Faster, Deterministic and Space Efficient Subtrajectory Clustering

Ivor van der Hoog
Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

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

Tim Ophelders
Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

Given a trajectory T and a distance Δ, we wish to find a set C of curves of complexity at most ℓ, such that we can cover T with subcurves that each are within Fréchet distance Δ to at least one curve in C. We call C an (ℓ, Δ)-clustering and aim to find an (ℓ, Δ)-clustering of minimum cardinality. This problem variant was introduced by Akitaya *et al.* (2021) and shown to be NP-complete. The main focus has therefore been on bicriteria approximation algorithms, allowing for the clustering to be an $(\ell, \Theta(\Delta))$-clustering of roughly optimal size.

We present algorithms that construct $(\ell, 4\Delta)$-clusterings of $\mathcal{O}(k \log n)$ size, where k is the size of the optimal (ℓ, Δ)-clustering. We use $\mathcal{O}(n^3)$ space and $\mathcal{O}(kn^3 \log^4 n)$ time. Our algorithms significantly improve upon the clustering quality (improving the approximation factor in Δ) and size (whenever $\ell \in \Omega(\log n / \log k)$). We offer deterministic running times improving known expected bounds by a factor near-linear in ℓ. Additionally, we match the space usage of prior work, and improve it substantially, by a factor super-linear in $n\ell$, when compared to deterministic results.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Fréchet distance, clustering, set cover

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.133

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2402.13117

Funding *Ivor van der Hoog*: This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 899987, and from the Carlsberg Foundation via Eva Rotenberg's Young Researcher Fellowship CF21-0302 "Graph Algorithms with Geometric Applications".
Tim Ophelders: Partially supported by the Dutch Research Council (NWO) under the project number VI.Veni.212.260.

Acknowledgements We thank Jacobus Conradi for pointing out an error in a preprint of this paper.

1 Introduction

In subtrajectory clustering, the goal is to partition an input trajectory T with n vertices into subtrajectories and group them into *clusters* such that all subtrajectories within a cluster have low Fréchet distance to one another. Clustering under the Fréchet distance is a natural application of the Fréchet distance and a well-studied topic [9, 10, 11, 15, 16] with applications

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis
Article No. 133; pp. 133:1–133:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Prior work and our result. The first two (red) rows indicate randomized results. k denotes the smallest (ℓ, Δ)-clustering size of T. λ denotes the arc length of T relative to Δ.

# Clusters	$\Delta' =$	Time	Space	Source
$\mathcal{O}(k\ell^2 \log(k\ell))$	19Δ	$\tilde{\mathcal{O}}(k\ell^4\lambda^2 + n\lambda)$	$\mathcal{O}(n + \lambda)$	[3]
$\mathcal{O}(k\ell \log k)$	11Δ	$\tilde{\mathcal{O}}(kn^3\ell)$	$\tilde{\mathcal{O}}(n^3)$	[5]
$\mathcal{O}(k \log n)$	11Δ	$\tilde{\mathcal{O}}(kn^4\ell + n^4\ell^2)$	$\tilde{\mathcal{O}}(n^4\ell)$	[13]
$\mathcal{O}(k \log n)$	4Δ	$\mathcal{O}(kn^3 \log^4 n)$	$\mathcal{O}(n^3)$	Thm. 16

in, for example, map reconstruction [6,7]. In recent years, several variants of this algorithmic problem have been proposed [1,3,5,8]. Regardless of the variant, the subtrajectory clustering problem has been shown to be NP-complete [1,3,8].

We focus on the problem variant proposed by Akitaya, Brüning, Chambers, and Driemel [3]. Given a trajectory T and a distance Δ, and some ℓ, they compute what we call an (ℓ, Δ)-*clustering* C of T. Each cluster $Z \in C$ is a set of subtrajectories together with a center curve (the "reference curve" P_Z) of complexity at most ℓ. Each curve in a cluster must have Fréchet distance at most Δ to the center and each point on T must be present in at least one cluster. The goal is to compute an (ℓ, Δ)-clustering of minimum cardinality. Note that the parameter ℓ is necessary to not trivialize the problem. Indeed, if P_Z may have arbitrary complexity, then a trivial $(n, 0)$-clustering exists consisting of a single cluster Z where $Z = \{T\}$.

Akitaya et al. [3] propose a bicriteria approximation scheme: Given ℓ and Δ, let k be the minimum size of an (ℓ, Δ)-clustering of T. The goal is to compute an $(\ell, \Theta(\Delta))$-clustering of size $\mathcal{O}(f(k))$. This paradigm was studied in [3,5,13] and previous results are summarised in Table 1. [3] computes an $(\ell, 19\Delta)$-clustering of $\mathcal{O}(k\ell^2 \log(k\ell))$ size. The running time and space bounds depend on the *arc length* of T relative to Δ. Brüning, Conradi and Driemel [5] compute an $(\ell, 11\Delta)$-clustering of $\mathcal{O}(k\ell \log k)$ size (where the hidden constant is exceptionally large). Their algorithm uses $\tilde{\mathcal{O}}(n^3)$ space and has $\tilde{\mathcal{O}}(kn^3)$ expected running time. Recently, Conradi and Driemel [13] improve both the size and the quality of the clustering. They compute an $(\ell, 11\Delta)$-clustering of $\mathcal{O}(k \log n)$ size in $\tilde{\mathcal{O}}(n^4\ell)$ space and $\tilde{\mathcal{O}}(kn^4\ell + n^4\ell^2)$ time.

Results. We present a bicriteria approximation algorithm that uses $\mathcal{O}(kn^3 \log^4 n)$ time and $\mathcal{O}(n^3)$ space, and computes an $(\ell, 4\Delta)$-clustering of size $\mathcal{O}(k \log n)$. When compared to previous works [3,5,13] our results:

- obtain deterministic results and improve the running time by a factor near-linear in ℓ,
- match the space usage,
- improve the approximation in Δ from a factor 11 to 4,
- asymptotically match the clustering size (whenever $\ell \in \Omega(\log n/ \log k)$).

Compared exclusively to deterministic results [13], we instead improve time by a factor near-linear in $n\ell$, space by a factor super-linear in $n\ell$, and obtain asymptotically equal clustering size for all ℓ (see also Table 1).

Methodology and contribution. Our algorithm constructs a clustering iteratively by greedily adding a cluster that covers an approximately-maximum set of uncovered points on T. The challenge is to compute such a cluster. Previous work [3,5] presented randomized algorithms for constructing a cluster based on ε-net sampling over the set of all candidate clusters. They shatter the set of candidate clusters and show that it has bounded VC dimension, which leads to their asymptotic approximation of k – the minimum size of an (ℓ, Δ)-clustering. The algorithm of Conradi and Driemel [13] is more similar to ours. They also simplify the input and iteratively select the cluster with the (exact) maximum coverage to obtain an (ℓ, Δ)-clustering of size $\mathcal{O}(k \log n)$. The key difference lies in finding the next cluster. Conradi and Driemel [13] explicitly consider a set of $\mathcal{O}(n^3\ell)$ candidate clusters, which requires $\mathcal{O}(n^4\ell)$ time and space to construct.

We make two key contributions that distinguish us from prior works: First, we present a novel simplification algorithm that computes a curve S such that we may restrict potential reference curves of clusters to be subcurves of S. This new curve simplification technique allows us to create a clustering where clusters have radius at most 4Δ as opposed to 11Δ. Second, we prove that we may restrict the reference curves to be one of two types:

- vertex-subcurves of S, which are subcurves that start and end at a vertex of S, (we may furthermore only consider subcurves whose complexity is a power of 2)
- and subedges of S, which are subcurves that are a subsegment of a single edge of S.

We prove that a greedy algorithm that exclusively adds maximal clusters where the reference curve is of one of these two types creates a clustering of size $\mathcal{O}(k \log n)$. This characterization reduces the set of candidate clusters from $\tilde{\mathcal{O}}(n^3\ell)$ to $\tilde{\mathcal{O}}(n^2)$ which significantly reduces the time spent compared to [13]. The geometric characterization of these subcurves allow us to compute candidate clusters on the fly, significantly reducing space usage.

2 Preliminaries

A *(polygonal) curve* with n vertices is a piecewise-linear map $P\colon [1, n] \to \mathbb{R}^d$ whose breakpoints (called *vertices*) are at each integer parameter, and whose pieces are called *edges*. We denote by $P[a, b]$ the subcurve of P that starts at $P(a)$ and ends at $P(b)$. If a and b are integers, we call $P[a, b]$ a *vertex subcurve* of P. Let $|P|$ denote the number of vertices of P.

Fréchet distance. A *reparameterization* of $[1, n]$ is a non-decreasing surjection $f\colon [0, 1] \to [1, n]$. Two reparameterizations f and g of $[1, m]$ and $[1, n]$, respectively, describe a *matching* (f, g) between two curves P and Q with n and m vertices, where for any $t \in [0, 1]$, point $P(f(t))$ is matched to $Q(g(t))$. A matching (f, g) is said to have *cost* $\max_t \|P(f(t)) - Q(g(t))\|$, where $\|\cdot\|$ denotes the Euclidean norm. A matching with cost at most Δ is called a Δ-*matching*. The (continuous) *Fréchet distance* $d_F(P, Q)$ between P and Q is the minimum cost over all matchings.

Free space diagram. The *parameter space* of curves P and Q with m and n vertices, respectively, is given by the orthogonal rectangle $[1, m] \times [1, n]$. This parameter space is associated with a regular grid whose cells are the squares $[i, i + 1] \times [j, j + 1]$ for integers i and j. A point (x, y) in the parameter space corresponds to the pair of points $P(x)$ and $Q(y)$. We say that (x, y) is Δ-*free* if $\|P(x) - Q(y)\| \leq \Delta$. The Δ-*free space diagram* Δ-FSD(P, Q) of P and Q is the set of Δ-free points in the parameter space of P and Q. The *obstacles* of Δ-FSD(P, Q) are the connected components of $([1, m] \times [1, n]) \setminus \Delta$-FSD$(P, Q)$.

Alt and Godau [4] observe that the Fréchet distance between $P[x_1, x_2]$ and $Q[y_1, y_2]$ is at most Δ if and only if there is a bimonotone path in Δ-FSD(P, Q) from (x_1, y_1) to (x_2, y_2) (and $x_1 \leq x_2$ and $y_1 \leq y_2$).

Input and output. Our input is a curve T with n vertices, which we will call the *trajectory*, some integer parameter $\ell \geq 2$, and some distance parameter $\Delta \geq 0$. We consider *clustering* subtrajectories of T using *pathlets*:

▶ **Definition 1** (Pathlet). *An* (ℓ, Δ)-pathlet *is a tuple* $(P, \mathcal{I})$ *where* P *is a curve with* $|P| \leq \ell$ *and* $\mathcal{I}$ *is a set of intervals in* $[1, n]$*, where* $d_F(P, T[a, b]) \leq \Delta$ *for all* $[a, b] \in \mathcal{I}$*. We call* P *the* reference curve *of* $(P, \mathcal{I})$*.*

We can see a pathlet $(P, \mathcal{I})$ as a cluster, where the center is P and all subtrajectories induced by $\mathcal{I}$ get mapped to P. See Figure 1. An (ℓ, Δ)-clustering of T is defined as follows:

▶ **Definition 2.** *An* (ℓ, Δ)-clustering C *is a set of* (ℓ, Δ)*-pathlets with* $\bigcup_{(P,\mathcal{I}) \in C} \bigcup_{I \in \mathcal{I}} I = [1, n]$*.*

Throughout this paper, we let $k_\ell(\Delta)$ denote the smallest integer for which there exists an (ℓ, Δ)-clustering of size $k_\ell(\Delta)$. The goal is to find an (ℓ, Δ')-clustering C where $|C|$ is not too large compared to $k_\ell(\Delta)$, and $\Delta' \in \mathcal{O}(\Delta)$.

Weighting a cluster. We define a *Universe* $\mathcal{U}$ as any set of interior-disjoint closed intervals that together cover $[1, n]$. Given a fixed universe $\mathcal{U}$, we can weigh each pathlet by what we call its *coverage*:

▶ **Definition 3.** *The* coverage *over* $\mathcal{U}$ *of a pathlet* $(P, \mathcal{I})$ *is* $\mathrm{Cov}_{\mathcal{U}}(P, \mathcal{I}) = \{I \in \mathcal{U} \mid I \subseteq \mathrm{Cov}(P, \mathcal{I})\}$*. The coverage of a set* C *of pathlets is* $\mathrm{Cov}_{\mathcal{U}}(C) = \sum_{(P,\mathcal{I}) \in C} \mathrm{Cov}_{\mathcal{U}}(P, \mathcal{I})$*.*

Whenever $\mathcal{U}$ is clear from context we denote the coverage of a pathlet $(P, \mathcal{I})$ by $\mathrm{Cov}(P, \mathcal{I})$.

▶ **Definition 4** (Reference optimal). *Let the universe* $\mathcal{U}$ *be fixed and let* C *be a set of* (ℓ, Δ)*-pathlets. An* (ℓ, Δ)*-pathlet* $(P, \mathcal{I})$ *is* reference-optimal *if its coverage over* $\mathcal{U} \setminus \mathrm{Cov}(C)$*, i.e.,* $|\mathrm{Cov}(P, \mathcal{I}) \backslash \mathrm{Cov}(C)|$*, is maximum over all* (ℓ, Δ)*-pathlets with the same reference curve.*

▶ **Definition 5.** *Let the universe* $\mathcal{U}$ *be fixed and let* C *be a set of* (ℓ, Δ)*-pathlets. An* (ℓ, Δ)*-pathlet* $(P, \mathcal{I})$ *is* optimal *whenever* $|\mathrm{Cov}(P, \mathcal{I}) \backslash \mathrm{Cov}(C)|$ *is maximum over all* (ℓ, Δ)*-pathlets.*

3 Algorithmic outline

Our algorithmic input is a trajectory T, an integer $\ell \geq 2$, and value $\Delta \geq 0$. We provide a high-level overview of our algorithm here. Our approach can be decomposed as follows:

1. Reference curves may lie anywhere in the ambient space. Our first step is to restrict where these reference curves may lie. In Section 4 we construct a 2Δ-simplification S of T, and prove that for any (ℓ, Δ)-pathlet $(P, \mathcal{I})$, there exists a subcurve $S[a, d]$ of S for which $(S[a, d], \mathcal{I})$ is an $(\ell + 2 - |\mathbb{N} \cap \{a, d\}|, \Delta')$-pathlet, where $\Delta' = 4\Delta$. Hence we may restrict our attention to pathlets where the reference curve is a subcurve of S, if we allow for a slightly higher complexity. This higher complexity is circumvented later on, to still give an (ℓ, Δ')-clustering.

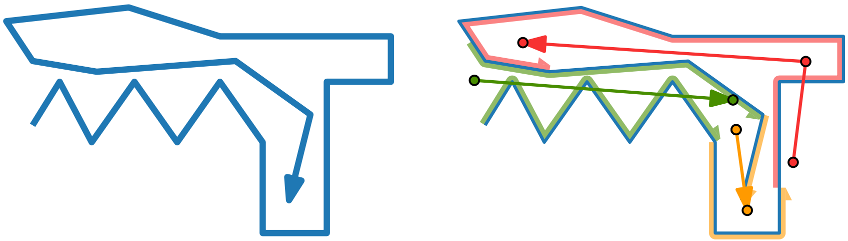

Figure 1 The trajectory T (blue, left) is covered by three pathlets. Each pathlet is defined by a reference curve (green, red, yellow) and the subcurve(s) of T the curve covers.

2. In Section 5, Given S and T, we smartly create some universe $\mathcal{U}$. We prove, by adapting the argument for greedy set cover, that any algorithm that iteratively computes an optimal (ℓ, Δ)-pathlet outputs a clustering of size $\mathcal{O}(k_\ell(\Delta) \log n)$.
3. In Section 6 we give the general algorithm. We choose some $\Delta' \in \Theta(\Delta)$. We iteratively construct an (ℓ, Δ')-clustering of size $\mathcal{O}(k_\ell(\Delta) \log n)$. Our greedy iterative algorithm maintains a set C of pathlets and adds an (ℓ, Δ')-pathlet $(P, \mathcal{I})$ to C at every iteration. Consider having a set of pathlets $C = \{(P_i, \mathcal{I}_i)\}$. We greedily select a pathlet $(P, \mathcal{I})$ that covers as much of $\mathcal{U} \setminus \mathrm{Cov}(C)$ as possible, and add it to C. Formally, we select a $(\Delta, \frac{1}{17})$-*maximal* (ℓ, Δ')-pathlet: an (ℓ, Δ')-pathlet $(P, \mathcal{I})$ such that

$$|\mathrm{Cov}(P, \mathcal{I}) \setminus \mathrm{Cov}(C)| \geq \frac{1}{17} |\mathrm{Cov}(P', \mathcal{I}') \setminus \mathrm{Cov}(C)|$$

for all (ℓ, Δ)-pathlets $(P', \mathcal{I}')$.
4. The subsequent goal is to compute $(\Delta, \frac{1}{17})$-maximal pathlets. We restrict pathlets to two types: those where the reference curve is 1) a vertex subcurve of S, or 2) a subsegment of an edge of S. Then we give algorithms for constructing pathlets of these types with a certain quality guarantee, i.e., pathlets that cover at least a constant fraction of what the optimal pathlet of that type covers. These algorithms are given in Sections 8 and 9.

Reachability graph. We introduce the *reachability graph* in Section 7. This graph is defined on a subcurve W of S and a set Z of points in Δ'-FSD(W, T). The reachability graph $G(W, T, Z)$ is a directed acyclic graph whose vertices are the set of points Z, together with certain boundary points of the free space Δ'-FSD(W, T) and a collection of *steiner points*. Given two points (x, y) and (x', y') in Z, the graph contains a directed path from (x, y) to (x', y') if and only if $d_F(W[x, x'], T[y, y']) \leq \Delta'$.

We treat the free space diagram as a rectilinear polygon $\mathcal{R}$ with rectilinear holes, obtained by reducing all obstacles of Δ'-FSD(W, T) to their intersections with the parameter space grid. We show that a bimonotone path between two points p and q exists in Δ'-FSD(W, T) if and only if a rectilinear shortest path between p and q in $\mathcal{R}$ has length $\|p - q\|_1$, the L_1-distance between p and q. The reachability graph $G(W, T, Z)$ is defined as the *shortest paths preserving graph* [20] for the set Z with respect to $\mathcal{R}$, made into a directed graph by directing edges, which are all horizontal or vertical, to the right or top. This graph has $\mathcal{O}((|W|n + |Z|) \log(n|Z|))$ complexity, and a shortest path in the graph between points in Z is also a rectilinear shortest path between the corresponding points in $\mathcal{R}$.

Vertex-to-vertex pathlets. In Section 8 we construct a pathlet where the reference curve is a vertex subcurve of S. For a given vertex $S(i)$ of S, we construct reference-optimal (ℓ, Δ')-pathlets $(S[i, i + j], \mathcal{I}_j)$ for all $j \in [\ell]$. We first identify a set Z of $\mathcal{O}(n\ell)$ *critical points* in Δ'-FSD$(S[i, i + \ell], T)$. We show that for every reference curve $S[i, i + j]$, there is a reference-optimal (ℓ, Δ')-pathlet $(S[i, i+j], \mathcal{I}_j)$ where for each interval $[y, y'] \in \mathcal{I}_j$, the points (i, y) and $(i + j, y')$ are critical points. We construct the intervals $\mathcal{I}_j$ through a sweepline algorithm over the reachability graph $G(S[i, i + \ell], T, Z)$, which has $\mathcal{O}(n\ell \log n)$ complexity. Our sweepline computes, for all $j \in [\ell]$, a reference-optimal (j, Δ')-pathlet $(S[i, i + j], \mathcal{I}_j)$ by iterating over all in-edges to critical points $(i + j, y)$ in $G(S[i, i + \ell], T, Z)$. Doing this for all i (and remembering the optimum) thereby takes $\mathcal{O}(n^2 \ell \log^2 n)$ time and $\mathcal{O}(n\ell \log n)$ space.

Subedge pathlets. In Section 9 we construct a pathlet where the reference curve is a subsegment of an edge of S. For a given edge e of S, we again first identify a set Z of $\mathcal{O}(n^2)$ *critical points* in Δ'-FSD(e, T). However, rather than restricting the intervals in pathlets

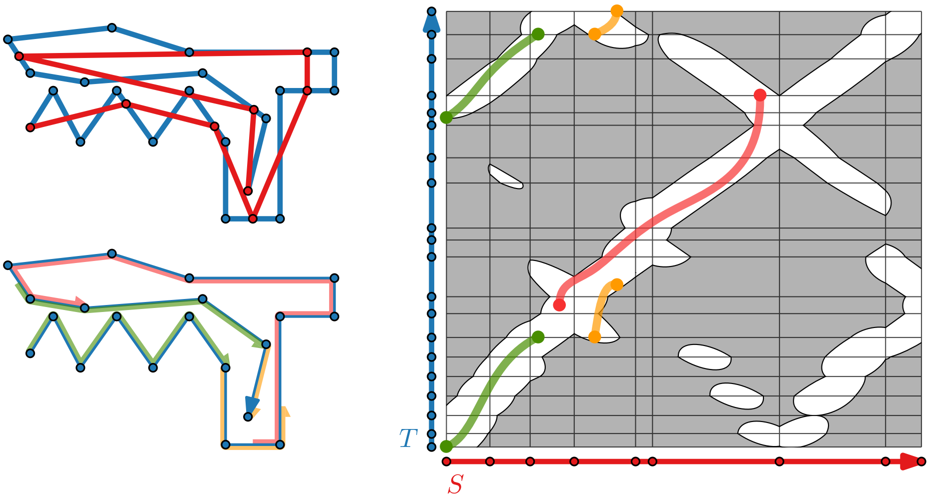

Figure 2 Top left: A simplification S (red) of the trajectory T (blue). **Right:** The diagram Δ'-FSD(S,T) in white. The obstacles of the diagram are colored in gray. The clustering (bottom left) corresponds to a set of colored bimonotone paths, where paths of a given color are horizontally aligned, and the paths together span the entire vertical axis.

based on these critical points, we restrict the reference curves based on these critical points. Specifically, there are $m = \mathcal{O}(n)$ unique x-coordinates of points in Z, which we order as $x_1, \ldots, x_m$. We show that by allowing for pathlets to use subsegments of the reversal $\overleftarrow{e}$ of e as reference curves, we may restrict reference curves to be of the form $e[x_i, x_{i'}]$ or $\overleftarrow{e}[x_i, x_{i'}]$ to not lose much coverage. That is, the optimal $(2,\Delta')$-pathlet with such a reference curve covers at least one-fourth of what any other $(2,\Delta')$-pathlet using a subsegment of e as a reference curve covers.

The remainder of our subedge pathlet construction algorithm follows the same procedure as for vertex-to-vertex pathlets, though with the following optimization. We consider every x_i separately, for $i \in [m]$. However, rather than considering all reference curves $e[x_i, x_{i'}]$, of which there are $m-i$, we consider only $\mathcal{O}(\log(m-i))$ reference curves. The main observation is that we may split a pathlet $(e[x_i, x_{i'}], \mathcal{I})$ into two: $(e[x_i, x_{i+2^j}], \mathcal{I}_1)$ and $(e[x_{i'-2^j}, x_{i'}], \mathcal{I}_2)$, for some $j \leq \log(m-i)$. One of the two pathlets covers at least half of what $(e[x_i, x_{i'}], \mathcal{I})$ covers, so an optimal $(2,\Delta')$-pathlet $(e[x_i, x_{i+2^j}], \mathcal{I})$ that is defined by critical points covers at least one-eighth of any other subedge $(2,\Delta')$-pathlet $(e[x, x'], \mathcal{I}')$.

For every $i \in [m]$, we let $Z_i \subseteq Z$ be the subset of critical points with x-coordinate equal to x_i or x_{i+2^j} for some $j \leq \log(m-i)$. We construct the reachability graph $G(e, T, Z_i)$, which has $\mathcal{O}(n \log^2 n)$ complexity. We then proceed as with the vertex-to-vertex pathlets, using a sweepline through the reachability graph. Doing this for all i (and remembering the optimal pathlet) thereby takes $\mathcal{O}(n^2 \log^3 n)$ total time and $\mathcal{O}(n \log^2 n)$ space. Taken over all edges of S, we obtain a subedge pathlet in $\mathcal{O}(n^3 \log^3 n)$ time and $\mathcal{O}(n \log^2 n)$ space.

4 Pathlet-preserving simplifications

We first aim to limit our attention to $(\ell, 4\Delta)$-pathlets $(P, \mathcal{I})$ whose reference curves P are subcurves of some universal curve S. This way, we may design an algorithm that considers all subcurves of S, rather than all curves in $\mathbb{R}^d$. This has the additional benefit of allowing the use of the free space diagram 4Δ-FSD(S,T) to construct pathlets, as seen in Figure 2.

For any (ℓ, Δ)-pathlet $(P, \mathcal{I})$ there exists an $(n, 2\Delta)$-pathlet $(P', \mathcal{I})$ where P' is a subcurve of T. Indeed, consider any interval $[a, b] \in \mathcal{I}$ and choose $P' = T[a, b]$. However, restricting the subcurves of T to have complexity at most ℓ may significantly reduce the maximum coverage, see for example Figure 3. Instead of restricting pathlets to be subcurves of T, we restrict them to be subcurves of a different curve S. We enforce the following property:

▶ **Definition 6.** *For a trajectory T and value $\Delta \geq 0$, a pathlet-preserving simplification is a curve S together with a 2Δ-matching (f, g), where for any subtrajectory $T[a, b]$ of T and all curves P with $d_F(P, T[a, b]) \leq \Delta$, the subcurve $S[s, t]$ matched to $T[a, b]$ by (f, g) has complexity $|S[s, t]| \leq |P| + 2 - |\mathbb{N} \cap \{s, t\}|$.*

▶ **Theorem 7.** *Let (S, f, g) be a pathlet-preserving simplification of T. For any (ℓ, Δ)-pathlet $(P, \mathcal{I})$, there exists a subcurve $S[s, t]$ such that $(S[s, t], \mathcal{I})$ is an $(\ell + 2 - |\mathbb{N} \cap \{s, t\}|, 4\Delta)$-pathlet.*

Proof. Consider any (ℓ, Δ)-pathlet $(P, \mathcal{I})$ and choose some interval $[a, b] \in \mathcal{I}$. For all $[c, d] \in \mathcal{I}$, via the triangle inequality, $d_F(T[a, b], T[c, d]) \leq 2\Delta$. Let $S[s, t]$ be the subcurve of S matched to $T[a, b]$ by (f, g). Naturally, $d_F(S[s, t], T[a, b]) \leq 2\Delta$, and so by the triangle inequality $d_F(S[s, t], T[c, d]) \leq 4\Delta$. By the definition of a pathlet-preserving simplification, we obtain that for every curve P' with $d_F(P', T[a, b]) \leq \Delta$, we have $|P'| \geq |S[s, t]| - 2 + |\mathbb{N} \cap \{s, t\}|$. In particular, setting $P' \leftarrow P$ implies that $|S[s, t]| \leq \ell + 2 - |\mathbb{N} \cap \{s, t\}|$. Thus $(S[s, t], \mathcal{I})$ is an $(\ell + 2 - |\mathbb{N} \cap \{s, t\}|, 4\Delta)$-pathlet. ◀

Prior simplifications. The curve S that we construct is a *curve-restricted $\alpha\Delta$-simplification* of T; a curve whose vertices lie on T, where for every edge $s = \overline{T(a)T(b)}$ of S we have $d_F(s, T[a, b]) \leq \alpha\Delta$. Various $\alpha\Delta$-simplification algorithms have been proposed [2, 14, 17, 19].

If T is a curve in $\mathbb{R}^2$, Guibas et al. [17] provide an $\mathcal{O}(n \log n)$ time algorithm that constructs a 2Δ-simplification S for which there is no Δ-simplification S' with $|S'| < |S|$. Their algorithm is not efficient in higher dimensions however.

Agarwal et al. [2] also construct a 2Δ-simplification S of T in $\mathcal{O}(n \log n)$ time. This was applied by Akitaya et al. [3] for their subtrajectory clustering algorithm under the discrete Fréchet distance. The simplification S has a similar guarantee as the simplification of [17]: there exists no *vertex-restricted* Δ-simplification S' with $|S'| < |S|$. This guarantee is weaker than that of [17], as vertex-restricted simplifications are simplifications formed by taking a subsequence of vertices of T as the vertices of the simplification. It can, however, be constructed efficiently in higher dimensions.

As we show in Figure 3, the complexity of a vertex-restricted Δ-simplification can be arbitrarily bad compared to the (unrestricted) Δ-simplification with minimum complexity. Brüning et al. [5] note that for the subtrajectory problem under the continuous Fréchet distance, one requires an $\alpha\Delta$-simplification whose complexity has guarantees with respect

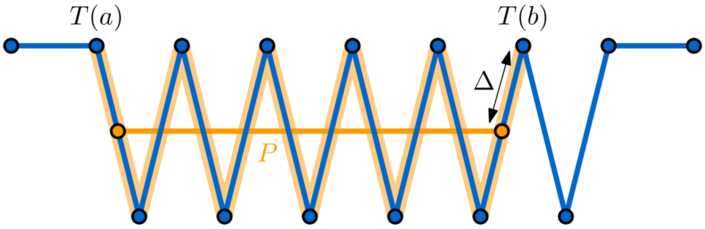

Figure 3 There exists a segment P where $d_F(P, T[a, b]) \leq \Delta$. In contrast, for any vertex-restricted S with $d_F(T[a, b], S) \leq \Delta$, the complexity of S is $\Theta(|T[a, b]|)$.

to the optimal (unrestricted) simplification. They present a 3Δ-simplification S (whose definition was inspired by de Berg, Gudmundsson and Cook [14]) with the following property: for any subcurve $T[a, b]$ of T within Fréchet distance Δ of some line segment, there exists a subcurve $S[s, t]$ of S with complexity at most 4 that has Fréchet distance at most 3Δ to $T[a, b]$. Thus, there exists no Δ-simplification S' with $|S'| < |S|/2$.

Our new curve simplification. In Definition 6 we presented yet another curve simplification under the Fréchet distance for curves in $\mathbb{R}^d$. Our simplification has a stronger property than the one that is realized by Brüning et al. [5]: for any subcurve $T[a, b]$ and *any* curve P with $d_F(P, T[a, b]) \leq \Delta$, we require that there exists a subcurve $S[s, t]$ with $d_F(S[s, t], T[a, b]) \leq 2\Delta$ that has at most two more vertices than P. This implies both the property of Brüning et al. [5] and ensures that no Δ-simplification S' exists with $|S'| < |S| - 2$.

In the full version of our paper we provide an efficient algorithm for constructing pathlet-preserving simplifications. The algorithm is an extension of the vertex-restricted simplification of Agarwal et al. [2] to construct a curve-restricted simplification instead. For this, we use the techniques of Guibas et al. [17] to quickly identify if an edge of T is suitable to place a simplification vertex on. We combine this check with the algorithm of [2] and obtain:

▶ **Theorem 8.** *For any trajectory T with n vertices and any $\Delta \geq 0$, we can construct a pathlet-preserving simplification S in $\mathcal{O}(n \log n)$ time.*

5 The universe $\mathcal{U}$ and greedy set cover

Subtrajectory clustering is closely related to the *set cover* problem. In this problem, we have a discrete universe $\mathcal{U}$ and a family of sets $\mathcal{S}$ in this universe, and the goal is to pick a minimum number of sets in $\mathcal{S}$ such that their union is the whole universe. The decision variant of set cover is NP-complete [18]. However, the following greedy strategy gives an $\mathcal{O}(\log |\mathcal{U}|)$ approximation of the minimal set cover size [12]. Suppose we have picked a set $\hat{\mathcal{S}} \subseteq \mathcal{S}$ that does not yet cover all of $\mathcal{U}$. The idea is then to add a set $S \in \mathcal{S}$ that maximizes $|S \cap (\mathcal{U} \setminus \bigcup \hat{\mathcal{S}})|$, and to repeat the procedure until $\mathcal{U}$ is fully covered.

Defining the universe $\mathcal{U}$. We apply this greedy strategy to subtrajectory clustering, putting the focus on constructing a pathlet that covers the most of some universe $\mathcal{U}$. For subtrajectory clustering, the universe is, in principle, infinite. We therefore first define a discrete universe $\mathcal{U}$ consisting of $\mathcal{O}(n^3)$ intervals that together cover $[1, n]$. We choose this universe carefully, as an optimal covering of $\mathcal{U}$ with pathlets must have roughly the same size as an optimal covering of $[1, n]$. We define $\mathcal{U}$ using the following set of *critical points* in Δ'-FSD(S, T):

▶ **Definition 9.** *For $i \in [|S| - 1]$ and $j \in [n - 1]$, consider their corresponding cell (the area $[i, i + 1] \times [j, j + 1]$) and the following six extreme points:*

- *A leftmost point of Δ'-FSD$(S, T) \cap ([i, i + 1] \times [j, j + 1])$,*
- *A rightmost point of Δ'-FSD$(S, T) \cap ([i, i + 1] \times [j, j + 1])$,*
- *The leftmost and rightmost points of Δ'-FSD$(S, T) \cap ([i, i + 1] \times \{j\})$, and*
- *The leftmost and rightmost points of Δ'-FSD$(S, T) \cap ([i, i + 1] \times \{j + 1\})$.*

Let $X_{i,j}$ be the set of corresponding x-coordinates and $X := \bigcup_{i,j} X_{i,j}$. For each $x \in X$, we call every point (x, y) that is an endpoint of a connected component (vertical segment) of Δ'-FSD$(S, T) \cap (\{x\} \times [1, n])$ a critical point.

▶ **Definition 10.** *Let Y^* be the set of critical points, sorted by their y-coordinate. We define the set $\mathcal{U}$ as the set of intervals in $[1, n]$ between two consecutive y-coordinates in X. Since there are at most $6n$ critical points in $[i, i+1] \times [j, j+1]$ for each $i \in [|S| - 1]$ and $j \in [n-1]$, it follows that $|\mathcal{U}| \leq 6n^3 - 1 = \mathcal{O}(n^3)$.*

▶ **Lemma 11.** *We can construct $\mathcal{U}$ in $\mathcal{O}(n^3)$ time.*

Proof. Fix an integer $i \in [|S| - 1]$. We compute the critical points inside the cells $[i, i+1] \times [j, j+1]$, for all $j \in [n-1]$, in $\mathcal{O}(n^2)$ time altogether. For this, we compute the sets $X_{i,j}$ of Definition 9 in $\mathcal{O}(1)$ time each. Let $X_i = \bigcup_j X_{i,j}$. Then, we compute the intersections of each vertical line $\{x\} \times [1, n]$, for $x \in X_i$, in $\mathcal{O}(n)$ time each. The critical points inside the cells $[i, i+1] \times [j, j+1]$, for $j \in [n-1]$, are the endpoints of connected components of these intersections, and can be computed in $\mathcal{O}(n)$ time per line, totalling $\mathcal{O}(n^2)$ time. Summing over all integers i completes the proof. ◀

Applying greedy set cover. In the remainder of this paper, we let $\mathcal{U}$ denote this discrete universe. For notational convenience, we denote for a pathlet $(P, \mathcal{I})$ by $\text{Cov}(P, \mathcal{I})$ the set $\text{Cov}_{\mathcal{U}}(P, \mathcal{I})$. We generalize the analysis of the greedy set cover argument to pathlets that cover a (constant) fraction of what the optimal pathlet covers. This relaxes the requirements on the pathlets and helps reduce complexity of the problem. For this, we introduce the following:

▶ **Definition 12** (Maximal pathlets). *Given a set C of pathlets, a $(\Delta, \frac{1}{c})$-maximal (ℓ, Δ')-pathlet $(P', \mathcal{I}')$ is a pathlet such that there exists no (ℓ, Δ)-pathlet $(P, \mathcal{I})$ with*

$$\frac{1}{c}|\text{Cov}(P, \mathcal{I}) \setminus \text{Cov}(C)| \geq |\text{Cov}(P', \mathcal{I}') \setminus \text{Cov}(C)|.$$

In Lemma 13, we show that if we keep greedily selecting $(\Delta, \frac{1}{c})$-maximal pathlets for our clustering, the size of the clustering stays relatively small compared to the optimum size. The bound closely resembles the bound obtained by the argument for greedy set cover.

▶ **Lemma 13.** *Iteratively adding $(\Delta, \frac{1}{c})$-maximal pathlets yields a clustering of size at most $3c \cdot k_\ell(\Delta) \ln(6n) + 1$.*

Proof. Let $C^* = \{(P_i, \mathcal{I}_i)\}_{i=1}^k$ be an (ℓ, Δ)-clustering of T of minimal size. Then $k := |C^*| = k_\ell(\Delta)$. Consider iteration j of the algorithm, where we have some set of (ℓ, Δ')-pathlets C_j. Denote by $W_j = |\mathcal{U}| \setminus \text{Cov}(C_j)$ the "size" of the part of the universe that still needs to be covered. Since C^* covers $\mathcal{U}$, it must cover $\mathcal{U} \setminus \text{Cov}(C_j)$. It follows via the pigeonhole principle that there is at least one (ℓ, Δ)-pathlet $(P_i, \mathcal{I}_i) \in C^*$ that covers at least W_j/k intervals in $\text{Cov}(P_i, \mathcal{I}_i) \setminus \text{Cov}(C_j)$. Per definition of being $(\Delta, \frac{1}{c})$-maximal, our greedy algorithm finds a pathlet $(P_j, \mathcal{I}_j)$ that covers at least $\frac{W_j}{ck}$ uncovered intervals. Thus:

$$W_{j+1} = |\mathcal{U}| - |\text{Cov}(C_j) \cup \text{Cov}(P_j, \mathcal{I}_j)| \leq W_j - \frac{W_j}{c \cdot k} = W_j \cdot (1 - \frac{1}{c \cdot k}).$$

We have that $W_0 = |\mathcal{U}|$. Suppose it takes $k' + 1$ iterations to cover all of T' with the greedy algorithm. Then before the last iteration, at least one edge of T' remained uncovered. That is, $|\mathcal{U}| \cdot \left(1 - \frac{1}{c \cdot k}\right)^{k'} \geq 1$. We apply that $e^x \geq 1 + x$ for all real x to obtain:

$$\frac{1}{e} \geq \left(1 - \frac{1}{x}\right)^x$$

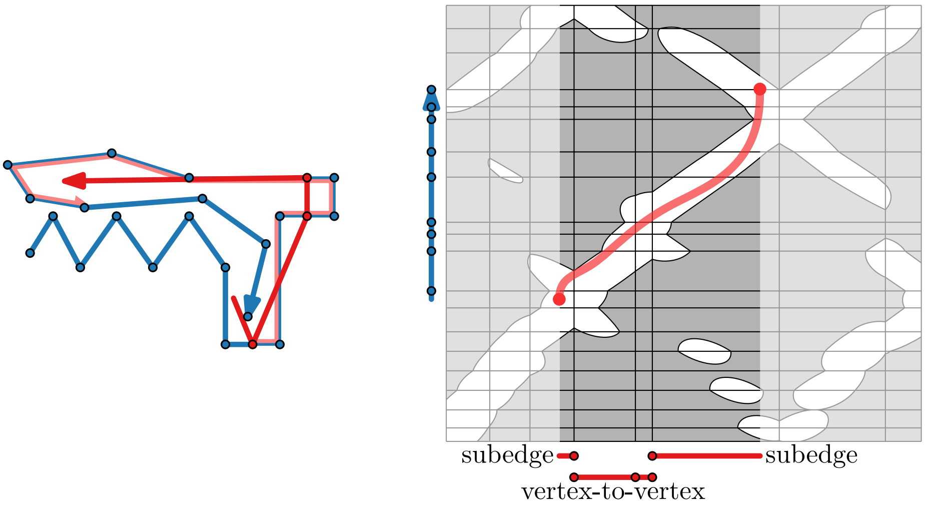

Figure 4 A pathlet (left), corresponding to the red Δ'-matching (right), gets split into a vertex-to-vertex and two subedge pathlets. The new pathlets correspond to the parts of the red matching that are vertically above the part of the x-axis corresponding to the new reference curve.

for all $x \geq 1$. Plugging in $x \leftarrow c \cdot k$, it follows that

$$1 \leq |\mathcal{U}| \cdot \left(1 - \frac{1}{c \cdot k}\right)^{k'} = |\mathcal{U}| \cdot \left(1 - \frac{1}{c \cdot k}\right)^{c \cdot k \cdot \frac{k'}{c \cdot k}} \leq |\mathcal{U}| \cdot e^{-\frac{k'}{c \cdot k}}.$$

Hence $e^{\frac{k'}{c \cdot k}} \leq |\mathcal{U}| - 1$, showing that $k' \leq c \cdot k \ln(|\mathcal{U}| - 1)$. Thus after $k' + 1 \leq c \cdot k_\ell(\Delta) \ln(|\mathcal{U}| - 1) + 1$ iterations, all of T', and therefore T, is covered. Using that $|\mathcal{U}| \leq 6n^3$ completes the proof. ◀

6 Subtrajectory clustering

In this section we present our algorithm for subtrajectory clustering. We first restrict our attention to reference curves of two types.

Recall that using the pathlet-preserving simplification S of T, we may already restrict our attention to reference curves that are subcurves of S. Still, the space of possible reference curves remains infinite. We wish to discretize this space by identifying certain finite classes of reference curves that contain a "good enough" reference curve, i.e., one with which we can construct a pathlet that is $(\Delta, \frac{1}{c})$-maximal for some small constant c.

We distinguish between two types of pathlets, based on their reference curves (note that not all pathlets fit into a class, and that some may fit into both classes):

1. Vertex-to-vertex pathlets: pathlets $(P, \mathcal{I})$ where P is a vertex subcurve of S.
2. Subedge pathlets: pathlets $(P, \mathcal{I})$ where P is a subsegment of an edge of S.

We construct pathlets of the above types, ensuring that they all cover at least some constant fraction of the optimal coverage for pathlets of the same type. Let $(P_{\text{ver}}, \mathcal{I}_{\text{ver}})$ and $(P_{\text{sub}}, \mathcal{I}_{\text{sub}})$ respectively be a vertex-to-vertex and subedge (ℓ, Δ')-pathlet, that respectively cover at least a factor $\frac{1}{c_{\text{ver}}}$ and $\frac{1}{c_{\text{sub}}}$ of an optimal pathlet of the same type. We show that one of these two pathlets is a $(\Delta, \frac{1}{c})$-maximal pathlet, for $c = c_{\text{ver}} + 2c_{\text{sub}}$. For intuition, refer to Figure 4.

▶ **Lemma 14.** *Given a collection C of pathlets, let*

$$(P, \mathcal{I}) \in \{(P_{\text{ver}}, \mathcal{I}_{\text{ver}}), (P_{\text{sub}}, \mathcal{I}_{\text{sub}})\}$$

be a pathlet with maximal coverage among the uncovered points. Then $(P, \mathcal{I})$ is $(\Delta, \frac{1}{c})$-maximal with respect to C, for $c = c_{\text{ver}} + 2c_{\text{sub}}$.

Next we combine the previous ideas on simplification and greedy algorithms and present our algorithm for subtrajectory clustering. The algorithm uses subroutines for constructing the two types of pathlets described above, as well as a data structure for comparing their coverages to select the best pathlet for the clustering.

Our pathlet construction algorithms guarantee that $c_{\text{ver}} = 1$ and $c_{\text{sub}} = 8$. By Lemma 14, the pathlet with the most coverage is therefore $(\Delta, \frac{1}{c})$-maximal with respect to the uncovered points, for $c = 1 + 2 \cdot 8 = 17$. By Lemma 13, the resulting (ℓ, Δ')-clustering has a size of at most $17k_\ell(\Delta)\ln(|\mathcal{U}| - 1) + 1$. We show in our constructions of the pathlets that $|\mathcal{U}| \leq 2n^2 + 4n^3 \leq 6n^3$.

A data structure for comparing pathlets. Recall that we fixed some discrete universe $\mathcal{U}$ of $\mathcal{O}(n^3)$ intervals, and that we denote $\text{Cov}(P,\mathcal{I}) = \text{Cov}_{\mathcal{U}}(P,\mathcal{I})$. In each iteration of our greedy algorithm, we select one of two pathlets whose coverage is the maximum over $\mathcal{U} \setminus \text{Cov}(C)$, given the current set of picked pathlets C. To compare the coverages of pathlets, we make use of binary search trees built on $\mathcal{U}$ and $\text{Cov}(C)$:

▶ **Lemma 15.** *In $\mathcal{O}(n^3 \log n)$ time, we can preprocess $\mathcal{U}$ and $\text{Cov}(C)$ into a data structure of $\mathcal{O}(n^3)$ size, such that given a pathlet $(P,\mathcal{I})$, the value $|\text{Cov}(P,\mathcal{I}) \setminus \text{Cov}(C)|$ can be computed in $\mathcal{O}(|\mathcal{I}| \log n)$ time.*

Proof. We make use of a general data structure for storing a set $\mathcal{I}$ of m interior-disjoint intervals, such that given a query interval I, the number of intervals in $\mathcal{I}$ that are fully contained in I can be reported efficiently. For the data structure, we store the (multiset of) endpoints of intervals in $\mathcal{I}$ in a balanced binary search tree. The tree uses $\mathcal{O}(m)$ space and is constructed in $\mathcal{O}(m \log m)$ time.

We report the number of intervals in $\mathcal{I}$ contained in a query interval I as follows. An interval $[a, b] \in \mathcal{I}$ is contained in I if and only if both a and b are. Furthermore, there are $k' \leq 2$ intervals in $\mathcal{I}$ that I intersects but does not contain. Thus, if I contains k endpoints stored in the binary search tree, then it contains $(k - k')/2$ intervals of $\mathcal{I}$. We compute k' by reporting the intervals of $\mathcal{I}$ containing the endpoints of I in $\mathcal{O}(\log m)$ time. Computing k and then reporting $(k - k')/2$ takes an additional $\mathcal{O}(\log m)$ time. Thus we answer a query in $\mathcal{O}(\log m)$ time.

We use the above data structure to efficiently compute $|\text{Cov}(P,\mathcal{I}) \setminus \text{Cov}(C)|$ for a query pathlet $(P,\mathcal{I})$. For this, we preprocess both $\mathcal{U}$ and $\text{Cov}(C)$ into the above data structure. Since $\text{Cov}(C) \subseteq \mathcal{U}$ and $|\mathcal{U}| = \mathcal{O}(n^3)$, this takes $\mathcal{O}(n^3 \log n)$ time, and the data structures use $\mathcal{O}(n^3)$ space. With the two data structures, we report the values $|\text{Cov}_{\mathcal{U}}(P,\mathcal{I}) \cap \mathcal{U}|$ and $|\text{Cov}(P,\mathcal{I}) \cap \text{Cov}(C)|$ in $\mathcal{O}(\log n)$ time. We then report

$$|\text{Cov}(P,\mathcal{I}) \setminus \text{Cov}(C)| = |\text{Cov}(P,\mathcal{I}) \cap \mathcal{U}| - |\text{Cov}(P,\mathcal{I}) \cap \text{Cov}(C)|. \qquad ◀$$

Asymptotic complexities. Our algorithm iteratively constructs a set C of $\mathcal{O}(k_\ell(\Delta) \log n)$ pathlets. Before we start constructing pathlets, we compute the universe $\mathcal{U}$ of $\mathcal{O}(n^3)$ intervals. This takes $\mathcal{O}(n^3)$ time (Lemma 11).

In each iteration, we construct the data structure of Lemma 15 on the universe $\mathcal{U}$ and current set of pathlets C. This takes $\mathcal{O}(n^3 \log n)$ time and uses $\mathcal{O}(n^3)$ space. Constructing the vertex-to-vertex pathlet then takes $\mathcal{O}(n^2 \ell \log^2 n)$ time and uses $\mathcal{O}(n\ell \log n)$ space (Theorem 19). The subedge pathlet takes $\mathcal{O}(n^3 \log^3 n)$ time and $\mathcal{O}(n \log^2 n)$ space to construct (Theorem 21).

To decide which pathlet to use in the clustering, we make further use of the data structure of Lemma 15. All constructed pathlets $(P, \mathcal{I})$ have $|\mathcal{I}| \leq n$, and so we compute the coverages of the two pathlets in $\mathcal{O}(n \log n)$ time. By summing up all complexities, we derive our main theorem:

▶ **Theorem 16.** *Given a trajectory T with n vertices, an integer $\ell \geq 2$, and a value $\Delta \geq 0$, we can construct an $(\ell, 4\Delta)$-clustering of size at most $51k_\ell(\Delta)\ln(6n)+1$ in $\mathcal{O}(k_\ell(\Delta)n^3 \log^4 n)$ time and using $\mathcal{O}(n^3)$ space.*

7 The reachability graph

Let $\Delta' = 4\Delta$. For any subcurve W of S and a set of points Z in Δ'-FSD(W, T) we define the *reachability graph* $G(W, T, Z)$. The vertices of this graph are the set of points Z, together with some Steiner points in $[1, |W|] \times [1, |T|]$. The reachability graph $G(W, T, Z)$ is a directed graph where, for any two $\mu_1, \mu_2 \in Z$, there exists a directed path from μ_1 to μ_2 if and only if μ_2 is reachable from μ_1 in the free space Δ'-FSD(W, T).

A more detailed description of the reachability graph is provided in the full version. There we show that our definition of the graph has $\mathcal{O}((n|W| + |Z|)\log(n|Z|))$ vertices and edges, and can be constructed in $\mathcal{O}((n|W| + |Z|)\log(n|Z|))$ time.

▶ **Theorem 17.** *Let W be a subcurve of S and Z a set of points in Δ'-FSD(W, T). The reachability graph $G(W, T, Z)$ has $\mathcal{O}((n|W| + |Z|)\log(n|Z|))$ vertices and edges, and can be constructed in $\mathcal{O}((n|W| + |Z|)\log(n|Z|))$ time.*

8 Vertex-to-vertex pathlets

Let $\Delta' = 4\Delta$, and let C be a set of pathlets. Recall that we can compute $|\mathrm{Cov}(P, \mathcal{I}) \setminus \mathrm{Cov}(C)|$, for a given pathlet $(P, \mathcal{I})$, in $\mathcal{O}(|\mathcal{I}| \log n)$ time (Lemma 15). We fix some integer i. We then give an algorithm for constructing a vertex-to-vertex (ℓ, Δ')-pathlet $(P, \mathcal{I})$ where P starts at the i'th vertex of S, and its coverage over $\mathcal{U} \setminus \mathrm{Cov}(C)$ is maximum.

We find for each subcurve S' of S of length at most ℓ a reference-optimal (ℓ, Δ')-pathlet. To this end, we consider each vertex $S(i)$ of S separately. We construct a set of reference-optimal pathlets $(S[i, i+1], \mathcal{I}_1), \ldots, (S[i, i+j], \mathcal{I}_j), \ldots, (S[i, i+\ell], \mathcal{I}_\ell)$. We let each interval $\mathcal{I}_j$ contain all maximal intervals $[y, y']$ for which $d_F(S[i, i+j], T[y, y']) \leq \Delta'$, and thus all maximal intervals for which (i, y) can reach $(i+j, y')$ by a bimonotone path in Δ'-FSD(S, T).

Recall that in Definition 9 we defined a set of *critical points*. Let Z denote all critical points in Δ'-FSD$(S[i, i+\ell], T)$ of the form $(i+j, y)$, for integers $j \in [\ell]$. That is, Z contains for all $j \in [\ell]$ the endpoints of all connected components (vertical line segments) of Δ'-FSD$(S, T) \cap (\{i+j\} \times [1, n])$. Since each cell has at most $\mathcal{O}(1)$ such critical points, it follows that $|Z| \in \mathcal{O}(n\ell)$. Observe that for any Δ'-matching between T and a vertex-to-vertex subcurve, we can always extend each curve in the matching to start and end at a point in Z:

▶ **Observation 18.** *Let $(P, \mathcal{I})$ be a vertex-to-vertex (ℓ, Δ')-pathlet where P starts at the i'th vertex. Then there exists an (ℓ, Δ')-pathlet $(P, \mathcal{I}')$ with $\mathrm{Cov}(P, \mathcal{I}) \subseteq \mathrm{Cov}(P, \mathcal{I}')$ such that for each interval in $\mathcal{I}'$, the corresponding bimonotone path in Δ'-FSD(S, T) starts and ends at a point in Z.*

We create a sweepline algorithm that, for each $j \in [\ell]$, constructs a reference-optimal (ℓ, Δ')-pathlet $(S[i, j], \mathcal{I}_j)$. We let each interval $\mathcal{I}_j$ contain all maximal intervals $[y, y']$ for which $d_F(S[i, i+j], T[y, y']) \leq \Delta'$, and thus all maximal intervals for which (i, y) can reach

$(i+j, y')$ by a bimonotone path in Δ'-FSD(S,T). Note that both (i,y) and $(i+j,y')$ are critical points. Thus we aim to find all maximal intervals $[y,y']$ for which Δ'-FSD(S,T) contains a bimonotone path between critical points (i,y) and $(i+j,y')$.

To this end, we construct, for each $i \in [n]$, the reachability graph $G(S[i,i+\ell],T,Z)$ from Section 7, which encodes reachability between all critical points. This graph takes $\mathcal{O}((n\ell + |Z|)\log(n|Z|)) = \mathcal{O}(n\ell \log n)$ time to construct and has complexity $\mathcal{O}(n\ell \log n)$ (see Theorem 17). We aim to annotate each vertex μ (which does not necessarily have to be a critical point) in $G(S[i,i+\ell],T,Z)$ with the minimum y, such that there exists a critical point (i,y) that can reach μ. We annotate μ with ∞ if no such value y exists.

Annotating vertices. We begin by annotating the vertices (i,y) in $\mathcal{O}(n)$ time, by scanning over them in order of increasing y-coordinate. We go over the remaining vertices in yx-lexicographical order, where we go over the vertices based on increasing y-coordinate, and increasing x-coordinate when ties arise. Each vertex μ that we examine has only incoming arcs originating from vertices below and left of μ. By our lexicographical ordering, each of these vertices are already annotated. The minimal y for which there exists a path from (i,y) to μ, must be the minimum over all its incoming arcs which we compute in time proportional to the in-degree of μ. If μ has no incoming arcs, we annotate it with ∞.

Let V and A be the sets of $\mathcal{O}(n\ell \log n)$ vertices and arcs of $G(S[i,i+\ell],T,Z)$. For the above annotation procedure, we first compute the yx-lexicographical ordering of the vertices, based on their corresponding points in the parameter space. This takes $\mathcal{O}(|V|\log|V|)$ time. Afterwards, we go over each vertex and each incoming arc exactly once, which take an additional $\mathcal{O}(|V|+|A|)$ time. In total, we annotate all vertices in $\mathcal{O}(n\ell \log^2 n)$ time.

Constructing the pathlets. With the annotations, constructing the pathlets becomes straightforward. For each $j \in [\ell]$, we construct $\mathcal{I}_j$ as follows. We iterate over all critical point $(i+j,y')$ in the graph $G(S[i,i+\ell],T,Z)$. For each critical point $(i+j,y')$ with a finite annotation y, we add the interval $[y,y']$ to $\mathcal{I}_j$. This procedure ensures that $\mathcal{I}_j$ contains all maximal intervals $[y,y']$ for which $d_F(S[i,i+j],T[y,y']) \leq \Delta'$, creating an optimal pathlet $(S[i,i+j],\mathcal{I}_j)$ with respect to its reference curve. Since there are $\mathcal{O}(n)$ critical points per j, this algorithm uses $\mathcal{O}(n\ell)$ time. Storing the pathlets takes $\mathcal{O}(n\ell)$ space. We conclude:

▶ **Theorem 19.** *Suppose* Cov(C) *is preprocessed by Lemma 15. In $\mathcal{O}(n^2\ell \log^2 n)$ time and using $\mathcal{O}(n\ell \log n)$ space, we can construct an optimal vertex-to-vertex (ℓ,Δ')-pathlet $(P,\mathcal{I})$.*

Proof. For a given vertex $S(i)$, we compute optimal pathlets $(S[i,i+j],\mathcal{I}_j)$ with respect to their reference curves for $j \in [\ell]$ in $\mathcal{O}(n\ell \log^2 n)$ time, using $\mathcal{O}(n\ell \log n)$ space. Using the data structure of Lemma 15, we subsequently compute the coverage of one of these pathlets $\mathcal{O}(n \log n)$ time, so $\mathcal{O}(n\ell \log n)$ time for all. We pick the best pathlet and remember its coverage. Doing so for all vertices $S(i)$ of S, we obtain $|S|$ pathlets, of which we report the best. By only keeping the best pathlet in memory, rather than all $|S|$, the space used by these pathlets is lowered from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. ◀

9 Subedge pathlets

Let $\Delta' = 4\Delta$, and let C be a set of pathlets. We assume that Cov(C) has at most $\mathcal{O}(n^2 \log n)$ connected components. We construct a subedge $(2,\Delta')$-pathlet $(P,\mathcal{I})$ whose coverage over $\mathcal{U} \setminus$ Cov(C) is at least $\frac{1}{8}$'th of the the optimum.

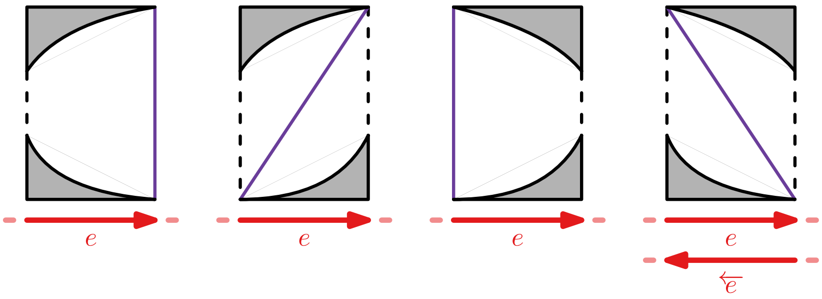

Figure 5 The connected components of Δ'-FSD(e', T) fall into these four cases, based on where the minima and maxima of the bottom and top parabolic arcs lie. In the first three cases, a clear matching with optimal coverage exists (purple). In the fourth case, the matching is only valid when mirroring the free space, achieved by using $\overleftarrow{e}$ instead of e.

Recall that a subedge pathlet $(P, \mathcal{I})$ is a pathlet where $P = e[x, x']$ is a subsegment of some edge e of S. We construct a subedge pathlet given the edge e. To this end, recall that in Definition 9 we defined a set of *critical points*. Let Z denote all critical points in Δ'-FSD(e, T). Since there are at most $\mathcal{O}(n)$ critical points per cell in the free-space diagram, there are at most $\mathcal{O}(n^2)$ critical points in Z. We fix Z throughout this section and compute subedge pathlet $(e[x, x'], \mathcal{I})$ with at least one-fourth the coverage of any pathlet that uses a subedge of e as a reference curve, and where for all $[y, y'] \in \mathcal{I}$, the points (x, y) and (x', y') are both critical points.

Before we restrict pathlets to be defined by these critical points, we initially allow a broader range of pathlets. We consider the edge $\overleftarrow{e}$, obtained by reversing the direction of e, and look at constructing a pathlet that is a subedge of either e or $\overleftarrow{e}$. We show that by restricting pathlets to be defined by Z, allowing for reference curves that are subcurves of $\overleftarrow{e}$, we lose only a factor four in the maximum (discrete) coverage.

▶ **Lemma 20.** *Let C be a set of pathlets. For any subedge $(2, \Delta')$-pathlet $(e[x, x'], \mathcal{I})$, there exists a subedge $(2, \Delta')$-pathlet $(P, \mathcal{I}')$ with*

$$|\mathrm{Cov}(P, \mathcal{I}') \setminus \mathrm{Cov}(C)| \geq \frac{1}{4}|\mathrm{Cov}(e[x, x'], \mathcal{I}) \setminus \mathrm{Cov}(C)|,$$

where P is equal to $e[x_i, x_j]$ or $\overleftarrow{e}[x_i, x_j]$ for some i and j, and for every interval $[y, y'] \in \mathcal{I}'$, the points (x_i, y) and (x_j, y') are contained in Z.

Proof. Consider a subedge $(2, \Delta')$-pathlet $(e[x, x'], \mathcal{I})$. Any interval $[a, b] \in \mathcal{I}$ corresponds to a bimonotone path from (x, a) to (x', b) in Δ'-FSD(e, T). Consider such an interval $[a, b]$ and a corresponding path π.

Suppose first that $x_i \leq x \leq x' \leq x_{i+1}$ for some i. Observe that every connected component of Δ'-FSD$(e, T) \cap ([x_i, x_{i+1}] \times [1, n])$ is bounded on the left and right by (possibly empty) vertical line segments, and that the bottom and top chains are parabolic curves whose extrema are the endpoints of these segments. In particular, these connected components are convex. Thus there is a straight line segment e' from (x, a) to (x', b) in the free space. The line segment e^* connecting the extrema of the parabolic curves bounding the connected component containing (x, a) and (x', b) is longer than e'. The endpoints of e^* are both critical points, and e^* describes a valid matching between a subcurve of T and either a subcurve of e, or a subcurve of e'. See Figure 5. As there are four different reference curves we choose from, the resulting intervals are spread over four different pathlets. Therefore, one of the pathlets must have at least one-fourth the coverage of any subedge pathlet.

Next suppose that $x_i \leq x \leq x_{i+1} < x'$ for some i. At some point, π reaches a point (x_{i+1}, a'). Let (x^*, y^*) be the lowest point in the connected component containing (x, a). This point is a critical point. By convexity, the segment e^* from (x^*, y^*) to (x_{i+1}, a') lies in the free space. Because $y^* \leq a$ by our choice of (x^*, y^*), we may connect e^* to the suffix of π that starts at (x_{i+1}, a') to obtain a matching that starts at a critical point and that covers at least as much of T as the original matching. Applying a symmetric procedure to the end (x', b) of π yields a matching that starts and ends at critical points without losing coverage. Again, since there are four different reference curves to choose from for the intervals in $\mathcal{I}$, the resulting intervals are spread over four different pathlets. One of the pathlets must therefore have at least one-fourth the coverage of any subedge pathlet. ◀

We find for each point $e(x_i)$ of e corresponding to a critical point a subedge pathlet whose reference curve starts at $e(x_i)$ and ends at some point $e(x_j)$ that also corresponds to a critical point. To this end, we consider each point $e(x_i)$ separately. We proceed akin to the construction for vertex-to-vertex pathlets Section 8, with some optimization steps.

It proves too costly to consider each reference curve $e[x_i, x_{i'}]$ for every x_i we consider. By sacrificing the quality of the pathlet slightly, settling for a pathlet with at least one-eighth the coverage of any subedge pathlet rather than one-fourth, we can reduce the number of reference curves we have to consider from $\Theta(m^2) = \mathcal{O}(n^2)$ to $\mathcal{O}(m \log m)$. Let $(e[x_i, x_{i'}], \mathcal{I})$ be a subedge pathlet. We can split $e[x_i, x_{i'}]$ into two subedges $e[x_i, x_{i+2^j}]$ and $e[x_{i'-2^j}, x_{i'}]$. The matchings corresponding to $\mathcal{I}$ decompose into two sets of matchings (whose matched subcurves may overlap), giving rise to two pathlets $(e[x_i, x_{i+2^j}], \mathcal{I}_1)$ and $(e[x_{i'-2^k}, x_{i'}], \mathcal{I}_2)$ with $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}$. Thus at least one of these pathlets has at least half the coverage that $(e[x_i, x_{i'}], \mathcal{I})$ has. By Lemma 20, a pathlet $(e[x_i, x_{i+2^j}], \mathcal{I})$ that has maximum coverage out of all such pathlets then covers at least one-eighth of what any other subedge pathlet covers.

Using a sweepline algorithm analogous to that of Section 8, we construct reference-optimal (ℓ, Δ')-pathlets using each $e[x_i, x_{i+2^j}]$ and $\overleftarrow{e}[x_i, x_{i+2^j}]$ with $j \leq \log(m-i)$ as reference curves. This leads to the following result (for details, see the full version):

▶ **Theorem 21.** *Suppose that the universe $\mathcal{U}$ and the coverage* $\mathrm{Cov}(C)$ *is preprocessed into the data structure of Lemma 15. In $\mathcal{O}(n^3 \log^3 n)$ time and using $\mathcal{O}(n \log^2 n)$ space, we can construct a $(2, \Delta')$-pathlet $(P, \mathcal{I})$ with*

$$|\mathrm{Cov}(P, \mathcal{I}) \setminus \mathrm{Cov}(C)| \geq \frac{1}{8} |\mathrm{Cov}(P', \mathcal{I}') \setminus \mathrm{Cov}(C)|$$

for any subedge $(2, \Delta')$-pathlet $(P', \mathcal{I}')$.

10 Conclusion

In this work, we presented an improved approximation algorithm for subtrajectory clustering. We discuss our technical contribution, and how it differs from previous works, our asymptotic improvements and finally interesting directions for future work.

Technical contribution. Our technical contributions are threefold:

First, we introduced a new type of curve simplification in Section 4. This simplification allows us to construct a curve S, such that our clustering needs to consider only pathlets whose reference curve is a subcurve of S. Although numerous similar curve-simplification algorithms exist, our method distinguishes itself by lying significantly closer to the input curve T.

Consequently, our approximation algorithm is a 4-approximation in Δ, compared to the 11-approximations of prior works. We consider this simplification to be of independent interest, as future works may immediately use our simplification method to obtain 4-approximations in Δ also.

Secondly, we considered in Section 6 an extension to the greedy set cover algorithm wherein each iteration adds an approximately-maximum covering element, rather than a maximum one. Observe that P can always be divided into at most three subcurves, where at most one of them starts and ends at a vertex of S (a vertex-subcurve) and at most two of them are subcurves of an edge of S (a subedge of S). We design a greedy meta-algorithm, that in each iteration computes an (ℓ, Δ)-pathlet $(P, \mathcal{I})$ with approximately-maximum coverage, whose reference curve is a vertex-subcurve or subedge of S. Our approximately greedy set cover analysis shows that our meta-algorithm computes a clustering of size $\mathcal{O}(k \log n)$. A key takeaway from our construction is that by restricting our attention to vertex-subcurves and subedges of S, we significantly reduce the set of candidate pathlets from $\tilde{\mathcal{O}}(n^3\ell)$ to $\tilde{\mathcal{O}}(n^2)$. We consider this fact to also be of independent interest. Indeed, our subsequent algorithm spends near-linear time per candidate pathlet but future works may discover more efficient algorithms over the same smaller candidate set.

Finally, we presented algorithms in Sections 8 and 9 that compute the corresponding candidate pathlet for a candidate reference curve in near-linear time and near-linear space. The key observation to this contribution is that we show that it suffices to compute all candidate pathlets on the fly, significantly reducing the space.

Asymptotic improvements. Compared to the best prior deterministic work [13], our algorithm improves the running time by a factor near-linear in $n\ell$, improves the space by a factor near-linear in $n^2\ell$, and improves the approximation in Δ from a factor 11 to 4, all whilst asymptotically matching the size of the clustering. We consider this a significant improvement over the state-of-the-art.

When we compare to previous randomized work [3, 5] we improve the running time by a factor near-linear in ℓ, improve space by a factor n, and improve the approximation in Δ from a factor 11 to 4. A downside of our approach is that, compared to randomised works, we only asymptotically match the clustering size whenever ℓ is relatively large (i.e., $\ell \in \Omega(\log n / \log k)$). However, we note that on all other algorithmic quality measures, we still offer a substantial improvement whilst also being deterministic. In addition, when considering algorithmic performance in practice, we note that these previous randomized results [3, 5] use ε-net sampling. Such a sampling procedures leads to very high hidden constants in the asymptotic clustering size which makes such an approach impractical.

Future work. We think it remains an interesting open problem whether one can obtain a clustering size of $\mathcal{O}(k\ell \log k)$ in a deterministic manner. We also note that, currently, our algorithm considers a set of $\tilde{\mathcal{O}}(n^2)$ reference curves P, and computes an (ℓ, Δ)-pathlet $(P, \mathcal{I})$ with approximately-maximum coverage for each reference curve independently, in near-linear time. We think it is an interesting open problem whether one can present an algorithm that is overall more efficient whenever these maximum pathlets are considered simultaneously rather than independently.

References

1 Pankaj K. Agarwal, Kyle Fox, Kamesh Munagala, Abhinandan Nath, Jiangwei Pan, and Erin Taylor. Subtrajectory clustering: Models and algorithms. In *proc. 37th ACM SIGMOD-*

SIGACT-SIGAI Symposium on Principles of Database Systems (PODS), pages 75–87, 2018. doi:10.1145/3196959.3196972.

2 Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42(3):203–219, 2005. doi:10.1007/s00453-005-1165-y.

3 Hugo A. Akitaya, Frederik Brüning, Erin Chambers, and Anne Driemel. Subtrajectory clustering: Finding set covers for set systems of subcurves. *Computing in Geometry and Topology*, 2(1):1:1–1:48, 2023. doi:10.57717/cgt.v2i1.7.

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

5 Frederik Brüning, Jacobus Conradi, and Anne Driemel. Faster approximate covering of subcurves under the Fréchet distance. In *proc. 30th Annual European Symposium on Algorithms (ESA)*, pages 28:1–28:16, Dagstuhl, Germany, 2022. doi:10.4230/LIPIcs.ESA.2022.28.

6 Kevin Buchin, Maike Buchin, David Duran, Brittany Terese Fasy, Roel Jacobs, Vera Sacristan, Rodrigo I. Silveira, Frank Staals, and Carola Wenk. Clustering trajectories for map construction. In *proc. 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2017. doi:10.1145/3139958.3139964.

7 Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Jorren Hendriks, Erfan Hosseini Sereshgi, Vera Sacristán, Rodrigo I. Silveira, Jorrick Sleijster, Frank Staals, and Carola Wenk. Improved map construction using subtrajectory clustering. In *proc. 4th ACM SIGSPATIAL Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising*, pages 1–4, 2020. doi:10.1145/3423334.3431451.

8 Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Maarten Löffler, and Jun Luo. Detecting commuting patterns by clustering subtrajectories. *International Journal of Computational Geometry & Applications*, 21(03):253–282, 2011. doi:10.1142/S0218195911003652.

9 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In *proc. Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2922–2938, 2019. doi:10.1137/1.9781611975482.181.

10 Maike Buchin and Dennis Rohde. Coresets for (k, ℓ)-median clustering under the Fréchet distance. In *proc. 8th International Conference on Algorithms and Discrete Applied Mathematics (CALDAM)*, pages 167–180, 2022. doi:10.1007/978-3-030-95018-7_14.

11 Siu-Wing Cheng and Haoqiang Huang. Curve simplification and clustering under Fréchet distance. In *proc. 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1414–1432, 2023. doi:10.1137/1.9781611977554.ch51.

12 Vasek Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/MOOR.4.3.233.

13 Jacobus Conradi and Anne Driemel. Finding complex patterns in trajectory data via geometric set cover. *arXiv preprint arXiv:2308.14865*, 2023. doi:10.48550/arXiv.2308.14865.

14 Mark de Berg, Atlas F. Cook, and Joachim Gudmundsson. Fast Fréchet queries. *Computational Geometry*, 46(6):747–755, 2013. doi:10.1016/j.comgeo.2012.11.006.

15 Anne Driemel, Amer Krivošija, and Christian Sohler. Clustering time series under the Fréchet distance. In *proc. twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 766–785, 2016. doi:10.1137/1.9781611974331.ch5.

16 Joachim Gudmundsson and Sampson Wong. Cubic upper and lower bounds for subtrajectory clustering under the continuous Fréchet distance. In *proc. 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 173–189, 2022. doi:10.1137/1.9781611977073.9.

17 Leonidas J. Guibas, John Hershberger, Joseph S. B. Mitchell, and Jack Snoeyink. Approximating polygons and subdivisions with minimum link paths. *International Journal of Computational Geometry & Applications*, 3(4):383–415, 1993. doi:10.1142/S0218195993000257.

18 Richard M. Karp. Reducibility among combinatorial problems. In *proc. Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9.

19 Mees van de Kerkhof, Irina Kostitsyna, Maarten Löffler, Majid Mirzanezhad, and Carola Wenk. Global curve simplification. *European Symposium on Algorithms (ESA)*, 2019.

20 Peter Widmayer. On graphs preserving rectilinear shortest paths in the presence of obstacles. *Annals of Operations Research*, 33(7):557–575, 1991. doi:10.1007/BF02067242.