Polyline Simplification has Cubic Complexity

Karl Bringmann
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
kbringma@mpi-inf.mpg.de

Bhaskar Ray Chaudhury
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
Graduate School of Computer Science Saarbrücken, Saarland Informatics Campus, Germany
braycha@mpi-inf.mpg.de

Abstract

In the classic polyline simplification problem we want to replace a given polygonal curve P, consisting of n vertices, by a subsequence P' of k vertices from P such that the polygonal curves P and P' are “close”. Closeness is usually measured using the Hausdorff or Fréchet distance. These distance measures can be applied *globally*, i.e., to the whole curves P and P', or *locally*, i.e., to each simplified subcurve and the line segment that it was replaced with separately (and then taking the maximum). We provide an $\mathcal{O}(n^3)$ time algorithm for simplification under Global-Fréchet distance, improving the previous best algorithm by a factor of $\Omega(kn^2)$. We also provide evidence that in high dimensions cubic time is essentially optimal for all three problems (Local-Hausdorff, Local-Fréchet, and Global-Fréchet). Specifically, improving the cubic time to $O(n^{3-\epsilon}\text{poly}(d))$ for polyline simplification over $(\mathbb{R}^d, L_p)$ for $p = 1$ would violate plausible conjectures. We obtain similar results for all $p \in [1, \infty), p \neq 2$. In total, in high dimensions and over general L_p-norms we resolve the complexity of polyline simplification with respect to Local-Hausdorff, Local-Fréchet, and Global-Fréchet, by providing new algorithms and conditional lower bounds.

2012 ACM Subject Classification Theory of computation; Theory of computation → Design and analysis of algorithms; Theory of computation → Mathematical optimization

Keywords and phrases Polyline simplification, Fréchet distance, Hausdorff distance, Conditional lower bounds

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.18

Related Version A full version of the paper is available at `https://arxiv.org/abs/1810.00621`.

1 Introduction

We revisit the classic problem of polygonal line simplification, which is fundamental to computational geometry. The most frequently implemented algorithms for curve simplification go back to the 70s (Douglas and Peucker [11]) and 80s (Imai and Iri [18]). A *polyline* is given by a sequence $P = \langle v_0, v_1, \dots, v_n \rangle$ of points $v_i \in \mathbb{R}^d$, and represents the continuous curve walking along the line segments $\overline{v_i v_{i+1}}$ in order. Given such a polyline P and a number $\delta > 0$, we want to compute $P' = \langle v_{i_0}, \dots, v_{i_{k-1}} \rangle$, with $0 = i_0 < \dots < i_{k-1} = n$, of minimal length k such that P and P' have “distance” at most δ.

Several distance measures have been used for the curve simplification problem. The most generic distance measure on *point sets* A, B is the *Hausdorff distance*. However, the most popular distance measure for curves in computational geometry is the *Fréchet distance* δ_F. In comparison to Hausdorff distance, it takes the ordering of the vertices along the curves into account, and thus better captures an intuitive notion of distance among curves.

For both of these distance measures $\delta_* \in \{\delta_H, \delta_F\}$, we can apply them *locally* or *globally* to measure the distance between the original curve P and its simplification P'. In the global variant, we consider the distance $\delta_*(P, P')$, i.e., we use the Hausdorff or Fréchet distance of P and P'. In the local variant, we consider the distance $\max_{1 \le \ell < k} \delta_*(P[i_{\ell-1} \dots i_\ell], \overline{v_{i_{\ell-1}} v_{i_\ell}})$, i.e., for each simplified subcurve $P[i_{\ell-1} \dots i_\ell]$ of P we compute the distance to the line segment $\overline{v_{i_{\ell-1}} v_{i_\ell}}$ that we simplified the subcurve to, and we take the maximum over these distances. This gives rise to four problem variants, depending on the distance measure: Local-Hausdorff, Local-Fréchet, Global-Hausdorff, and Global-Fréchet.

Among the variants, Global-Hausdorff is unreasonable as it does not take the ordering of vertices of the curve into account and it was recently shown that curve simplification under Global-Hausdorff is NP-hard [20]. Hence, we do not consider this measure in this paper.

The classic $\hat{\mathcal{O}}(n^3)$[1] time algorithm by Imai and Iri [18] was designed for Local-Hausdorff simplification. By changing the distance computation in this algorithm for the Fréchet distance, one can obtain an $\hat{\mathcal{O}}(n^3)$-time algorithm for Local-Fréchet [15]. There are improvements for Local-Hausdorff simplification in small dimension d [19, 8, 6]; the fastest running times are $2^{O(d)} n^2$ for L_1-norm, $\hat{\mathcal{O}}(n^2)$ for L_∞-norm, and $\hat{\mathcal{O}}(n^{3-\Omega(1/d)})$ for L_2-norm [6].

The remaining variant, Global-Fréchet, has only been studied very recently [20], although it is a reasonable measure: The Local constraints (i.e., matching each v_{i_ℓ} to itself) are not necessary to enforce ordering along the curve, since Fréchet distance already takes the ordering of the vertices into account – in contrast to Hausdorff distance. Van Kreveld et al. [20] presented an $\hat{\mathcal{O}}(k^* \cdot n^5)$ time algorithm for Global-Fréchet simplification where k^* is the size of the optimal simplification.

1.1 Contribution 1: Algorithm for Global-Fréchet simplification

One could get the impression that Global-Fréchet simplification is a well-motivated, but computationally expensive curve simplification problem, in comparison to the local variants. We show that the latter intuition is wrong, by designing an $\hat{\mathcal{O}}(n^3)$-time algorithm for Global-Fréchet simplification improving the previously best $\hat{\mathcal{O}}(k^* \cdot n^5)$-time algorithm [20].

▶ **Theorem 1** (Section 3). *Global-Fréchet simplification can be solved in time $\hat{\mathcal{O}}(n^3)$.*

This shows that all three problem variants (Local-Hausdorff, Local-Fréchet, and Global-Fréchet) can be solved in time $\hat{\mathcal{O}}(n^3)$, and thus the choice of which problem variant to apply should not be made for computational reasons, at least in high dimensions.

1.2 Contribution 2: Conditional lower bound

Since all three variants can be solved in time $\hat{\mathcal{O}}(n^3)$, the question arises whether any of them can be solved in time $\hat{\mathcal{O}}(n^{3-\varepsilon})$. Tools to (conditionally) rule out such algorithms have been developed in recent years in the area of *fine-grained complexity*, see, e.g., the survey [21]. One of the most widely used fine-grained hypotheses is the following.

***k*-OV Hypothesis.** *Problem:* Given sets $A_1, \dots, A_k \subseteq \{0,1\}^d$ of size n, determine whether there exist vectors $a_1 \in A_1, \dots, a_k \in A_k$ that are orthogonal, i.e., for each dimension $j \in [d]$ there is an $i \in [k]$ with $a_i[j] = 0$.
Hypothesis: For any $k \geq 2$ and $\varepsilon > 0$ the problem is not in time $\hat{\mathcal{O}}(n^{k-\varepsilon})$.

Naively, k-OV can be solved in time $\hat{\mathcal{O}}(n^k)$, and the hypothesis asserts that no polynomial improvement is possible, at least not with polynomial dependence on d. Buchin et al. [7] used the 2-OV hypothesis to rule out $\hat{\mathcal{O}}(n^{2-\varepsilon})$-time algorithms for Local-Hausdorff[2] in the

[1] In $\hat{\mathcal{O}}$-notation we *hide any polynomial factors in d*, but we make exponential factors in d explicit.
[2] Their proof can be adapted to also work for Local-Fréchet and Global-Fréchet.

L_1, L_2, and L_∞ norm. This yields a tight bound (in $\hat{\mathcal{O}}$ sense) for L_∞, since an $\hat{\mathcal{O}}(n^2)$-time algorithm is known [6]. However, for all other L_p-norms ($p \in [1, \infty)$), the question remained open whether $\hat{\mathcal{O}}(n^{3-\varepsilon})$-time algorithms exist. To answer this question, one could try to generalize the conditional lower bound by Buchin et al. [7] to start from 3-OV. However, curve simplification problems seem to have the wrong "quantifier structure" for such a reduction (we will make this clearer when we give a brief overview of the reduction.) For similar reasons, Abboud et al. [2] introduced the Hitting Set Hypothesis, in which they essentially consider a variant of 2-OV where we have a universal quantifier over the first set of vectors and an existential quantifier over the second one (∀∃-OV). From their hypothesis, however, it is not known how to prove higher lower bounds than quadratic. We therefore consider the following natural extension of their hypothesis. This problem was studied in a more general context by Gao et al. [14].

∀∀∃-OV Hypothesis. *Problem:* Given sets $A, B, C \subseteq \{0,1\}^d$ of size n, determine whether for all $a \in A, b \in B$ there is $c \in C$ such that a, b, c, are orthogonal, i.e., $\sum_{\ell \in [d]} a[\ell]b[\ell]c[\ell] = 0$.
Hypothesis: For any $\varepsilon > 0$ the problem is not in time $\hat{\mathcal{O}}(n^{3-\varepsilon})$.

No algorithm violating this hypothesis is known, and even for much stronger hypotheses on variants of k-OV and Satisfiability no such algorithms are known, see Section 5 in the full version for details. This shows that the hypothesis is plausible, in addition to being a natural generalization of the hypothesis of Abboud et al. [2].

We now give an brief overview of an ∀∀∃-OV-based lower bound for curve simplification. Given a ∀∀∃-OV instance on vectors $A, B, C \subseteq \{0,1\}^d$, we construct corresponding point sets $\tilde{A}, \tilde{B} \subset \mathbb{R}^{d'}$ (for some $d' = O(d)$), forming two clusters that are very far apart from each other. We add a start- and an endpoint, which can be chosen far away from these clusters (in a new direction). Near the midpoint between $\tilde{A}$ and $\tilde{B}$, another set of points $\tilde{C}$ is constructed. The final curve then starts in the startpoint, walks through all points in $\tilde{A}$, then through all points in $\tilde{C}$, then through all points in $\tilde{B}$, and ends in the endpoint. Choosing a size-4 simplification implements an existential quantifier over $a \in A, b \in B$. The constraints that all $\tilde{c} \in \tilde{C}$ are close to the line segment from $\tilde{a}$ to $\tilde{b}$ implements a universal quantifier over $c \in C$. Naturally, we want the distance from $\tilde{c}$ to the line segment $\overline{\tilde{a}\tilde{b}}$ to be large if a, b, c are orthogonal, and to be small otherwise. This simulates the negation of ∀∀∃-OV, so any curve simplification algorithm can be turned into an algorithm for ∀∀∃-OV.

▶ **Theorem 2** (Section 4). *Over $(\mathbb{R}^d, L_p)$ for any $p \in [1, \infty)$ with $p \neq 2$, Local-Hausdorff, Local-Fréchet, and Global-Fréchet simplification have no $\hat{\mathcal{O}}(n^{3-\varepsilon})$-time algorithm for any $\varepsilon > 0$, unless the* ∀∀∃-OV *Hypothesis fails. This holds even for the problem of deciding whether the optimal simplification has size ≤ 4 or ≥ 5.*

In particular, this rules out improving the $2^{O(d)}n^2$-time algorithm for Local-Hausdorff over L_1 [6] to a polynomial dependence on d. Note that the theorem statement excludes two interesting values for p, namely ∞ and 2. For $p = \infty$, an $\hat{\mathcal{O}}(n^2)$-time algorithm is known for Local-Hausdorff [6], so proving the above theorem also for $p = \infty$ would immediately yield an algorithm breaking the ∀∀∃-OV Hypothesis.

For $p = 2$, we do not have such a strong reason why it is excluded, however, we now argue that at least a significantly different proof would be necessary in this case. We want the points $\tilde{C}$ to lie in the middle between $\tilde{A}$ and $\tilde{B}$, which essentially means that we want to consider the distance from $(\tilde{a} + \tilde{b})/2$ to $\tilde{c}$. Now consider just a single dimension of ∀∀∃-OV. Then our task boils down to constructing points a_0, a_1 and b_0, b_1 and c_0, c_1, corresponding to the bits in this dimension, such that $\|(a_i + b_j)/2 - c_k\|_p = \beta_1$ if $i = j = k = 1$ and β_0 otherwise, with $\beta_1 < \beta_0$. Writing $a'_i = a_i/2$ and $b'_j = b_j/2$ for simplicity, in the case $p = 2$ we can simplify

$$\begin{aligned}
&\|a'_i + b'_j - c_k\|_2^2 = \sum_{\ell=1}^{d'} (a'_i[\ell] + b'_j[\ell] - c_k[\ell])^2 \\
&= \sum_{\ell=1}^{d'} \Big((a'_i[\ell] + b'_j[\ell])^2 + (a'_i[\ell] - c_k[\ell])^2 + (b'_j[\ell] - c_k[\ell])^2 - a'_i[\ell]^2 - b'_j[\ell]^2 - c_k[\ell]^2 \Big) \\
&= \|a'_i + b'_j\|_2^2 + \|a'_i - c_k\|_2^2 + \|b'_j - c_k\|_2^2 - \|a'_i\|_2^2 - \|b'_j\|_2^2 - \|c_k\|_2^2 \\
&= f_1(i,j) + f_2(j,k) + f_3(i,k),
\end{aligned}$$

for some functions$f_1, f_2, f_3 \colon \{0,1\} \times \{0,1\} \to \mathbb{R}$. Note that by assumption this is equal to β_1^2 if $i = j = k = 1$ and β_0^2 otherwise, with $\beta_1 < \beta_0$. However, it can be checked that such functions do not exist[3]. Therefore, for $p = 2$ *our outlined reduction cannot work - provably!* We nevertheless make this reduction work for $p \in [1, \infty)$, $p \neq 2$. The above argument shows that the construction is necessarily subtle. Indeed, constructing the right points requires some technical effort, see Section 4. This leaves open the possibility of a faster curve simplification algorithm for L_2, but such a result would need to exploit the Euclidean norm very heavily.

1.3 Further related work

Curve simplification has been studied in a variety of different formulations and settings. To list some examples, it was shown that the algorithm by Douglas and Peucker [11] can be implemented in time $O(n \log n)$ [17], and that the classic $O(n^3)$-time algorithm for Local-Hausdorff simplification by Imai and Iri [18] can be implemented in time $O(n^2)$ in two dimensions [8, 19]. More topics include curve simplification without self-intersections [10], Local-Hausdorff simplification with angular constraints between consecutive line segments [9], approximation algorithms [3], streaming algorithms [1], and the use of curve simplification in subdivision algorithms [16, 12, 13].

1.4 Organization

In Section 2 we formally define the problems studied in this paper. In Section 3 we present our new algorithm for Global-Fréchet, and in Section 4 we show our conditional lower bounds.

2 Preliminaries

Our ambient space is the metric space $(\mathbb{R}^d, L_p)$, where the distance between points $x, y \in \mathbb{R}^d$ is the L_p-norm of their difference, i.e., $\|x - y\|_p = \big(\sum_{i=1}^d (x[i] - y[i])^p\big)^{1/p}$. A *polyline* P of *size* $n+1$, given by a sequence of points $\langle v_0, v_1, \ldots, v_n \rangle$ is the continuous curve that starts in v_0, walks along the line segments $\overline{v_i v_{i+1}}$ for $i = 0, \ldots, n-1$ in order, and ends in v_n. We also interpret P as a function $P \colon [0, n] \to \mathbb{R}^d$ where $P[i + \lambda] = (1 - \lambda)v_i + \lambda v_{i+1}$ for any $\lambda \in [0, 1]$ and $i \in \{0, \ldots, n-1\}$. We use the notation $P[t_1 \ldots t_2]$ to represent the sub-polyline of P between $P[t_1]$ and $P[t_2]$. Formally for integers $0 \le i \le j \le n$ and reals $\lambda_1, \lambda_2 \in [0, 1)$,

$$P[i + \lambda_1 \ldots j + \lambda_2] = \langle (1 - \lambda_1)v_i + \lambda_1 v_{i+1}, v_{i+1}, \ldots, v_j, (1 - \lambda_2)v_j + \lambda_2 v_{j+1} \rangle$$

[3] We can express this situation by a linear system of equations in 12 variables (4 image values for each function f_i) and 8 equations (for the values of f on $i, j, k \in \{0, 1\}$) and verify that it has no solution.

A *simplification* of P is a curve $Q = \langle v_{i_0}, v_{i_1} \ldots, v_{i_m} \rangle$ with $0 = i_0 < i_1 < \ldots < i_m = n$. The size of the simplification Q is $m+1$. Our goal is to determine a simplification of minimal size that "very closely" represents P. To this end we define two popular measures of similarity between the curves, namely the Fréchet and Hausdorff distances.

▶ **Definition 3** (Fréchet distance). *The (continuous) Fréchet distance $\delta_F(P_1, P_2)$ between two curves P_1 and P_2 of size n and m respectively is*

$$\delta_F(P_1, P_2) = \inf_f \max_{t \in [0,n]} \|P_1[t] - P_2[f(t)]\|_p$$

where $f\colon [0, n] \to [0, m]$ is monotone with $f(0) = 0$ and $f(n) = m$.

Alt and Godau [5] characterize the Fréchet distance in terms of the "free-space diagram".

▶ **Definition 4** (Free-Space). *Given two curves P_1, P_2 and $\delta \geq 0$, the free-space $FS_\delta(P_1, P_2) \subseteq \mathbb{R}^2$ is the set $\{(x, y) \in ([0, n] \times [0, m]) \mid \|P_1[x] - P_2[y]\|_p \leq \delta\}$.*

Consider the following decision problem. Given two curves P_1, P_2 of size n and m, respectively, and given $\delta \geq 0$, decide whether $\delta_F(P_1, P_2) \leq \delta$. The answer to this question is yes if and only if (n, m) is reachable from $(0, 0)$ by a monotone path through $FS_\delta(P_1, P_2)$. This "reachability" problem is known to be solvable by a dynamic programming algorithm in time $\mathcal{O}(nm)$. In particular, if either P_1 or P_2 is a line segment, then the decision problem can be solved in linear time.

The Hausdorff distance between curves ignores the ordering of the points along the curve. Intuitively, if we remove the monotonicity condition from function f in Definition 3 we obtain the directed Hausdorff distance between the curves. Formally, it is defined as follows.

▶ **Definition 5** (Hausdorff distance). *The (directed)* Hausdorff *distance $\delta_H(P_1, P_2)$ between curves P_1 and P_2 of size n and m, respectively, is*

$$\delta_H(P_1, P_2) = \max_{t_1 \in [0,n]} \min_{t_2 \in [0,m]} \|P_1[t_1] - P_2[t_2]\|_p$$

In order to measure the "closeness" between a curve and its simplification, these above similarity measures can be applied either *globally* to the whole curve and its simplification, or *locally* to each simplified subcurve $P[i_\ell \ldots i_{\ell+1}]$ and the segment $\overline{v_{i_\ell}, v_{i_{\ell+1}}}$ to which it was simplified (taking the maximum over all ℓ). This gives rise to the following measures for curve simplification.

▶ **Definition 6** (Similarity for Curve Simplification). *Given a curve $P = \langle v_0, v_1, \ldots, v_n \rangle$ and a simplification $Q = \langle v_{i_0}, v_{i_1} \ldots, v_{i_m} \rangle$ of P, we define their Global-Hausdorff distance as $\delta_H(P, Q)$ and their Local-Hausdorff distance[4] as $\max_{0 \leq \ell \leq m-1} \delta_H(P[i_\ell \ldots i_{\ell+1}], \overline{v_{i_\ell} v_{i_{\ell+1}}})$. Their Global-Fréchet and Local-Fréchet distance are defined similarly, with δ_H replaced by δ_F.*

3 Algorithms for Global-Fréchet simplification

In this section we present an $\mathcal{O}(n^3)$ time algorithm for curve simplification under Global-Fréchet distance, i.e., we prove Theorem 1.

[4] It can be checked that in this expression directed and undirected Hausdorff distance have the same value, and so for Local-Hausdorff we can without loss of generality use the directed Hausdorff distance.

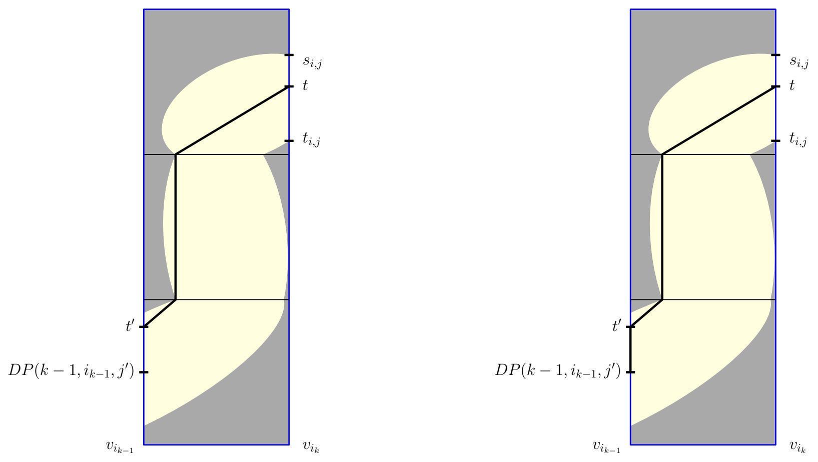

Figure 1 There is a monotone path from $(0, t')$ to $(1, t)$ in $FS_\delta(P, \overline{v_{i_{k-1}} v_{i_k}})$ (left). Since $\mathrm{DP}(k-1, i_{k-1}, j') \le t' \le t$, there is a monotone path in from $(0, \mathrm{DP}(k-1, i_{k-1}, j'))$ (right) to $(1, t)$ by moving from $(0, \mathrm{DP}(k-1, i_{k-1}, j'))$ to $(0, t')$ and then following the previous monotone path.

3.1 An $O(kn^5)$ algorithm for Global Fréchet simplification

We start by describing the previously best algorithm by [20]. Let P be the polyline $\langle v_0, v_1, \dots v_n \rangle$. Let $\mathrm{DP}(k, i, j)$ represent the earliest reachable point on $\overline{v_j v_{j+1}}$ with a length k simplification of the polyline $P[0 \dots i]$, i.e, $\mathrm{DP}(k, i, j)$ represents the smallest t such that $P[t]$ lies on the line-segment $\overline{v_j v_{j+1}}$ (i.e. $j \le t \le j+1$) and there is a simplification $\tilde{Q}$ of the polyline $P[0 \dots i]$ of size at most k such that $\delta_F(\tilde{Q}, P[0 \dots t]) \le \delta$. If such a point does not exist then we set $\mathrm{DP}(k, i, j) = \infty$. To solve Global-Fréchet simplification, we need to return the minimum k such that $\mathrm{DP}(k, n, n-1) \ne \infty$. Let $P[t_{i,j}]$ and $P[s_{i,j}]$ be the first point and the last point respectively on the line segment $\overline{v_j v_{j+1}}$ such that $\|v_i - P[t_{i,j}]\|_p \le \delta$ and $\|v_i - P[s_{i,j}]\|_p \le \delta$. Observe that if $\mathrm{DP}(k, i, j) \ne \infty$ then $t_{i,j} \le \mathrm{DP}(k, i, j) \le s_{i,j}$ for all k. We will crucially make use of the following characterization of the DP table entries.

▶ **Lemma 7.** $\mathrm{DP}(k, i, j)$ *is the minimal* $t \in [t_{i,j}, s_{i,j}]$*, such that for some* $i' < i$ *and* $j' \le j$*, we have* $\mathrm{DP}(k-1, i', j') \ne \infty$ *and* $\delta_F(P[\mathrm{DP}(k-1, i', j') \dots t], \overline{v_{i'} v_i}) \le \delta$*. If no such* t *exists then* $\mathrm{DP}(k, i, j) = \infty$*.*

Proof Sketch. $\mathrm{DP}(k, i, j)$ is the minimal $t \in [t_{i,j}, s_{i,j}]$ such that for some $i' < i$ and $t' \le t$ we have a simplification $\hat{Q}$ of the polyline $P[0 \dots i']$ and $\delta_F(\hat{Q}, P[0 \dots t']) \le \delta$ and $\delta_F(P[t' \dots t], \overline{v_{i'} v_i}) \le \delta$, and $\mathrm{DP}(k, i, j) = \infty$ if no such t exists. Let $j' \le t' \le j' + 1$ and $i' < i$, then $\mathrm{DP}(k-1, i', j') \ne \infty$ and by inspecting $FS_\delta(P, \overline{v_{i'} v_i})$, it is clear that there also exists a monotone path from $(0, \mathrm{DP}(k-1, i', j'))$ to $(1, t)$ (see Figure 1). Thus it suffices to consider only $\mathrm{DP}(k-1, i', j')$ as candidates for t'. ◀

A dynamic programming algorithm follows more or less directly from Lemma 7. Note that for fixed $i' < i$ and $j' \le j$ such that $\mathrm{DP}(k-1, i', j') \ne \infty$ we can determine the minimal t such that $(1, t)$ is reachable from $(0, \mathrm{DP}(k-1, i', j'))$ by a monotone path in $FS_\delta(P, \overline{v_{i'} v_i})$ in $\mathcal{O}(n)$ time. This follows from the standard algorithm for the decision version of the Fréchet distance between two polygonal curves of length at most n (in particular here one of the curves is of length 1). To determine $\mathrm{DP}(k, i, j)$ we enumerate over all $i' < i$ and $j' \le j$ such that $\mathrm{DP}(k-1, i', j') \ne \infty$ and determine the minimum t that is reachable. The running time to determine $\mathrm{DP}(k, i, j)$ is thus $\mathcal{O}(n^3)$. As there are $\mathcal{O}(k^* n^2)$ DP-cells to fill, the algorithm runs in time $\mathcal{O}(k^* n^5)$ where k^* is the size of the optimal simplification.

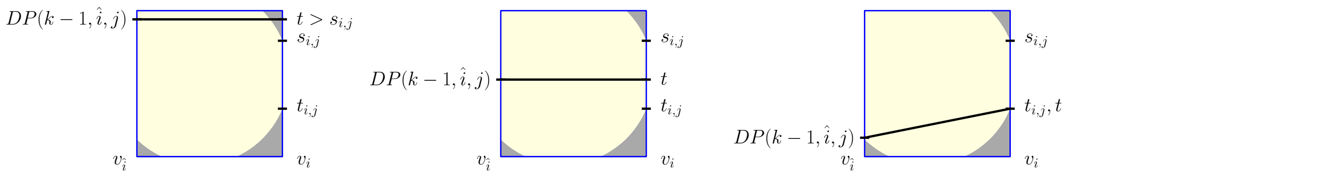

Figure 2 For $\hat{i} < i$, $t \in [t_{i,j}, s_{i,j}]$ is minimal such that $(1,t)$ is reachable from $(0, \mathrm{DP}(k-1,\hat{i},j))$ by a monotone path in fbox_j. If $\mathrm{DP}(k-1,\hat{i},j) > s_{i,j}$ (left) then no such t exists. If $t_{i,j} \leq \mathrm{DP}(k-1,\hat{i},j) \leq s_{i,j}$ (middle) then $t = \mathrm{DP}(k-1,\hat{i},j)$. If $\mathrm{DP}(k-1,\hat{i},j) < t_{i,j}$ (right) then $t = t_{i,j}$.

3.2 An $\mathcal{O}(n^3)$ algorithm for Global-Fréchet simplification

Now we improve the running time by a more careful understanding of the monotone paths through $FS_\delta(P, \overline{v_{i'}v_i})$ to $(1, \mathrm{DP}(k,i,j))$ for fixed i,j and i'. Let fbox_j denote the intersection of the free-space $FS_\delta(P, \overline{v_{i'}v_i})$ with the square with corner vertices $(0,j)$ and $(1,j+1)$. The following fact will be useful later.

▶ **Fact 8.** *fbox_j is convex for all $j \in [n-1]$.*

Let ver_j be the free space on the vertical line segment with endpoints $(0,j)$ and $(0,j+1)$ and let hor_j be the free space on the horizontal line segment $(0,j)$ to $(1,j)$ in the free space $FS_\delta(P, \overline{v_{i'}v_i})$. We consider the point $(0,j)$ to belong to ver_j, but not hor_j, to avoid certain corner cases. We split the monotone paths from $(0, \mathrm{DP}(k-1,i',j'))$ for $i' < i$ and $j' \leq j$ to $(1, \mathrm{DP}(k,i,j))$ in $FS_\delta(P, \overline{v_{i'}v_i})$ into two categories: the ones that intersect ver_j and the ones that intersect hor_j. We first look at the monotone paths that intersect ver_j. If the monotone path intersects ver_j then $j' = j$. Let $\overline{\mathrm{DP}}_1(k,i,j) = \min_{i'<i} \mathrm{DP}(k-1,i',j)$. We define

$$\mathrm{DP}_1(k,i,j) = \begin{cases} \max(\overline{\mathrm{DP}}_1(k,i,j), t_{i,j}) & \text{if } \overline{\mathrm{DP}}_1(k,i,j) \leq s_{i,j} \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

We show a characterization of DP_1 similar to the characterization of DP in Lemma 7 and thus establish that DP_1 correctly handles all paths intersecting ver_j.

▶ **Observation 9.** *$\mathrm{DP}_1(k,i,j)$ is the minimal $t \in [t_{i,j}, s_{i,j}]$ such that $\mathrm{DP}(k-1,i',j) \neq \infty$ and $\delta_F(P[\mathrm{DP}(k-1,i',j)\dots t], \overline{v_{i'}v_i}) \leq \delta$ for some $i' < i$. If no such t exists then $\mathrm{DP}_1(k,i,j) = \infty$.*

Proof Sketch. Observe that for any $\hat{i} < i$, the minimal $t \in [t_{i,j}, s_{i,j}]$ such that there is a monotone path from $(0, \mathrm{DP}(k-1,\hat{i},j))$ to $(1,t)$ in $FS_\delta(P, \overline{v_{\hat{i}}v_i})$ or equivalently in fbox_j is $\max(\mathrm{DP}(k-1,\hat{i},j), t_{i,j})$ if $\mathrm{DP}(k-1,\hat{i},j) \leq s_{i,j}$ (see Figure 2 middle and right) and t does not exist when $\mathrm{DP}(k-1,\hat{i},j) > s_{i,j}$ (see Figure 2 left). Therefore, if $\mathrm{DP}_1(k,i,j) = \min_{i'<i} \mathrm{DP}(k-1,i',j) \leq s_{i,j}$ then the minimal $t \in [t_{i,j}, s_{i,j}]$ that is reachable from $(0, \mathrm{DP}(k-1,i',j)$ for $i' < i$ is $\max(\min_{i'<i}(\mathrm{DP}(k-1,i',j), t_{i,j}) = \max(\mathrm{DP}_1(k,i,j), t_{i,j})$. Otherwise (if $\mathrm{DP}_1(k,i,j) = \min_{i'<i} \mathrm{DP}(k-1,i',j) > s_{i,j}$) then no such t exists. ◀

We now look at the monotone paths that intersect hor_j. Observe that if the monotone path intersects hor_j then $j' < j$. Along this line, we define $\overline{\mathrm{DP}}_2(k,i,j) = 1$ if there exists some $i' < i$ and $j' < j$, such that $\mathrm{DP}(k-1,i',j') \neq \infty$ and there exists a monotone path from $(0, \mathrm{DP}(k-1,i',j'))$ to $(1, t_{i,j})$ in the free-space $FS_\delta(P, \overline{v_{i'}v_i})$, and otherwise we set $\overline{\mathrm{DP}}_2(k,i,j) = 0$. Using this, we define

$$\mathrm{DP}_2(k,i,j) = \begin{cases} t_{i,j} & \text{if } \overline{\mathrm{DP}}_2(k,i,j) = 1 \\ \infty & \text{otherwise} \end{cases}$$

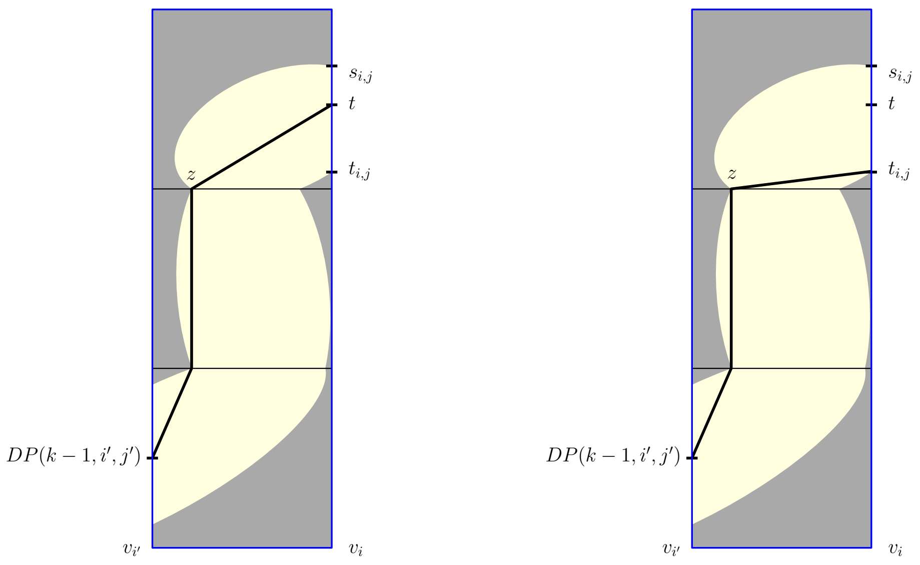

Figure 3 For $t \geq t_{i,j}$, there is a monotone path from $(0, \mathrm{DP}(k-1,i',j'))$ to $(1,t)$ in $FS_\delta(P, \overline{v_{i'}v_i})$ (left) for $i' < i$ and $j' < j$ that intersect hor_j at z. Then there is also a monotone path from $(0, \mathrm{DP}(k-1,i',j'))$ to $(1,t_{i,j})$ (right) following the previous monotone path upto z and then from z to $(1,t_{i,j})$.

We show a characterization of DP_2 similar our characterization of DP in Lemma 7, thus establishing that DP_2 correctly handles all paths intersecting hor_j.

▶ **Observation 10.** $\mathrm{DP}_2(k,i,j)$ *is the minimal* $t \in [t_{i,j}, s_{i,j}]$ *such that* $\mathrm{DP}(k-1,i',j') \neq \infty$ *and* $\delta_F(P[\mathrm{DP}(k-1,i',j')\ldots t], \overline{v_{i'}v_i}) \leq \delta$ *for some* $i' < i$ *and* $j' < j$. *If no such* t *exists then* $\mathrm{DP}_2(k,i,j) = \infty$.

Proof Sketch. The key idea is that if any $t \in [t_{i,j}, s_{i,j}]$, $(1,t)$ is reachable by a monotone path in $FS_\delta(P, \overline{v_{i'}v_i})$ from $(0, \mathrm{DP}(k-1,i',j'))$ for $i' < i$, then so is $(1, t_{ij})$ (see Figure 3). ◀

▶ **Lemma 11.** $\mathrm{DP}(k,i,j) = \min(\mathrm{DP}_1(k,i,j), \mathrm{DP}_2(k,i,j))$.

Proof. Follows directly from Observations 7, 9, and 10. ◀

In particular this yields a dynamic programming formulation for $\mathrm{DP}(k,i,j)$, since both $\mathrm{DP}_1(k,i,j)$ and $\mathrm{DP}_2(k,i,j)$ depend on values of $\mathrm{DP}(k',i',j')$ with $k' < k$, $i' < i$ and $j' \leq j$.

We define $\kappa(i,j)$ as the minimal k such that $\mathrm{DP}(k,i,j) \neq \infty$. Similarly we define $\kappa_1(i,j)$ and $\kappa_2(i,j)$ as the minimal k such that $\mathrm{DP}_1(k,i,j) \neq \infty$ and $\mathrm{DP}_2(k,i,j) \neq \infty$, respectively. Note that $\kappa(i,j) = \min(\kappa_1(i,j), \kappa_2(i,j))$ (by Lemma 11). Also note that both $\kappa_1(i,j)$ and $\kappa_2(i,j)$ depend only on the values of $\mathrm{DP}(k',i',j')$ with $k' < k$, $i' < i$ and $j' \leq j$.

With these preparations can now present our dynamic programming algorithm, except for one subroutine κ_2-*subroutine*(i) that we describe in Section 3.3. In particular, for any i, κ_2-subroutine(i) determines $\kappa_2(i,j)$ for all $j \in [n]$ only using the values of $\kappa(i',j)$ for all $i' < i$ and all $0 \leq j \leq n-1$. Now we show how to compute $\mathrm{DP}_1(k,i,j)$. Observe that for any i, j and k we can compute $\overline{\mathrm{DP}}_1(k,i,j)$ from $\overline{\mathrm{DP}}_1(k,i-1,j)$ and $\mathrm{DP}(k-1,i-1,j)$ as $\overline{\mathrm{DP}}_1(k,i,j) = \min(\overline{\mathrm{DP}}_1(k,i-1,j), \mathrm{DP}(k-1,i-1,j))$. Then we can compute $\mathrm{DP}_1(k,i,j)$ by using equation (1) and set $\kappa_1(i,j)$ to the minimal k such that $\mathrm{DP}_1(k,i,j) \neq \infty$. We compute $\mathrm{DP}_2(k,i,j)$ by setting $\mathrm{DP}_2(k,i,j) = t_{i,j}$ if $k \geq \kappa_2(i,j)$ and $\mathrm{DP}_2(k,i,j) = \infty$ otherwise. Also, we set $\kappa(i,j)$ as $\min(\kappa_1(i,j), \kappa_2(i,j))$. After that we can update $\mathrm{DP}(k,i,j)$ by the formulation in Lemma 11.

Algorithm 1 Solving curve simplification under Global-Fréchet distance.

```
1: Precompute all $t_{i,j}$ and $s_{i,j}$ and initialize all $\overline{\mathrm{DP}}_1(k,0,j)$, $\mathrm{DP}(k,0,j)$, $\kappa(0,j)$ and $\mathrm{DP}(0,i,j)$.
2: for all $i = 1$ to $n$ do
3:     Determine $\kappa_2(i,j)$ for all $0 \le j \le n-1$ using $\kappa_2$-subroutine($i$)
4:     for $j = 0$ to $n-1$ do
5:         for every $k \in [n+1]$ set $\overline{\mathrm{DP}}_1(k,i,j)$ to $\min(\overline{\mathrm{DP}}_1(k,i-1,j), \mathrm{DP}(k-1,i-1,j))$
6:         for every $k \in [n+1]$ set $\mathrm{DP}_1(k,i,j)$ to $\max(\overline{\mathrm{DP}}_1(k,i,j), t_{i,j})$ if $\overline{\mathrm{DP}}_1(k,i,j) \le s_{i,j}$ and to $\infty$ otherwise
7:         set $\kappa_1(i,j)$ to the smallest $k$ such that $\mathrm{DP}_1(k,i,j) \neq \infty$
8:         set $\kappa(i,j) = \min(\kappa_1(i,j), \kappa_2(i,j))$
9:         for every $k \in [n+1]$ set $\mathrm{DP}_2(k,i,j)$ to $t_{i,j}$ if $k \ge \kappa_2(i,j)$ and to $\infty$ otherwise
10:        for every $k \in [n+1]$ set $\mathrm{DP}(k,i,j)$ to $\min(\mathrm{DP}_1(k,i,j), \mathrm{DP}_2(k,i,j))$
```

Denote the running time of κ_2-subroutine(i) by $T(n)$. Since we fill each of $O(n^3)$ DP cells in time $O(1)$, the total running time of Algorithm 1 is $O(n^3 + n \cdot T(n))$.

3.3 Implementing the κ_2-subroutine(i)

In this subsection we show how to implement the κ_2-subroutine(i) in time $T(n) = \mathcal{O}(n^2)$. Then in total we have $\mathcal{O}(n^3)$ for solving Global-Fréchet simplification.

3.3.1 Cell Reachability

We introduce an auxiliary problem *Cell Reachability*. We shall see later that an $\mathcal{O}(n)$ time solution to this problem ensures that the κ_2-subroutine(i) can be implemented in time $\mathcal{O}(n^2)$.

▶ **Definition 12.** *In an instance of the Cell Reachability problem, we are given:*

- *A set of n* cells. *Each cell j with $1 \le j \le n$ is a unit square with corner points $(0,j)$ and $(1,j+1)$. We say that cells j and $j+1$ are* consecutive.
- *An integral* entry-cost *$\lambda_j > 0$ for every cell j.*
- *A set of $n-1$* passages *between consecutive cells. The passage p_j is the horizontal line segment with endpoints (j,a_j) and (j,b_j) where $b_j > a_j$.*

A cell j is reachable *from a cell j' with $j' < j$ if and only if there exists $x_{j'+1} \le x_{j'+2} \ldots \le x_j$ such that $x_k \in [a_k, b_k]$ for every $j' < k \le j$. Intuitively, cell j is reachable from cell j' if and only if there is a monotone path through the passages from cell j' to cell j. Let the* exit-cost *μ_j of a cell j as the minimal $\lambda_{j'}$ such that j is reachable from cell j', $j' < j$. The goal is to determine the sequence $\langle \mu_1, \mu_2, \ldots, \mu_n \rangle$. See Figure 4 for an illustration.*

We make a more refined notion of reachability. For any cells j and $j' < j$ we define the *first reachable point frp*(j,j') on cell j from cell j' as the minimal t such that there exist $x_{j'+1} \le x_{j'+2} \le \ldots \le x_j$ such that $x_k \in [a_k, b_k]$ for every $j' < k \le j$ and $x_j = t$, and we set *frp*$(j,j') = \infty$ if there exists no such t. Let $t_j(k)$ be the first reachable point on cell j from any cell j' with entry-cost at most k i.e. $t_j(k) = \min\{\mathit{frp}(j,j') \mid j' < j, \lambda_{j'} \le k\}$. Note that μ_j is the minimal k such that $t_j(k) \neq \infty$. Thus, it suffices to show how to determine $t_j(\cdot)$ and μ_j from $t_j(\cdot)$ for all $j \in [n]$ in $\mathcal{O}(n)$ time. To this end we generalize an algorithm by Alt et al. [4, Lemma 2.3]; the details can be found in the full version of this paper.

▶ **Theorem 13.** *Cell Reachability can be solved in $\mathcal{O}(n)$ time.*

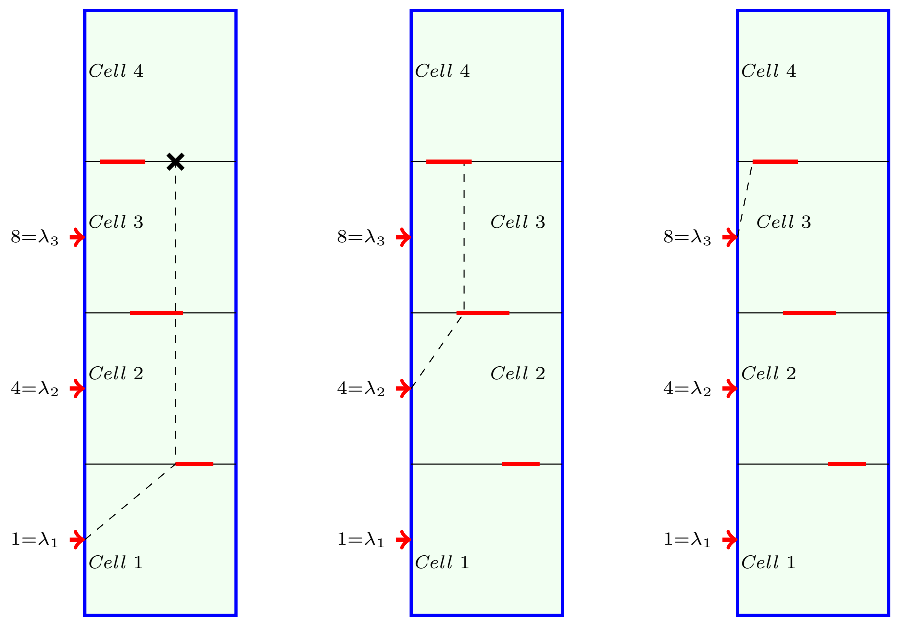

Figure 4 The red horizontal line segments between the cells indicate the passages. Cell 4 is only reachable from cells 2 and 3. Thus $\mu_4 = \min(\lambda_2, \lambda_3) = \min(4, 8) = 4$.

3.3.2 Implementing κ_2-subroutine(i) using Cell Reachability

Recall the definition of $\kappa_2(\cdot,\cdot)$ and what our goal is now: For a fixed $i' < i$, let $\kappa(i, j, i')$ be the minimal k such that for some $j' < j$, we have $\mathrm{DP}(k-1, i', j') \neq \infty$ and $\delta_F(P[\mathrm{DP}(k-1, i', j') \ldots t_{i,j}], \overline{v_{i'}v_i}) \leq \delta$. Note that $\kappa_2(i,j) = \min_{i'<i} \kappa(i, j, i')$. In order to show that the κ_2-subroutine(i) can be implemented in $\mathcal{O}(n^2)$, it suffices to show that for fixed $i' < i$ we can determine $\kappa(i, j, i')$ for all $j \in [n-1]$ in $\mathcal{O}(n)$ time.

Let the line segment between (a_j, j) and (b_j, j) denote the free-space on hor_j. Due to the convexity of every cell in the free space $FS_\delta(P, \overline{v_{i'}v_i})$, it suffices to look at reachability through the free space at the boundary of the cells (hor_j). Thus for any $j' < j$ there is a monotone path from $(0, \mathrm{DP}(\kappa(i', j'), i', j'))$ to $(1, t_{i,j})$ in the free-space $FS_\delta(P, \overline{v_{i'}v_i})$ if and only if there exist $x_{j'+1} \leq x_{j'+2} \leq \ldots \leq x_j$ with each $x_k \in [a_k, b_k]$ for all $j' < k \leq j$. Similarly, it suffices to look at the reachability only from the points $(0, \mathrm{DP}(\kappa(i', j'), i', j'))$. This yields the following observation, whose proof is illustrated in Figure 5.

▶ **Observation 14.** *For any $i' < i$ if there is a monotone path from $(0, \mathrm{DP}(k, i', j'))$ to $(1, t_{i,j})$ in the free-space $FS_\delta(P, \overline{v_{i'}v_i})$ intersecting hor_j, then there is also a monotone path from $(0, \mathrm{DP}(\kappa(i', j'), i', j'))$ to $(1, t_{i,j})$ in the free-space $FS_\delta(P, \overline{v_{i'}v_i})$ intersecting hor_j.*

Observation 14 implies that $\kappa(i, j, i')$ is the minimal value of $1 + \kappa(i', j')$ over all $j' < j$ such that there exist $x_{j'+1} \leq x_{j'+2} \leq \ldots \leq x_j$ with $x_k \in [a_k, b_k]$ for $j' < k \leq j$. Note that now we are in an instance of Cell Reachability. Thus for any fixed i' we can determine $\kappa(i, j, i')$ in $\mathcal{O}(n)$ time and therefore we can implement κ_2-subroutine(i) in $\mathcal{O}(n^2)$ time.

4 Conditional lower bound for curve simplification

In this section we show that an $\mathcal{O}(n^{3-\varepsilon}\mathrm{poly}(d))$ time algorithm for Global-Fréchet, Local-Fréchet or Local-Hausdorff simplification over $(\mathbb{R}^d, \|\|_p)$ for any $p \in [1, \infty)$, $p \neq 2$, would yield an $\mathcal{O}(n^{3-\varepsilon}\mathrm{poly}(d))$ algorithm for ∀∀∃-OV.

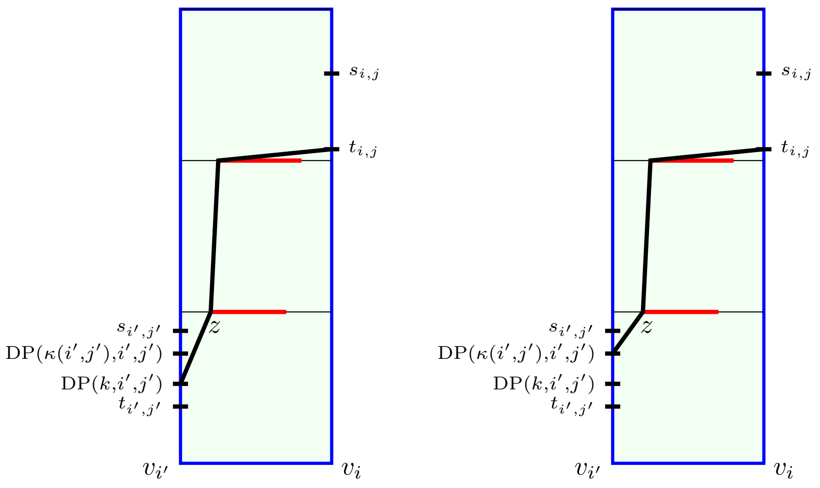

Figure 5 Illustration of the proof of Observation 14. For any $i' < i$, $j' < j$ and any k, there is a monotone path from $(0, \mathrm{DP}(k, i', j'))$ to $(1, t_{i,j})$ in $FS_\delta(P, \overline{v_{i'}v_i})$ (left) that intersects hor_j at z. Then there is a monotone path from $(0, \mathrm{DP}(\kappa(i', j'), i', j'))$ to $(1, t_{i,j})$ in $FS_\delta(P, \overline{v_{i'}v_i})$ (right) by walking from $(0, \mathrm{DP}(\kappa(i', j'), i', j')$ to z and then following the previous path from z to $(1, t_{i,j})$.

4.1 Overview of the reduction

Consider any instance (A, B, C) of ∀∀∃-OV where A,B,$C \subseteq \{0, 1\}^d$ have size n. We write $A = \{a_1, a_2, \dots a_n\}$, $B = \{b_1, b_2, \dots b_n\}$ and $C = \{c_1, c_2, \dots c_n\}$. We will efficiently construct $3n + 1$ points in $\mathbb{R}^D$ with $D \in \mathcal{O}(d)$ namely the sets of points $\tilde{A} = \{\tilde{a}_1, \tilde{a}_2, \dots \tilde{a}_n\}$, $\tilde{B} = \{\tilde{b}_1, \tilde{b}_2, \dots \tilde{b}_n\}$ and $\tilde{C} = \{\tilde{c}_1, \tilde{c}_2, \dots \tilde{c}_n\}$ and one more point s. We also determine $\delta \geq 0$ such that the following properties are satisfied.

(P$_1$) For any $\tilde{a} \in \tilde{A}, \tilde{b} \in \tilde{B}, \tilde{c} \in \tilde{C}$, there is a point x on the line segment $\overline{\tilde{a}\tilde{b}}$ with $\|x - \tilde{c}\|_p \leq \delta$ if and only if $\|\frac{\tilde{a}+\tilde{b}}{2} - \tilde{c}\|_p \leq \delta$.

(P$_2$) For any $\tilde{a} \in \tilde{A}, \tilde{b} \in \tilde{B}, \tilde{c} \in \tilde{C}$, we have $\|\frac{\tilde{a}+\tilde{b}}{2} - \tilde{c}\|_p \leq \delta$ if and only if $\sum_{\ell \in [d]} a[\ell] \cdot b[\ell] \cdot c[\ell] \neq 0$.

(P$_3$) $\|x - y\|_p \leq \delta$ holds for all $x, y \in \tilde{A}$, and for all $x, y \in \tilde{B}$ and for all $x, y \in \tilde{C}$.

(P$_4$) For any $y_1, y_2 \in \{s\} \cup \tilde{B} \cup \tilde{C}$ and any point x on the line segment $\overline{y_1y_2}$ we have $\|x - \tilde{a}\|_p > \delta$ for all $\tilde{a} \in \tilde{A}$.

(P$_5$) For any $y_1, y_2 \in \{s\} \cup \tilde{A} \cup \tilde{C}$ and any point x on the line segment $\overline{y_1y_2}$ we have $\|x - \tilde{b}\|_p > \delta$ for all $\tilde{b} \in \tilde{B}$.

(P$_6$) For any $y \in \tilde{B} \cup \tilde{A}$ and any point x on the line segment $\overline{sy}$ we have $\|x - \tilde{c}\|_p > \delta$ for all $\tilde{c} \in \tilde{C}$.

We postpone the exact construction of these points. Our hard instance for curve simplification will be $Q = \langle s, \tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_n, \tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_n, \tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n, s\rangle$.

▶ **Fact 15.** *Let $r_1, r_2, s_1, s_2 \in \mathbb{R}^D$ and $\|r_k - s_k\|_p \leq \delta$ for $k \in [2]$. Then $\delta_F(\overline{r_1r_2}, \overline{s_1s_2}) \leq \delta$.*

▶ **Lemma 16.** *Let $\hat{Q} = \langle s, \tilde{a}_i, \tilde{b}_j, s\rangle$ for some $\tilde{a}_i \in \tilde{A}$ and $\tilde{b}_j \in \tilde{B}$. If $\|\frac{\tilde{a}_i+\tilde{b}_j}{2} - \tilde{c}\|_p \leq \delta$ for all $\tilde{c} \in \tilde{C}$ then the Local-Frechet distance between Q and $\hat{Q}$ is at most δ.*

Proof. Both Q and $\hat{Q}$ have the same starting point s. By property **P$_3$** we have $\|\tilde{a} - \tilde{a}_i\|_p \leq \delta$ for all $\tilde{a} \in \tilde{A}$, and $\|\tilde{b} - \tilde{b}_j\|_p \leq \delta$ for all $\tilde{b} \in \tilde{B}$. Thus it follows that $\delta_F(\langle s, \tilde{a}_1, \dots, \tilde{a}_i\rangle, \overline{s\tilde{a}_i}) \leq \delta$ and $\delta_F(\langle \tilde{b}_j, \dots, \tilde{b}_n, s\rangle, \overline{\tilde{b}_j s}) \leq \delta$. It remains to show that $\delta_F(Q_{ij}, \overline{\tilde{a}_i\tilde{b}_j}) \leq \delta$ where $Q_{ij} =$

$\langle \tilde{a}_i, \dots, \tilde{a}_n, \tilde{c}_1, \dots, \tilde{c}_n, \tilde{b}_1, \dots, \tilde{b}_j \rangle$. To this end first note that both Q_{ij} and $\overline{\tilde{a}_i \tilde{b}_j}$ have the same endpoints. We outline monotone walks within distance δ on both Q_{ij} and $\overline{\tilde{a}_i \tilde{b}_j}$.

(1) Walk on Q_{ij} from $\tilde{a}_i$ to $\tilde{a}_n$ and remain at $\tilde{a}_i$ on $\overline{\tilde{a}_i \tilde{b}_j}$. (by Property $\mathbf{P}_3$)

(2) Walk uniformly on both polylines, up to $\frac{\tilde{a}_i + \tilde{b}_j}{2}$ on $\overline{\tilde{a}_i \tilde{b}_j}$ and up to $\tilde{c}_1$ on Q_{ij}. (by Fact 15)

(3) Walk on Q_{ij} from $\tilde{c}_1$ to $\tilde{c}_n$ and remain at $\frac{\tilde{a}_i + \tilde{b}_j}{2}$ on $\overline{\tilde{a}_i \tilde{b}_j}$. (by assumption)

(4) Walk uniformly on both curves up to $\tilde{b}_j$ on $\overline{\tilde{a}_i \tilde{b}_j}$ and up to $\tilde{b}_1$ on Q_{ij}. (by Fact 15)

(5) Walk on Q_{ij} until $\tilde{b}_j$ and remain at $\tilde{b}_j$ on $\overline{\tilde{a}_1 \tilde{b}_j}$. (by Property $\mathbf{P}_3$) ◀

Observe that property $\mathbf{P_3}$ implies that there is a simplification of size five namely $\hat{Q} = \langle s, \tilde{a}, \tilde{c}, \tilde{b}, s \rangle$ for any $\tilde{a} \in \tilde{A}$, $\tilde{b} \in \tilde{B}$, and $\tilde{c} \in \tilde{C}$, such that the distance between $\hat{Q}$ and Q is at most δ under all three distance measures. We now show that a smaller simplification is only possible if there exist $a \in A$, $b \in B$ such that for all $c \in C$ we have $\sum_{\ell \in [d]} a[\ell] \cdot b[\ell] \cdot c[\ell] \neq 0$.

▶ **Lemma 17.** *Let $\hat{Q}$ be a simplification of the polyline Q of size 4. Then the following statements are equivalent:*

(1) *The Global-Fréchet distance between Q and $\hat{Q}$ is at most δ.*

(2) *The Local-Fréchet distance between Q and $\hat{Q}$ is at most δ.*

(3) *The Local-Hausdorff distance between Q and $\hat{Q}$ is at most δ.*

(4) *There exists $\tilde{a} \in \tilde{A}$, $\tilde{b} \in \tilde{B}$, such that $\hat{Q} = \langle s, \tilde{a}, \tilde{b}, s \rangle$ and $\| \frac{\tilde{a} + \tilde{b}}{2} - \tilde{c} \|_p \leq \delta$ for every $\tilde{c} \in \tilde{C}$.*

(5) *There exist $a \in A$, $b \in B$ such that for all $c \in C$ we have $\sum_{\ell \in [d]} a[\ell] \cdot b[\ell] \cdot c[\ell] \neq 0$.*

A detailed proof can be found in the full version of this paper. Assuming that we can determine Q and δ in $\mathcal{O}(nd)$ time, the above lemma directly yields the following theorem.

▶ **Theorem 18.** *For any $\varepsilon > 0$, there is no $\mathcal{O}(n^{3-\varepsilon} \mathrm{poly}(d))$ algorithm for Global-Fréchet, Local-Fréchet and Local-Hausdorff simplification over $(\mathbb{R}^d, \|\|_p)$ for any $p \in [1, \infty)$, $p \neq 2$, unless* ∀∀∃-OV *Hypothesis fails. This holds even for the problem of deciding whether the optimal simplification has size ≤ 4 or ≥ 5.*

Proof. given an instance A,B,C of ∀∀∃-OV, we can construct the curve Q and δ in time $\mathcal{O}(nd)$. By Lemma 17, the simplification problem on (Q, δ) is equivalent to ∀∀∃-OV on A,B,C. Thus, any $\mathcal{O}(n^{3-\varepsilon} \mathrm{poly}(d))$ time algorithm for the curve simplification problem would yield an $\mathcal{O}(n^{3-\varepsilon} \mathrm{poly}(d))$ time algorithm for ∀∀∃-OV. ◀

It remains to construct the point s, the sets $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ and δ. We first introduce some notation. For vectors x and y and $\alpha \in [-\frac{1}{2}, \frac{1}{2}]$, we define $P_{xy}(\alpha)$ as $(\frac{1}{2} - \alpha)x + (\frac{1}{2} + \alpha)y$. Moreover let $u_i \in \mathbb{R}^d$. We write $v = \left[u_1 u_2 \dots u_m \right]$ for the vector $v \in \mathbb{R}^{md}$ with $v[(j-1)d+k] = u_j[k]$ for any $j \in [m]$ and $k \in [d]$.

▶ **Fact 19.** *Let $u_1, u_2, \dots, u_m \in \mathbb{R}^d$ and $v = \left[u_1 u_2 \dots u_m \right]$. Then we have $\|v\|_p^p = \sum_{i \in [m]} \|u_i\|_p^p$.*

4.2 Cordinate gadgets

In this section, our aim is to construct points $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ for $i \in \{0, 1\}$ such that the distance $\|\mathbf{C}_i - P_{\mathbf{A}_j \mathbf{B}_k}(0)\|_p$ only depends on whether the bits $i, j, k \in \{0, 1\}$ seen as cordinates of vectors are orthogonal. In other words, the points $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ form a *cordinate gadget*. Formally we will prove the following lemma,

▶ **Lemma 20.** *For any $p \neq 2$*

$$\|\mathbf{C}_i - P_{\mathbf{A}_j\mathbf{B}_k}(0)\|_p^p = \begin{cases} \beta_1 & \textit{if } i=1, j=1, k=1 \\ \beta_2 & \textit{otherwise} \end{cases}$$

where $\beta_1 < \beta_2$.

In Section 4.3 we will use this lemma to construct the final point sets . Let θ_1, θ_2, θ_3, θ_4 and θ_5 be positive constants. We construct the points $\mathbf{A}_1$,$\mathbf{B}_1$,$\mathbf{C}_1$ and $\mathbf{A}_0$,$\mathbf{B}_0$,$\mathbf{C}_0$ in $\mathbb{R}^9$.

$$\begin{aligned}
\mathbf{A}_0 &= [\ -\theta_1, \ 0, \ -\theta_2, \ 0, \ \theta_3, \ 2\theta_3, \ \theta_4, \ -2\theta_4, \ 0] \\
\mathbf{A}_1 &= [\ \theta_1, \ 2\theta_1, \ \theta_2, \ -2\theta_2, \ -\theta_3, \ 0, \ -\theta_4, \ 0, \ 0] \\
\mathbf{B}_0 &= [\ -\theta_1, \ 0, \ \theta_2, \ 2\theta_2, \ \theta_3, \ -2\theta_3, \ -\theta_4, \ 0, \ 0] \\
\mathbf{B}_1 &= [\ \theta_1, \ -2\theta_1, \ -\theta_2, \ 0, \ -\theta_3, \ 0, \ \theta_4, \ 2\theta_4, \ 0] \\
\mathbf{C}_0 &= [\ 0, \ 0, \ 0, \ 0, \ 0, \ 0, \ 0, \ 0, \ \theta_5] \\
\mathbf{C}_1 &= [\ -\theta_1, \ 0, \ -\theta_2, \ 0, \ -\theta_3, \ 0, \ -\theta_4, \ 0, \ 0]
\end{aligned}$$

From these points we can compute the points $P_{\mathbf{A}_i\mathbf{B}_j}(0)$ for all $i,j \in \{0,1\}$.

$$\begin{aligned}
P_{\mathbf{A}_0\mathbf{B}_0}(0) &= [\ -\theta_1, \ 0, \ 0, \ \theta_2, \ \theta_3, \ 0, \ 0, \ -\theta_4, \ 0] \\
P_{\mathbf{A}_1\mathbf{B}_0}(0) &= [\ 0, \ \theta_1, \ \theta_2, \ 0, \ 0, \ -\theta_3, \ -\theta_4, \ 0, \ 0] \\
P_{\mathbf{A}_1\mathbf{B}_1}(0) &= [\ \theta_1, \ 0, \ 0, \ -\theta_2, \ -\theta_3, \ 0, \ 0, \ \theta_4, \ 0] \\
P_{\mathbf{A}_0\mathbf{B}_1}(0) &= [\ 0, \ -\theta_1, \ -\theta_2, \ 0, \ 0, \ \theta_3, \ \theta_4, \ 0, \ 0]
\end{aligned}$$

Observe that $\|\mathbf{C}_0 - P_{\mathbf{A}_i\mathbf{B}_j}(0)\|_p^p = \sum_{r\in[5]} \theta_r^p$ for all $i,j \in \{0,1\}$. Thus all the points $P_{\mathbf{A}_i\mathbf{B}_j(0)}$ are equidistant from $\mathbf{C}_0$ irrespective of the exact values of θ_r for $r \in [5]$. Note that when $\theta_r = \theta$ for all $r \in [5]$, then $\|\mathbf{C}_1 - P_{\mathbf{A}_i\mathbf{B}_j}(0)\|_p^p = 4\theta^p + 2^p\theta^p$ for all $i,j \in \{0,1\}$. Thus all the points $P_{\mathbf{A}_i\mathbf{B}_j}(0)$ are equidistant from $\mathbf{C}_1$ when all the θ_r are the same. We now determine θ_r for $r \in [5]$ such that all but one point in $\{P_{\mathbf{A}_i\mathbf{B}_j}(0) | i,j \in \{0,1\}\}$ are equidistant and far from $\mathbf{C}_1$. More precisely,

$$\|\mathbf{C}_1 - P_{\mathbf{A}_i\mathbf{B}_j}(0)\|_p^p = \begin{cases} \beta_1 & \text{if } i=1, j=1 \\ \beta_2 & \text{otherwise} \end{cases}$$

and $\beta_1 < \beta_2$. We first quantify the distances from $\{\mathbf{C}_0, \mathbf{C}_1\}$ to each of the points in $\{P_{\mathbf{A}_j\mathbf{B}_k}(0) \mid j,k \in \{0,1\}\}$.

▶ **Lemma 21.** *We have*

$$\|\mathbf{C}_i - P_{\mathbf{A}_j\mathbf{B}_k}(0)\|_p^p = \begin{cases} \sum_{r\in[5]} \theta_r^p & \textit{if } i=0 \\ 2\theta_2^p + 2^p\theta_3^p + 2\theta_4^p & \textit{if } i=1, j=0, k=0 \\ 2\theta_1^p + 2^p\theta_2^p + 2\theta_3^p & \textit{if } i=1, j=1, k=0 \\ 2\theta_1^p + 2\theta_3^p + 2^p\theta_4^p & \textit{if } i=1, j=0, k=1 \\ 2^p\theta_1^p + 2\theta_2^p + 2\theta_4^p & \textit{if } i=1, j=1, k=1 \end{cases}$$

We set the values of θ_r for $r \in [5]$ depending on p. When $1 \leq p < 2$ we set

$$\theta_1 = (2^{p-1} - 1)^{\frac{1}{p}}, \theta_2 = 0, \theta_3 = 1, \theta_4 = 0, \theta_5 = 2^{\frac{p-1}{p}}$$

Substituting these values of θ_r for all $r \in [5]$ in Lemma 21 we make the following observation.

▶ **Observation 22.** *When $1 \le p < 2$, then*

$$\|\mathbf{C}_i - P_{\mathbf{A}_j\mathbf{B}_k}(0)\|_p^p = \begin{cases} 2^p(2^{p-1}-1) & \textit{if } i=1, j=1, k=1 \\ 2^p & \textit{otherwise} \end{cases}$$

In case $p > 2$. Then we set

$$\theta_1 = 0, \theta_2 = (2^p-2)^{\frac{1}{p}}, \theta_3 = (2^p-4)^{\frac{1}{p}}, \theta_4 = (2^p-2)^{\frac{1}{p}}, \theta_5 = (2^{2p} - 3\cdot 2^p)^{\frac{1}{p}}$$

Substituting these values of θ_r for all $r \in [5]$ in Lemma 21 we make the following observation.

▶ **Observation 23.** *When $p > 2$, then*

$$\|\mathbf{C}_i - P_{\mathbf{A}_j\mathbf{B}_k}(0)\|_p^p = \begin{cases} 2^{p+2}-8 & \textit{if } i=1, j=1, k=1 \\ 2^{2p}-8 & \textit{otherwise} \end{cases}$$

Combining Observations 22 and 23 we arrive at Lemma 20.

4.3 Vector gadgets

For every $a \in A$, $b \in B$ and $c \in C$ we introduce vectors a', b', c' and $a''b'', c''$ and then concatenate the respective vectors to form $\tilde{a}$, $\tilde{b}$ and $\tilde{c}$ respectively. Intuitively a', b', c' help us to ensure properties $\mathbf{P}_1$ and $\mathbf{P}_2$, while a'', b'', c'' help us ensure the other properties.

4.3.1 The vectors a', b', c', and s'

We construct the vector s' and the vectors a', b' and c' for every $a \in A$, $b \in B$ and $c \in C$ respectively, in $\mathbb{R}^{9d}$ as follows:

$$a' = \left[\mathbf{A}_{a[1]}, \mathbf{A}_{a[2]}, \dots \mathbf{A}_{a[d]}\right] \qquad b' = \left[\mathbf{B}_{b[1]}, \mathbf{B}_{b[2]}, \dots \mathbf{B}_{b[d]}\right]$$
$$c' = \left[\mathbf{C}_{c[1]}, \mathbf{C}_{c[2]}, \dots \mathbf{C}_{c[d]}\right] \qquad s' = \left[0, 0, \dots, 0\right]$$

We also define the sets $A' = \{a' \mid a \in A\}$, $B' = \{b' \mid b \in B\}$ and $C' = \{c' \mid c \in C\}$. We now make a technical observation about the vectors in A', B', and C', that will be useful later. We set $\eta_1 = \max_{i \in [5]} \theta_i$.

▶ **Observation 24.** *For any $x, y \in A' \cup B' \cup C'$, we have $\|x - y\|_p \le \eta_2$ where $\eta_2 := 36d\eta_1$.*

Proof Sketch. Note that the absolute value of every cordinate of the vectors a', b', and c', is bounded by $2\eta_1$ (Since every cordinate is of the form $\pm\theta_r$ or $\pm 2\theta_r$ or 0). Combined with the fact that the total number of cordinates is $9d$, this immediately implies the observation. ◀

Note that $a \in A$, $b \in B$ and $c \in C$ are non orthogonal if and only if $\#_{111}^{c,a,b} > 0$ where $\#_{111}^{c,a,b} = |\{i \mid i \in [d], a[i] = b[i] = c[i] = 1\}|$. The following lemma shows a connection between non-orthogonality and small distance $\|c' - P_{a'b'}(0)\|_p$.

▶ **Lemma 25.** *For any $a \in A$, $b \in B$ and $c \in C$, $\|c' - P_{a'b'}(0)\|_p^p = d\beta_2 - (\beta_2 - \beta_1)\#_{111}^{c,a,b}$.*

Proof. By Observation 19 we have

$$\begin{aligned}\|c' - P_{a'b'}(0)\|_p^p &= \sum_{\ell \in [d]} \|\mathbf{C}_{c[\ell]} - P_{\mathbf{A}_{a[\ell]}\mathbf{B}_{b[\ell]}}(0)\|_p^p \\ &= \beta_2(d - \#_{111}^{c,a,b}) + \beta_1\#_{111}^{c,a,b} \quad \text{(by Lemma 20)} \\ &= d\beta_2 - (\beta_2 - \beta_1)\#_{111}^{c,a,b}.\end{aligned}$$

◀

4.3.2 The vectors a'',b'', c'', and s''

We construct the vector s'' and the vectors a'',b'', and c'' for every $a \in A$,$b \in B$, and $c \in C$, respectively, in $\mathbb{R}^3$ as follows:

$$a'' = \left[\gamma_1, 0, 0\right] \quad b'' = \left[\gamma_1, \gamma_2, 0\right] \quad c'' = \left[0, \frac{\gamma_2}{2}, 0\right] \quad s'' = \left[0, \frac{\gamma_2}{2}, \gamma_2\right]$$

where γ_1, γ_2 are positive constants. We are now ready to define the final points of our construction, s and $\tilde{a}$, $\tilde{b}$ and $\tilde{c}$ for any $a \in A$, $b \in B$ and $c \in C$, respectively.

$$\tilde{a} = \left[a', a''\right] \quad \tilde{b} = \left[b', b''\right] \quad \tilde{c} = \left[c', c''\right] \quad s = \left[s', s''\right]$$

We set

$$\gamma_1 = \eta_1,\ \delta = (\gamma_1^p + d\beta_2 - (\beta_2 - \beta_1))^{\frac{1}{p}},\ \gamma_2 = \max\left(4\delta, \eta_2\left(1 + \frac{(\gamma_1^p + d\beta_2)^{\frac{1}{p}}}{(\gamma_1^p + d\beta_2)^{\frac{1}{p}} - \delta}\right)\right)$$

Note that we have constructed the point sets $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and the point s and determined δ in total time $\mathcal{O}(nd)$. Therefore now it suffices to show that our point set and δ satisfy the properties $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, $\mathbf{P}_5$, and $\mathbf{P}_6$. To this end, we first show how the distance $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(\alpha)\|_p$ is related with $\#_{111}^{c,a,b}$ (the non-orthogonality of the vectors a,b, and c).

▶ **Lemma 26.** *For any $a \in A$, $b \in B$ and $c \in C$ we have,*

- $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(0)\|_p^p = \gamma_1^p + \beta_2 d - (\beta_2 - \beta_1)\#_{111}^{c,a,b}$.
- *If* $\#_{111}^{c,a,b} = 0$ *then* $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(\alpha)\|_p^p > \delta$ *for all* $\alpha \in [-\frac{1}{2}, \frac{1}{2}]$.

Proof Sketch. Note that $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(0)\|_p = \|c' - P_{a'b'}(0)\|_p + \|c'' - P_{a''b''}(0)\|_p$. Using Lemma 25 and substituting vectors a'',b'' and c'' we arrive at $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(0)\|_p^p = \gamma_1^p + \beta_2 d - (\beta_2 - \beta_1)\#_{111}^{c,a,b}$. Since we choose our γ_2 to be sufficiently large the closest point on the line-segment $\overline{\tilde{a}\tilde{b}}$ to $\tilde{c}$ is sufficiently near $P_{\tilde{a}\tilde{b}}(0)$. Thus when $\#_{111}^{c,a,b} = 0$, we have $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(0)\|_p^p = \gamma_1^p + \beta_2 d > \delta$ and no point sufficiently near $P_{\tilde{a}\tilde{b}}(0)$ has distance smaller than δ to $\tilde{c}$. ◀

We now show that the properties $\mathbf{P}_1$ and $\mathbf{P}_2$ hold. The first result of Lemma 26 implies that for any $a \in A$, $b \in B$ and $c \in C$ we have $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(0)\|_p \leq \delta$ if and only if $\#_{111}^{c,a,b} \geq 1$, or equivalently if $\sum_{\ell \in [d]} a[\ell] \cdot b[\ell] \cdot c[\ell] \neq 0$. By the second result of Lemma 26, it follows that for any $\alpha \in [-\frac{1}{2}, \frac{1}{2}]$ if $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(\alpha)\|_p^p \leq \delta$, then $\#_{111}^{c,a,b} = 0$ which implies $\|\tilde{c} - P_{\tilde{a}\tilde{b}}(0)\|_p^p \leq \delta$. The remaining properties are guaranteed primarily by the component vectors a'', b'' and c'' of $\tilde{a}$, $\tilde{b}$ and $\tilde{c}$, and the detailed proof is in the full version of the paper.

References

1 Mohammad Ali Abam, Mark de Berg, Peter Hachenberger, and Alireza Zarei. Streaming Algorithms for Line Simplification. *Discrete & Computational Geometry*, 43(3):497–515, 2010.

2 Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and Fixed Parameter Subquadratic Algorithms for Radius and Diameter in Sparse Graphs. In *SODA*, pages 377–391. SIAM, 2016.

3 Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-Linear Time Approximation Algorithms for Curve Simplification. *Algorithmica*, 42(3-4):203–219, 2005.

4 Helmut Alt, Alon Efrat, Günter Rote, and Carola Wenk. Matching planar maps. In *SODA*, pages 589–598. ACM/SIAM, 2003.

5 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1–2):78–99, 1995.

6 Gill Barequet, Danny Z. Chen, Ovidiu Daescu, Michael T. Goodrich, and Jack Snoeyink. Efficiently Approximating Polygonal Paths in Three and Higher Dimensions. *Algorithmica*, 33(2):150–167, 2002.

7 Kevin Buchin, Maike Buchin, Maximilian Konzack, Wolfgang Mulzer, and André Schulz. Fine-grained analysis of problems on curves. *EuroCG, Lugano, Switzerland*, 2016.

8 W.S. Chan and F. Chin. APPROXIMATION OF POLYGONAL CURVES WITH MINIMUM NUMBER OF LINE SEGMENTS OR MINIMUM ERROR. *International Journal of Computational Geometry & Applications*, 06(01):59–77, 1996.

9 Danny Z. Chen, Ovidiu Daescu, John Hershberger, Peter M. Kogge, Ningfang Mi, and Jack Snoeyink. Polygonal path simplification with angle constraints. *Comput. Geom.*, 32(3):173–187, 2005.

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

11 David H. Douglas and Thomas 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.

12 Regina Estkowski and Joseph S. B. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *SoCG*, pages 40–49. ACM, 2001.

13 Stefan Funke, Thomas Mendel, Alexander Miller, Sabine Storandt, and Maria Wiebe. Map Simplification with Topology Constraints: Exactly and in Practice. In *ALENEX*, pages 185–196. SIAM, 2017.

14 Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and R. Ryan Williams. Completeness for First-Order Properties on Sparse Structures with Algorithmic Applications. In *SODA*, pages 2162–2181. SIAM, 2017.

15 Michael Godau. A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In *STACS*, pages 127–136, 1991.

16 Leonidas J. Guibas, John Hershberger, Joseph S. B. Mitchell, and Jack Snoeyink. Approximating Polygons and Subdivisions with Minimum Link Paths. *Int. J. Comput. Geometry Appl.*, 3(4):383–415, 1993.

17 John Hershberger and Jack Snoeyink. An $O(n \log n)$ Implementation of the Douglas-Peucker Algorithm for Line Simplification. In *SoCG*, pages 383–384. ACM, 1994.

18 Hiroshi Imai and Masao Iri. Polygonal Approximations of a Curve — Formulations and Algorithms. In *Computational Morphology*, volume 6 of *Machine Intelligence and Pattern Recognition*, pages 71–86. North-Holland, 1988.

19 Avraham Melkman and Joseph O'Rourke. On Polygonal Chain Approximation. In *Computational Morphology*, volume 6 of *Machine Intelligence and Pattern Recognition*, pages 87–95. North-Holland, 1988.

20 Marc J. van Kreveld, Maarten Löffler, and Lionov Wiratma. On Optimal Polyline Simplification Using the Hausdorff and Fréchet Distance. In *SoCG*, volume 99 of *LIPIcs*, pages 56:1–56:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

21 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, 2018.