Global Curve Simplification

Mees van de Kerkhof
Utrecht University, The Netherlands
m.a.vandekerkhof@uu.nl

Irina Kostitsyna
TU Eindhoven, The Netherlands
i.kostitsyna@tue.nl

Maarten Löffler
Utrecht University, The Netherlands
m.loffler@uu.nl

Majid Mirzanezhad
Tulane University, New Orleans, USA
mmirzane@tulane.edu

Carola Wenk
Tulane University, New Orleans, USA
cwenk@tulane.edu

Abstract

Due to its many applications, *curve simplification* is a long-studied problem in computational geometry and adjacent disciplines, such as graphics, geographical information science, etc. Given a polygonal curve P with n vertices, the goal is to find another polygonal curve P' with a smaller number of vertices such that P' is sufficiently similar to P. Quality guarantees of a simplification are usually given in a *local* sense, bounding the distance between a shortcut and its corresponding section of the curve. In this work we aim to provide a systematic overview of curve simplification problems under *global* distance measures that bound the distance between P and P'. We consider six different curve distance measures: three variants of the *Hausdorff* distance and three variants of the *Fréchet* distance. And we study different restrictions on the choice of vertices for P'. We provide polynomial-time algorithms for some variants of the global curve simplification problem, and show NP-hardness for other variants. Through this systematic study we observe, for the first time, some surprising patterns, and suggest directions for future research in this important area.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Curve simplification, Fréchet distance, Hausdorff distance

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.67

Related Version A full version of the paper is available at `http://arxiv.org/abs/1809.10269`.

Funding *Mees van de Kerkhof*: Supported by the Netherlands Organisation for Scientific Research (NWO) under project number 628.011.005.
Maarten Löffler: Partially supported by the Netherlands Organisation for Scientific Research (NWO) under project numbers 614.001.504 and 628.011.005.
Majid Mirzanezhad: Supported by the National Science Foundation grant CCF-1637576.
Carola Wenk: Supported by the National Science Foundation grant CCF-1637576.

1 Introduction

Due to its many applications, *curve simplification* (also known as *line simplification*) is a long-studied problem in computational geometry and adjacent disciplines, such as graphics, geographical information science, etc. Given a polygonal curve P with n vertices, the goal is to find another polygonal curve P' with a smaller number of vertices such that P' is

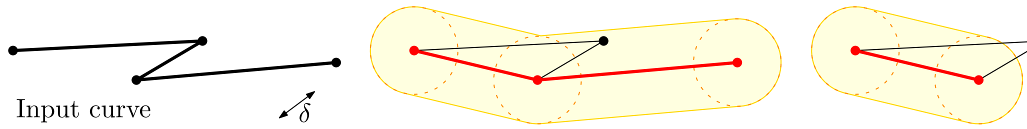

Figure 1 For a target distance δ, the red curve (middle) is a global simplification of the input curve (left), but it is not a local simplification, since the first shortcut does not closely represent its corresponding curve section (right). The example works for both Hausdorff and Fréchet distance.

sufficiently similar to P. Classical algorithms for this problem famously include a simple recursive scheme by Douglas and Peucker [16], and a more involved dynamic programming approach by Imai and Iri [21]; both are frequently implemented and cited. Since then, numerous further results on curve simplification, often in specific settings or under additional constraints, have been obtained [1, 2, 6, 11, 12, 14, 8, 18, 20].

Despite its popularity, the Douglas-Peucker algorithm comes with no provable quality guarantees. The method by Imai and Iri, though slower, was introduced as an alternative which does supply guarantees: it finds an optimal shortest path in a graph in which potential shortcuts are marked as either *valid* or *invalid*, based on their distance to the corresponding sections of the input curve. However, Agarwal et al. [2] note that the Imai-Iri algorithm does not actually globally optimize any distance measure between the original curve P and the simplification P'. This work initiated a more formal study of curve simplification; van Kreveld et al. [24] systematically show that both Douglas-Peucker and Imai-Iri may indeed produce far-from-optimal results.

This raises a question of what it means for a simplification to be optimal. We may view it as a dual-optimization problem: we wish to minimize the number of vertices of P' given a constraint on its similarity to P. This depends on the distance measure used; popular curve distance measures include the *Hausdorff* and *Fréchet* distances (variants and formal definitions are discussed in Section 2.1). However, the difference in interpretation between Agarwal et al. and Imai and Iri lies not so much in the choice of distance measure, but rather what exactly it is applied to. In fact, the Imai-Iri algorithm is optimal in a *local* sense: it outputs a subsequence of the vertices of P such that the Hausdorff distance between each shortcut and *its corresponding section of the input* is bounded: each shortcut approximates the section of P between the vertices of the shortcut.

In this work, we underline this difference by using the term *global* simplification when a bound on a distance measure must be satisfied between P and P' (formal definition in Section 2.3), and *local* simplification when a bound on a distance measure must be satisfied between each edge of P' and its corresponding section of P. Clearly, a local simplification is also a global simplification, but the reverse is not necessarily true, see Figure 1. Both local and global simplifications have their merits: one can imagine situations where it is important that each segment of a simplified curve is a good representation of the curve section it replaces, but in other applications (e.g., visualization) it is really the similarity of the overall result to the original that matters. Most existing work on curve simplification falls in the *local* category. In this work, we focus on *global* curve simplification.

1.1 Existing Work on Global Curve Simplification

Surprisingly, only a few results on simplification under global distance measures are known [2, 7, 10, 24]; consequently, what makes the problem difficult is not well understood.

Agarwal et al. [2] first consider the idea of global simplification. They introduce what they call a *weak simplification*: a model in which the vertices of the simplification are not restricted to be a subset of the input vertices, but can lie anywhere in the ambient space.[1] Interestingly, they compare this to a *local* simplification where vertices are restricted to be a subset of the input. We may interpret a combination of two of their results (Theorem 1.2 and Theorem 4.1) as an approximation algorithm for global curve simplification with unrestricted vertices under the Fréchet distance: for a given curve P and threshold δ one can compute, in $O(n \log n)$ time, a simplification P' which has at most the number of vertices of an optimal simplification with threshold $\delta/8$.

Bereg et al. [7] first explicitly consider global simplification in the setting where vertices are restricted to be a subsequence of input vertices, but using the *discrete Fréchet distance*: a variant of the Fréchet distance which only measures distances between vertices (refer to Section 2.1). They show how to compute an optimal simplification where vertices are restricted to be a subsequence in $O(n^2)$ time, and they give an $O(n \log n)$ time algorithm for the setting where vertices may be placed freely.

Van Kreveld et al. [24] consider the same (global distance, but vertices should be a subsequence) setting, but for the continuous Fréchet and Hausdorff distances. They give polynomial-time algorithms for the Fréchet distance and directed Hausdorff distance (from simplification curve to input curve), but they show the problem is NP-hard for the directed Hausdorff distance in the opposite direction and for the undirected Hausdorff distance. Recently, Bringmann and Chaudhury [10] improved their result for the Fréchet distance to $O(n^3)$, and also give a conditional cubic lower bound.

Finally, we mention there is earlier work which does not explicitly study simplification under global distance measures, but contains results that may be reinterpreted as such. Guibas et al. [19] provide algorithms for computing minimum-link paths that stab a sequence of regions in order. One of the variants, presented in Theorems 10 and 14 of [19], computes what may be seen as an optimal simplification under the Fréchet distance with no vertex restrictions, i.e., the same setting that was studied by Agarwal et al., in $O(n^2 \log^2 n)$ time.

2 Classification

We aim to provide a systematic overview of curve simplification problems under global distance measures. To this end, we have collected known results and arranged them in a table (Table 1), and provide several new results to complement these (refer to Section 2.4). This allows us for the first time to observe some surprising patterns, and it suggests directions for future research in this important area. We first discuss the dimensions of the table.

2.1 Distance Measures

For our study, we consider six different curve distance measures: three variants of the *Hausdorff* distance and three variants of the *Fréchet* distance. These are among the most popular curve distance measures in the algorithms literature. The Hausdorff distance captures the maximum distance from a point on one curve to a point on the other curve. The variants of the Hausdorff distance we consider are the directed Hausdorff distance from the input to the output, the directed Hausdorff distance from the output to the input, and the undirected

[1] We choose not to adopt the terms *weak* and *strong* in this context because we will also distinguish an intermediate model, and to avoid confusion with the *weak Fréchet* distance; refer to Section 2.2.

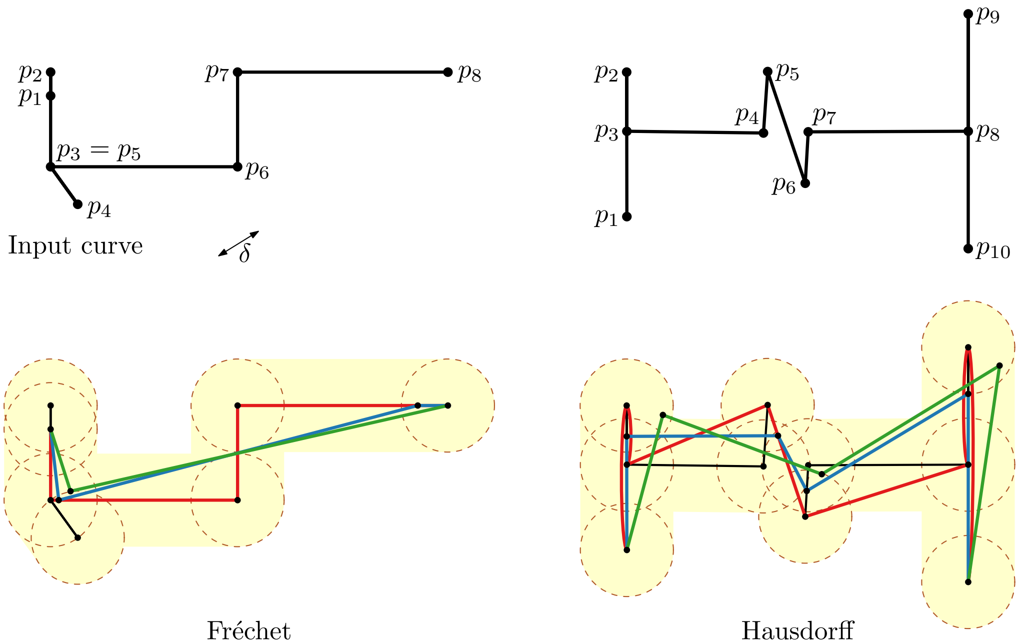

Figure 2 Globally simplified curves under Fréchet distance (left) and Hausdorff distance (right). The vertex-restricted case (in red) requires 5 vertices for Fréchet distance and 8 vertices for the Hausdorff distance. The curve-restricted case (in blue) requires 4 vertices for Fréchet distance and 6 vertices for the Hausdorff distance. The non-restricted case (in green) requires only 3 vertices for Fréchet distance and only 5 vertices for the Hausdorff distance. The δ-neighborhoods for the original curves are shown in yellow.

(or bidirectional) Hausdorff distance. The Fréchet distance captures the maximum distance between a pair of points traveling along the two curves simultaneously. We now formally define all six distance measures.

Let $P = \langle p_1, p_2, \cdots, p_n \rangle$ be the input polygonal curve. We treat P as a continuous map $P : [1, n] \to \mathbb{R}^d$, where $P(i) = p_i$ for integer i, and the i-th edge is linearly parametrized as $P(i + \lambda) = (1 - \lambda)p_i + \lambda p_{i+1}$. We write $P[s, t]$ for the subcurve between $P(s)$ and $P(t)$ and denote the *shortcut*, i.e., the straight line connecting them, by $\langle P(s)P(t) \rangle$.

The *Fréchet distance* between two polygonal curves P and Q, with n and m vertices, respectively, is $\mathsf{F}(P, Q) = \inf_{(\sigma,\theta)} \max_t \|P(\sigma(t)) - Q(\theta(t))\|$, where σ and θ are continuous non-decreasing functions from $[0, 1]$ to $[1, n]$ and $[1, m]$, respectively. If σ and θ are continuous but not necessarily monotone, the resulting infimum is called the *weak Fréchet distance*. Finally, the *discrete Fréchet distance* is a variant where σ and θ are discrete functions from $\{1, \ldots, k\}$ to $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$ with the property that $|\sigma(i) - \sigma(i + 1)| \leq 1$.

The *directed Hausdorff distance* between two polygonal curves (or more generally, compact sets) P and Q is defined as $\overrightarrow{\mathsf{H}}(P, Q) = \max_{p \in P} \min_{q \in Q} \|p - q\|$. The *undirected Hausdorff distance* is then simply the maximum over the two directions: $\mathsf{H}(P, Q) = \max\{\overrightarrow{\mathsf{H}}(P, Q), \overrightarrow{\mathsf{H}}(Q, P)\}$.

2.2 Vertex Restrictions

Once we have fixed the distance measure and agreed that we wish to apply it globally, one important design decision still remains to be made. Traditional curve simplification algorithms consider the (polygonal) input curve P to be a sequence of points, and produce

Table 1 Known and new results (in blue) for the GCS problem under global distance measures.

Distance	Vertex-restricted ($\mathcal{V}$)	Curve-restricted ($\mathcal{C}$)	Non-restricted ($\mathcal{N}$)
$\overleftarrow{\mathsf{H}}(P,\delta)$	strongly NP-hard [24]	weakly NP-hard (Thm 5)	?
$\overrightarrow{\mathsf{H}}(P,\delta)$	$O(n^4)$ [24] $\mathrm{O}(n^3 \log\ n)$ (Thm 3)	weakly NP-hard (Thm 5)	poly(n) [23]
$\mathsf{H}(P,\delta)$	strongly NP-hard [24]	strongly NP-hard (Cor 14)	strongly NP-hard (Thm 13)
$\mathsf{F}(P,\delta)$	$O(mn^5)$ [24] $O(n^3)$ [22] $O(n^3)$ [10]	$O(n)$ in $\mathbb{R}^1$ (Thm 7) weakly NP-hard in $\mathbb{R}^2$ (Thm 5)	$O(n^2 \log^2 n)$ in $\mathbb{R}^2$ [19] $O(n \log n)$ $(1,8)$-approx [2] $O^*(n^2 \log n \log\log n)$ $(2, 1+\varepsilon)$-approx (Thm 11)
$\mathsf{dF}(P,\delta)$	$O(n^2)$ [7]	$O(n^3)$ (Thm 6)	$O(n \log n)$ [7]
$\mathsf{wF}(P,\delta)$	$O(n^3)$ (Thm 2)	weakly NP-hard (Thm 5)	$(2, 1+\varepsilon)$-approx (Cor 12)

as output P' a subsequence of this sequence. However, if we measure the distance globally, there may be no strong reason to restrict the family of acceptable output curves so much: the distance measure already ensures the similarity between input and output curves, so perhaps we may allow a more free choice of vertex placement. Indeed, several results under this more relaxed viewpoint exist, as discussed in Section 1.1. Here, we choose to investigate three increasing levels of freedom: (1) *vertex-restricted* ($\mathcal{V}$), where vertices of P' have to be a subsequence of vertices of P; (2) *curve-restricted* ($\mathcal{C}$), where vertices of P' can lie anywhere on P but have to respect the order along P; and (3) *non-restricted* ($\mathcal{N}$), where vertices of P' can be anywhere in the ambient space. Figure 2 illustrates the difference between the three models. The third category does not make sense for local curve simplification, but is very natural for global curve simplification. Observe that when the vertices of a simplified curve have more freedom, the optimal simplified curve never has more, but may have fewer, vertices.

2.3 Global Curve Simplification Overview

We are now ready to formally define a class of global curve simplification problems. When $D(\cdot,\cdot)$ denotes a distance measure between curves (e.g., the *Hausdorff* or *Fréchet* distance), the *global curve simplification (GCS)* problem asks what is the smallest number k such that there exists a curve P' with at most k vertices, chosen either as a subsequence of the vertices of P (variant $\mathcal{V}$), as a sequence of points on the edges of P in the correct order along P (variant $\mathcal{C}$), or chosen anywhere in $\mathbb{R}^d$ (variant $\mathcal{N}$) and such that $D(P,P') \leq \delta$, for a given threshold δ. In all cases, we require that P and P' start at the same point and end at the same point.

Table 1 summarizes results for the different variants of the GCS problem obtained by instantiating D with the Hausdorff or Fréchet distance measures and by applying a vertex restriction R. Here $R \in \{\mathcal{V},\mathcal{C},\mathcal{N}\}$, and D is either the undirected Hausdorff distance H, the directed Hausdorff distance $\overleftarrow{\mathsf{H}}(P,\delta)$ from P to P', the directed Hausdorff distance $\overrightarrow{\mathsf{H}}(P,\delta)$ from P' to P, the Fréchet distance F, the discrete Fréchet distance dF, or the weak Fréchet distance wF. Throughout the paper we use $D_R(P,\delta)$ to denote a curve P' that is the optimal R-restricted simplification of P with $D(P,P') \leq \delta$.

Since GCS is a dual-optimization problem, we call an algorithm an (α, β)-*approximation* if it computes a solution with distance at most $\beta\delta$ and uses at most α times more shortcuts than the optimal solution for distance δ.

2.4 New Results

In order to provide a thorough understanding of the different variants of the GCS problem we provide several new results. In some cases these are straightforward adaptations of known results, in other cases they require deeper ideas. Additional lemmas, theorems, and proofs are available in the full version of this paper [22]. We give polynomial time algorithms for finding $\mathsf{wF}_{\mathcal{V}}(P, \delta)$, the vertex-restricted GCS under the weak Fréchet distance (Section 3, Theorem 2), and $\mathsf{wF}_{\mathcal{V}}(P, \delta)$, the vertex-restricted GCS under the strong Fréchet distance (see [22]). In Section 4 we consider the vertex-restricted problem under the directed Hausdorff distance from P' to P (that is, to find $\overrightarrow{\mathsf{H}}_{\mathcal{V}}(P, \delta)$), originally considered by van Kreveld et al. [24], and we provide an algorithm with an improved runtime of $O(n^3 \log n)$ (Theorem 3). In Section 5 we prove that solving the curve-restricted GCS is NP-hard for almost all distance measures considered in this paper except for the discrete Fréchet distance (Theorem 6) and strong Fréchet distance in $\mathbb{R}^1$ (Theorem 7) for which we present polynomial time algorithms. To the best of our knowledge, these are the first results in the curve-restricted setting under global distance measures. Finally, in Section 8, we give a $(2, 1+\varepsilon)$-approximation algorithm for $\mathsf{F}_{\mathcal{N}}(P, \delta)$, the non-restricted GCS under the Fréchet distance, which runs in $O^*(n^2 \log n \log \log n)$ time, where O^* hides factors polynomial in $1/\varepsilon$ (Theorem 11). The same result also holds for $\mathsf{wF}_{\mathcal{N}}(P, \delta)$ (Corollary 12). In Section 9 we show that the non-restricted GCS problem becomes NP-hard when we consider the Hausdorff distance (Theorem 13).

2.5 Discussion

With both the existing work and our new results in place, we now have a good overview of the complexity of the different variants of the GCS problem, see Table 1.

Observe that the curve-restricted variants seem to generally be harder than both the vertex-restricted and the non-restricted variants. That means that, on the one hand, broadening the search space from the vertex-restricted to the curve-restricted case makes the problem harder. But on the other hand it does not give unrestricted freedom of choice, which in turn enables the development of efficient algorithms for the unrestricted case.

Another interesting pattern can be observed for the Hausdorff distance measures. The direction of the Hausdorff distance makes a significant difference in whether the corresponding GCS problem is NP-hard or polynomially solvable. The GCS problem for the undirected Hausdorff distance is at least as hard as for the directed Hausdorff distance from the input curve to the simplification.

Drawing upon the above observations we make the following conjecture:

▶ **Conjecture 1.** *The curve-restricted and non-restricted GCS problems for $\overleftarrow{\mathsf{H}}(P, \delta)$ are strongly NP-hard.*

3 Freespace-Based Algorithms for Fréchet Simplification

We use the free space diagram between P and its shortcut graph G to solve the vertex-restricted GCS problem under the weak and strong Fréchet distances in $O(n^3)$ time and space. This is related to *map-matching* [4, 9], however in our case we need to compute *shortest* paths in the free space that correspond to *simple* paths in G. While map-matching for closed simple paths is NP-complete [25], we exploit the DAG property of G to develop efficient algorithms. The proof for the strong Fréchet distance can be found in [22].

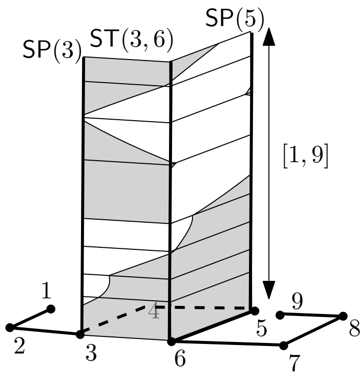

Figure 3 A free space diagram $\mathsf{FSD}_\delta(P, G)$ with strips and spines.

3.1 Shortcut DAG and Free Space Diagram

Let $G = (V, E)$ be the *shortcut DAG* of P, where $V = \{1, \dots, n\}$ and $E = \{(u, v) \mid 1 \leq u < v \leq n\}$. Each $v \in V$ is embedded at p_v and each edge $e = (u, v) \in E$ as a straight line shortcut is linearly parameterized as $e(t) = (1 - t)p_u + tp_v$ for $t \in [0, 1]$. We consider the parameter space of G to be $E \times [0, 1]$.

Now, let $\delta > 0$, and consider the joint parameter space $[1, n] \times E \times [0, 1]$ of P and G. Any $(s, e, t) \in [1, n] \times E \times [0, 1]$ is called *free* if $\|P(s) - e(t)\| \leq \delta$, and the union of all free points is referred to as the *free space*. For brevity, we write $(s, e(t))$ instead of (s, e, t), and if $e(t) = v \in V$ we write (s, v). The *free space diagram* $\mathsf{FSD}_\delta(P, G)$ consists of all points in $[1, n] \times E \times [0, 1]$ together with an annotation for each point whether it is free or not. In the special case that the graph is a polygonal curve Q with m vertices, then $\mathsf{FSD}_\delta(P, Q)$ consists of $(n - 1) \times (m - 1)$ cells in the domain $[1, n] \times [1, m]$. A monotone path from $(1, 1)$ to (n, m) that lies entirely within the free space corresponds to a pair of monotone reparameterizations (σ, θ) that witness $\mathsf{F}(P, Q) \leq \delta$. Such a *reachable path* can be computed using dynamic programming in $O(mn)$ time [5]. If one drops the monotonicity requirement for the path, one obtains a witness for $\mathsf{wF}(P, Q) \leq \delta$.

The free space diagram $\mathsf{FSD}_\delta(P, G)$ consists of one *cell* for each edge in P and each edge in G. The free space in such a cell is convex. The boundary of a cell comprises four line segments that each contain at most one *free space interval*. $\mathsf{FSD}_\delta(P, G)$ is composed of spines and strips. For any $v \in V$ and $e \in E$ we call $\mathsf{SP}(v) = [1, n] \times v$ a *spine* and $\mathsf{ST}(e) = [1, n] \times e \times [0, 1]$ a *strip*. We denote the free space within spines and strips as $\mathsf{SP}_\delta(v) = \{(s, v) \mid 1 \leq s \leq n,\ ||P(s) - p_v|| \leq \delta\}$ and $\mathsf{ST}_\delta(e) = \{(s, e(t)) \mid 1 \leq s \leq n,\ 0 \leq t \leq 1,\ ||P(s) - e(t)|| \leq \delta\}$. For $(u, v) \in E$, both spines centered at the vertices of the edge are subsets of the strip: $\mathsf{SP}(u), \mathsf{SP}(v) \subseteq \mathsf{ST}(u, v)$, and $\mathsf{SP}(u)$ is a subset of all strips with respect to edges incident on u. See Figure 3 for an illustration.

3.2 Weak Fréchet Distance $\mathsf{wF}_\mathcal{V}(P, \delta)$ in Polynomial Time

Let $P' = \mathsf{wF}_V(P, \delta)$ and let $n' = \#P'$ be the number of vertices in P'. Then P' is a path in G, and P' visits an increasing subsequence of vertices in P (or V). From the fact that $\mathsf{wF}(P, P') \leq \delta$ we know that there is a path $\mathcal{P} = (\sigma, \theta)$ from $(1, 1)$ to (n, n') in $\mathsf{FSD}_\delta(P, P')$ that lies entirely within free space. And since $\mathsf{FSD}_\delta(P, P')$ is a subset of $\mathsf{FSD}_\delta(P, G)$, the path $\mathcal{P} = (\sigma, \theta)$ is also a path in $\mathsf{FSD}_\delta(P, G)$. Here, σ is a reparameterization of P, and θ is a reparameterization of P', and P' is simple. We call (s, d) in $\mathsf{FSD}_\delta(P, G)$ *weakly reachable* if there exists a path $\mathcal{P} = (\sigma, \theta)$ from $(1, 1)$ to (s, d) in $\mathsf{FSD}_\delta(P, G)$ that lies in free space such that θ is a reparameterization of a simple path from p_1 to some point on an edge in G. We denote the number of vertices on this simple path by $\#\mathcal{P}$, and we call $\mathcal{P}$ *weakly reachable*. We define the cost function $\phi : [1, n] \times V \to \mathbb{N}$ as $\phi(z, v) = \min_{\mathcal{P}} \#\mathcal{P}$, where the minimum ranges over all weakly reachable paths to (z, v) in the free space diagram. If no such path exists then $\phi(z, v) = \infty$. Note that all points in a free space interval (on the boundary of a free space cell) have the same ϕ-value.

▶ **Observation 1.** *There is a weakly reachable path $\mathcal{P}$ in $\mathsf{FSD}_\delta(P, G)$ from $(1, 1)$ to (n, n) with $\#\mathcal{P} = \#\mathsf{wF}_\mathcal{V}(P, \delta)$ if and only if $\phi(n, n) = \#\mathsf{wF}_\mathcal{V}(P, \delta)$.*

Since ϕ is the length of a shortest path, it seems as if one could compute it by simply using a breadth-first propagation. However, one has to be careful because a weakly reachable path $\mathcal{P}$ is only allowed to backtrack *along the path in G that it has already traversed*. We therefore carefully combine two breadth-first propagations to compute the ϕ values for all $I \in \mathcal{I}$, where $\mathcal{I}$ is the set of all (non-empty) free space intervals on all spines $\mathsf{SP}(v)$ for all $v \in V$. For the primary breadth-first propagation, we initialize a queue Q by enqueuing the interval $I \subseteq \mathsf{SP}_\delta(1)$ that contains $(1, 1)$. Once an interval has been enqueued it is considered *visited*, and it can never become unvisited again. Then we repeatedly extract the next interval I from Q. Assume $I \subseteq \mathsf{SP}_\delta(u)$. For each v from $u+1$ to n we consider $\mathsf{ST}(u, v)$ and we compute all unvisited intervals $J \subseteq \mathsf{SP}_\delta(u) \cup \mathsf{SP}_\delta(v)$ that are reachable from I with a path in $\mathsf{ST}_\delta(u, v)$. These J can be reached using one more vertex, therefore we set $\phi(J) = \phi(I) + 1$, we insert J into Q, and we store the predecessor $\pi(J) = I$. For each $J \in \mathsf{SP}_\delta(u)$ we then launch a secondary breadth-first traversal to propagate $\phi(J)$ to all unvisited intervals $J' \in \mathcal{I}$ that are reachable from J within the free space of $\mathsf{FSD}_\delta(P, G(\pi(J)))$. Here, $G(\pi(J))$ denotes the projection of the predecessor DAG rooted at $\pi(J)$ onto G, i.e., each interval I in the predecessor DAG is projected to u if $I \subseteq \mathsf{SP}_\delta(u)$. This allows $\mathcal{P}$ to backtrack along the path in G that it has already traversed, *without* increasing ϕ. This secondary breadth-first traversal uses a separate queue Q', and sets $\phi(J') = \phi(J)$ and $\pi(J') = J$. When this secondary traversal is finished, Q' is prepended to Q, and then the primary breadth-first propagation continues. Once Q is empty, i.e., all intervals have been visited, $\phi(I) = \#\mathsf{wF}_\mathcal{V}(P, \delta)$, where $I \subseteq \mathsf{SP}_\delta(n)$ is the interval that contains (n, n). Backtracking a path $\mathcal{P}$ from n to 1 in the predecessor DAG rooted at $\pi(I)$, and projecting $\mathcal{P}$ onto G, yields the simplified curve P'. This algorithm visits each interval in $\mathcal{I}$ once using nested breadth-first traversals. Since there are $O(n^3)$ free space intervals this takes $O(n^3)$ time and space.

▶ **Theorem 2.** *Given a polygonal curve P with n vertices and $\delta > 0$, an optimal solution to the vertex-restricted GCS problem under the weak Fréchet distance can be computed in $O(n^3)$ time and space.*

4 Vertex-Restricted GCS under Directed Hausdorff from P' to P

In this section we revisit the GCS problem for $\overrightarrow{\mathsf{H}}_\mathcal{V}(P, \delta)$ considered by Kreveld et al. [24]. We improve on the running time of their $O(n^4)$ time algorithm. First we thicken the input curve P by width δ. This induces a polygon $\mathcal{P}$ with $h = O(n^2)$ holes. Now all we need is to decide whether each shortcut $\langle p_i p_j \rangle$ for all $1 \le i < j \le n$ lies entirely within $\mathcal{P}$ or not. To this end, we preprocess $\mathcal{P}$ into a data structure such that for any straight line query ray ρ originating from a point inside $\mathcal{P}$ one can efficiently compute the first point on the boundary of $\mathcal{P}$ hit by ρ. We use the data structure proposed by [13] of size $O(N)$ which can be constructed in time $O(N\sqrt{h} + h^{3/2} \log h + N \log N)$ and which answers queries in

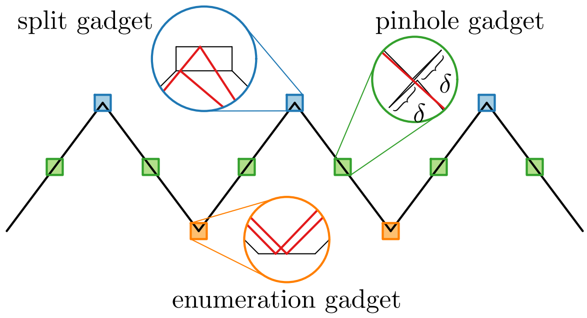

Figure 4 Sketch of the template curve.

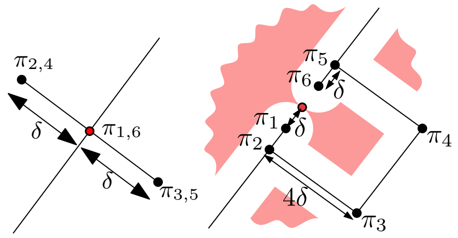

Figure 5 Pinhole gadgets π for $\mathsf{F}_{\mathcal{C}}(P, \delta)$ (left) and $\overleftarrow{\mathsf{H}}_{\mathcal{C}}(P, \delta)$ (right). The pinhole is shown in red.

$O(\sqrt{h} \log N)$ time. We have $\Theta(n^2)$ shortcuts $\langle p_i p_j \rangle$ to process and need to examine whether each shortcut lies inside $\mathcal{P}$ or not. If a shortcut lies inside $\mathcal{P}$ then we include it in the edge set of the shortcut graph proposed by Imai and Iri [20]. Otherwise we eliminate the shortcut. We originate a ray at p_i and compute the first point x on the boundary of $\mathcal{P}$ hit by the ray in $O(\sqrt{h} \log N)$ time. If $\|p_i - x\| \geq \|p_i - p_j\|$ then the shortcut lies inside $\mathcal{P}$, otherwise it does not. Once the edge set of the shortcut graph is constructed, we compute the shortest path in it. As a result we have the following theorem:

▶ **Theorem 3.** *Given a polygonal curve P with n vertices and $\delta > 0$, an optimal solution to the curve-restricted GCS problem for $\overrightarrow{\mathsf{H}}_{\mathcal{V}}(P, \delta)$ can be computed in $O(n^3 \log n)$ time using $O(n^2)$ space.*

5 NP-Hardness of Several Curve-Restricted Variants

In this section we construct a template that we use to prove NP-hardness of the curve-restricted GCS problems for most of the distance measures discussed in this paper. The template takes inspiration from the NP-hardness proofs of minimum-link path problems [23]. We believe that this template can be adapted to show hardness of other similar problems.

The template reduces from the Subset Sum problem. Given a set of m integers $A = \{a_1, a_2, \ldots, a_m\}$ and an integer M, we will construct an instance of the curve-restricted GCS problem such that there exists a subset $B \subset A$ with the total sum of its integers equal to M if and only if there exists a simplified polygonal curve with at most $2m + 1$ vertices.

The input curve P we construct has a zig-zag pattern. It has m *split gadgets* at every other bend of the pattern, $m + 1$ *enumeration gadgets* at the other bends, and $2m$ *pinhole gadgets* halfway through each zig-zag segment (refer to Figure 4).

The construction forces any optimal simplification P' to follow a zig-zag pattern with a vertex on each split and enumeration gadget and no other vertices. The pinhole gadget is named as such because any segment of P' that goes through it is forced to pass through a specific point, called the *pinhole*. This limits the placements of P''s vertices. The choice of where to place the vertex on each split gadget then corresponds to the choice of including or excluding a given integer in the subset B and the x-coordinate of the vertex on each enumeration gadget encodes the sum of integers in B up to that point. We ensure that the endpoint of P is reachable with at most $2m + 1$ vertices only if B sums to exactly M.

The split and enumeration gadgets always have the same shape, but the shape of the pinhole gadget depends on the distance measure. Pinhole gadgets must be chosen so that the following properties hold:

1. Any segment of P' starting before a pinhole gadget and ending after the pinhole gadget must pass through the pinhole gadget's *pinhole*.
2. It must be impossible to have a segment of P' traverse multiple pinhole gadgets at once.
3. Any segment of P' where the starting vertex u is on a split or enumeration gadget, the segment goes through a pinhole, and the ending vertex v is on the next enumeration or split gadget, must have distance $\leq \delta$ to $P[u, v]$.
4. P must be polynomial in size. Specifically, only a polynomial number of polyline segments can be used and all vertices must have rational coordinates.

In Figure 5 we show pinhole gadgets for Fréchet distance and directed Hausdorff distance directed from P' to P. The gadget for Fréchet distance also works for weak Fréchet distance, undirected Hausdorff distance and directed Hausdorff distance in the other direction. For these latter three distance measures, we note that the pinhole gadget here does not force P' to go through the pinhole but to pass close enough by it instead. This results in there being reachable intervals on the split and enumeration gadgets rather than reachable points. This leads to an expanded version of the first property:

1. The endpoint of any segment of P' starting before a pinhole gadget and ending after the pinhole gadget must have distance less than $\frac{0.5}{2m}$ to the endpoint of the segment with the same starting point that passes exactly through the pinhole and ends on the same segment of P.

If this property holds (as it does for the gadget in Figure 5 (left) under weak Fréchet and Hausdorff distance) the reachable intervals on the gadgets are so small they never overlap, so the reduction still holds. This leads to the following theorems:

▶ **Theorem 4.** *Given a curve distance measure, if there exists a pinhole gadget that can be inserted in the described template such that the listed properties hold, the curve-restricted GCS problem for that distance measure is NP-hard.*

▶ **Theorem 5.** *The GCS problem for* $\overleftarrow{\mathsf{H}}_{\mathcal{C}}(P, \delta), \overrightarrow{\mathsf{H}}_{\mathcal{C}}(P, \delta), \mathsf{H}_{\mathcal{C}}(P, \delta), \mathsf{F}_{\mathcal{C}}(P, \delta), \mathsf{wF}_{\mathcal{C}}(P, \delta)$ *is NP-hard.*

Theorem 4 implies this template may be used to prove curve-restricted simplification under other distance measures NP-hard as well in the future. Since the template reduces from Subset Sum it proves the above problems weakly to be NP-hard. For undirected Hausdorff distance, we also prove strong NP-hardness in Corollary 14.

6 Curve-Restricted GCS under Discrete Fréchet Distance

Next we present an $O(n^3)$-time algorithm for the GCS problem for $\mathsf{dF}_{\mathcal{C}}(P, \delta)$. Observe that, given an input curve P, there is only a discrete set of candidate points we need to consider for vertices of the output curve. Let $\mathcal{A}$ be the arrangement of n disks of radius δ centered on the vertices of P, and let $C = \langle c_1, \ldots, c_m \rangle$, with $m \in O(n^2)$, be the sequence of intersections between P and $\mathcal{A}$, in order along P. Observe that under the discrete Fréchet distance, if there exists a curve-restricted simplification $P' = \langle q_1, \ldots, q_k \rangle$ of P, then there exists a subsequence of C of length k which is a simplification of P.

Although the approach of Bereg et al. [7] to compute the minimal vertex-restricted simplification of $\mathcal{A}$ does not apply in our case, we can design a dynamic programming algorithm in a similar fashion. Define $K(i, j)$ to be the minimum value k such that there exists a subsequence $\langle c_1, \ldots, c_j \rangle$ of length k that has discrete Fréchet distance at most δ to the sequence $\langle p_1, \ldots, p_i \rangle$. We will design a dynamic program to calculate all nm values

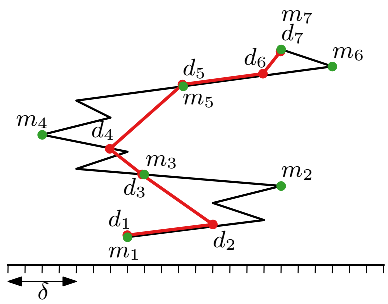

Figure 6 The time-stamped traversals made by the man m and the dog d. The red lines indicate the dog's jumps.

$K(i,j)$. Specifically, if p_{i-1} and c_j are within distance δ, then

$$K(i,j) = \min\left(K(i-1,j), \min_{1\le j'<j}(K(i-1,j')+1)\right),$$

and $K(i,j) = \infty$ otherwise. This definition immediately gives an $O(n^4)$ algorithm to compute $K(n,m)$. We can improve on this by maintaining a second table with prefix minima $M(i,j) = \min_{1\le j'\le j} K(i,j)$, which can be calculated in constant time per table entry and overall saves a linear factor. The full proof of the following theorem can be found in [22].

▶ **Theorem 6.** *Given a polygonal curve P with n vertices and $\delta > 0$, an optimal solution to the $\mathsf{dF}_{\mathcal{C}}(P,\delta)$ can be computed in $O(n^3)$ time and $O(n^2)$ space.*

7 GCS Fréchet Distance in One Dimension

In this section we provide a greedy algorithm for the curve-restricted GCS problem in $\mathbb{R}^1$ under the Fréchet distance. We describe our algorithm using the man-dog terminology that is often used in the literature on Fréchet distance: Initially a man and his dog start at p_1. The man walks along P until his distance to the dog exceeds δ. Now if there is a turn between the man and the dog, the dog marks its current position and jumps over the turn and stays at distance exactly δ away from the man. If there is no turn in between, the dog just follows the man at distance exactly δ and stops when the man arrives at the next turn or the end. Once they both end the walk at p_n we report the positions marked by the dog as P'. See Figure 6. More details are given in [22].

▶ **Theorem 7.** *Given a polygonal curve P in $\mathbb{R}^1$ with n vertices and $\delta > 0$, an optimal solution to the curve-restricted GCS problem under the Fréchet distance can be computed in linear time.*

8 Approximation of Non-Restricted GCS under Fréchet Distance

In this section we present an approximation algorithm for the non-restricted GCS problem that discretizes the feasible space for the vertices of the simplified curve. The idea is to compute a polynomial number of shortcuts in the discretized space, and (approximately) validate for each shortcut whether it is within Fréchet distance δ to a subcurve of P. For every subcurve of P we incrementally add the valid shortcuts to the edge set of a graph G until all the shortcuts have been processed. Once G is built, we compute the shortest path in G and

Algorithm 1 Non-restricted GCS problem for the Fréchet distance.

```
1 forall i ∈ {1,··· ,n} do Compute C_i(δ, (εδ/(4√d)) and G_i ;
2 E ← ∅, V ← ∅, C_1 ← p_1 ∪ C_1, C_n ← p_n ∪ C_n;
3 forall C_i and C_j, with 1 ≤ i ≤ j ≤ n do
4     forall c_1 ∈ C_i and c_2 ∈ C_j do
5         if VALIDATE(⟨c_1c_2⟩, P[i, j]) = true then E ← E ∪ ⟨c_1c_2⟩, V ← V ∪ {c_1, c_2};
6 return the shortest path between p_1 and p_n in G = (V, E).
```

return P'. To speed up the validation for each shortcut, we use a data structure to decide whether the Fréchet distance between a shortcut and a subcurve of P is at most δ. For a better understanding of our algorithm, we introduce some notation. Consider a ball $B(o, r)$ of radius $r > 0$ centered at $o \in \mathbb{R}^d$. Let $\mathsf{Prt}(\mathbb{R}^d, l)$ be a *partitioning* of $\mathbb{R}^d$ into a set of disjoint cells (hypercubes) of side length l that is induced by axis parallel hyperplanes placed consecutively at distance l. For any $1 \leq i \leq n$ we call $\mathcal{C}_i = \mathcal{C}_i(r, l) = \{c \in \mathsf{Prt}(\mathbb{R}^d, l) \mid c \cap B(p_i, r) \neq \emptyset\}$ a *discretization* of $B(p_i, r)$. Let $\mathcal{G}_i$ be the set of *corners* of all cells in $\mathcal{C}_i$.

As we can see, Algorithm 1 is a straightforward computation of valid shortcuts and shortest path in the graph G. The VALIDATE procedure takes a shortcut $\langle c_1c_2\rangle$ and a subcurve $P[i, j]$ as arguments and decides (approximately) if $\mathsf{F}(\langle c_1c_2\rangle, P[i, j]) \leq \delta$. In particular, it returns *true* if $\mathsf{F}(\langle c_1c_2\rangle, P[i, j]) \leq (1 + \varepsilon/2)\delta$ and *false* if $\mathsf{F}(\langle c_1c_2\rangle, P[i, j]) > (1 + \varepsilon)\delta$. We implement the VALIDATE procedure (line 5) using the data structure in [17]. Let $\#P'$ denote the number of vertices of the polygonal curve P'. The following lemmas imply Theorem 11. More details and proofs are provided in [22].

▶ **Lemma 8.** *The shortest path P'_{alg} returned by Algorithm 1 exists and $\mathsf{F}(P, P'_{\mathsf{alg}}) \leq (1 + \varepsilon)\delta$.*

▶ **Lemma 9.** *Let $P' = \mathsf{F}_{\mathcal{N}}(P, \delta)$ and let P'_{alg} be the curve returned by Algorithm 1. Then $\#P'_{\mathsf{alg}} \leq 2(\#P' - 1)$.*

▶ **Lemma 10.** *Algorithm 1 runs in $O\Big(\varepsilon^{-d} n \log n \big(\log^2(1/\varepsilon) \log n + \varepsilon^{-(d+2)} n \log\log n\big)\Big)$ time and uses $O\big((\varepsilon^{-d} \log^2(1/\varepsilon))n\big)$ space.*

▶ **Theorem 11.** *Let P be a polygonal curve with n vertices in $\mathbb{R}^d$, $\delta > 0$, and $P' = \mathsf{F}_{\mathcal{N}}(P, \delta)$. For any $0 < \varepsilon \leq 1$, one can compute in $O^*(n^2 \log n \log\log n)$ time and $O^*(n)$ space a non-restricted simplification P^* of P such that $\#P^* \leq 2(\#P' - 1)$ and $\mathsf{F}(P, P^*) \leq (1 + \varepsilon)\delta$. Here, O^* hides factors polynomial in $1/\varepsilon$.*

▶ **Corollary 12.** *Theorem 11 also holds for the non-restricted GCS under the weak Fréchet distance.*

9 Strong NP-Hardness for Non-Restricted GCS under Undirected Hausdorff Distance

Van Kreveld et al. [24] showed that the vertex-restricted GCS problem is NP-hard for undirected Hausdorff distance by a reduction from Hamiltonian cycle in segment intersection graphs. Their proof can be extended to the curve-restricted and non-restricted case; however, because of the increased freedom in vertex placement we must take care when exact embedding the segment graph: e.g., segments that intersect at arbitrarily small angles could potentially cause coordinates with unbounded bit complexity. For this reason, we here reduce from a more restricted class of graphs: *orthogonal* segment intersections graphs. Czyzowicz et al. [15] show that Hamiltonian cycle remains NP-complete in 2-connected cubic bipartite planar graphs, and Akiyama et al. [3] prove that every bipartite planar graph has a representation as an intersection graph of orthogonal line segments. Hence, Hamiltonian cycle in orthogonal segment intersection graphs is NP-complete.

We sketch the adapted proof; the full proof can be found in [22]. Let S be a set of n horizontal or vertical line segments in the plane with integer-coordinate endpoints such that $\bigcup S$ forms one connected component. Furthermore, assume that all intersections of segments in S are proper, that is no endpoints of segments in S coincide. Let the input polygonal curve P consist of the subsegments of S, and let P cover all the segments of S (possibly multiple times). That is, the vertices of P are chosen from the set of endpoints and the intersection points of segments in S, and the union of all the links of P equals to the union of the segments in S. Set $\delta = \frac{1}{8}$, and let $D \subset \mathbb{R}^2$ be the Minkowski sum of S and a closed ball of radius δ. A simplification P' with Hausdorff distance at most δ to P must visit the δ-disks around all endpoints of S, while staying inside D. A Hamiltonian path in the intersection graph of S corresponds to a simplification P' with $3n-1$ vertices. Indeed, since no two δ-disks around the endpoints of the segments in S are visible to each other within D (unless they are endpoints of the same segment), an optimal solution visits the two endpoints of each segment consecutively and has one extra bend to switch to the next segment. This results in three links of P' per segment, except for the first and the last segment to be covered, for which only two links each are needed.

▶ **Theorem 13.** *The non-restricted GCS problem under undirected Hausdorff distance is strongly NP-hard.*

Since a solution to the reduction never benefits from placing vertices not on P, we also immediately obtain an improvement over Theorem 5 for the case of $\mathsf{H}_{\mathcal{C}}(P, \delta)$.

▶ **Corollary 14.** *The curve-restricted GCS problem under undirected Hausdorff distance is strongly NP-hard.*

References

1 M. A. Abam, M. de Berg, P. Hachenberger, and A. Zarei. Streaming Algorithms for Line Simplification. *Discrete & Computational Geometry*, 43(3):497–515, 2010.

2 P. K. Agarwal, S. Har-Peled, N. Mustafa, and Y. Wang. Near Linear time approximation algorithm for curve simplification. *Algorithmica*, 42(3–4):203–219, 2005.

3 T. Akiyama, T. Nishizeki, and N. Saito. NP-completeness of the hamiltonian cycle problem for bipartite graphs. *Journal of Information Processing*, 3:73–76, 1980.

4 H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262—283, 2003.

5 H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5(1–2):75–91, 1995.

6 G. Barequet, D. Z. Chen, O. Daescu, M. T. Goodrich, and J. Snoeyink. Efficiently Approximating Polygonal Paths in Three and Higher Dimensions. *Algorithmica*, 33(2):150–167, 2002.

7 S. Bereg, M. Jiang, W. Wang, B. Yang, and B. Zhu. Simplifying 3D polygonal chains under the discrete Fréchet distance. In *Proc. Latin American Symposium on Theoretical Informatics (LATIN)*, Lecture Notes in Computer Science (LNCS, volume 4957), pages 630–641, 2008.

8 M. de Berg, M. van Kreveld, and S. Schirra. Topologically Correct Subdivision Simplification Using the Bandwidth Criterion. *Cartography and Geographic Information Systems*, 25(4):243–257, 1998.

9 S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On Map-Matching Vehicle Tracking Data. In *Proc. 31st Conference on Very Large Data Bases (VLDB)*, pages 853–864, 2005.

10 K. Bringmann and B.R. Chaudhury. Polyline Simplification has Cubic Complexity, 2018. `arXiv:1810.00621`.

11 L. Buzer. Optimal Simplification of Polygonal Chain for Rendering. In *Proc. 23rd Annual ACM Symposium on Computational Geometry (SoCG)*, pages 168–174, 2007.

12 S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *International Journal of Computational Geometry & Applications*, 6(1):59–77, 1996.

13 B. Chazelle, H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12(1):54–68, 1994.

14 D. Z. Chen, O. Daescu, J. Hershberger, P. M. Kogge, N. Mi, and J. Snoeyink. Polygonal path simplification with angle constraints. *Computational Geometry*, 32(3):173–187, 2005.

15 J. Czyzowicz, E. Kranakis, and J. Urrutia. A Simple Proof of the Representation of Bipartite Planar Graphs As the Contact Graphs of Orthogonal Straight Line Segments. *Information Processing Letters*, 66(3):125–126, 1998.

16 D. H. Douglas and T. K. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica*, 10(2):112–122, 1973.

17 A. Driemel and S. Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42:1830–1866, 2013.

18 M. Godau. A natural metric for curves – computing the distance for polygonal chains and approximation algorithms. In *Proc. 8th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science book series (LNCS, volume 480), pages 127–136, 1991.

19 L. Guibas, J. Hershberger, J. Mitchell, and J. Snoeyink. Approximating polygons and subdivisions with minimum-link paths. *International Journal of Computational Geometry & Applications*, 3(4):383–415, 1993.

20 H. Imai and M. Iri. An optimal algorithm for approximating a piecewise linear function. *Journal of Information Processing*, 9(3):159–162, 1986.

21 H. Imai and M. Iri. Polygonal Approximations of a Curve – Formulations and Algorithms. In Godfried T. Toussaint, editor, *Computational Morphology: A Computational Geometric Approach to the Analysis of Form.* North-Holland, Amsterdam, 1988.

22 M. van de Kerkhof, I. Kostitsyna, M. Löffler, M. Mirzanezhad, and C. Wenk. Global Curve Simplification, 2019. `arXiv:1809.10269`.

23 I. Kostitsyna, M. Löffler, V. Polishchuk, and F. Staals. On the complexity of Minimum-Link path problems. *Journal of Computational Geometry*, 8(2):80–108, 2017.

24 M. van Kreveld, M. Löffler, and L. Wiratma. On optimal polyline simplification using the Hausdorff and Fréchet distance. In *Proc. 34th International Symposium on Computational Geometry (SoCG)*, volume 56, pages 1–14, 2018.

25 W. Meulemans. Map Matching with Simplicity Constraints, 2013. `arXiv:1306.2827`.