the point in the parametrisation where we are between vertices i and $i + 1$ on $\mathcal{U}$ and at j on $\mathcal{V}$; we have not restricted the range to u_{i+1} yet. The allowed directions for parameter s now depend on how we reached this point in the parametrisation, since segments connecting realisations are straight line segments, and the direction needs to be kept consistent once chosen. From this description the reader can deduce what the other sets capture by symmetry. See also Fig. 1a, where the sets are positioned as in a regular free-space diagram, replacing the edges, vertices, and cells.

To solve the decision problem, we must decide whether $\mathcal{R}_{m,n}$ is non-empty. If so, then there are realisations of $\mathcal{U}_m \equiv \mathcal{U}$ and $\mathcal{V}_n \equiv \mathcal{V}$ that, after placing all the previous realisations, result in curves with the Fréchet distance at most δ. We compute $\mathcal{R}_{\cdot,\cdot}$ using dynamic programming. We illustrate the propagation dependencies in Fig. 1b and make them explicit in Lemma 2.

Lemma 2. *Let* $\star(A) := \{(p + \lambda s, q + \mu t, s, t) \mid (p, q, s, t) \in A \text{ and } \lambda, \mu \geq 0\}$. *We have*

$$
\begin{aligned}
\mathcal{R}_{\cdot,0} &= \mathcal{R}_{0,\cdot} = \emptyset, \\
\mathcal{R}_{i+1,j^*} &= \{(p, q, s, t) \in u_{i+1} \times \phantom{v_{}} Y \phantom{_{+1}} \times S \times S \mid (p, q, \cdot, t) \in \mathcal{R}_{i^*,j^*}\}, \\
\mathcal{R}_{i^*,j+1} &= \{(p, q, s, t) \in \phantom{u_{+1}} Y \times v_{j+1} \times S \times S \mid (p, q, s, \cdot) \in \mathcal{R}_{i^*,j^*}\}, \\
\mathcal{R}_{i+1,j+1} &= \{(p, q, s, t) \in u_{i+1} \times v_{j+1} \times S \times S \mid (p, q, \cdot, \cdot) \in \mathcal{R}_{i^*,j^*}\}, \\
\mathcal{R}_{0^*,0^*} &= \mathcal{F}_\delta \times S \times S, \\
\mathcal{R}_{i^*,j^*} &= (\mathcal{F}_\delta \times S \times S) \cap \star(\mathcal{R}_{i,j} \cup \mathcal{R}_{i^*,j} \cup \mathcal{R}_{i,j^*}) \quad \text{for } i > 0 \text{ or } j > 0.
\end{aligned}
$$

Proof. The first equation holds because the empty function has no parametrisation, so the Fréchet distance of any pair of realisations is infinite. The equation for $\mathcal{R}_{i+1,j^*}$ holds because for $\pi \in \mathcal{U}_{i+1}$, $\mathbf{d}_{i+1}(\pi) = S$, and the only additional constraint that a realisation of $\mathcal{U}_{i+1}$ has over one of $\mathcal{U}_i^*$ is that the final vertex lies in u_{i+1}. Using symmetric properties on $\mathcal{V}$, we obtain the equations for $\mathcal{R}_{i^*,j+1}$ and $\mathcal{R}_{i+1,j+1}$. The equation for $\mathcal{R}_{0^*,0^*}$ concerns curves π and σ consisting of a single vertex, so $\mathbf{d}_0(\pi) = \mathbf{d}_0(\sigma) = S$, and $d_F(\pi, \sigma) \leq \delta$ if and only if $(\pi(1), \sigma(1)) \in \mathcal{F}_\delta$. The equation for $\mathcal{R}_{i^*,j^*}$ remains. First we show that
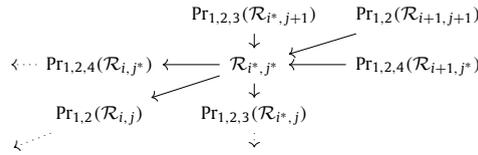
Fig. 2. Simplified dependencies with projections as follows from Lemma 2.

the right-hand side is contained in $\mathcal{R}_{i^*,j^*}$. Suppose that π and σ form a witness for $(p,q,s,t) \in \mathcal{R}_{i,j} \cup \mathcal{R}_{i^*,j} \cup \mathcal{R}_{i,j^*}$. We obtain realisations $\pi^* \Subset \mathcal{U}_i^*$ and $\sigma^* \Subset \mathcal{V}_j^*$ by extending the last edge of π and σ in the direction it is already going (or adding a new edge in an arbitrary direction if $\pi \Subset \mathcal{U}_i$ or $\sigma \Subset \mathcal{V}_j$), to $(p + \lambda s, q + \mu t)$. If $(p + \lambda s, q + \mu t) \in \mathcal{F}_\delta$, then, by convexity of $\mathcal{F}_\delta$, the extensions of the last edges have Fréchet distance at most δ (since the points at which the extension starts have distance at most δ), so $(p + \lambda s, q + \mu t, s, t) \in \mathcal{R}_{i^*,j^*}$. Conversely, we show that the right-hand side contains $\mathcal{R}_{i^*,j^*}$. Let $\pi^* \Subset \mathcal{U}_i^*$ and $\sigma^* \Subset \mathcal{V}_j^*$ together with parametrisations $\alpha : [0,1] \to [1, i+1]$ and $\beta : [0,1] \to [1, j+1]$ form a witness that $(p,q,s,t) \in \mathcal{R}_{i^*,j^*}$. Then, for any $x \in [0,1]$, the restrictions π_x of π^* and σ_x of σ^* to the domains $[1, \alpha(x)]$ and $[1, \beta(x)]$ have Fréchet distance at most δ. Because α and β are non-decreasing surjections, whenever $i > 0$ or $j > 0$, there exists some x such that

1. $\alpha(x) = i$ and $\beta(x) = j$, in which case $\pi_x \Subset \mathcal{U}_i$ and $\sigma_x \Subset \mathcal{V}_j$, or
2. $\alpha(x) > i$ and $\beta(x) = j$, in which case $\pi_x \Subset \mathcal{U}_i^*$ and $\sigma_x \Subset \mathcal{V}_j$, or
3. $\alpha(x) = i$ and $\beta(x) > j$, in which case $\pi_x \Subset \mathcal{U}_i$ and $\sigma_x \Subset \mathcal{V}_j^*$.

Note that if $i = 0$, only the second case applies, and if $j = 0$, only the third case applies. In each case, the last edge of π^* and σ^* extends the i-th and j-th edge of π_x and σ_x, respectively. So (π_x, σ_x) forms a witness that (p,q,s,t) is contained in the right-hand side. □

Simplifying the approach. Due to their dimension, the sets above can be impractical to work with. However, for the majority of these sets, at least one of the factors S carries no additional information, as formulated below. Denote by Pr_c the projection map of the c-th component, so that $\mathrm{Pr}_1 : (p,q,s,t) \mapsto p$, and in general $\mathrm{Pr}_{c_1,\ldots,c_k}(x) = (\mathrm{Pr}_{c_1}(x), \ldots, \mathrm{Pr}_{c_k}(x))$. The equations of Lemma 2 imply the equivalences

$$(p,q,s,t) \in \mathcal{R}_{i,j} \iff (p,q) \in \mathrm{Pr}_{1,2}(\mathcal{R}_{i,j}),$$

$$(p,q,s,t) \in \mathcal{R}_{i^*,j} \iff (p,q,s) \in \mathrm{Pr}_{1,2,3}(\mathcal{R}_{i^*,j}),$$

$$(p,q,s,t) \in \mathcal{R}_{i,j^*} \iff (p,q,t) \in \mathrm{Pr}_{1,2,4}(\mathcal{R}_{i,j^*}).$$

Consequently, to find $\mathcal{R}_{i,j}$, $\mathcal{R}_{i^*,j}$, and $\mathcal{R}_{i,j^*}$, it suffices to compute the projections above. This simplifies the prior dependencies as shown in Fig. 2.

$$\begin{aligned}
\mathrm{Pr}_{1,2}(\mathcal{R}_{i+1,j+1}) &= (u_{i+1} \times v_{j+1}) \cap \mathrm{Pr}_{1,2}(\mathcal{R}_{i^*,j^*}) \\
&= (u_{i+1} \times \quad Y \quad) \cap \mathrm{Pr}_{1,2}(\mathcal{R}_{i^*,j+1}) \\
&= (\quad Y \quad \times v_{j+1}) \cap \mathrm{Pr}_{1,2}(\mathcal{R}_{i+1,j^*}).
\end{aligned}$$

Instantiating the approach. The dynamic program of Lemma 2 can naturally be adapted to constrained realisations whose edge directions are to be drawn from a subset $S' \subseteq S^{d-1}$, by replacing S by S', so the framework can be used for restricted settings in 2D. For $S = S^{d-1}$ the complexity of $\mathcal{R}_{i,j}$ can be exponential, so it can be useful to restrict the problem.

We can look at the construction used to prove NP-hardness of the problem in 2D [25] as an example for our approach. There the curve $\mathcal{V}$ is precise, so each v_j is a single point and each t is predetermined, and curve $\mathcal{U}$ consists of uncertainty regions that are vertical line segments, so each u_i has a fixed x-coordinate and a range of y-coordinates. If we now exclude the fixed values from our propagation, we get to track pairs (y,s) of the feasible y-coordinates on the current interval and the directions. We start with a single region. The hardness construction uses gadgets on the precise curve to force the uncertain curve to go through certain points. In our approach, this means that we keep restricting the set of feasible directions while passing by vertices on $\mathcal{V}$, and eventually each point in the starting region gives rise to two disjoint reachable points on one of the following uncertainty regions. So we can use our algorithm to correctly track the feasible y-coordinates through the construction; however, we would need to keep track of regions of exponential complexity, which is, predictably, inefficient. Therefore, it is important to analyse the complexity of the propagated regions to determine whether our approach gives rise to an efficient algorithm. To illustrate our approach, we use it in the 1D case to devise an efficient algorithm in Section 4.

4. Lower bound Fréchet distance: one dimension

In this section we instantiate the approach of Section 3 in 1D and analyse its efficiency. We first show the formal definitions that result from this process, and then give some intuition for how the resulting algorithm works in 1D.

In our case, $S = S^0$, so there are only two directions: positive x-direction and negative x-direction. We make use of the projections interpretation and split the projections into two regions based on the value of the relevant direction; then all the regions we maintain are in $\mathbb{R}^2$ and have a geometric interpretation as feasible combinations of realisations of the last uncertain points on the prefixes of the curves. We omit $\mathcal{R}_{i,j}$ from our computations except for checking whether $\mathcal{R}_{m,n}$ is non-empty. As follows from the definition of the sets, $\mathcal{R}_{i,j} \subseteq \mathcal{R}_{i^*,j}$ and $\mathcal{R}_{i,j} \subseteq \mathcal{R}_{i,j^*}$, so we can simplify the computation of $\mathcal{R}_{i^*,j^*}$, and then we do not need the explicit computation of $\mathcal{R}_{i,j}$. Furthermore, we do not compute any of $\mathcal{R}_{i^*,j^*}$ explicitly, opting instead to substitute them into the relevant expressions. Therefore, we maintain the sets $\mathcal{R}_{i,j^*}$ and $\mathcal{R}_{i^*,j}$, splitting each into two based on the relevant direction. Based on our earlier free space cell interpretation (see Fig. 1a), call the directions along $\mathcal{U}$ *right* and *left* and call the directions along $\mathcal{V}$ *up* and *down*. We then have the following mapping from the regions of Section 3 to the simpler intuitive regions of this section.

$$U_{i,j} = \{(p,q) \mid (p,q,\cdot,t) \in \mathcal{R}_{i,j^*} \wedge t = 1\},$$
$$D_{i,j} = \{(p,q) \mid (p,q,\cdot,t) \in \mathcal{R}_{i,j^*} \wedge t = -1\},$$
$$R_{i,j} = \{(p,q) \mid (p,q,s,\cdot) \in \mathcal{R}_{i^*,j} \wedge s = 1\},$$
$$L_{i,j} = \{(p,q) \mid (p,q,s,\cdot) \in \mathcal{R}_{i^*,j} \wedge s = -1\}.$$

It is also easier to express the $\star$ operator of Lemma 2 in this setting. Depending on which of the directions we consider fixed because we already committed to a direction, the propagation through the cell interior works by adding either a quadrant or a half-plane to every point in the starting region; we can denote this with a Minkowski sum. Based on these considerations, we give the following simplified definition.

Formal definition. Denote $\mathbb{R}^{\leq 0} = \{x \in \mathbb{R} \mid x \leq 0\}$ and $\mathbb{R}^{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$. Consider the space $\mathbb{R} \times \mathbb{R}$ of the coordinates of the two curves in 1D. We are interested in what is feasible within the *interval free space,* which in this space turns out to be a band around the line $y = x$ of width 2δ in L_1-distance called $\mathcal{F}_\delta$. For notational convenience, define the following regions (see Fig. 3):

$$\mathcal{F}_\delta = \{(x,y) \in \mathbb{R}^2 \mid |x - y| \leq \delta\}, \qquad I_i = (u_i \times \mathbb{R}) \cap \mathcal{F}_\delta, \qquad J_j = (\mathbb{R} \times v_j) \cap \mathcal{F}_\delta.$$

We use dynamic programming, similarly to the standard free-space diagram for the Fréchet distance; however, we propagate reachable subsets of uncertainty regions on the two curves. The propagation in the interval-free-space diagram consists of starting anywhere within the current region and going in restricted directions, since we need to distinguish between going in the positive and the negative x-direction along both curves. We introduce the notation for restricting the directions in the form of quadrants, half-planes, and slabs:

$$Q_{LD} = \mathbb{R}^{\leq 0} \times \mathbb{R}^{\leq 0}, \quad Q_{LU} = \mathbb{R}^{\leq 0} \times \mathbb{R}^{\geq 0}, \quad Q_{RD} = \mathbb{R}^{\geq 0} \times \mathbb{R}^{\leq 0}, \quad Q_{RU} = \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0},$$
$$H_L = \mathbb{R}^{\leq 0} \times \mathbb{R}, \qquad H_R = \mathbb{R}^{\geq 0} \times \mathbb{R}, \qquad H_D = \mathbb{R} \times \mathbb{R}^{\leq 0}, \qquad H_U = \mathbb{R} \times \mathbb{R}^{\geq 0}.$$
$$S_L = \mathbb{R}^{\leq 0} \times \{0\}, \qquad S_R = \mathbb{R}^{\geq 0} \times \{0\}, \qquad S_D = \{0\} \times \mathbb{R}^{\leq 0}, \qquad S_U = \{0\} \times \mathbb{R}^{\geq 0}.$$

We introduce notation for propagating in these directions from a region by taking the appropriate Minkowski sum, denoted with $\oplus$. For $a, b \in \{L, R, U, D\}$ and a region X,

$$X^a = X \oplus H_a, \qquad X^{ab} = X \oplus Q_{ab}, \qquad X^{a0} = X \oplus S_a.$$

Now we can discuss the propagation. We start with the base case, where we compute the feasible combinations for the boundaries of the cells of a regular free-space diagram corresponding to the first vertex on one of the curves. For the sake of better intuition we do not use $(0,0)$ as the base case here. So, we fix our position to the first vertex on $\mathcal{U}$ and see how far we can go along $\mathcal{V}$; and the other way around. As we are bound to the same vertex on $\mathcal{U}$, as we go along $\mathcal{V}$, we keep restricting the feasible realisations of u_1. Thus, we cut off unreachable parts of the interval as we propagate along the other curve. We do not care about the direction we were going in after we cross a vertex on the curve where we move. So, if we stay at u_1 and we cross over v_j, then we are free to go both in the negative and the positive direction of the x-axis to reach a realisation of v_{j+1}. We get the following expressions, where $U_{i,j}$ denotes the propagation upwards from the pair of vertices u_i and v_j and propagation down, left, and right is defined similarly:
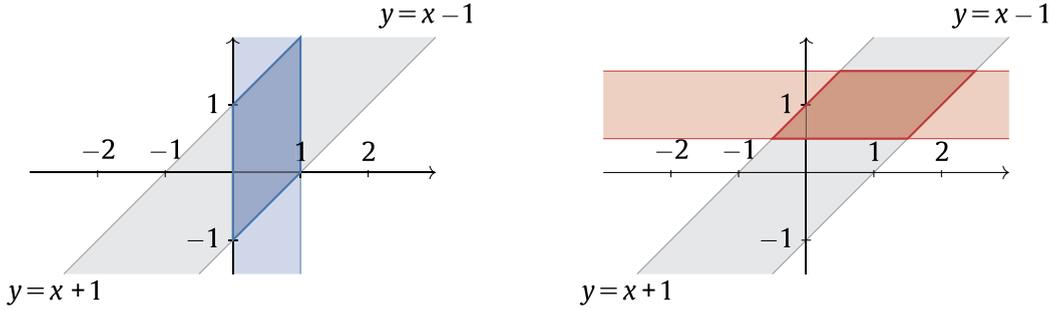
Fig. 3. On the left, the filled region is $I_i = (u_i \times \mathbb{R}) \cap \mathcal{F}_\delta$ for $u_i = [0, 1]$. On the right, the filled region is $J_j = (\mathbb{R} \times v_j) \cap \mathcal{F}_\delta$ for $v_j = [0.5, 1.5]$. In both cases $\delta = 1$.
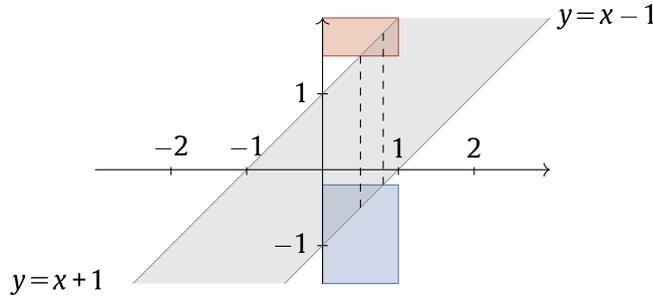

Fig. 4. An interval-free-space diagram for $u_i = [0, 1]$, $v_j = [-1.5, -0.2]$, and $v_{j+1} = [1.5, 2]$ with $\delta = 1$. Note that the feasible realisations for u_i are $[0.5, 0.8]$.

$$U_{1,1} = (I_1 \cap J_1)^{U0} \cap \mathcal{F}_\delta, \qquad D_{1,1} = (I_1 \cap J_1)^{D0} \cap \mathcal{F}_\delta,$$
$$R_{1,1} = (I_1 \cap J_1)^{R0} \cap \mathcal{F}_\delta, \qquad L_{1,1} = (I_1 \cap J_1)^{L0} \cap \mathcal{F}_\delta,$$
$$U_{1,j+1} = ((U_{1,j} \cup D_{1,j}) \cap J_{j+1})^{U0} \cap \mathcal{F}_\delta, \qquad D_{1,j+1} = ((U_{1,j} \cup D_{1,j}) \cap J_{j+1})^{D0} \cap \mathcal{F}_\delta,$$
$$R_{i+1,1} = ((R_{i,1} \cup L_{i,1}) \cap I_{i+1})^{R0} \cap \mathcal{F}_\delta, \qquad L_{i+1,1} = ((R_{i,1} \cup L_{i,1}) \cap I_{i+1})^{L0} \cap \mathcal{F}_\delta.$$

Once the boundary regions are computed, we can proceed with propagation:

$$U_{i+1,j} = (U_{i,j}^U \cup R_{i,j}^{RU} \cup L_{i,j}^{LU}) \cap I_{i+1}, \qquad D_{i+1,j} = (D_{i,j}^D \cup R_{i,j}^{RD} \cup L_{i,j}^{LD}) \cap I_{i+1},$$
$$R_{i,j+1} = (R_{i,j}^R \cup U_{i,j}^{RU} \cup D_{i,j}^{RD}) \cap J_{j+1}, \qquad L_{i,j+1} = (L_{i,j}^L \cup U_{i,j}^{LU} \cup D_{i,j}^{LD}) \cap J_{j+1}.$$

To solve the decision problem, we check if the last vertex combination is feasible:

$$((R_{m-1,n} \cup L_{m-1,n}) \cap I_m) \cup ((U_{m,n-1} \cup D_{m,n-1}) \cap J_n) \neq \emptyset.$$

Intuition. If the consecutive regions are always disjoint, we do not need to consider the possible directions: we always know (in 1D) where the next region is, and thus what direction we take. However, if the regions may overlap, it may be that for different realisations of a curve a segment goes in the positive or in the negative direction. The propagation we compute is based on the parameter space where we look at whether we have reached a certain vertex on each curve yet, inspired by the traditional free-space diagram. It may be that we pass by several vertices on, say, $\mathcal{V}$ while moving along a single segment on $\mathcal{U}$. The direction we choose on $\mathcal{U}$ needs to be kept consistent as we compute the next regions, otherwise we might include realisations that are invalid as feasible solutions. Therefore, we need to keep track of the chosen direction, reflected by the pair (s, t) in the general approach and the separate sets in this section. Otherwise, these regions in 1D are simply the feasible pairs of realisations of the last vertices on the prefixes of the curves.

It is helpful to think of the approach in terms of interval-free-space diagrams. Consider a combination of specific vertices on the two curves, say, u_i and v_j, and suppose that we want to stay at u_i but move to v_{j+1}. Which realisations of u_i, v_j, and v_{j+1} can we pick that allow this move to stay within the 2δ-band?

Suppose the x-coordinate of the diagram corresponds to the x-coordinate of $\mathcal{U}$. Then we may pick a realisation for u_i anywhere in the vertical slab corresponding to the uncertainty interval for u_i, namely, in the slab $u_i \times \mathbb{R}$. The fixed realisation for u_i would then yield a vertical line. Now suppose the y-coordinate of the diagram corresponds to the x-coordinate of $\mathcal{V}$. For v_j, picking a realisation corresponds to picking a horizontal line from the slab $\mathbb{R} \times v_j$; for v_{j+1}, it corresponds to picking a horizontal line from $\mathbb{R} \times v_{j+1}$. Picking a realisation for the pair (u_i, v_j) thus corresponds to a point in $u_i \times v_j$.
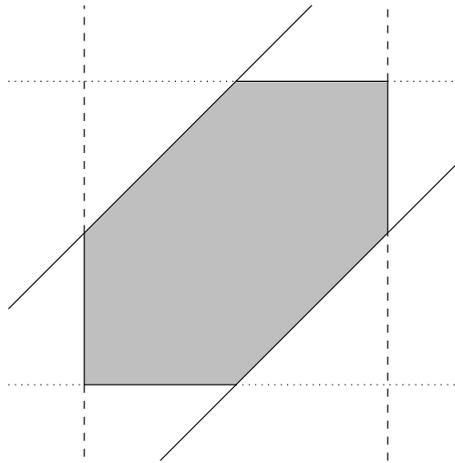
Fig. 5. An example simple region. We get less general ones by setting any side length to 0.

Of course, we may only maintain the matching as long the distance between the matched points is at most δ. For a fixed point on $\mathcal{U}$, this corresponds to a 2δ window for the coordinates along $\mathcal{V}$. Therefore, the allowed matchings are contained within the band defined by $y = x \pm \delta$, and when we pick the realisations for (u_i, v_j), we only pick points from $u_i \times v_j$ for which $|y - x| \leq \delta$ holds.

As we consider the propagation to v_{j+1}, note that we may not move within u_i, so the allowed realisations for the pair (u_i, v_{j+1}) are limited. In particular, we can find that region by taking the subset of $u_i \times v_{j+1}$ for which $|y - x| \leq \delta$ holds and restricting the x-coordinate further to be feasible for the pair (u_i, v_j). See Fig. 4 for an illustration of this. In this figure, we know that v_{j+1} lies above v_j; if we did not know that, we would have to attempt propagation both upwards and downwards. For the second curve, the same holds.

Complexity. We now discuss the complexity of the regions we are propagating to analyse the efficiency of the algorithm presented above. We will perform the following steps:

1. Define complexity of the regions and establish the complexity of the base case.
2. Study the possible complex regions that can arise from all simple regions.
3. Study what happens to the complex regions as we propagate and conclude that the complexity is bounded by a constant.

The boundaries of the regions are always horizontal, vertical, or coincide with the boundaries of $\mathcal{F}_\delta$. A region can be thus represented as a union of (possibly unbounded) axis-aligned rectangular regions, further intersected with the interval free space. We define the *complexity* of a region as the minimal required number of such rectangular regions. Define a *simple* region as a region of complexity at most 1. Observe that a simple region is necessarily convex; and a non-simple region has to be non-convex. The illustration in Fig. 5 shows the most general example of a simple region. An empty region is also a simple region. To enumerate the possible non-simple regions, we need to examine where higher region complexity may come from in our algorithm. To that aim, we first prove some simple statements about the propagation procedure.

First, we discuss the complexity of the regions we can get in the base case of the propagation.

Lemma 3. *For all $i \in [m-1]$ and $j \in [n-1]$, regions $U_{1,j}$, $D_{1,j}$, $R_{i,1}$, and $L_{i,1}$ are simple.*

Proof. Consider first the intersection $I_1 \cap J_1$. It is the intersection of a vertical slab, a horizontal slab, and the diagonal slab (interval free space). All three are convex sets, hence their intersection is also convex and uses only vertical, horizontal, and diagonal line segments, so the result is a simple region. To obtain $U_{1,1}$, $D_{1,1}$, $R_{1,1}$, and $L_{1,1}$, we take the Minkowski sum of the region with the corresponding half-slab. Both are convex, so the result again is convex; we then intersect it with the interval free space again, getting a simple region.

Now assume that $U_{1,j}$ is simple; we show that $U_{1,j+1}$ is simple. Note that for some region X, $U_{1,j} = X^{U0} \cap \mathcal{F}_\delta$ and $D_{1,j} = X^{D0} \cap \mathcal{F}_\delta$. Then

$$U_{1,j} \cup D_{1,j} = (X^{U0} \cup X^{D0}) \cap \mathcal{F}_\delta = (X \oplus (\{0\} \times \mathbb{R})) \cap \mathcal{F}_\delta .$$

So, we get a vertical slab the width of X, intersected with $\mathcal{F}_\delta$, so the result is convex. We then intersect the region with the simple region J_{j+1}; take Minkowski sum with a slab; and again intersect with the interval free space. Clearly, the result is convex and uses only the allowed boundaries, so we get a simple region.
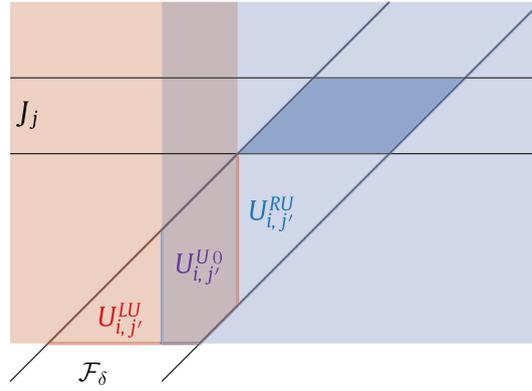
Fig. 6. Propagation of $U_{i,j'}$ to $L_{i,j}$ and $R_{i,j}$. Note $J_j \subset \mathcal{F}_\delta$. Observe that $U_{i,j'}^{LU} \cap U_{i,j'}^{RU} = U_{i,j'}^{U0}$, and if $U_{i,j'}^{U0} \cap J_j = \emptyset$ but J_j does not lie below $U_{i,j'}$, then $U_{i,j'}^{LU} \cap J_j = \emptyset$ and $U_{i,j'}^{RU} \cap J_j = J_j$, so one of the regions is empty and the other covers the entire feasible region.

The argument for $D_{1,j}$ is symmetric; the arguments for $R_{i,1}$ and $L_{i,1}$ are equally straightforward. Hence, all the regions we get in the base case are simple. □

To proceed, we need to make the relation in pairs (U, D) and (R, L) clear, so we know where the complexity may come from. Denote a half-plane with a vertical or horizontal boundary starting at coordinate s and going in direction X by H_s^X. For example, a half-plane bounded on the left by the line $x = 2$ is denoted H_2^R.

Lemma 4. *Take two imprecise curves $\mathcal{U}$ and $\mathcal{V}$ of lengths m and n, respectively, and let $i \in [m-1]$ and $j \in [n-1]$. Consider the pair $R_{i,j}, L_{i,j}$ and assume the regions are simple. Then exactly one of the following options holds:*

1. *$R_{i,j} = L_{i,j} = \emptyset$, so both regions are empty;*
2. *$R_{i,j} = J_j \cap H_s^R \neq \emptyset \wedge L_{i,j} = \emptyset$ for some s, so one region is empty and the other spawns the entire feasible range, except that it may be cut with a vertical line on the left;*
3. *$L_{i,j} = J_j \cap H_s^L \neq \emptyset \wedge R_{i,j} = \emptyset$ for some s, so one region is empty and the other spawns the entire feasible range, except that it may be cut with a vertical line on the right;*
4. *$L_{i,j} \cap R_{i,j} \neq \emptyset$, so both regions are non-empty, and they intersect.*

We can make the same statement for the pair $U_{i,j}, D_{i,j}$, replacing the half-planes with H_s^U and H_s^D.

Proof. We show the statement for the pair $R_{i,j}, L_{i,j}$. We prove the statement by induction on j. First of all, for $j = 1$ we know that either both regions are empty (case 1), or they are both non-empty and intersect (case 4), showing the claim. So let $j = j' + 1$ for the rest of the proof and assume that the lemma holds for the pair $R_{i,j'}, L_{i,j'}$.

We go over the possible combinations of the previous regions that are combined in the propagation and show that for any such combination we end up in one of the cases. Recall that $R_{i,j} = R_{i,j'+1} = (R_{i,j'}^R \cup U_{i,j'}^{RU} \cup D_{i,j'}^{RD}) \cap J_{j'+1}$. Similarly, $L_{i,j} = (L_{i,j'}^L \cup U_{i,j'}^{LU} \cup D_{i,j'}^{LD}) \cap J_{j'+1}$. Consider the following cases:

- $U_{i,j'} \neq \emptyset$. Note that $U_{i,j'}^{LU} \cap U_{i,j'}^{RU} = U_{i,j'}^{U0}$, so a vertical half-slab from a lower boundary. If $U_{i,j'}^{U0} \cap J_{j'+1} \neq \emptyset$, then both $L_{i,j}$ and $R_{i,j}$ are non-empty and intersect, landing in case 4. Otherwise, suppose $U_{i,j'}^{U0} \cap J_{j'+1} = \emptyset$. This intersection can be empty due to two reasons. Firstly, $U_{i,j'}^{U0}$ may lie entirely above $J_{j'+1}$. Then $U_{i,j'}^{LU} \cap J_{j'+1} = U_{i,j'}^{RU} \cap J_{j'+1} = \emptyset$, so $U_{i,j'}$ does not contribute anything to either region; this case is considered below. Secondly, $U_{i,j'}^{U0}$ may lie entirely to the left of $J_{j'+1}$. Then we get the situation shown in Fig. 6: it must be that $U_{i,j'}^{LU} \cap J_{j'+1} = \emptyset$ and $U_{i,j'}^{RU} \cap J_{j'+1} = J_{j'+1}$. This means, in particular, that $R_{i,j} = J_{j'+1} = J_j$. It might be that $L_{i,j}$ and $R_{i,j}$ are both non-empty; as $L_{i,j} \subseteq J_j$, they intersect, and so we end up in case 4. Otherwise, $L_{i,j}$ must be empty, ending up in case 2. So, whenever $U_{i,j'}$ contributes, we end up in one of the cases.
- $D_{i,j'} \neq \emptyset$. We can make arguments symmetric to the previous case, landing us in either case 4 or case 3. If $D_{i,j'}$ does not contribute to either region, we consider the next case.
- Neither $U_{i,j'}$ nor $D_{i,j'}$ contribute to $L_{i,j}$ or $R_{i,j}$, meaning we can simplify the expressions to $R_{i,j'+1} = R_{i,j'}^R \cap J_{j'+1}$ and $L_{i,j'+1} = L_{i,j'}^L \cap J_{j'+1}$. We use the induction hypothesis and distinguish between the cases for the pair $R_{i,j'}, L_{i,j'}$. Starting in case 1, we get that $L_{i,j} = R_{i,j} = \emptyset$, ending up in case 1. Starting in case 2, we get $L_{i,j} = \emptyset$, and $R_{i,j} = R_{i,j'+1} = J_{j'+1} \cap R_{i,j'}^R$. Observe that $R_{i,j'}^R$ is a half-plane that can be denoted by H_s^R for some appropriate s; depending on whether
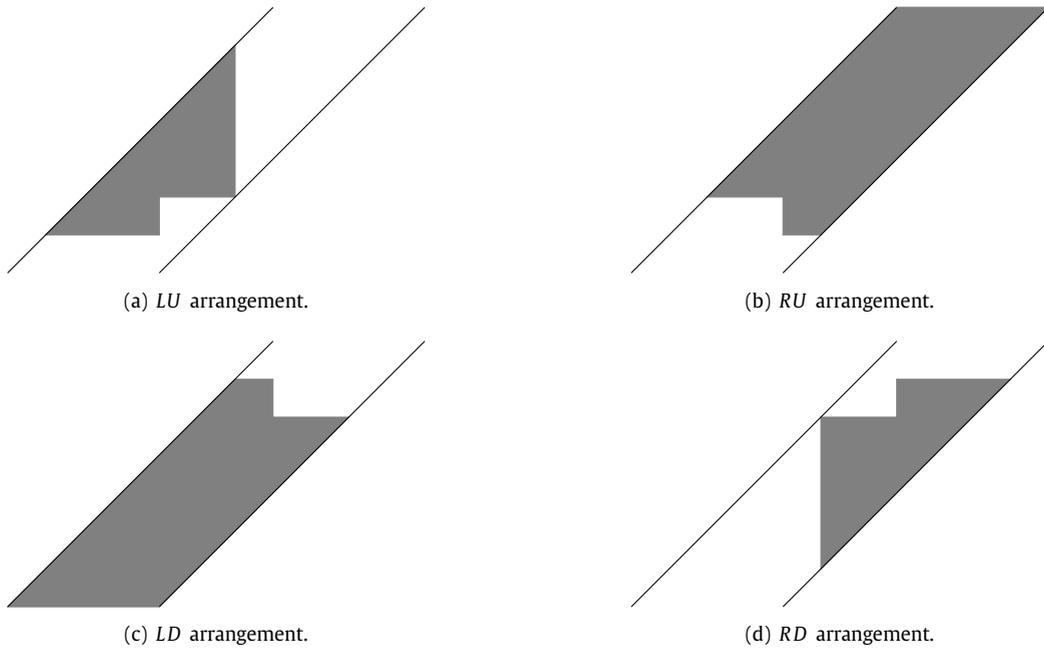
(a) *LU* arrangement.

(b) *RU* arrangement.

(c) *LD* arrangement.

(d) *RD* arrangement.

Fig. 7. All possible combinations for a single-step staircase. Each can be further intersected by a vertical or horizontal slab (I_i or J_j) or shifted so that the boundary is affected by $\mathcal{F}_\delta$.

the intersection is empty, we end in either in case 1 or in case 2. Starting in case 3 is symmetric and lands us in either case 1 or case 3. If we start in case 4, then the half-planes $R^R_{i,j'}$ and $L^L_{i,j'}$ intersect, and so for the pair $R_{i,j}$, $L_{i,j}$, we end up in case 4; or in case 2 or 3 if $L^R_{i,j'} \cap J_{j'+1}$ or $R^R_{i,j'} \cap J_{j'+1}$ is empty.

This covers all the cases. By induction, we conclude that the lemma holds. The proof for U, D is symmetric. $\square$

Let us now introduce the higher complexity regions.

Definition 5. A *staircase with k steps* is an otherwise simple region with k cut-outs on the same side of the region, each consisting of a single horizontal and a single vertical segments, introducing higher complexity. All the options for a staircase with one step (regions of complexity 2) are illustrated in Fig. 7.

We should note that a staircase with k steps, when intersected with $\mathcal{F}_\delta$, can yield up to $k+1$ disjoint simple regions. More specifically, every step that extends outside $\mathcal{F}_\delta$ splits a staircase of k steps into two staircases of at most $k-1$ steps in total.

We make the following observation relating the regions in pairs $R_{i,j}$, $L_{i,j}$ and $U_{i,j}$, $D_{i,j}$.

Lemma 6. *Take two imprecise curves $\mathcal{U}$ and $\mathcal{V}$ of lengths m and n, respectively, and let $i \in [m-1]$ and $j \in [n-1]$. Consider the pair $R_{i,j}$, $L_{i,j}$ and assume both regions are non-empty. If $j = 1$, the regions have the same y-coordinate for their lower and upper boundaries. If $j = j' + 1$ and the regions $U_{i,j'}$, $D_{i,j'}$ are simple, then the union $R_{i,j} \cup L_{i,j}$ is either simple or a staircase with one step. Furthermore, both $R_{i,j}$ and $L_{i,j}$ are either simple or staircases with one step.*

Proof. First of all, for $j = 1$, Lemma 3 implies that $R_{i,j}$ and $L_{i,j}$ are simple; furthermore, the propagation starts from the same region, so the y-range is the same and the statement holds. For the rest of the proof assume that $j = j' + 1$ and regions $U_{i,j'}$ and $D_{i,j'}$ are simple.

Consider the region $R_{i,j} = (R^R_{i,j'} \cup U^{RU}_{i,j'} \cup D^{RD}_{i,j'}) \cap J_j$. In principle, the union of the two quadrants may create a staircase with a single step. However, as the reader may verify, adding the half-plane to the union cannot add a step, since doing so would require a horizontal ray forming the top or the bottom boundary of the union of quadrants, which is impossible. Symmetrical arguments can be made for $L_{i,j}$. So, under the given assumptions the regions are always either simple or staircases with one step. Figs. 8a and 8b show some examples.

Now consider the union of regions $R_{i,j} \cup L_{i,j}$:

$$R_{i,j} \cup L_{i,j} = (R^R_{i,j'} \cup L^L_{i,j'} \cup U^U_{i,j'} \cup D^D_{i,j'}) \cap J_j.$$
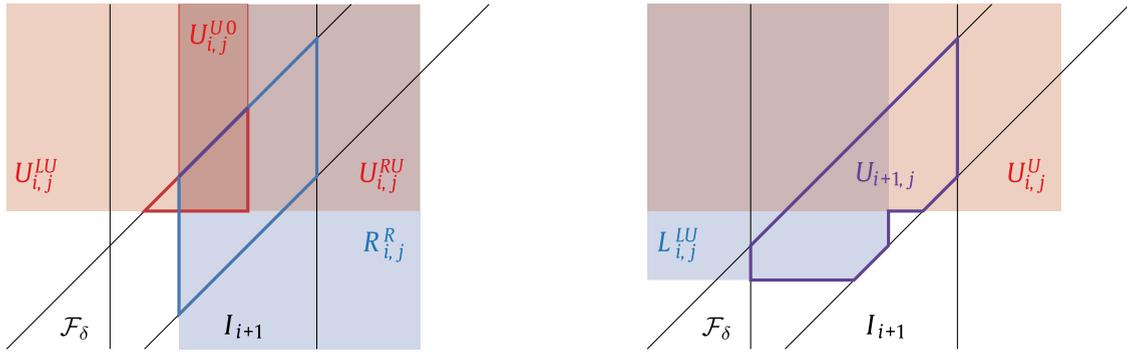
(a) Taking the union of $R^R_{i,j}$ and $U^{RU}_{i,j}$ creates a simple region. The other region is also simple, but the union of resulting regions is a staircase.

(b) Taking the union of $L^{LU}_{i,j}$ and $U^U_{i,j}$ creates a staircase. Intersection with I_{i+1} preserves it: see the coloured outline of the resulting region for $U_{i+1,j}$.

Fig. 8. Examples of staircase arrangements.

The only source of higher complexity is the union operator in the propagation. This is the union of four half-planes. If both $R_{i,j'}$ and $L_{i,j'}$ are non-empty, we know from Lemma 4 that they intersect, so $J_j \subseteq R^R_{i,j'} \cup L^L_{i,j'} = \mathbb{R}^2$. The same holds for the pair $U_{i,j'}$, $D_{i,j'}$. Now assume that at least one region from each pair is empty, say, $L_{i,j'}$ and $D_{i,j'}$. If one more region is empty, then one of $L_{i,j}$, $R_{i,j}$ is empty, which contradicts our assumption. Note that the union of two half-planes with perpendicular boundaries, intersected with a horizontal strip and the interval free space, can create a staircase with one step. In our particular setting we get the staircase in the RU arrangement, shown in Fig. 7b. Other choices for empty regions will give one of the other arrangements of Fig. 7. There are no other options, so the statement about the union $R_{i,j} \cup L_{i,j}$ is proven. □

Now consider the propagation when we start from not necessarily simple regions or regions that do not match in their y-range (or x-range), as described in Lemma 6. Consider the complexity contribution when propagating across a cell—say, $U_{i,j}$ to $U_{i+1,j}$. To perform the propagation, we take

$$U^U_{i,j} = U_{i,j} \oplus H_U = U_{i,j} \oplus (\mathbb{R} \times \mathbb{R}^{\geq 0}).$$

From the definition of the Minkowski sum, it is easy to see that for non-empty $U_{i,j}$ this results in an upper half-plane with respect to the lowest point in $U_{i,j}$. Therefore, when propagating a region across the cell, it either contributes nothing if it is empty, or it contributes a half-plane otherwise. Therefore, to establish if we can arrive at progressively more complex regions, we need to consider the other boundaries as source of complexity. This insight together with the previous results informs the following argument.

Lemma 7. *The regions that we propagate are either simple, or staircases with one step, so the regions have constant complexity.*

Proof. As shown in Lemma 3, the base regions are always simple. Consider the regions $R_{i,j}$, $L_{i,j}$ for some i and $j = j' + 1$. The proof for $U_{i,j}$, $D_{i,j}$ is symmetric. As we have just observed, the complexity of $R_{i,j'}$ and $L_{i,j'}$ is irrelevant for $R_{i,j}$, $L_{i,j}$, as they contribute a half-plane in the worst case. Furthermore, we have shown in Lemma 6 that if $U_{i,j'}$ and $D_{i,j'}$ are simple, then regions $R_{i,j}$, $L_{i,j}$ are at worst single-step staircases.

It remains to consider what happens as we propagate further from the regions obtained in Lemma 6. So suppose $R_{i,j}$, $L_{i,j}$ are obtained as in Lemma 6. Again, their complexity is irrelevant for the complexity of $R_{i,j+1}$, $L_{i,j+1}$, so it remains to answer the following question. Assuming no restrictions on $U_{i,j}$, $D_{i,j}$, what is the possible complexity of $U_{i+1,j}$, $D_{i+1,j}$? Consider the propagation for e.g. $U_{i+1,j} = (U^U_{i,j} \cup R^{RU}_{i,j} \cup L^{LU}_{i,j}) \cap I_{i+1}$. As follows from Lemma 6 and the mechanics of propagation, the region $L^{LU}_{i,j} \cup R^{RU}_{i,j}$ is either a simple region or a staircase with a single step, unbounded horizontally. Therefore, adding the half-plane of $U^U_{i,j}$ cannot increase the complexity. A symmetric argument holds for $D_{i+1,j}$. Hence, both $U_{i+1,j}$ and $D_{i+1,j}$ are again either simple or staircases with a single step.

Finally, consider the propagation through the next cell to the pair $R_{i+1,j+1}$, $L_{i+1,j+1}$. For the region $R_{i+1,j+1}$ we need to compute $U^{RU}_{i+1,j} \cup D^{RD}_{i+1,j}$. Note that

$$U_{i+1,j} \cup D_{i+1,j} = (U^U_{i,j} \cup D^D_{i,j} \cup R^R_{i,j} \cup L^L_{i,j}) \cap I_{i+1},$$

and as both $R_{i,j}$ and $L_{i,j}$ are non-empty and intersect, as follows from Lemmas 4 and 6, we conclude $U_{i+1,j} \cup D_{i+1,j} = I_{i+1}$. Therefore, the region $R_{i+1,j+1}$ is formed with a union of two half-planes with parallel boundaries, and so the region is simple. The same holds for $L_{i+1,j+1}$. So, within two propagation steps we may go from simple regions to staircase regions

with one step before returning to simple regions. As there are no other possibilities for the propagation, the statement of the lemma holds. □

The operations we use during propagation can be done in constant time for constant-complexity arguments. Using Lemma 7, we state the main result.

Theorem 8. *We can solve the decision problem for lower bound Fréchet distance on imprecise curves of lengths m and n in 1D in time $\Theta(mn)$.*

5. Upper bound Fréchet distance

Until this point, we have been discussing the lower bound Fréchet distance. We now turn our attention to the upper bound. The problem is known to be NP-hard in 2D in all variants we consider [25]; we show that this remains true even in 1D. Define the following problems for the discrete and continuous Fréchet distance.

Problem 9. UPPER BOUND (DISCRETE) FRÉCHET: Given two uncertain trajectories $\mathcal{U}$ and $\mathcal{V}$ in 1D of lengths m and n, respectively, and a threshold $\delta > 0$, determine if $d_F^{\max}(\mathcal{U}, \mathcal{V}) \leq \delta$ ($d_{dF}^{\max}(\mathcal{U}, \mathcal{V}) \leq \delta$).

We show that these problems are NP-hard both for indecisive and imprecise models by giving a reduction from CNF-SAT. The construction we use is similar to that used in 2D; however, in 2D the desired alignment of subcurves is achieved by having one of the curves be close enough to $(0, 0)$ at all times. Here making a curve close to 0 will not work, so we need to add extra gadgets instead that can 'eat up' the alignment of the subcurves that we do not care about. We start by describing the construction and then show how it leads to the NP-hardness argument.

Suppose we are given a CNF-SAT formula C on n clauses and m variables:

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

We define an *assignment* as a function $a \colon \{x_1, \ldots, x_m\} \to \{\text{True}, \text{False}\}$ that assigns a value to each variable, $a(x_j) = \text{True}$ or $a(x_j) = \text{False}$ for any $j \in [m]$. $C[a]$ then denotes the result of substituting $x_j \mapsto a(x_j)$ in C for all $j \in [m]$. We construct two curves: curve $\mathcal{U}$ is an uncertain curve that represents the variables, and curve $\mathcal{V}$ is a precise curve that represents the structure of the formula.

Literal level.   Define a *literal gadget* for curve $\mathcal{V}$:

$$\text{LG}_{i,j} = \begin{cases} 0 & \sqcup\, 1.5 & \text{if } x_j \text{ is a literal of } C_i, \\ -1.5 & \sqcup\, 1.5 & \text{if } \neg x_j \text{ is a literal of } C_i, \\ -0.75 & \sqcup\, 1.5 & \text{otherwise.} \end{cases}$$

Consider for now the indecisive uncertainty model. The curve $\mathcal{U}$ has an indecisive point per variable, each with two options, corresponding to True and False assignments. Define a *variable gadget* for curve $\mathcal{U}$:

$$\text{VG}_j = \{-1.5, 0\} \sqcup 2.5 \, .$$

Here the notation $\{-1.5, 0\}$ denotes an indecisive point with two possible locations -1.5 and 0. We interpret the position -1.5 as assigning $x_j = \text{True}$ and the position 0 as assigning $x_j = \text{False}$. Observe the relationship between $\text{LG}_{i,j}$ and VG_j for any given $i \in [n]$: the distance between the first points of the gadgets is large if the given variable assignment turns the clause true. For instance, if a clause has the literal x_j, then the choice of $x_j = \text{True}$ makes the distance between the first points $1.5 > 1$; if the literal is $\neg x_j$ and we make the same choice, then the distance is 0; and if the literal does not occur in C_i, then whichever realisation we pick, the distance is $0.75 < 1$.

Clause level.   We now aggregate the literal gadgets into *clause gadgets*. Similarly, we aggregate the variable gadgets into the *variable section*:

$$\text{CG}_i = 3.5 \sqcup \bigsqcup_{j \in [m]} \text{LG}_{i,j}, \qquad \text{VS} = 4.5 \sqcup \bigsqcup_{j \in [m]} \text{VG}_j \, .$$

Suppose that we pick some realisation for all the variables with some function a. Pick a clause C_i. Suppose that $C_i[a] = \text{True}$. This means there is at least one x_j assigned in a way that makes C_i turn true. In our construction, this means that there is at least one pair of $\text{LG}_{i,j}$ and VG_j that gives a large distance between the first two points. If we are interested in just the Fréchet distance between CG_i and VS for some fixed i, we can state the following.

Lemma 10. *For some fixed $i \in [n]$, the (discrete) Fréchet distance between* CG_i, *corresponding to clause* C_i, *and a realisation* $\pi \Subset VS$, *corresponding to an assignment a, is 1 iff $C_i[a] =$ False, and is 1.5 iff $C_i[a] =$ True, and there are no other possible values.*

Proof. First of all, note that the points 4.5 and 3.5 must be matched, yielding the distance of at least 1 between the curves. Furthermore, the only point within distance 1.5 of the point 2.5 that occurs at the end of every VG_j is the last point of every $LG_{i,j}$, namely, 1.5. Observe that simply walking along both curves, matching point k on one curve to point k on the other curve for every k, gives us (discrete) Fréchet distance of at most 1.5. Thus, the optimal matching will always match the point 2.5 to one of the points at 1.5. Furthermore, the optimal solution will always match the first point of $LG_{i,j}$ to the indecisive point of VG_j, as the point at 2.5 is always too far. Therefore, both for Fréchet and discrete Fréchet distance the optimal matching is one-to-one, i.e. we advance along both curves on every step. The initial synchronisation points yield the distance 1, as do the second points in the literal level gadgets; each indecisive point is matched at distance of either 0, 0.75, or 1.5. The latter case only occurs if the assignment of the variable makes the clause satisfied. So, indeed, we conclude that we can only get the distance of either 1 or 1.5, and the latter is only possible if some variable turns the clause to true, so if $C_i[a] =$ True. Otherwise, the clause is false, and the distance is 1. $\square$

Formula level. We can now paste the clause gadgets together. Once we do that, we would like to have a way to freely choose a clause to align with the variable section: then, if there is a clause that is not satisfied, choosing that clause would yield a small overall distance; and if all clauses are satisfied, then any one of them will give a large distance, and so we can distinguish between whether the formula is satisfied or not. As a starting point, it is clear that we need to prepend and append something to the variable section that would catch the clauses that are not aligned with the variable section. We devise the following gadget for that:

$$\text{abs} = 2.5 \sqcup \bigsqcup_{j \in [m]} (-0.5 \sqcup 0.5).$$

We show that this gadget may indeed be satisfactorily aligned with any CG_i.

Lemma 11. *The (discrete) Fréchet distance between* abs *and any* CG_i *is 1.*

Proof. First of all, note that we must match the first synchronisation point of CG_i at 3.5 to some point on the other curve, and the only point in abs that is close enough is the point at 2.5 in the beginning. This establishes the lower bound of 1. Furthermore, we can always get the distance of 1 by walking step-by-step along both curves: the distance between any of -1.5, -0.75, and 0 is at most 1 to -0.5, and the distance between 1.5 and 0.5 is 1. Thus, the statement holds. $\square$

We need as many of these gadgets as there may be misaligned clauses. In the worst case, we may align CG_1 or CG_n with VS, and so we need $n-1$ of the catch gadgets before and after VS. However, the new problem we get is that now the extra abs clauses need to be aligned with something. To that end, we devise the following gadget:

$$\text{abs}^2 = 1.5 \sqcup 0.5.$$

Again, we show that it can perform its function.

Lemma 12. *The (discrete) Fréchet distance between* abs^2 *and* abs *is 1.*

Proof. First of all, note that we must match the first synchronisation point of abs at 2.5 to the point at 1.5 on abs^2, giving the lower bound of 1. Furthermore, we can always get the distance of 1 by stepping to the second point on both curves and staying at 0.5 on abs^2 while alternating between -0.5 and 0.5 on abs. Thus, the statement holds. $\square$

Finally, we need to align these gadgets with something, but that is not too difficult, as they only have the length of 1. We define our final uncertain curves:

$$\mathcal{U} = 1 \sqcup \left(\bigsqcup_{i \in [n-1]} \text{abs} \right) \sqcup VS \sqcup \left(\bigsqcup_{i \in [n-1]} \text{abs} \right) \sqcup 1, \qquad \mathcal{V} = \left(\bigsqcup_{i \in [n-1]} \text{abs}^2 \right) \sqcup \left(\bigsqcup_{i \in [n]} CG_i \right) \sqcup \left(\bigsqcup_{i \in [n-1]} \text{abs}^2 \right).$$

We illustrate the curves in Fig. 9. With these definitions, we can show the following.

Theorem 13. *The problem* Upper Bound (Discrete) Fréchet *is NP-hard in the indecisive model.*

Proof. First of all, notice that in our construction the synchronisation points at the start of the clauses gadgets must be matched to the synchronisation points at the start of the variable section and the abs gadgets in the optimal matching, as hinted at in the proofs of the previous lemmas. Furthermore, note that any number of abs^2 at the start can be matched to
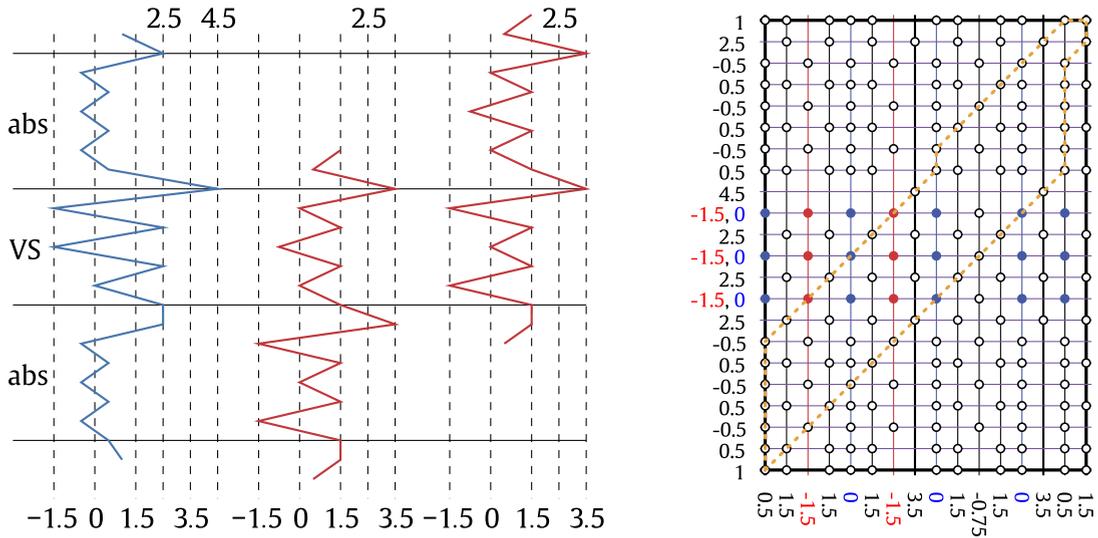
Fig. 9. Left: The realisation of $\mathcal{U}$ for assignment $x_1 = \text{True}$, $x_2 = \text{True}$, $x_3 = \text{False}$ and curve $\mathcal{V}$ for the formula $C = (x_1 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$. Note that $C = \text{True}$ with this assignment, and that both feasible alignments give (discrete) Fréchet distance of 1.5. Right: The corresponding free space. White dots are accessible, spots without a dot are never accessible. Blue (red) dots are only accessible if the corresponding variable is set to True (False). Yellow dashed paths indicate potential paths through the free space; the goal is to determine if the variables can be set such that all potential paths are blocked. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

the point at 1, and any number of abs^2 can be matched to the point at 1 at the end. Putting these observations together with Lemmas 10 to 12, it is easy to see the following. Choose some assignment a and consider the corresponding realisation $\pi \Subset \mathcal{U}$. Suppose that $C[a] = \text{False}$; this means that there is at least one i for which $C_i[a] = \text{False}$. Our construction allows us to consider the alignment where we match CG_i to VS, and the rest of clauses to one of abs; the remaining abs are matched to the abs^2, and the remaining abs^2 are matched to 1. In this matching, the (discrete) Fréchet distance between the curves is 1, which is optimal, and the formula is not satisfied. Now suppose that $C[a] = \text{True}$; this means that for all i, $C_i[a] = \text{True}$. So, no matter which CG_i we choose to align with VS, we get the distance of 1.5; therefore, the formula is satisfied, and the optimal distance is 1.5.

Recall that the upper bound distance takes the maximum distance over all realisations. Therefore, if the upper bound distance is 1, then all the realisations yield the distance 1, and so all assignments a yield $C[a] = \text{False}$, and the formula is not satisfiable. On the other hand, if the upper bound distance is 1.5, then there is some realisation that yields this distance, and it corresponds to an assignment a with $C[a] = \text{True}$, so the formula is satisfiable. Thus, our construction with the threshold $\delta = 1$ solves CNF-SAT. Curve $\mathcal{U}$ has length $2 + 2 \cdot (n-1) \cdot (1 + 2m) + 1 + 2m = 2n + 4mn - 2m + 1$; curve $\mathcal{V}$ has length $4 \cdot (n-1) + n \cdot (1 + 2m) = 5n + 2mn - 4$. Clearly, the construction takes polynomial time. Therefore, the problem both for discrete and continuous Fréchet distance is NP-hard. □

We can easily extend this result to the imprecise curves. We replace the indecisive points at $\{-1.5, 0\}$ with intervals $[-1.5, 0]$. The following observation is key.

Observation 14. *Any upper bound solution that can be found as a certificate in the construction with the indecisive points can also be found in the imprecise construction.*

Furthermore, note that no realisation can yield a distance above 1.5 with an optimal matching. Thus, if the formula is satisfiable, the upper bound distance is still 1.5, and this distance cannot be obtained otherwise. We conclude that the problem is NP-hard.

Theorem 15. *The problem* UPPER BOUND (DISCRETE) FRÉCHET *is NP-hard in the imprecise model.*

6. Weak Fréchet distance

In this section, we investigate the weak Fréchet distance for uncertain curves. In general, since weak matchings can revisit parts of the curve, the dynamic program for the regular Fréchet distance cannot easily be adapted, as it relies on the fact that only the realisation of the last few vertices is tracked. In particular, when computing the weak Fréchet distance for uncertain curves, one cannot simply forget the realisations of previously visited vertices, as the matching might revisit them. Surprisingly, we can show that for the continuous weak Fréchet distance between uncertain one-dimensional curves,

we can still obtain a polynomial-time dynamic program, as shown in Section 6.1. One may expect that the discrete weak Fréchet distance for uncertain curves in 1D is also solvable in polynomial time; however, in Section 6.2 we show that this problem is NP-hard. We also show that computing the continuous weak Fréchet distance is NP-hard for uncertain curves in 2D.

6.1. Algorithm for continuous setting

We first introduce some definitions. Consider two polygonal one-dimensional curves $\pi : [1, m] \to \mathbb{R}$ and $\sigma : [1, n] \to \mathbb{R}$ with vertices at the integer parameters. Let π^{-1} denote the reversal of a polygonal curve π. Denote by $\pi|_{[a,b]}$ the restriction of π to the domain $[a, b]$. For integer values of a and b, note that $\pi|_{[a,b]} \equiv \pi[a : b]$. Finally, define the *image* of a curve as the set of points in $\mathbb{R}$ that belong to the curve, $\mathrm{Im}(\pi) \equiv \{\pi(x) \mid x \in [1, m]\}$ for $\pi : [1, m] \to \mathbb{R}$. For any polygonal curve π, define the *growing curve* $\overrightarrow{\pi}$ of π as the sequence of local minima and maxima of the sequence $\langle \pi(i) \mid \pi(i) \notin \mathrm{Im}(\pi|_{[1,i)}) \rangle_{i=1}^{m}$. Thus, the vertices of a growing curve alternate between local minima and maxima, the subsequence of local maxima is strictly increasing, and the subsequence of local minima is strictly decreasing.

It has been shown that for precise one-dimensional curves, the weak Fréchet distance can be computed in linear time [29]. For uncertain curves, it is unclear how to use that linear-time algorithm; however, we can apply some of the underlying ideas. A *relaxed matching* between π and σ is defined by parametrisations $\alpha : [0, 1] \to [1, m]$ and $\beta : [0, 1] \to [1, n]$ with $\alpha(0) = 1$, $\alpha(1) = x \in [m - 1, m]$ and $\beta(0) = 1$, $\beta(1) = y \in [n - 1, n]$. Observe that the final points of parametrisations have to be on the last segments of the curves, but not necessarily at the endpoints of those segments. Moreover, define a relaxed matching (α, β) to be *cell-monotone* if for all $t \le t'$, we have $\min(\lfloor \alpha(t) \rfloor, m - 1) \le \alpha(t')$ and $\min(\lfloor \beta(t) \rfloor, n - 1) \le \beta(t')$. In other words, once we pass by a vertex to the next segment on a curve, we do not allow going back to the previous segment; backtracking within a segment is allowed. Let $rm(\pi, \sigma)$ be the minimum matching cost over all cell-monotone relaxed matchings:

$$rm(\pi, \sigma) = \inf_{\text{cell-monotone relaxed matching } \mu} \mathrm{cost}_\mu(\pi, \sigma).$$

It has been shown for precise curves [29] that

$$d_{\mathrm{wF}}(\pi, \sigma) = \max \left(rm(\overrightarrow{\pi}, \overrightarrow{\sigma}), rm(\overrightarrow{\pi^{-1}}, \overrightarrow{\sigma^{-1}}) \right).$$

Let $rm(\pi, \sigma)[i, j] \equiv rm(\pi[1 : i], \sigma[1 : j])$. Then the value of $rm(\pi, \sigma)$ can be computed in quadratic time as $rm(\pi, \sigma)[m, n]$ using the following dynamic program:

$$rm(\pi, \sigma)[0, \cdot] = rm(\pi, \sigma)[\cdot, 0] = \infty,$$

$$rm(\pi, \sigma)[1, 1] \qquad = |\pi(1) - \sigma(1)|, \text{ and for } i > 0 \text{ or } j > 0,$$

$$rm(\pi, \sigma)[i + 1, j + 1] \qquad = \min \begin{cases} \max \left(rm(\pi, \sigma)[i, j + 1], d(\pi(i), \mathrm{Im}(\sigma[j : j + 1])) \right), \\ \max \left(rm(\pi, \sigma)[i + 1, j], d(\sigma(j), \mathrm{Im}(\pi[i : i + 1])) \right). \end{cases}$$

If π is a growing curve, we have $\mathrm{Im}(\pi[i : i + 1]) = \mathrm{Im}(\pi[1 : i + 1])$, so the following dynamic program is equivalent if π and σ are growing curves:

$$r(\pi, \sigma)[0, \cdot] = r(\pi, \sigma)[\cdot, 0] = \infty,$$

$$r(\pi, \sigma)[1, 1] \qquad = |\pi(1) - \sigma(1)|, \text{ and for } i > 0 \text{ or } j > 0,$$

$$r(\pi, \sigma)[i + 1, j + 1] \qquad = \min \begin{cases} \max \left(r(\pi, \sigma)[i, j + 1], d(\pi(i), \mathrm{Im}(\sigma[1 : j + 1])) \right), \\ \max \left(r(\pi, \sigma)[i + 1, j], d(\sigma(j), \mathrm{Im}(\pi[1 : i + 1])) \right). \end{cases}$$

Let $r(\pi, \sigma) := r(\pi, \sigma)[m, n]$ when executing the dynamic program above for curves $\pi : [1, m] \to \mathbb{R}$ and $\sigma : [1, n] \to \mathbb{R}$. We have $rm(\overrightarrow{\pi}, \overrightarrow{\sigma}) = r(\overrightarrow{\pi}, \overrightarrow{\sigma})$. Moreover, observe that the final result of computing r is the same whether we apply it to the original or the growing curves. In other words, $r(\pi, \sigma) = r(\overrightarrow{\pi}, \overrightarrow{\sigma})$, so

$$d_{\mathrm{wF}}(\pi, \sigma) = \max \left(rm(\overrightarrow{\pi}, \overrightarrow{\sigma}), rm(\overrightarrow{\pi^{-1}}, \overrightarrow{\sigma^{-1}}) \right)$$

$$= \max \left(r(\overrightarrow{\pi}, \overrightarrow{\sigma}), r(\overrightarrow{\pi^{-1}}, \overrightarrow{\sigma^{-1}}) \right)$$

$$= \max \left(r(\pi, \sigma), r(\pi^{-1}, \sigma^{-1}) \right).$$

With regard to computing the minimum weak Fréchet distance over realisations of uncertain curves, this roughly means that we only need to keep track of the image of the prefix (and the suffix) of π and σ. To formalise this, we split up the computation over the prefix and the suffix. Let $i_{\min}, i_{\max} \in [m]$, $j_{\min}, j_{\max} \in [n]$, $[x_{\min}, x_{\max}] \subseteq \mathbb{R}$, and $[y_{\min}, y_{\max}] \subseteq \mathbb{R}$. Abbreviate the pairs $I := (i_{\min}, i_{\max})$, $J := (j_{\min}, j_{\max})$ and the intervals $X := [x_{\min}, x_{\max}]$, $Y := [y_{\min}, y_{\max}]$, and call a

realisation π of an uncertain curve *I-respecting* if $\pi(i_{\min})$ is a global minimum of π and $\pi(i_{\max})$ is a global maximum of π. Moreover, say that π is *(I, X)-respecting* if additionally $\pi(i_{\min}) = x_{\min}$ and $\pi(i_{\max}) = x_{\max}$. Let $\pi' \Subset \mathcal{U}_I$ and $\pi'' \Subset \mathcal{U}_I^X$ denote some *I*- and *(I, X)*-respecting realisations of an uncertain curve $\mathcal{U}$, respectively. Consider the minimum weak Fréchet distance between (I, X)- and (J, Y)-respecting realisations $\pi \in \mathcal{U}_I^X$ and $\sigma \in \mathcal{V}_J^Y$:

$$d_{\mathsf{wF}}^{\min}(\mathcal{U}_I^X, \mathcal{V}_J^Y) \equiv \min_{\pi \in \mathcal{U}_I^X, \sigma \in \mathcal{V}_J^Y} d_{\mathsf{wF}}(\pi, \sigma) = \min_{\pi \in \mathcal{U}_I^X, \sigma \in \mathcal{V}_J^Y} \max\left(r(\pi, \sigma), r(\pi^{-1}, \sigma^{-1})\right).$$

Lemma 16. *Among (I, X)- and (J, Y)-respecting realisations, the prefix and the suffix are independent:*

$$d_{\mathsf{wF}}^{\min}(\mathcal{U}_I^X, \mathcal{V}_J^Y) = \max \begin{cases} \min_{\pi \in \mathcal{U}_I^X, \sigma \in \mathcal{V}_J^Y} r(\pi, \sigma), \\ \min_{\pi' \in \mathcal{U}_I^X, \sigma' \in \mathcal{V}_J^Y} r(\pi'^{-1}, \sigma'^{-1}). \end{cases}$$

Proof. If we take $\pi = \pi'$ and $\sigma = \sigma'$, the right-hand side becomes a lower bound on $d_{\mathsf{wF}}^{\min}(\mathcal{U}_I^X, \mathcal{V}_J^Y)$. To show that it is also an upper bound, consider (I, X)-respecting realisations π and π', and define π_c as the prefix of π up to $i_{\min}$ concatenated with the suffix of π' starting from $i_{\min}$. Then π_c is an (I, X)-respecting realisation of $\mathcal{U}$. Moreover, the growing curves $\overrightarrow{\pi}$ and $\overrightarrow{\pi_c}$ are the same (this is obvious if $i_{\min} > i_{\max}$, and follows from the fact that the value of the $i_{\max}$-th vertex is $x_{\max}$ otherwise). Symmetrically, $\overrightarrow{\pi'^{-1}} = \overrightarrow{\pi_c^{-1}}$. We can similarly define a (J, Y)-respecting realisation σ_c of $\mathcal{V}$ based on some σ and σ'. Since $\overrightarrow{\pi} = \overrightarrow{\pi_c}$ and $\overrightarrow{\sigma} = \overrightarrow{\sigma_c}$, we have $r(\pi, \sigma) = r(\pi_c, \sigma_c)$, and symmetrically, $r(\pi^{-1}, \sigma^{-1}) = r(\pi_c^{-1}, \sigma_c^{-1})$. We can therefore use $\pi_c \Subset \mathcal{U}_I^X$ and $\sigma_c \Subset \mathcal{V}_J^Y$ in the definition of $d_{\mathsf{wF}}(\mathcal{U}_I^X, \mathcal{V}_J^Y)$ to obtain the desired upper bound. $\square$

The remainder of this section is guided by observations based on Lemma 16.

1. If we can compute $\min_{\pi \in \mathcal{U}_I^X, \sigma \in \mathcal{V}_J^Y} r(\pi, \sigma)$, we can compute $d_{\mathsf{wF}}^{\min}(\mathcal{U}_I^X, \mathcal{V}_J^Y)$.
2. To compute $d_{\mathsf{wF}}^{\min}(\mathcal{U}_I, \mathcal{V}_J)$, we must find an optimal pair of images X and Y for π and σ.
3. We can find $d_{\mathsf{wF}}^{\min}(\mathcal{U}, \mathcal{V})$ by computing $d_{\mathsf{wF}}^{\min}(\mathcal{U}_I, \mathcal{V}_J)$ for all $O(m^2 n^2)$ values for (I, J).

Instead of computing $\min_{\pi \in \mathcal{U}_I^X, \sigma \in \mathcal{V}_J^Y} r(\pi, \sigma)$ for a specific value of (X, Y), we compute the function $(X, Y) \mapsto \min_{\pi \in \mathcal{U}_I^X, \sigma \in \mathcal{V}_J^Y} r(\pi, \sigma)$ using a dynamic program that effectively simulates the dynamic program $r(\pi, \sigma)$ for all *I*- and *J*-respecting realisations simultaneously. So let

$$R_{I,J}[i, j](x, y, X, Y) := \inf_{\substack{\pi \in \mathcal{U}_I, \mathrm{Im}(\pi[1:i]) = X, \pi(i) = x \\ \sigma \in \mathcal{V}_J, \mathrm{Im}(\sigma[1:j]) = Y, \sigma(j) = y}} r(\pi, \sigma)[i, j], \quad \text{then}$$

$$R_{I,J}[m, n](x, y, X, Y) = \inf_{\substack{\pi \in \mathcal{U}_I^X, \pi(m) = x \\ \sigma \in \mathcal{V}_J^Y, \sigma(n) = y}} r(\pi, \sigma).$$

We derive

$$R_{I,J}[0, \cdot](x, y, X, Y) = R_{I,J}[\cdot, 0](x, y, X, Y) = \infty,$$

$$R_{I,J}[1, 1](x, y, X, Y) = \inf_{\substack{\pi \in \mathcal{U}_I, \{x\} = X, \pi(1) = x \\ \sigma \in \mathcal{V}_J, \{y\} = Y, \sigma(1) = y}} |\pi(1) - \sigma(1)|, \text{ and for } (i, j) \neq (1, 1)$$

$$R_{I,J}[i, j](x, y, X, Y)$$

$$= \inf_{\substack{\pi \in \mathcal{U}_I, \mathrm{Im}(\pi[1:i]) = X, \pi(i) = x \\ \sigma \in \mathcal{V}_J, \mathrm{Im}(\sigma[1:j]) = Y, \sigma(j) = y}} \min \begin{cases} \max\{r(\pi, \sigma)[i-1, j], d(\pi(i-1), Y)\}, \\ \max\{r(\pi, \sigma)[i, j-1], d(\sigma(j-1), X)\} \end{cases}$$

$$= \min \begin{cases} \inf_{\substack{\pi \in \mathcal{U}_I, \mathrm{Im}(\pi[1:i]) = X, \pi(i) = x \\ \sigma \in \mathcal{V}_J, \mathrm{Im}(\sigma[1:j]) = Y, \sigma(j) = y \\ \pi(i-1) = x'}} \max\{r(\pi, \sigma)[i-1, j], d(x', Y)\}, \\ \inf_{\substack{\pi \in \mathcal{U}_I, \mathrm{Im}(\pi[1:i]) = X, \pi(i) = x \\ \sigma \in \mathcal{V}_J, \mathrm{Im}(\sigma[1:j]) = Y, \sigma(j) = y \\ \sigma(j-1) = y'}} \max\{r(\pi, \sigma)[i, j-1], d(y', X)\} \end{cases}$$

$$= \min \begin{cases} \inf_{\substack{\pi \in \mathcal{U}_I, \mathrm{Im}(\pi[1:i]) = X, \pi(i) = x \\ \mathrm{Im}(\pi[1:i-1]) = X', \pi(i-1) = x'}} \max\{R_{I,J}[i-1, j](x', y, X', Y), d(x', Y)\}, \\ \inf_{\substack{\sigma \in \mathcal{V}_J, \mathrm{Im}(\sigma[1:j]) = Y, \sigma(j) = y \\ \mathrm{Im}(\sigma[1:j-1]) = Y', \sigma(j-1) = y'}} \max\{R_{I,J}[i, j-1](x, y', X, Y'), d(y', X)\}, \end{cases}$$

where, crucially, the conditions on x', y', X', and Y' can be checked purely in terms of $\mathcal{U}_I$ and $\mathcal{V}_J$, so the recurrence does not depend on any particular π or σ. This yields a dynamic program that constructs the function $R_{I,J}[i,j]$ based on the functions $R_{I,J}[i-1,j]$ and $R_{I,J}[i,j-1]$.

The recurrence has the parameters I, J, i, j, x, y, X, and Y. The first four are easy to handle, since $i \in [m]$, $j \in [n]$, $I \in [m]^2$, and $J \in [n]^2$. The other parameters are continuous. X can be represented by $x_{\min}$ and $x_{\max}$, Y by $y_{\min}$ and $y_{\max}$. To prove that we can solve the recurrence in polynomial time, it is sufficient to prove that we can restrict the computation to a polynomial number of different $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, x and y.

We assume that each of the u_i and v_j is given as a set of intervals. This includes the cases of uncertain curves with imprecise vertices (where each of these is just one interval) and with indecisive vertices (where each interval is just a point; but in this case we get by definition only a polynomial number of different values for the parameters).

Consider the realisations $\pi = \langle p_1, \ldots, p_m \rangle$ and $\sigma = \langle q_1, \ldots, q_n \rangle$ of the curves that attain the lower bound weak Fréchet distance $d_{\mathrm{wF}}^{\min}(\mathcal{U}, \mathcal{V}) =: \delta$. In these realisations, we need to have a sequence of vertices $r_1 \leq r_2 \leq \cdots \leq r_\ell$ with the r_k alternately from the set of p_i and the set of q_j such that r_1 is at a right interval endpoint, r_ℓ is at a left interval endpoint, and $r_{k+1} - r_k = \delta$. Since $1 \leq \ell \leq m + n$, this implies that there are only $O(N^2 \cdot (m+n))$ candidates for δ, where N is the total number of interval endpoints. We can compute these candidates in time $O(N^2 \cdot (m+n))$.

Now assume that we have chosen π and σ such that none of the p_i or q_j can be increased (i.e. moved to the right) without increasing the weak Fréchet distance. Then for every p_i (and likewise q_j) there is a sequence $r_1 \leq r_2 \leq \cdots \leq r_\ell = p_i$, where r_1 is the endpoint of an interval and $r_{k+1} - r_k = \delta$. There are $O(N)$ possibilities for r_1, $O(m+n)$ possibilities for ℓ, and $O(N^2 \cdot (m+n))$ possibilities for δ, thus the total number of positions to consider for p_i is polynomial.

Theorem 17. *The continuous weak Fréchet distance between uncertain one-dimensional curves can be computed in polynomial time.*

6.2. Hardness of discrete setting

In this section, we prove that minimising the discrete weak Fréchet distance is NP-hard, already in one-dimensional space. We show this both for indecisive and imprecise points. In the constructions in this section, the lower bound Fréchet distance is never smaller than 1. The goal is to determine whether it is equal to 1 or greater than 1.

Indecisive points. We reduce from 3SAT. Consider an instance with n variables and m clauses. We assign each variable a unique *height*: variable x_i gets assigned height $10i + 5$. We use slightly higher heights ($10i + 6$ and $10i + 7$) to interact with the positive state of the variable, and slightly lower heights to interact with a negative state.

We construct two uncertain curves, one which represents the variables and one which represents the clauses. The first curve, $\mathcal{U}$, consists of $n + 2$ vertices. The first and last vertex are certain points, both at height 0. The remaining vertices are uncertain points, with two possible heights each:

$$\mathcal{U} = \langle 0, \{14, 16\}, \{24, 26\}, \ldots, \{10n + 4, 10n + 6\}, 0 \rangle.$$

The second curve, $\mathcal{V}$, consists of $nm + n + m + 2$ vertices. For a clause $c_j = \ell_a \vee \ell_b \vee \ell_c$, let C_j be the set $\{10a + 3/7, 10b + 3/7, 10c + 3/7\}$, where for each literal we choose $+7$ if $\ell_i = x_i$ or $+3$ if $\ell_i = \neg x_i$. Let S be the set $S = \{15, 25, \ldots, 10n + 5\}$ of 'neutral' variable heights. Then $\mathcal{V}$ is the curve that starts and ends at 0, has a vertex for each C_j, and has sufficiently many copies of S between them:

$$\mathcal{V} = \langle 0, S, \ldots, S, C_1, S, \ldots, S, C_2, S, \ldots, S, \ldots \ldots, C_m, 0 \rangle.$$

Consider the free-space diagram, with a 'spot' (i, j) corresponding to a pair of vertices u_i and v_j. The discrete weak Fréchet distance is equal to 1 if and only if there is an assignment to each uncertain vertex such that the there is a path from the bottom left to the top right of the diagram that uses only accessible spots, where a spot is accessible if the assigned heights of the corresponding row and column are within 1. Fig. 10 shows an example.

We can only cross a column corresponding to clause c_j if at least one of the corresponding literals is set to true. The remaining columns can always be crossed at any row. Note that the repetition is necessary: although all spots are in principle reachable, only one spot in each column can be reachable at the same time. If we have at least n columns between each pair of clauses, this will always be possible.

Theorem 18. *Given two uncertain curves $\mathcal{U}$ and $\mathcal{V}$, each given by a sequence of values and sets of values in $\mathbb{R}$, the problem of choosing a realisation of $\mathcal{U}$ and $\mathcal{V}$ such that the weak discrete Fréchet distance between $\mathcal{U}$ and $\mathcal{V}$ is minimised is NP-hard.*

Imprecise points. The construction above relies on the ability to select arbitrary sets of values as uncertainty regions. We now show that this is not required. We strengthen the proof in two ways: we restrict the uncertainty regions to intervals and we use uncertainty in only one curve.

The main idea of the adaptation is to encode clauses not by a single uncertain vertex, but by sets of globally distinct paths through the free-space diagram. To facilitate this, we need a global *frame* to guide the possible solution paths, and we need more copies of the variable vertices (though only one copy will be uncertain) to facilitate the paths.
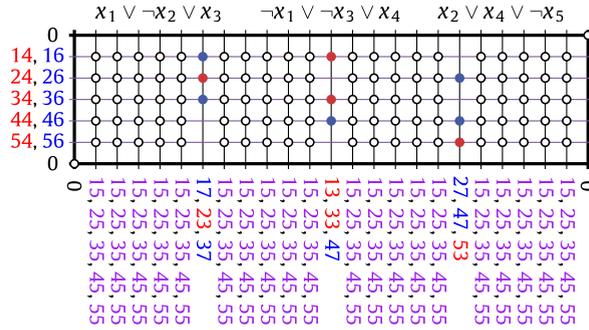
Fig. 10. An example with five variables and three clauses. White dots are always accessible, no matter the state of the variables (however, note that only one white dot per column can be used). Red / blue dots are accessible only if the corresponding variable is set to False / True. Spots without a dot are never accessible.
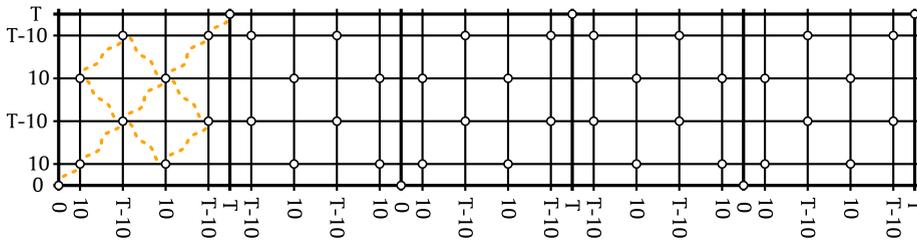

Fig. 11. The global frame. White dots are accessible, spots without a dot are never accessible. Within each block, there are three potential paths between its two accessible corners.

Let $T = 10(n + 2)$. We build a frame for the construction using four unique heights: 0, 10, $T - 10$ and T.[4] Let $S = \langle 0, 10, ?, T - 10, ?, 10, ?, T - 10, T \rangle$ be a partial sequence—the question marks indicate gaps where we insert other vertices later. Globally, the curves have the structure $\mathcal{U} = S$ and $\mathcal{V} = S \sqcup S^{-1} \sqcup S \sqcup S^{-1} \sqcup \ldots \sqcup S$: one copy or reversed copy of S for each clause (if the number of clauses is even, simply add a trivial clause). In the free-space diagram, this creates a frame that every path needs to adhere to. The frame consists of one block per clause, and inside each block, there are three potential paths from the bottom left to top right corner (or from the top left to bottom right corner for reversed blocks). See Fig. 11.

Next, we fill in the gaps. Let $\mathcal{U} = \langle 0 \rangle \sqcup \mathcal{U}_1 \sqcup \mathcal{U}_2^{-1} \sqcup \mathcal{U}_1 \sqcup \langle T \rangle$, where

$$\mathcal{U}_1 = \langle 10, 14, 16, 24, 26, \ldots, 10n + 4, 10n + 6, T - 10 \rangle,$$

$$\mathcal{U}_2 = \langle 10, [14, 16], [24, 26], \ldots, [10n + 4, 10n + 6], T - 10 \rangle.$$

Let $\mathcal{V} = \bigsqcup_{1 \leq j \leq m} C_j^{(-1)^{j-1}}$ be concatenation of clause sequences, where each even clause sequence is reversed. For a clause $c_j = \ell_a \vee \ell_b \vee \ell_c$, the sequence C_j is of the form

$$C_j = \langle 0, 10 \rangle \sqcup L_a \sqcup \langle T - 10 \rangle \sqcup L_b^{-1} \sqcup \langle 10 \rangle \sqcup L_c \sqcup \langle T - 10, T \rangle,$$

where the literal sequence L_i corresponding to $\ell_i = x_i$ (positive literals) or $\ell_i = \neg x_i$ (negative literals) is respectively

$$L_i = \langle 15, 25, \ldots, 10(i - 1) + 5, 10i + 5, 10i + 7, 10(i + 1) + 5, \ldots, 10n + 5 \rangle, \quad \text{or}$$

$$L_i = \langle 15, 25, \ldots, 10(i - 1) + 5, 10i + 3, 10i + 5, 10(i + 1) + 5, \ldots, 10n + 5 \rangle.$$

See Fig. 12 for an example of the resulting free-space diagram.

The construction relies on the following.

Observation 19. L_i can always be matched to $\mathcal{U}_1$. L_i can be matched to $\mathcal{U}_2$ if and only if $\ell_i = x_i$ and x_i is set to True, or $\ell_i = \neg x_i$ and x_i is set to False.

---

[4] The actual values are, in fact, irrelevant for the construction—they simply need to be unique numbers sufficiently removed from the values we will use for encoding the variables.
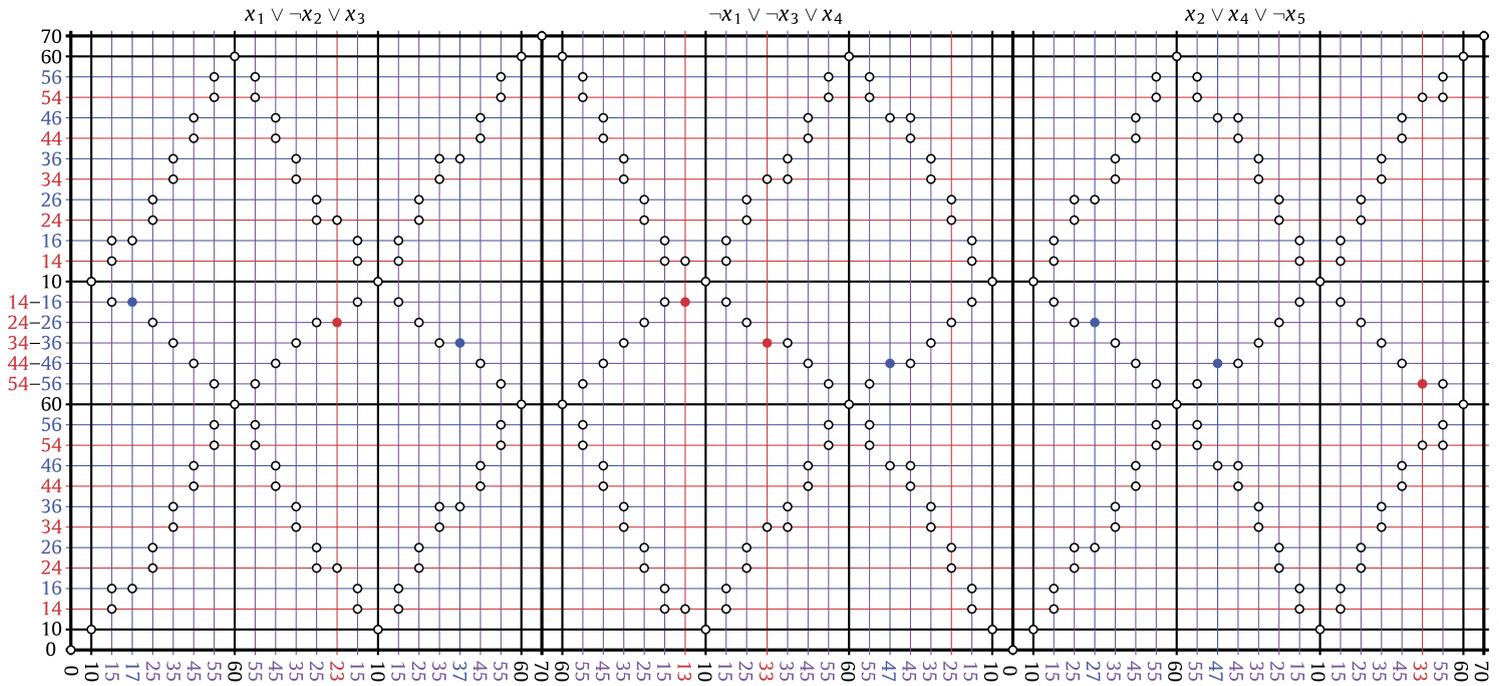
Fig. 12. An example with five variables and three clauses. White dots are always accessible, no matter the state of the variables. Red / blue dots are accessible only if the corresponding variable is set to False / True. Spots without a dot are never accessible.

Theorem 20. *Given an uncertain curve $\mathcal{U}$, given by a sequence of values and intervals in $\mathbb{R}$, and a certain curve $\mathcal{V}$, given by a sequence of values in $\mathbb{R}$, the problem of choosing a realisation of $\mathcal{U}$ such that the weak discrete Fréchet distance between $\mathcal{U}$ and $\mathcal{V}$ is minimised is NP-hard.*

Continuous weak Fréchet distance in $\mathbb{R}^2$.   Finally, we mention that the results in this section carry over to continuous weak Fréchet distance in one dimension higher. We simply construct the same curves as described above on the x-axis, and intersperse each curve with the point at $(0, \infty)$.

Corollary 21. *Given an uncertain curve $\mathcal{U}$, given by a sequence of points and regions in $\mathbb{R}^2$, and a certain curve $\mathcal{V}$, given by a sequence of points in $\mathbb{R}^2$, the problem of choosing a realisation of $\mathcal{U}$ such that the weak Fréchet distance between $\mathcal{U}$ and $\mathcal{V}$ is minimised is NP-hard.*

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P.K. Agarwal, R.B. Avraham, H. Kaplan, M. Sharir, Computing the discrete Fréchet distance in subquadratic time, SIAM J. Comput. 43 (2) (2014) 429–449, https://doi.org/10.1137/130920526.
[2] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, Int. J. Comput. Geom. Appl. 5 (1) (1995) 75–91, https://doi.org/10.1142/S0218195995000064.
[3] K. Bringmann, Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails, in: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, IEEE, Piscataway, NJ, USA, 2014, pp. 661–670, https://doi.org/10.1109/FOCS.2014.76.
[4] K. Bringmann, W. Mulzer, Approximability of the discrete Fréchet distance, J. Comput. Geom. 7 (2) (2016) 46–76, https://doi.org/10.20382/jocg.v7i2a4.
[5] K. Buchin, M. Buchin, W. Meulemans, W. Mulzer, Four Soviets walk the dog: improved bounds for computing the Fréchet distance, Discrete Comput. Geom. 58 (1) (2017) 180–216, https://doi.org/10.1007/s00454-017-9878-7.
[6] A. Driemel, S. Har-Peled, C. Wenk, Approximating the Fréchet distance for realistic curves in near linear time, Discrete Comput. Geom. 48 (1) (2012) 94–127, https://doi.org/10.1007/s00454-012-9402-z.
[7] S. Har-Peled, B. Raichel, The Fréchet distance revisited and extended, ACM Trans. Algorithms 10 (1) (2014) 3:1–3:22, https://doi.org/10.1145/2532646.
[8] M. Jiang, Y. Xu, B. Zhu, Protein structure: structure alignment with discrete Fréchet distance, J. Bioinform. Comput. Biol. 6 (1) (2008) 51–64, https://doi.org/10.1142/s0219720008003278.
[9] J. Zheng, X. Gao, E. Zhan, Z. Huang, Algorithm of on-line handwriting signature verification based on discrete Fréchet distance, in: International Symposium on Intelligence Computation and Applications, in: Lecture Notes in Computer Science, vol. 5370, Springer Berlin Heidelberg, Berlin, Germany, 2008, pp. 461–469, https://doi.org/10.1007/978-3-540-92137-0_5.
[10] S. Brakatsoulas, D. Pfoser, R. Salas, C. Wenk, On map-matching vehicle tracking data, in: Proceedings of the 31st International Conference on Very Large Data Bases, Association for Computing Machinery, New York, NY, USA, 2005, pp. 853–864, https://doi.org/10.5555/1083592.1083691.
[11] M. Ahmed, C. Wenk, Constructing street networks from GPS trajectories, in: Proceedings of the 20th Annual European Symposium on Algorithms, in: Lecture Notes in Computer Science, vol. 7501, Springer, Berlin Heidelberg, Berlin, Germany, 2012, pp. 60–71, https://doi.org/10.1007/978-3-642-33090-2_7.
[12] K. Buchin, M. Buchin, D. Duran, B.T. Fasy, R. Jacobs, V. Sacristán, R.I. Silveira, F. Staals, C. Wenk, Clustering trajectories for map construction, in: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '17), Association for Computing Machinery, New York, NY, USA, 2017, pp. 14:1–14:10, https://doi.org/10.1145/3139958.3139964.
[13] K. Buchin, A. Driemel, N. van de L'Isle, A. Nusser, Klcluster: center-based clustering of trajectories, in: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19), Association for Computing Machinery, New York, NY, USA, 2019, pp. 496–499, https://doi.org/10.1145/3347146.3359111.
[14] J. Gudmundsson, T. Wolle, Football analysis using spatio-temporal tools, Comput. Environ. Urban Syst. 47 (2014) 16–27, https://doi.org/10.1016/j.compenvurbsys.2013.09.004.
[15] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop, V. Sacristán, Smallest color-spanning objects, in: Algorithms – ESA 2001, in: Lecture Notes in Computer Science, vol. 2161, Springer Berlin Heidelberg, Berlin, Germany, 2001, pp. 278–289, https://doi.org/10.1007/3-540-44676-1_23.
[16] K. Buchin, M. Löffler, P. Morin, W. Mulzer, Preprocessing imprecise points for Delaunay triangulation: simplified and extended, Algorithmica 61 (3) (2011) 674–693, https://doi.org/10.1007/s00453-010-9430-0.
[17] C. Fan, J. Luo, B. Zhu, Tight approximation bounds for connectivity with a color-spanning set, in: Algorithms and Computation (ISAAC 2013), in: Lecture Notes in Computer Science, vol. 8283, Springer Berlin Heidelberg, Berlin, Germany, 2013, pp. 590–600, https://doi.org/10.1007/978-3-642-45030-3_55.
[18] C. Knauer, M. Löffler, M. Scherfenberg, T. Wolle, The directed Hausdorff distance between imprecise point sets, Theor. Comput. Sci. 412 (32) (2011) 4173–4186, https://doi.org/10.1016/j.tcs.2011.01.039.
[19] M. Löffler, Data imprecision in computational geometry, Ph.D. thesis, Universiteit Utrecht, 2009, cited 2019-06-15, https://dspace.library.uu.nl/bitstream/handle/1874/36022/loffler.pdf.
[20] M. Löffler, W. Mulzer, Unions of onions: preprocessing imprecise points for fast onion decomposition, J. Comput. Geom. 5 (1) (2014) 1–13, https://doi.org/10.20382/jocg.v5i1a1.
[21] M. Löffler, J.S. Snoeyink, Delaunay triangulations of imprecise points in linear time after preprocessing, Comput. Geom. Theory Appl. 43 (3) (2010) 234–242, https://doi.org/10.1016/j.comgeo.2008.12.007.
[22] M. Löffler, M. van Kreveld, Largest and smallest tours and convex hulls for imprecise points, in: Algorithm Theory – SWAT 2006, in: Lecture Notes in Computer Science, vol. 4059, Springer Berlin Heidelberg, Berlin, Germany, 2006, pp. 375–387, https://doi.org/10.1007/11785293_35.
[23] M. van Kreveld, M. Löffler, J.S.B. Mitchell, Preprocessing imprecise points and splitting triangulations, SIAM J. Comput. 39 (7) (2010) 2990–3000, https://doi.org/10.1137/090753620.
[24] H.-K. Ahn, C. Knauer, M. Scherfenberg, L. Schlipf, A. Vigneron, Computing the discrete Fréchet distance with imprecise input, Int. J. Comput. Geom. Appl. 22 (1) (2012) 27–44, https://doi.org/10.1142/S0218195912600023.

[25] K. Buchin, C. Fan, M. Löffler, A. Popov, B. Raichel, M. Roeloffzen, Fréchet distance for uncertain curves, in: 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020), in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 168, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2020, pp. 20:1–20:20, https://doi.org/10.4230/LIPIcs.ICALP.2020.20.

[26] M. Buchin, S. Sijben, Discrete Fréchet distance for uncertain points, Lugano, Switzerland, presented at EuroCG 2016, http://www.eurocg2016.usi.ch/sites/default/files/paper_72.pdf, 2016.

[27] C. Fan, B. Zhu, Complexity and algorithms for the discrete Fréchet distance upper bound with imprecise input, arXiv:1509.02576v2, 2018.

[28] K. Buchin, M. Buchin, C. Knauer, G. Rote, C. Wenk, How difficult is it to walk the dog?, presented at EuroCG 2007, Graz, Austria, https://page.mi.fu-berlin.de/rote/Papers/pdf/How+difficult+is+it+to+walk+the+dog.pdf, 2007.

[29] K. Buchin, T. Ophelders, B. Speckmann, 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), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2019, pp. 2887–2901, https://doi.org/10.5555/3310435.3310614.