On the hardness of computing an average curve

Kevin Buchin
Department of Mathematics and Computing Science, TU Eindhoven, The Netherlands
k.a.buchin@tue.nl

Anne Driemel
University of Bonn, Hausdorff Center for Mathematics, Bonn, Germany
driemel@cs.uni-bonn.de

Martijn Struijs
Department of Mathematics and Computing Science, TU Eindhoven, The Netherlands
m.a.c.struijs@tue.nl

Abstract
We study the complexity of clustering curves under \(k\)-median and \(k\)-center objectives in the metric space of the Fréchet distance and related distance measures. Building upon recent hardness results for the minimum-enclosing-ball problem under the Fréchet distance, we show that also the \(1\)-median problem is NP-hard. Furthermore, we show that the \(1\)-median problem is \(W[1]\)-hard with the number of curves as parameter. We show this under the discrete and continuous Fréchet and Dynamic Time Warping (DTW) distance. This yields an independent proof of an earlier result by Bulteau et al. from 2018 for a variant of DTW that uses squared distances, where the new proof is both simpler and more general. On the positive side, we give approximation algorithms for problem variants where the center curve may have complexity at most \(\ell\) under the discrete Fréchet distance. In particular, for fixed \(k, \ell\) and \(\varepsilon\), we give \((1 + \varepsilon)\)-approximation algorithms for the \((k, \ell)\)-median and \((k, \ell)\)-center objectives and a polynomial-time exact algorithm for the \((k, \ell)\)-center objective.

1 Introduction
Clustering is an important tool in data analysis, used to split data into groups of similar objects. Their dissimilarity is often based on distance between points in Euclidean space. However, the dissimilarity of polygonal curves is more accurately measured by specialised measures: Dynamic Time Warping (DTW) [25], continuous and discrete Fréchet distance [1, 13].

We focus on centroid-based clustering, where each cluster has a center curve and the quality of the clustering is based on the similarity between the center and the elements inside the cluster. In particular, given a distance measure \(\delta\), we consider the following problems:

\[ \text{Problem 1 (}\textit{k}-\text{median for curves with distance } \delta) \]
Given a set \( \mathcal{G} = \{g_1, \ldots, g_m\} \) of polygonal curves, find a set \( \mathcal{C} = \{c_1, \ldots, c_k\} \) of polygonal curves that minimizes \( \sum_{g \in \mathcal{G}} \min_{i=1}^{k} \delta(c_i, g) \).

\[ \text{Problem 2 (}\textit{k}-\text{center for curves with distance } \delta) \]
Given a set \( \mathcal{G} = \{g_1, \ldots, g_m\} \) of polygonal curves, find a set \( \mathcal{C} = \{c_1, \ldots, c_k\} \) of polygonal curves that minimizes \( \max_{g \in \mathcal{G}} \min_{i=1}^{k} \delta(c_i, g) \).

For points in Euclidean space, the most widely-used centroid-based clustering problem is \(k\)-means, in which the distance measure \(\delta\) is the squared Euclidean distance. But also for general metric spaces the \(k\)-median problem is well studied, often in the context of the closely related facility location problem [20, 21, 22]. In general metric spaces usually, the \textit{discrete} \(k\)-\textit{median} problem is studied, where the centers must be selected from a finite set \(F\), and are called facilities.
On the hardness of computing an average curve

For clustering curves, limiting the possible centers to a finite set of ‘facilities’ is unnecessarily restrictive. In this paper, we are therefore interested in the unconstrained $k$-median problem, where a center can be any element of the metric space (as in the case of $k$-means). Often, we will simply write $k$-median problem to denote the unconstrained version. In this paper, we are in particular interested in the complexity of the 1-median problem, which we refer to as average curve problem.

Hardness of the average curve problem

While clustering on points for general $k$ in the plane or higher dimension is often NP-hard [24], many point clustering problems can be solved efficiently when $k = 1$ in low dimension. For instance, the 1-center problem in the plane can be solved in linear time [23], and there are practical algorithms for higher dimensional Euclidean space [15]. In contrast, the 1-center problem (i.e., the minimum enclosing ball problem) for curves under the discrete and continuous Fréchet distance is already NP-hard in 1D [6].

In this paper, we show that also the average curve problem, i.e. the 1-median problem, is NP-hard. We show this for the discrete and for the continuous Fréchet distance, and for the dynamic time-warping (DTW) distance. Variants of the DTW distance differ in the norm used for comparing pairs of points, and how that norm is used, see Section 1.1 for details. Our results apply to a large class of variants of DTW. For the frequently used variant of DTW using the squared Euclidean distance, Bulteau et al. [8] recently showed that the average curve problem is NP-hard and even W[1]-hard when parametrized in the number of input curves $m$ and there exists no $f(m) \cdot n^{o(m)}$-time algorithm unless the Exponential Time Hypothesis (ETH) fails. Because of its importance in time series clustering, there are many heuristics for the average curve problem under DTW [18, 19, 25]. Brill et al. showed that dynamic programming yields an exponential-time exact algorithm [4] and additionally show the problem can be solved in polynomial time when both the input curves and center curve use only vertices from \{0, 1\}.

Approximation algorithms

Since both the $k$-center and the $k$-median problem for curves are already NP-hard for $k = 1$ in 1D, we further study efficient approximation algorithms for these problems.

For approximation in metric spaces, the discrete and unconstrained $k$-median (likewise for $k$-center) are closely related: any set of curves that realises an $\alpha$-approximation for the discrete $k$-median problem realises a $2\alpha$-approximation for the unconstrained $k$-median problem. There is an elegant $O(kn)$ time 2-approximation algorithm for the $k$-center problem in metric spaces [17]. This approximation factor is tight for clustering curves under the discrete [6] and continuous [30] Fréchet distance. Finding approximate solutions for $k$-median is more challenging: the best known polynomial-time approximation algorithm for discrete $k$-median in general metric space achieves a factor of $3 + \varepsilon$ for any $\varepsilon > 0$ [2] and it is NP-hard to achieve an approximation factor of $1 + 2/e$ [20].

Unconstrained clustering of curves may result in centers of high complexity. To avoid overfitting and to obtain a compact representation of the data, we look at a variant of the clustering problems with center curves of at most a fixed complexity, denoted by $\ell$. More formally, the $(k, \ell)$-center problem is to find a set of curves $C = \{c_1, \ldots, c_k\}$, each of complexity at most $\ell$, that minimizes $\max_{g \in G} \min_{k=1}^{k} \delta(c_k, g)$. The $(k, \ell)$-median problem is

\footnote{See e.g. [11] for background on parametrized complexity}
Figure 1 $(k, \ell)$-center clustering of pigeon flight paths computed by the algorithm of [7].

defined analogously. Although the general case for this variant is still NP-hard, we can find efficient algorithms when $k$ and $\ell$ are fixed. The $(k, \ell)$-center and $(k, \ell)$-median problems were introduced by Driemel et al. [12], who obtained an $\tilde{O}(mn)$-time $(1 + \varepsilon)$-approximation algorithm for the $(k, \ell)$-center and $(k, \ell)$-median problem under the Fréchet distance for curves in 1D, assuming $k, \ell, \varepsilon$ are constant. In [6], Buchin et al. gave polynomial-time constant-factor approximation algorithms for the $(k, \ell)$-center problem under the discrete and continuous Fréchet distance for curves in arbitrary dimension. These approximation algorithms have lead to efficient implementations of heuristics for the center version showing that the considered clustering formulations are useful in practice [7], see Figure 1. This encourages further study of the median variants of the problem.

1.1 Definitions of distance measures

Let $x$ be a polygonal curve, defined by a sequence of vertices $x_1, \ldots, x_n$ from $\mathbb{R}^d$ where consecutive vertices are connected by straight line segments. We call the number of vertices of $x$ the complexity, denoted by $|x|$. Given a pair of polygonal curves $x, y$, a warping path between them is a sequence $W = (w_1, \ldots, w_L)$ of index pairs $w_l = (i_l, j_l)$ from $\{1, \ldots, |x|\} \times \{1, \ldots, |y|\}$ such that $w_1 = (1, 1)$, $w_L = (|x|, |y|)$, and $(i_{l+1} - i_l, j_{l+1} - j_l) \in \{(0, 1), (1, 0), (1, 1)\}$ for all $1 \leq l < L$. We say two vertices $x_i, y_j$ are matched if $(i, j) \in W$.

Denote the set of all warping paths between curves $x$ and $y$ by $W_{x,y}$. For any integers $p, q \geq 1$, we define the Dynamic Time Warping Distance between $x$ and $y$ as

$$\text{DTW}^q_p(x, y) := \left( \min_{W \in W_{x,y}} \sum_{(i,j) \in W} \|x_i - y_j\|^p \right)^{q/p},$$

where $\| \cdot \|$ denotes the Euclidean norm. In text, we refer to $\text{DTW}^q_p$ also as $(p, q)$-DTW. Similarly, define the discrete Fréchet distance between $x, y$ as

$$d_{DF}(x, y) := \min_{W \in W_{x,y}} \max_{(i,j) \in W} \|x_i - y_j\|.$$
4 On the hardness of computing an average curve

Table 1 Overview of results. In these tables, \( n \) denotes the length of the input curves, \( m \) denotes the number of input curves and \( d \) denotes the ambient dimension of the curves.

(a) Results on exact computation.

| Problem | Result | Restrictions | Reference |
|---------|--------|--------------|-----------|
| 1-median, DTW\(_d\) | \( O(n^m + 1)^m \) | \( d = 1 \) | Brill et al. [4] |
| | \( O(mn^3) \) | Binary | |
| | NP-hard | | Bulteau et al. [8] |
| | W[1]-hard in \( m \) | \( p = q = 2 \) | Theorem 7 |
| 1-median, Fréchet | NP-hard | | Buchin et al. [6] |
| 1-center, discrete Fréchet | NP-hard | | Theorem 4 |
| \((k,\ell)\)-center, discrete Fréchet | \( O((mn)^{k\ell}k\ell m\log(mn)) \) | \( d \leq 2 \) | Theorem 13 |

(b) Approximation algorithms. (In stating the running times we assume \( k \), \( \ell \), and \( \varepsilon \) are constants independent of \( n \) and \( m \).)

| Problem | Result | Approximation factor | Restrictions | Reference |
|---------|--------|---------------------|--------------|-----------|
| \((k,\ell)\)-median, continuous Fréchet | \( \tilde{O}(nm) \) | \((1 + \varepsilon)\) | \( d = 1 \) | Driemel et al. [12] |
| \((k,\ell)\)-median, discrete Fréchet | \( \tilde{O}(nm) \) | 65 | | |
| | \( \tilde{O}(m^2(m + n)) \) | 12 | | Theorem 12 |
| | \( \tilde{O}(nm) \) | \((1 + \varepsilon)\) | \( k = 1 \) | Theorem 10 |
| | \( \tilde{O}(nm^{d\ell+1}) \) | \((1 + \varepsilon)\) | \( k > 1 \) | Theorem 12 |
| \((k,\ell)\)-center, discrete Fréchet | \( \tilde{O}(nm) \) | 3 | | Buchin et al. [6] |
| | \( \tilde{O}(nm) \) | \((1 + \varepsilon)\) | | Theorem 9 |

1.2 Results

We show that the average curve problem for discrete and continuous Fréchet distance in 1D is NP-complete, W[1]-hard when parametrized in the number of curves \( m \), and admits no \( f(m) \cdot n^{o(m)} \)-time algorithm unless ETH fails. In addition, we prove the same hardness results of the average curve problem for the \((p,q)\)-DTW distance for any \( p, q \in \mathbb{N} \).

This is an independent proof that is simpler and more general than the result by Bulteau et al. [8]. Their hardness result holds for the case of the \((2,2)\)-DTW distance, which is widely-used. Other common variants, covered by our proof, are \((1,1)\)-DTW, i.e., (non-squared) Euclidean distance and Manhattan distance in 1D [16], \((2,1)\)-DTW, and more generally \((p,1)\)-DTW [28, 29]. Note that, while we define \((p,1)\)-DTW in terms of the \( p \)th power of the Euclidean norm, our hardness results also apply to the \( p \)th power of the \( L_p \)-norm, since these norms are equal in 1D.

Brill et al. [4] asked whether their result can be extended to obtain a polynomial time algorithm when all curves are restricted to sets of 3 vertices. In our NP-hardness construction, both the input curves and the center curve use only vertices from \( \{-1, 0, 1\} \), so we answer this question in the negative, unless \( P = \text{NP} \). (The hardness construction by Bulteau et al. [8] was based on the \((2,2)\)-DTW distance, which is widely-used, and it is not clear how to extend their construction to accommodate the \((p,1)\)-DTW distance, which is known to be hard in 1D.)
uses a center curve that is not restricted to a bounded set of vertices, and therefore does not resolve this question.

Since our and other hardness results exclude efficient algorithms for the \((k, \ell)\)-center or -median clustering without further assumptions, we investigate other approaches with provable guarantees. In particular, we give a \((1 + \varepsilon)\)-approximation algorithm that runs in \(\tilde{O}(mn)\) time and a polynomial-time exact algorithm to solve the \((k, \ell)\)-center problem for the discrete Fréchet distance, when \(k, \ell\), and \(\varepsilon\) are fixed. For the \((k, \ell)\)-median problem under the discrete Fréchet distance, we give a \(12\)-approximation algorithm, and an \((1 + \varepsilon)\)-approximation algorithm that runs in polynomial time when \(k, \ell\), and \(\varepsilon\) are fixed.

Table 1 gives an overview of our results.

2 Hardness of the average curve problem for discrete and continuous Fréchet

In this section, we will show that the \(1\)-median problem (or average curve problem) is NP-hard for the discrete and continuous Fréchet distance. The average curve problem for the discrete Fréchet distance is as follows: given a set of curves \(G\) and an integer \(r\), determine whether there exists a center curve \(c\) such that \(\sum_{g \in G} d_{\text{DF}}(c, g) \leq r\). We will show that this problem is NP-hard. To find a reasonable algorithm, we can look at a parametrized version of the problem. A natural parameter is the number of input curves, which we will denote by \(m\). However, we will show that this parametrized problem is \(W[1]\)-hard, which rules out any \(f(m) \cdot n^{O(1)}\)-time algorithm, unless \(\text{FPT} = \text{W}[1]\). To achieve these reductions, we create a reduction from a variant of the shortest common supersequence (SCS) problem.

2.1 The FCCS problem

To show the hardness of the average curve problem for the Fréchet and DTW distance, we reduce from a variant of the Shortest Common Supersequence (SCS) problem, which we will call the Fixed Character Common Supersequence (FCCS) problem. If \(s\) is a string and \(x\) is a character, \(\#_{x}(s)\) denotes the number of occurrences of \(x\) in \(s\).

\[\text{Problem 3 (Shortest Common Supersequence (SCS))}\]

Given a set \(S\) of \(m\) strings with length at most \(n\) over the alphabet \(\Sigma\) and an integer \(t\), does there exists a string \(s^*\) of length \(t\) that is a supersequence of each string \(s \in S\)?

\[\text{Problem 4 (Fixed Character Common Supersequence (FCCS))}\]

Given a set \(S\) of \(m\) strings with length at most \(n\) over the alphabet \(\Sigma = \{A, B\}\) and \(i, j \in \mathbb{N}\), does there exists a string \(s^*\) with \(\#_{A}(s^*) = i\) and \(\#_{B}(s^*) = j\) that is a supersequence of each string \(s \in S\)?

The SCS problem with a binary alphabet is known to be NP-hard \cite{27} and \(\text{W}[1]\)-hard \cite{28}. The same holds for our variant:

\[\text{Lemma 1.}\] The FCCS problem is NP-hard. The FCCS problem with \(m\) as parameter is \(\text{W}[1]\)-hard. There exists no \(f(m) \cdot n^{O(m)}\)-time algorithm for FCCS unless \(\text{ETH}\) fails.

\[\text{Proof.}\] We reduce from SCS with the binary alphabet \(\{A, B\}\) to FCCS. Given an instance \((S, t)\) of SCS, construct \(S' = \{s + AB^{2t}A + c(s) \mid s \in S\}\), where \(c(s)\) denotes the string constructed by replacing all \(A\) characters in \(s\) by \(B\) and vice versa, and \(+\) denotes string concatenation. We reduce to the instance \((S', t + 2, 3t)\) of FCCS and claim that \((S, t)\) is a true instance of SCS if and only if \((S', t + 2, 3t)\) is a true instance of FCCS.
If \((S, t)\) is a true instance of SCS, then there exists a string \(q\) of length \(t\) that is a supersequence of each string in \(S\). Therefore, the string \(q' = q + AB^{2t}A + c(q)\) is a supersequence of all strings in \(S'\). Since \(\#_A(q') = 2 + \#_A(q + c(q)) = 2 + t\) and \(\#_B(q') = 2t + \#_B(q + c(q)) = 3t\), \((S', t + 2, 3t)\) is a true instance of FCCS.

If \((S', t + 2, 3t)\) is a true instance of FCCS, there is string \(q'\) with \(\#_A(q') = t + 2\) and \(\#_B(q') = 3t\) that is a supersequence of each string \(s' \in S'\). Consider a pair of strings \(s'_1 = s_1 + AB^{2t}A + c(s_1)\) and \(s'_2 = s_2 + AB^{2t}A + c(s_2)\) from \(S'\). If there is no matching such that the first character of the \(AB^{2t}A\) substring in \(s'_1\) is matched to the same character of \(q'\) as the first character of that substring in \(s'_1\), then \(q'\) is a supersequence of \(AB^{2t}A\) and so \(\#_B(q') > 3t\), a contradiction. By symmetry, the same holds for the last character of the substring \(AB^{2t}A\) and therefore \(q = q_1 + q_2 + q_3\), where \(q_1\) is a supersequence of \(S\), \(q_2\) is a supersequence of \(AB^{2t}A\) and \(q_3\) is a supersequence of \(\{c(s) \mid s \in S\}\). Note that \(c(q_3)\) is a supersequence of \(S\). Also, \(\#_A(q_1) + \#_A(c(q_3)) = \#_A(q) - \#_A(q_2) \leq t\) and \(\#_B(q_1) + \#_B(c(q_3)) = \#_B(q) - \#_B(q_2) \leq t\). So, \(|q_1| + |c(q_3)| \leq 2t\), which means that \(|q_1| \leq t\) or \(|c(q_3)| \leq t\) and thus \((S, t)\) is a true instance of SCS.

Note that this reduction is both a polynomial-time reduction and a parametrized reduction in the parameter \(m\). Since the SCS problem over the binary alphabet \(\{A, B\}\) is NP-hard \cite{27} and \(\mathcal{W}[1]\)-hard when parametrized with the number of strings \(m\) \cite{26}, the first two parts of the claim follow. The final part of the claim follows from the fact that this reduction together with the reduction from \cite{26}, we have a parametrized reduction from \textsc{Clique} with a linear bound on the parameter, so the final part of the claim follows \cite{11} Obs. 14.22.

2.2 Complexity of the average curve problem under the discrete and continuous Fréchet distance

We will show the hardness of computing the average curve under the discrete and continuous Fréchet distance via the following reduction from FCCS. Given an instance \((S, i, j)\) of FCCS, we construct a set of curves using the following vertices in \(\mathbb{R}\): \(g_a = -1\), \(g_b = 1\), \(g_A = -3\), and \(g_B = 3\). For each string \(s \in S\), we map each character to a subcurve in \(\mathbb{R}\):

\[
A \rightarrow (g_ag_b)^{i+j} g_A(g_ag_b)^{i+j} \quad B \rightarrow (g_ag_b)^{i+j} g_B(g_ag_b)^{i+j}.
\]

The curve \(\gamma(s)\) is constructed by concatenating the subcurves resulting from this mapping, \(G = \{\gamma(s) \mid s \in S\}\) denotes the set of these curves. Additionally, we use the curves

\[
A_i = g_b(g_A g_b)^i \quad B_j = g_a(g_B g_a)^j.
\]

We will call subcurves containing only \(g_A\) or \(g_B\) vertices \textit{letter gadgets} and subcurves containing only \(g_a\) or \(g_b\) vertices \textit{buffer gadgets}. Let \(R_{i,j} = \{A_i, B_j\}\). We reduce to the instance \((G \cup R_{i,j}, r)\) of the average curve problem, where \(r = |S| + 2\). We use the same construction for the discrete and continuous case.

For an example of this construction, take \(S = \{ABB, BBA, ABA\}, i = 2, j = 2\). Then \(ABBA\) is a supersequence of \(S\) with the correct number of characters. Note that the curve with vertices \(0g_A0g_B0g_B0g_A0\) has a (discrete) Fréchet distance of at most \(1\) to the curves in \(G \cup R_{i,j}\), see Figure 2, so the sum of those distances is at most \(|S| + 2 = r\).

\[\textbf{Lemma 2.} \textit{If \((S, i, j)\) is a true instance of FCCS, then \((G \cup R_{i,j}, r)\) is a true instance of the average curve problem for discrete and continuous Fréchet.}\]
Figure 2 Five curves from $G \cup R_{i,j}$ in the reduction for the Fréchet average curve problem and a center curve constructed from $ABBA$ (purple) as in Lemma 2. Matchings are indicated by dotted lines. Note that each of these matchings achieves a (discrete) Fréchet distance of 1.

Figure 3 Five curves from $G \cup R_{i,j}$ in the reduction for the DTW average curve problem and a center curve constructed from the string $ABBA$ (purple) as in Lemma 3. Fat horizontal lines indicate $\beta$ consecutive vertices. Vertices that match at distance 0 touch, vertices that match at distance 1 are indicated by dotted lines. The center has 1 mismatch with the first 3 curves and 2 with the final two, so the total cost here is $3 \cdot (1^p)^{\eta/p} + 2\alpha \cdot (2 \cdot 1^p)^{\eta/p} = 3 + 2\alpha \cdot 2^\eta$.
8 On the hardness of computing an average curve

Proof. We will show the proof for the discrete Fréchet distance. Since the discrete Fréchet distance is an upper bound of the continuous version, this proves the continuous case as well.

Since $(S, i, j)$ is a true instance of FCCS, there exists a common supersequence $s^*$ of $S$ with $|A(s^*)| = i$ and $|B(s^*)| = j$. Construct the curve $c$ of complexity $2|s^*| + 1$, given by

$$c_l = \begin{cases} 
0 & \text{if } l \text{ is odd} \\
-2 & \text{if } l \text{ is even and } s_{l/2} = A \\
2 & \text{if } l \text{ is even and } s_{l/2} = B 
\end{cases}$$

for each $l \in \{1, \ldots, 2|s^*| + 1\}$. Let $s \in S$, then note that the sequence of letter gadgets in $\gamma(s)$ is a subsequence of the letter gadgets in $c$, because $s$ is a subsequence of $s^*$. So, all letter gadgets in $\gamma(s)$ can be matched with a letter gadget in $c$, the remaining letter gadgets in $c$ with a buffer gadget in $\gamma(s)$ and all remaining buffer gadgets with another buffer gadget, such that $d_F(c, \gamma(s)) \leq 1$. For the matching with $A_i$, note that $c$ has exactly $i$ $g_A$ vertices, so these can be matched with the $i$ $g_A$ vertices in $A_i$. All other vertices in $c$ have distance 1 to the remaining buffer gadgets in $A_i$, so $d_F(c, A_i) \leq 1$. Analogously, $d_F(c, B_j) \leq 1$. So, we get $\sum_{g \in G \cup R_i} d_F(c, g) = \sum_{g \in S} d_F(c, \gamma(s)) + d_F(c, A_i) + d_F(c, B_j) \leq |S| + 2 = r$, and $(G \cup R_i, r)$ is a true instance of average curve problem for discrete Fréchet.

Lemma 3. If $(G \cup R_{i,j})$ is a true instance of the average curve problem for discrete and continuous Fréchet, then $(S, i, j)$ is a true instance of FCCS.

Proof. We will show the proof for the continuous Fréchet distance. Since the continuous Fréchet distance is a lower bound of the discrete version, this proves the discrete case as well.

We call the interval of points $p$ on a subcurve $g_B A g_B$ with $p < -1$ an $A$-peak, and the interval of points $p$ on a subcurve $g_A B g_A$ with $p > 1$ a $B$-peak. A curve $\gamma(s)$ has exactly one peak for every letter in $s$.

Since $(G \cup R_{i,j}, r)$ is a true instance of the average curve problem for continuous Fréchet, there exists a curve $c^*$ such that $\sum_{g \in G \cup R_{i,j}} d_F(c^*, g) \leq r = |S| + 2$. We start by deriving bounds for the distance between $c^*$ and the individual curves in $G \cup R_{i,j}$.

Claim. $d_F(\gamma(s), \gamma(s')) \geq 2$ for all $s, s' \in S$ such that $s \neq s'$.

Proof. If a letter vertex $p$ on $\gamma(s)$ is matched with a point $p'$ that does not lie on a peak of the same letter in $\gamma(s')$, then $|p - p'| \geq 2$ and so $d_F(\gamma(s), \gamma(s')) \geq 2$. By symmetry, the same holds if we exchange $s$ and $s'$.

Otherwise, each letter vertex can be matched only with points on a peak of the same letter. Let $k$ be the first index such that $s[k] \neq s'[k]$. Then, the $k$-th letter vertex of $\gamma(s)$ cannot be matched to any point on the $k$-th peak of $\gamma(s')$ and must be matched to a point on another peak; the same holds with $s$ and $s'$ exchanged. It is not possible that on both curves the $k$-th letter vertex is matched with a peak of index larger than $k$, since the matching is monotone. So, one of the curves has its $k$-th letter vertex matched with a point on a peak of index smaller than $k$, we assume w.l.o.g. that this curve is $s$.

By monotonicity, the first $k$ letter vertices of $s$ are matched to the first $k - 1$ peaks of $s'$, so there are two letter vertices on $s$ that are both matched with a point on the same peak on $s'$. The interval between those two points on this peak on $s'$ must be matched with the interval between the letter vertices on $s$, so all points in the buffer gadget between the letter vertices on $s$ are matched to some point on the peak on $s'$. But then there is either a point on an $A$-peak matched to $g_A$ or a point on a $B$-peak matched to $g_B$, which in both cases has distance at least 2, so $d_F(\gamma(s), \gamma(s')) \geq 2$. 

<
\textbf{Proof.} Using the previous claim and the triangle inequality, we have
\[
d_F(c^*, \gamma(s_k)) + d_F(c^*, \gamma(s_{k+1})) \geq d_F(\gamma(s_k), \gamma(s_{k+1}) \geq 2
\]
for all \(k \in \{1, \ldots, m-1\}\) and \(d_F(c, \gamma(s_m)) + d_F(c, \gamma(s_1)) \geq 2\). The summation of these \(m\) inequalities has each \(s_k\) exactly twice on the lefthand side, so \(\sum_{k=1}^m 2d_F(c^*, \gamma(s_k)) = 2m\), hence \(\sum_{k=1}^m d_F(c^*, \gamma(s_k)) \geq m = |S|\). So, \(d_F(c^*, A_i) + d_F(c^*, B_j) \leq r - \sum_{k=1}^m d_F(c^*, \gamma(s_k)) \leq 2\).

\textbf{Claim.} \(d_F(c^*, A_i) \geq 1\) and \(d_F(c^*, B_j) \geq 1\).

\textbf{Proof.} Suppose \(d_F(c^*, A_i) < 1\). Then, all points \(p\) on \(c^*\) are matched to some point in \([-3, 1]\) with distance < 1, which means \(|p-g_0| > 1\). We can assume that each string in \(S\) contains at least one \(B\) character (if there is a string \(s\) with only \(A\) characters, any supersequence with \(i\) \(A\)-characters is a supersequence of \(s\) when \(|s| \leq i\) and none when \(|s| > i\), so we can remove such trivial strings from the instance and check if the instance is trivially false). Therefore, \(d_F(c^*, \gamma(s)) > 1\) for any \(s \in S\).

Since \(|g_a - g_b| = 2\), we have \(d_F(A_i, B_j) \geq 2\), so \(d_F(c^*, A_i) + d_F(c^*, B_j) \geq d_F(A_i, B_j) \geq 2\). But then \(r \geq \sum_{g \in G \cup R_{i,j}} d_F(c^*, g) > |S| + 2 = r\), a contradiction, so \(d_F(c^*, A_i) \geq 1\). The proof of \(d_F(c^*, B_j) \geq 1\) is analogous.

\textbf{Claim.} \(d_F(c^*, g) = 1\) for all \(g \in G \cup R_{i,j}\).

\textbf{Proof.} The last two claims together imply \(d_F(c^*, A_i) = d_F(c^*, B_j) = 1\). This means that for each point \(p\) on \(c^*, |p| \leq 2\) (otherwise, \(p\) has distance > 1 to all points on \(A_i\) or all points on \(B_j\)), so \(d_F(c^*, \gamma(s)) \geq 1\) for all \(s \in S\), since we can assume \(s\) contains at least one \(A\) and \(B\) character. Therefore, \(d_F(\gamma(s), c^*) \leq r - d_F(A_i, c^*) - d_F(B_j, c^*) - \sum_{s' \in S \setminus \{s\}} d_F(\gamma(s'), c^*) \leq |S| - (|S| - 1) = 1\) for all \(s \in S\).

Now we have shown that any center curve that achieves a cost of \(|S| + 2\) for the constructed \(k\)-median instance needs to have Fréchet distance equal to 1 to all curves in this instance. It remains to show that such a center curve encodes a solution to the initial FCCS instance. Note that such a center curve is also a solution to the 1-center problem for this set of curves. We can now apply the proof of Lemma 33 from [6], where the same gadgets were used in the reduction to the 1-center problem.

\textbf{Theorem 4.} \textit{The average curve problem for discrete and continuous Fréchet distance is NP-hard. When parametrized in the number of input curves \(m\), this problem is W[1]-hard. There exists no \(f(m) \cdot n^{o(m)}\) time algorithm for this problem unless ETH fails.}

\textbf{Proof.} By Lemmas 2 and 3, we have a valid reduction from FCCS to the average curve problem. Since this reduction runs in polynomial time and FCCS is NP-hard (Lemma 1), the average curve problem for discrete and continuous Fréchet is NP-hard. Note that the number of curves in the reduced average curve instance is \(k + 2\), where \(k\) is the number of input sequences of the FCCS instance. So, together with the reduction from Lemma 1 this reduction is also a parametrized reduction from CLIQUE with a linear bound on the parameter to the average curve problem for discrete and continuous Fréchet with the number of curves as a parameter, which implies the remainder of the theorem.
3 Hardness of the average curve problem for \((p, q)\)-DTW

We will show that the average curve problem under the \((p, q)\)-DTW distance is NP-hard for all \(p, q \in \mathbb{N}\). This generalises the result of [8], who use different methods to achieve the same hardness results for the \((2, 2)\)-DTW average curve problem only. We again reduce from FCCS instance \((S, i, j)\). Given a string \(s \in S\) over the binary alphabet \(\{A, B\}\), we map each character to a subcurve in \(\mathbb{R}\):

\[
A \rightarrow g_0^\beta g_a^\beta g_0^\beta \quad B \rightarrow g_0^\beta g_b^\beta g_0^\beta,
\]

where \(g_0 = 0, g_a = -1, g_b = 1\) as before and \(\beta\) is a large constant that will be determined later. The curve \(\gamma(s)\) is constructed by concatenating these subcurves and \(G = \{\gamma(s) | s \in S\}\). We additionally use the curves

\[
A_i = g_0^\beta (g_a^\beta g_0^\beta)^i \quad B_j = g_0^\beta (g_b^\beta g_0^\beta)^j.
\]

Call any subcurve consisting of \(g_a\) or \(g_b\) vertices a letter gadget and any subcurve consisting of \(g_0\) a buffer gadget. Let \(R_{i,j}\) contain curves \(A_i\) and \(B_j\), both with multiplicity \(\alpha\). We reduce to the instance \((G \cup R_{i,j}, r)\) of \((p, q)\)-DTW average curve, where \(r = \sum_{s \in S} (i + j - |s|)^{q/p} + \alpha (\delta^{p/q} + \epsilon^{q/p}), \beta = \lceil r/\varepsilon \rceil + 1, \alpha = |S|\) and \(\varepsilon = 1 - (1 - \min_{s \in S} (i, j)^{q/p} - \delta^{q/p})^{1/q}\). See Figure 4 for an example of this construction with \(S = \{ABB, BBA, ABA\}\) and \(i = j = 2\).

The following definitions are used to prove Lemma 5. Take a vertex \(p\) on some center curve \(c^*\). If \(|p - g_a| < \varepsilon\), we call \(p\) an \(A\)-signal vertex. If \(|p - g_b| < \varepsilon\) we call \(p\) a \(B\)-signal vertex. If \(p\) is not a signal vertex, then we call \(p\) a buffer vertex. Note that \(\varepsilon\) is chosen small enough such that no vertex is both an \(A\)- and \(B\)-signal vertex. We will show that the sequence of signal vertices in the curve satisfying \((G \cup R_{i,j}, r)\) is a supersequence satisfying \((S, i, j)\).

\textbf{Lemma 5.} If \((S, i, j)\) is a true instance of FCCS, then \((G \cup R_{i,j}, r)\) is a true instance of \((p, q)\)-DTW average curve.

\textbf{Proof.} If \((S, i, j)\) is a true instance of FCCS, then there exists a string \(s^*\) that is a supersequence of \(S\), with \(#A(s^*) = i\) and \(#B(s^*) = j\). Construct the curve \(c\) of length \(2(i + j) + 1\):

\[
c_l = \begin{cases} 
0 & \text{if } l \text{ is odd} \\
g_a & \text{if } l \text{ is even and } s_{i+j}^* = A \\
g_b & \text{if } l \text{ is even and } s_{i+j}^* = B
\end{cases}
\]

for each \(l \in \{1, \ldots, 2(i + j) + 1\}\). Analogously to Lemma 2, we can match the letter gadgets from \(\gamma(s)\) to \(g_a\) or \(g_b\) in \(c\) as \(s^*\) is a supersequence of \(s\), the letter gadgets of \(A_i, B_j\) to \(g_A, g_B\) in \(c\) as the number of curves match, and \(g_0\) vertices to buffer gadgets. This gives a matching such that \(\sum_{g \in G \cup R_{i,j}} \text{DTW}_p^n (c, g) \leq r\).

\textbf{Lemma 6.} If \((G \cup R_{i,j}, r)\) is a true instance of \((p, q)\)-DTW average curve, then \((S, i, j)\) is a true instance of FCCS.

\textbf{Proof.} If \((G \cup R_{i,j}, r)\) is a true instance of \((p, q)\)-DTW average curve, then there exists a curve \(c^*\) such that \(\sum_{g \in G \cup R_{i,j}} \text{DTW}_p^n (c^*, g) \leq r\). Take a curve \(g \in G \cup R_{i,j}\). First note that there is at least one signal vertex in \(c^*\) matched to each letter gadget in \(g\); otherwise, matching all \(\beta\) vertices in the gadget costs at least \(\varepsilon^q \cdot \beta = \varepsilon^q \cdot (r/\varepsilon^q + 1) > r\), which contradicts the choice of
There exists no $\frac{A-signal vertices and at least $\varepsilon$.

Proof. By Lemmas 5 and 6, we have a valid reduction from FCCS to the average curve problem. Since this reduction runs in polynomial time and FCCS is NP-hard (Lemma 1), the average curve problem for discrete and continuous Fréchet is NP-hard. Since the reduction runs in polynomial time (note that $1/\varepsilon$ can be bounded by a polynomial function in $n$, since...
Given an approximate \((1, ℓ)-center\) curve (red) for a set of curves (dashed), the vertices of the optimal center curve (black) will be close to the hypercube grids around the vertices of the approximate center.

\(p, q\) are constants, so \(β\) can be polynomially bounded) and the number of input curves is bounded by a linear function in \(|S|\), the claim follows.

4 Algorithms for \((k, ℓ)-center\) and -median curve clustering

4.1 \((1 + \varepsilon)\)-approximation for \((k, ℓ)-center\) clustering for discrete Frechét in \(\mathbb{R}^d\)

In this section, we develop a \((1 + \varepsilon)\)-approximation algorithm for the \((k, ℓ)-center\) problem under the discrete Frechét distance that runs in \(O(mn \log(n))\) time for fixed \(k, ℓ, ε\). In this algorithm, we use hypercube grids \(L_v(a, b)\) around a vertex \(v\) of width \(a\) and resolution \(b\): take the axis-parallel \(d\)-dimensional hypercube centered at \(v\) of side-length \(a\). Divide this hypercube into smaller hypercubes of side-length at most \(b\). The grid \(L_v(a, b)\) is the set of all vertices of the smaller hypercubes that intersect the ball of diameter \(a\) around \(v\). See figure 4 for an example.

The algorithm is as follows: First, we compute a set of curves \(C = \{c_1, \ldots, c_k\}\) that forms a 3-approximation for the \((k, ℓ)-center\) problem, using the algorithm by Buchin et al. \[6\]. Let \(Δ\) be the cost of \(C\).

Let \(V\) be the union of the hypercube grids \(L_v(4Δ, \frac{2Δ}{\sqrt{d}})\) over all vertices \(v\) of curves in \(C\). For every set of \(k\) center curves with complexity \(ℓ\) using only vertices from \(V\), compute the clustering and cost as centers for \(G\), and return the set with minimal cost.

In order to show this algorithm gives an \((1 + \varepsilon)\)-approximation, we use the following lemma to show that there is a set of \(k\) center curves that is close enough to the optimal solution:

\textbf{Lemma 8.} Let \(k, ℓ, ε, δ \in \mathbb{N}\) and \(X > 0\). Suppose there are two sets \(C = \{c_1, \ldots, c_k\}\) and \(C^* = \{c_1^*, \ldots, c_k^*\}\), both containing \(k\) curves in \(\mathbb{R}^d\) of complexity \(ℓ\). Additionally, suppose that for all curves \(c^* \in C^*\), there exists a curve \(c \in C\) such that \(d_{AF}(c, c^*) ≤ δ\). Let \(V = \{L_v(2δ, \frac{2X}{\sqrt{d}}) \mid v\ \text{is a vertex of a curve in } C\}\). Then there is a set of curves \(\bar{C} = \{\bar{c}_1, \ldots, \bar{c}_k\}\), using only vertices from \(V\), such that \(d_{AF}(\bar{c}_i^*, \bar{c}_i) ≤ X, |\bar{c}_i| = ℓ\), for all \(1 ≤ i ≤ k\).

\textbf{Proof.} Let \(v\) be a vertex of a curve in \(C\), and let \(p\) be a point such that \(\|p - v\| ≤ δ\). Then \(p\) lies inside one of the small hypercubes and so there is a vertex \(p' \in L_v(2δ, \frac{2X}{\sqrt{d}})\) (a vertex of that small hypercube) such that \(\|p - p'\| ≤ \frac{1}{2} X, \frac{2X}{\sqrt{d}} = X\).
Let \( c^* \in C^* \). There exists a curve \( c \in C \) with \( d_{DF}(c, c^*) \leq \delta \), which means that each vertex \( u \) of \( c^* \) has distance at most \( \delta \) to some vertex \( v \) of \( c \). So, there exists a vertex \( v' \in L_v(2\delta, 2\frac{\Delta}{\epsilon}) \) such that \( |u - v'| \leq X \). Construct the curve \( \tilde{c} \) by connecting all such vertices \( v' \) by line segments. By construction, \( d_{DF}(\tilde{c}, c^*) \leq \delta \), \( |\tilde{c}| = \ell \), and all vertices of \( \tilde{c} \) are in \( V \). So, we can take \( \tilde{C} = \{ \tilde{c} \mid c^* \in C^* \} \).

\[ \textbf{Theorem 9.} \quad \text{Given} \ m \text{ input curves in } \mathbb{R}^d, \text{ each of complexity at most } n, \text{ and positive integers } k, \ell \text{ and some } 0 < \epsilon \leq 1, \text{ we can compute an } (1 + \epsilon) \text{-approximation to the } (k, \ell) \text{-center problem for the discrete Fréchet distance in } O \left( (Ck\ell)^{k\ell} + \log((\ell + n)) \cdot k\ell \cdot mn \right) \text{ time, with } C = \left( \frac{6\sqrt{\ell}}{\epsilon} + 1 \right)^d. \]

\[ \text{Proof.} \quad \text{We first show that the algorithm above achieves this approximation ratio. Let } C^* \text{ be an optimal optimal solution for the } (k, \ell) \text{-center problem, and } O \text{ its cost. Let } c^* \in C^*, \text{ then there is a curve } g \in G \text{ such that that } d_{DF}(c^*, g) \leq O \text{ (assuming without loss of generality that its cluster is non-empty). Since the solution } C \text{ has cost } \Delta, \text{ there is a } c \in C \text{ such that } d_{DF}(c, g) \leq \Delta. \text{ So, } d_{DF}(c, c^*) \leq d_{DF}(c, g) + d_{DF}(g, c^*) \leq 2\Delta, \text{ and by Lemma 8 with } \delta = 2\Delta \text{ and } X = \epsilon \cdot \Delta/3 \leq O, \text{ there is a solution } \tilde{C} \text{ with the properties in the Lemma. Since for any } g \in G, \text{ there is a curve } c^* \in C^* \text{ such that } d_{DF}(g, c^*) \leq O, \text{ there is a } \tilde{c} \in \tilde{C} \text{ such that } d_{DF}(g, \tilde{c}) \leq d_{DF}(g, c^*) + d_{DF}(\tilde{c}, c^*) \leq (1 + \epsilon)O. \text{ Since the algorithm returns the best solution using only from } V, \text{ it returns a solution of cost at most that of } \tilde{C}, \text{ and is therefore an } 1 + \epsilon \text{-approximation.}

\text{For the running time, computing the } 3 \text{-approximation } C \text{ takes } O(k\ell mn \log((\ell + n))) \text{ time} \left[ 3 \right]. \text{ A grid } L_v(a, b) \text{ has at most } \left( \left\lceil \frac{X}{\epsilon} \right\rceil + 1 \right)^d \text{ vertices and the curves in } C \text{ have at most } k\ell \text{ vertices, so } |V| \leq k\ell \left( \left\lceil \frac{6\sqrt{\ell}}{\epsilon} \right\rceil + 1 \right)^d. \text{ There are } O(|V|^{k\ell}) \text{ solutions using only vertices from } V, \text{ and we can test each solution in } O(k\ell mn) \text{ time: computing the discrete Fréchet distance between an input curve and a center curve takes } O(\ell n) \text{ time using dynamic programming, which we do for all } km \text{ pairs of input and center curves. In total, we get a running time of } O \left( (|V|^{k\ell} + \log((\ell + n))) \cdot k\ell mn \right). \]
On the hardness of computing an average curve

from [12] for 1D curves, except we use different subroutines and derive a tighter approximation bound. We first introduce some techniques we will use to get a 12-approximation. Given a polygonal curve \( \gamma \), a simplification is a polygonal curve that is similar to \( \gamma \), but has only a few vertices. Specifically, a \textit{minimum error }\ell\text{-simplification }\bar{\gamma}\text{ of a curve }\gamma\text{ is a curve of complexity at most }\ell\text{ that has a minimum distance to }\gamma\text{ among all curves with complexity at most }\ell\text{. We can compute a minimum error }\ell\text{-simplification under the discrete Fréchet distance for a curve }\gamma\text{ of complexity }n\text{ in }O(n\ell\log n \log(n/\ell))\text{ time [3].}

The 12-approximation algorithms goes as follows: First, compute a minimum error \( \ell\)-simplification \( \bar{g} \) for each input curve \( g \) and let \( \mathcal{G} \) be the set of all simplified curves. Then, compute a 4-approximation for the \( (k, \ell)\text{-median problem with } F = \mathcal{G} \text{ and } C = G, \text{ using the algorithm by Jain et al. [20]. This yields a 12-approximation:}

\[ \text{Lemma 11. Given } m \text{ input curves in } \mathbb{R}^d, \text{ each of complexity at most } n, \text{ and positive integers } k, \ell, \text{ we can compute a } 42\text{-approximation for the } (k, \ell)\text{-median problem for the discrete Fréchet distance in } O(m^3 \log n \log(n/\ell)) \text{ time.} \]

\[ \text{Proof. The proof is similar to Theorem 10, but now simplifications are clustered instead of the original curves. We first show the approximation ratio. Given a cluster } G^*_i \subset \mathcal{G} \text{ from the optimal clustering with center } c^*_i, \text{ let } \bar{c}_i \text{ be the simplification of a curve } g \text{ in this cluster such that } d_{df}(\bar{c}_i, c^*_i) \text{ is minimal. The curves } \mathcal{C} = \{\bar{c}_1, \ldots, \bar{c}_k\} \text{ are a } 4\text{-approximation for the } (k, \ell)\text{-median problem: we have } \sum_{g \in G} \min_{i=1}^{k} d_{df}(\bar{g}_i, \bar{c}_i) \leq \sum_{i=1}^{k} \sum_{g \in G^*_i} d_{df}(g, c^*_i) \leq \sum_{i=1}^{k} \sum_{g \in G^*_i} 2d_{df}(\bar{g}_i, \bar{c}_i) \leq 2 \sum_{i=1}^{k} \sum_{g \in G^*_i} d_{df}(\bar{g}_i, g) + d_{df}(g, c^*_i) \leq 2 \sum_{i=1}^{k} \sum_{g \in G^*_i} 2d_{df}(g, c^*_i) = 4O, \text{ where } d_{df}(\bar{c}_i, c^*_i) \leq d_{df}(\bar{g}_i, c^*_i) \text{ by definition of } \bar{c}_i \text{ and } d_{df}(\bar{g}_i, g) \leq d_{df}(g, c^*_i) \text{ because } |c^*_i| = \ell \text{ and } \bar{g} \text{ is a minimum error } \ell\text{-simplification of } g. \text{ Since we compute a } 10.5\text{-approximation to the problem for which } \mathcal{G} \text{ is a solution, the approximation ratio } 10.5 \cdot 4 = 42. \]
Computing the simplification of all curves in \( G \) takes \( O(mn\ell \log n \log(n/\ell)) \) time. The algorithm by Chen \[1\] takes \( O(mk + k^2 \log^5 m) \) time, so it uses at most that number of distance computations between curves in \( G \), which take \( O(\ell^2) \) time each.

We now use the 42-approximation algorithm to compute an \((1 + \varepsilon)\)-approximation. Let \( \mathcal{C} = \{c_1, \ldots, c_k\} \) be the solution given by the approximation algorithm above, and \( \Delta \) its cost. If \( k = 1 \), let \( V \) be the union of the hypercube grids \( L_v(4\Delta/m, \frac{2\Delta}{3m\sqrt{d}}) \) over all vertices \( v \) of curves in \( \mathcal{C} \). If \( k > 1 \), let \( V \) be the union of the grids \( L_v(4\Delta, \frac{2\Delta}{21m\sqrt{d}}) \) over the same vertices, instead. For every set of \( k \) center curves with complexity \( \ell \) using only vertices from \( V \), compute the clustering and cost (using the median objective) as centers for \( \mathcal{C} \), and return the set with minimal cost.

**Theorem 12.** Given \( m \) input curves in \( \mathbb{R}^d \), each of complexity at most \( n \), and positive integers \( k, \ell \) and some \( 0 < \varepsilon \leq 1 \), we can compute an \((1 + \varepsilon)\)-approximation to the \((k, \ell)\)-center problem for the discrete Fréchet distance in \( O \left( mn\ell \left( (\mathcal{C} \ell)^\ell + \log n \log(n/\ell) \right) \right) \) time when \( k = 1 \) with \( \mathcal{C} = \left( \frac{84\sqrt{d}}{\varepsilon} \right)^d \). When \( k > 1 \), we require \( O \left( \left( Ck\ell + 1 \right) \cdot \ell \cdot k^{\ell+1} \cdot m + mn\ell \log n \log(n/\ell) + \ell^2 (mk + k^2 \log^5 m) \right) \) time.

**Proof.** We first show the approximation ratio. Let \( \mathcal{C}^* = \{c^*_1, \ldots, c^*_k\} \) be an optimal solution for the \((k, \ell)\)-median problem, \( G^*_\ell \subset G \) the cluster induced by the center \( c^*_i \), and \( O \) the total cost of this solution. Let \( \bar{C} = \{c_1, \ldots, c_k\} \) be a set of curves with complexity at most \( \ell \) such that for all \( 1 \leq i \leq k \), there is a curve \( \bar{c}_i \subset \bar{C} \) with \( d_{df}(c^*_i, \bar{c}_i) \leq \varepsilon O/m \). Since \( \sum_{g \in G^*_\ell} \min_{1 \leq i \leq k} d_{df}(g, \bar{c}_i) \leq \sum_{i=1}^k \sum_{g \in G^*_\ell} d_{df}(g, \bar{c}_i) \leq \sum_{i=1}^k \sum_{g \in G^*_\ell} d_{df}(g, c^*_i) + d_{df}(c^*_i, \bar{c}_i) \leq \sum_{g \in G} \min_{1 \leq i \leq k} d_{df}(g, c^*_i) + \varepsilon O/m = (1 + \varepsilon)O \), the set \( \bar{C} \) is an \((1 + \varepsilon)\)-approximation. We will show that there is such a set that uses only vertices of \( V \).

If \( k = 1 \), then \( d_{df}(c_1, c^*_1) = \frac{1}{m} \sum_{g \in G} d_{df}(c_1, c^*_1) \leq \frac{1}{m} \sum_{g \in G} d_{df}(c_1, g) + d_{df}(g, c^*_1) \leq (\Delta + O)/m \leq 2\Delta/m \). Applying Lemma 8 with \( \delta = 2\Delta/m \) and \( X = \varepsilon \Delta/(12m) \leq \varepsilon O/m \), there is a \((1 + \varepsilon)\)-approximation using only vertices of \( V \).

Otherwise, if \( k > 1 \), then for each \( c_i^* \) there is a \( c_j \) such that the clusters of these centers share some curve \( g \in G \). So, \( d_{df}(c^*_i, c_j) \leq d_{df}(c^*_i, g) + d_{df}(g, c_j) \leq O + \Delta \leq 2\Delta \). Applying Lemma 8 with \( \delta = 2\Delta \) and \( X = \varepsilon \Delta/(12m) \leq \varepsilon O/m \), there is a \((1 + \varepsilon)\)-approximation using only vertices of \( V \).

For the running time, we have \(|V| \leq k\ell (\lceil \frac{d}{b} \rceil + 1)^d \) when we use grids with width \( a \) and resolution \( b \). If \( k = 1 \), \( \frac{a}{b} = \frac{4\Delta/m}{\varepsilon \Delta/(21m\sqrt{d})} = \frac{84\sqrt{d}}{\varepsilon} \). If \( k > 1 \), \( \frac{a}{b} = \frac{84m\sqrt{d}}{\varepsilon} \). The rest of the analysis is similar to that in Theorem 8.

The additional \( m^{\ell+1} \) factor in our algorithm when \( k > 1 \) is due to the following: We need to find curves with distance at most \( \varepsilon \Delta/(42m) \) to an optimal solution in order to get an \((1 + \varepsilon)\)-approximation. When \( k = 1 \), the number of grid points is still independent of \( m \), since the distance of the curve in \( \mathcal{C} \) to an optimal curve is \( O(\Delta/m) \). However, when \( k > 1 \), it is possible that all optimal solutions have a cluster of constant size. Then the center curve of that cluster can have distance \( \Omega(\Delta) \) to all curves in \( \mathcal{C} \).

### 4.3 Exact algorithm for \((k, \ell)\)-center under discrete Fréchet in 2D

We give an algorithm that solves the \((k, \ell)\)-center problem for the discrete Fréchet distance in 2D in polynomial time for fixed \( k \) and \( \ell \). We first show how to solve the decision version of this problem and use it as a subroutine to solve the optimisation problem.
Figure 5 An example configuration of $G = (V, E)$. Crosses indicate the vertices from the curves in $G$, dots indicate vertices from $V$ and all bounded faces are numbered. The maximal intersection regions are the faces $f_1$ and $f_9$ and the vertex $v$ (in red). Note that while all arcs on the boundary of $f_2$ are convex for that face, $f_2$ is not maximal, since its boundary intersects the boundary of $f_3$ only at vertex $v$.

The main idea of the algorithm for the decision version is based on the following observation: for a given $r$, we have $\min_{c \in C} d_{DF}(c, g) \leq r$ for all $g \in G$ if and only if each vertex $p$ of a curve in $C$ lies in the intersection of the disks of radius $r$ around all vertices $q$ from curves in $G$ that $p$ is matched with. Furthermore, it does not matter where the vertex $p$ lies within the intersection region. This means we can select a vertex for each maximal overlapping region (i.e. each region such that the set of disks intersecting the region is not contained in another region) and exhaustively test all sets with $k$ curves of $\ell$ vertices that can be constructed by using only the selected vertices to determine if there exists a set of curves $C$ such that $\min_{c \in C} d_{DF}(c, g) \leq r$ for all $g \in G$.

To find all maximal intersection regions, we first compute the planar graph $G = (V, E)$, where $V$ is the set of all intersection points between boundaries of disks centred around a vertex from our input curves with radius $r$ and $E$ is the set of arcs on the boundary of those disks ending at two intersection points. This graph has $O((nm)^2)$ vertices and arcs and can be computed in $O((nm)^2)$ time [9], see Figure 5 for an example.

By traversing the intersection points and arcs on the boundary, we can find the at most $O((nm)^2)$ maximal intersection regions. So, we test $O((nm)^{2k\ell})$ sets of center curves, for which we can test whether an input curve has discrete Fréchet distance less than $r$ to a single curve among the $k$ center curves in $O(m\ell)$. This means the algorithm for the decision version takes $O((nm)^{3k\ell}k\ell m)$ time.

To find a minimum $r$ such that a $(k, \ell)$-center exists, note that we only have to consider the decision problem for those $r$ where the topology of the intersection regions in $G$ is different. If we start with $r = 0$ and gradually increase it, the topology of $G$ changes only when a new maximal intersection is created, which then consists of exactly one point $p$. This means that there is a subset of our disks such that point $p$ is the earliest point where all disks have a non-empty intersection. So, $p$ must be the center of the minimum enclosing disk for this subset of disks. Since a minimum enclosing disk is determined by at most 3 points, there can be at most one unique point for every triple in set of vertices of the input curves which give at most $O((mn)^3)$ distinct values of $r$ where the topology of $G$ changes. By performing
a binary search on these values, we can find the optimal value in $O(\log(mn))$ calls to the algorithm for the decision version and we get the following result:

\begin{itemize}
  \item \textbf{Theorem 13.} Given a set of $n$ curves $G$ in the plane with at most $m$ vertices each, we can find a solution to the $(k, \ell)$-center problem for the discrete Fréchet distance in $O((mn)^{2k\ell}k\ell m\log(mn))$ time.
\end{itemize}

\section{Conclusion}

In this paper, we have shown that the 1-median problem is computationally hard under the discrete Fréchet, continuous Fréchet, and DTW distance. A natural question is whether this problem is hard to approximate. Efficient constant factor approximation algorithms are known for the Fréchet distance (see Section 4.2), but not for DTW. If we extend our analysis in Lemma 3 to a solution $c^*$ with cost $(1 + \varepsilon)r$ for some $\varepsilon > 0$, we can show $d_{DF}(c^*, g) \leq 1 + O(\varepsilon m)$ for all input curves $g$ (where the constant is independent of other input parameters). Together with the approximation lower bound of 2 for 1-center under continuous Fréchet distance [30], this implies a lower bound of $1 + \Omega(\frac{1}{m})$ on the approximation factor for 1-median. If we do the same for Lemma 6, we get that it is hard to approximate 1-median under $(p, q)$-DTW for any factor $< 1 + 2(1 + \frac{1}{\min(i,j)})^{q/p - 1}$. So, it remains an open problem to find a constant lower bound for approximating 1-median for this distance measures.

On the positive side, we have given $(1 + \varepsilon)$-approximation algorithms for $(k, \ell)$-center and $(k, \ell)$-median problems under discrete Fréchet in Euclidean space and an exact algorithm for the $(k, \ell)$-center problem under discrete Fréchet in 2D that all run in polynomial time for fixed $k, \ell, \varepsilon$. It would be interesting to see if these algorithms can be adapted to the DTW or continuous Fréchet settings. Our approximation algorithms rely on the fact that good approximations have small distance to some optimal solution and that we can search a bounded space (the set of balls surrounding the vertices) for better approximations. The first property does not hold for DTW, since it is non-metric and the second property does not hold for continuous Fréchet, since the vertices of a curve with small continuous Fréchet distance do not have to be near the vertices of the other curve. The latter property is also crucial for the exact algorithm.

\section*{References}

1 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. \textit{International Journal of Computational Geometry & Applications}, 5:75–91, 1995. \texttt{doi:10.1142/9789814313939_0002}

2 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. \textit{SIAM Journal of Computing}, 33(3):544–562, 2004. \texttt{doi:10.1137/S0097539702418402}

3 Sergey Bereg, Minghui Jiang, Wencheng Wang, Boting Yang, and Binhai Zhu. Simplifying 3D polygonal chains under the discrete Fréchet distance. In \textit{Proceedings of the 8th Latin American Conference on Theoretical Informatics}, pages 630–641, 2008.

4 Markus Brill, Till Fluschnik, Vincent Froese, Brijnesh Jain, Rolf Niedermeier, and David Schultz. Exact mean computation in dynamic time warping spaces. \textit{Data Mining and Knowledge Discovery}, 33(1):252–291, 2019. \texttt{doi:10.1007/s10618-018-0604-8}

5 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitisyna, and Maarten Löffler. Approximating $(k, \ell)$-center clustering for curves. \textit{CoRR}, abs/1805.01547, 2018. URL: \texttt{http://arxiv.org/abs/1805.01547} \texttt{arXiv:1805.01547}
On the hardness of computing an average curve

6 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms, pages 2922–2938, 2019. doi:10.1137/1.9781611975482.181

7 Kevin Buchin, Anne Driemel, Natasa van de L’Isle, and André Nusser. Klcluster: Center-based clustering of trajectories. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019, Chicago, IL, USA, November 5-8, 2019, pages 496–499, 2019. URL: https://doi.org/10.1145/3347146.3359111

8 Laurent Bulteau, Vincent Froese, and Rolf Niedermeier. Tight hardness results for consensus problems on circular strings and time series. arXiv preprint arXiv:1804.02854, 2018. URL: http://arxiv.org/abs/1804.02854

9 Bernard Marie Chazelle and Der-Tsai Lee. On a circle placement problem. Computing, 36(1-2):1–16, 1986.

10 Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. SIAM J. Comput., 39(3):923–947, 2009. URL: https://doi.org/10.1137/070699007

11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Parameterized Algorithms. Springer, 2015. URL: https://doi.org/10.1007/978-3-319-21275-3

12 Anne Driemel, Amer Krivošija, and Christian Sohler. Clustering time series under the Fréchet distance. In Proceedings of the 37th ACM-SIAM Symposium on Discrete Algorithms, pages 766–785. Society for Industrial and Applied Mathematics, 2016.

13 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.

14 Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In Proceedings of the twentieth annual ACM symposium on Theory of computing, pages 434–444, 1988.

15 Kaspar Fischer, Bernd Gärtner, and Martin Kutz. Fast smallest-enclosing-ball computation in high dimensions. In Proceedings of the 11th Annual European Symposium on Algorithms, pages 630–641, 2003. doi:10.1007/978-3-540-39658-1_57

16 Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. Journal of Statistical Software, Articles, 31(7):1–24, 2009. URL: https://www.jstatsoft.org/v031/i07

17 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. Theoretical Computer Science, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5

18 Lalit Gupta, Dennis L Molfese, Ravi Tammana, and Panagiotis G Simos. Nonlinear alignment and averaging for estimating the evoked potential. IEEE Transactions on Biomedical Engineering, 43(4):348–356, 1996.

19 Ville Hautamäki, Pekka Nykänen, and Pasi Fränti. Time-series clustering by approximate prototypes. In Proceedings of the 19th International Conference on Pattern Recognition, pages 1–4, 2008. doi:10.1109/ICPR.2008.4761108

20 Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In Proceedings of the 34th ACM Symposium on Theory of Computing, pages 731–740, 2002. doi:10.1145/509907.510012

21 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. J. ACM, 48(2):274–296, 2001. doi:10.1145/375827.375845

22 Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. SIAM Journal of Computing, 45(2):530–547, 2016. doi:10.1137/130938645

23 Nimrod Megiddo. Linear-time algorithms for linear programming in r^3 and related problems. SIAM Journal of Computing, 12(4):759–776, 1983. doi:10.1137/0212052
Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal of Computing*, 13(1):182–196, 1984. doi:10.1137/0213014

François Petitjean and Pierre Gançarski. Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment. *Theoretical Computer Science*, 414(1):76–91, 2012. doi:10.1016/j.tcs.2011.09.029

Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4):757–771, 2003.

Kari-Jouko Räihä and Esko Ukkonen. The shortest common supersequence problem over binary alphabet is NP-complete. *Theoretical Computer Science*, 16(2):187–198, 1981. doi:10.1016/0304-3975(81)90075-X

Alexis Sardá-Espinosa. Comparing time-series clustering algorithms in R using the dtwclust package. *R package vignette*, 12:41, 2017.

Alexis Sardá-Espinosa. Time-Series Clustering in R Using the dtwclust Package. *The R Journal*, 11(1):22–43, 2019. URL: https://doi.org/10.32614/RJ-2019-023 doi:10.32614/RJ-2019-023

Martijn Struijs. Curve clustering: hardness and algorithms. Msc thesis, Eindhoven University of Technology, 2018. URL: https://research.tue.nl/files/125547043/thesis_Martijn_Struijs_IAM_311.pdf