

# Exact Mean Computation in Dynamic Time Warping Spaces

Markus Brill<sup>1</sup>, Till Fluschnik<sup>2</sup>, Vincent Froese<sup>2</sup>, Brijnesh Jain<sup>3</sup>,  
Rolf Niedermeier<sup>2</sup>, and David Schultz<sup>3</sup>

<sup>1</sup>Efficient Algorithms, TU Berlin, Germany, [brill@tu-berlin.de](mailto:brill@tu-berlin.de)

<sup>2</sup>Algorithmics and Computational Complexity, TU Berlin, Germany,  
{[till.fluschnik](mailto:till.fluschnik@tu-berlin.de),[vincent.froese](mailto:vincent.froese@tu-berlin.de),[rolf.niedermeier](mailto:rolf.niedermeier@tu-berlin.de)}@tu-berlin.de

<sup>3</sup>Distributed Artificial Intelligence Laboratory, TU Berlin, Germany,  
{[brijnesh-johannes.jain](mailto:brijnesh-johannes.jain@dai-labor.de),[david.schultz](mailto:david.schultz@dai-labor.de)}@dai-labor.de

## Abstract

Dynamic time warping constitutes a major tool for analyzing time series. In particular, computing a mean series of a given sample of series in dynamic time warping spaces (by minimizing the Fréchet function) is a challenging computational problem, so far solved by several heuristic and inexact strategies. We spot some inaccuracies in the literature on exact mean computation in dynamic time warping spaces. Our contributions comprise an exact dynamic program computing a mean (useful for benchmarking and evaluating known heuristics). Based on this dynamic program, we empirically study properties like uniqueness and length of a mean. Moreover, experimental evaluations reveal substantial deficits of state-of-the-art heuristics in terms of their output quality. We also give an exact polynomial-time algorithm for the special case of binary time series.

*Keywords:* time series analysis, Fréchet function, dynamic programming, exact exponential-time algorithm, empirical evaluation of heuristics

## 1 Introduction

Time series such as acoustic signals, electrocardiograms, and internet traffic data are time-dependent observations that vary in length and temporal dynamics. Given a sample of time series, to filter out the corresponding variations, one major direction to time series averaging applies *dynamic time warping* (dtw). Different dtw-averaging approaches have been applied to improve nearest neighbor classifiers and to formulate centroid-based clustering algorithms in dtw-spaces [1, 15, 19, 22, 27, 28, 33].

The most successful approaches pose time series averaging as an optimization problem [8, 15, 25, 30, 33]: Suppose that  $\mathcal{X} = (x^{(1)}, \dots, x^{(k)})$  is a sample of  $k$  time series  $x^{(i)}$ . Then a (weighted Fréchet) *mean* in dtw-spaces is usually defined as any time series  $z$  that minimizes the *weighted Fréchet function* [11]

$$F_w(z) := \sum_{i=1}^k w_i \cdot \text{dtw}(z, x^{(i)})^2,$$

where  $\text{dtw}(x, y)$  denotes the dtw-distance between time series  $x$  and  $y$  and  $w = (w_1, \dots, w_k)$  is a given  $k$ -dimensional weight vector (usually assumed to satisfy  $0 \leq w_i \leq 1$  for all  $i \in [k]$  and  $\sum_{i=1}^k w_i = 1$ ). We refer to the problem of minimizing  $F_w$  over the set  $\mathcal{T}$  of all finite time series as the WEIGHTED DTW-MEAN problem. In the special case of  $w_i = 1/k$  for all  $i \in [k]$ , we write  $F(z)$  and refer to the problem of minimizing  $F$  over the set  $\mathcal{T}$  as DTW-MEAN.

A variant of the DTW-MEAN problem constrains the solution set  $\mathcal{T}$  to the subset  $\mathcal{T}_m \subseteq \mathcal{T}$  of all length- $m$  time series. For both the constrained and unconstrained DTW-MEAN variant, solutions are guaranteed to exist, but are not unique in general [30].

As has been shown very recently, DTW-MEAN is NP-hard, W[1]-hard with respect to the sample size  $k$ , and not solvable in  $\rho(k) \cdot n^{o(k)}$  time for any computable function  $\rho$  (where  $n$  is the maximum length of any input time series) assuming a plausible complexity-theoretic hypothesis [6]. Algorithms with exponential time complexity have been falsely claimed to provide optimal solutions [15, 24, 25]. Existing heuristics approximately solve the constrained DTW-MEAN problem [8, 25, 30]. Thereby, the length  $m$  of feasible solutions is specified beforehand without any knowledge about whether the subset  $\mathcal{T}_m$  contains an optimal solution of the unconstrained DTW-MEAN problem. In summary, the development of nontrivial exact algorithms are to be considered widely open.

**Our Contributions.** We discuss several problematic statements in the literature concerning the computational complexity of exact algorithms for DTW-MEAN. We refute (supplying counterexamples) some false claims from the literature and clarify the known state of the art with respect to computing means in dtw-spaces (Section 3). We show that in case of binary time series (both input and mean) there is an exact polynomial-time algorithm for mean computation in dtw-spaces (Section 4). We develop a dynamic program as an exact algorithm for the (unconstrained) WEIGHTED DTW-MEAN problem on rational time series. The worst-case time complexity of the proposed dynamic program is  $O(n^{2k+1}2^k k)$ , where  $k$  is the sample size and  $n$  is the maximum length of a sample time series (Section 5). We apply the proposed exact dynamic program on small-scaled problems as a benchmark of how well the state-of-the-art heuristics approximate a mean. Our empirical findings reveal that all state-of-the-art heuristics suffer from relatively poor worst-case solution quality in terms of minimizing the Fréchet function, and the solution quality in general may vary quite a lot (Section 6).

## 2 Preliminaries

Throughout this paper, we consider only finite univariate time series with rational elements. A univariate *time series* of length  $n$  is a sequence  $x = (x_1, \dots, x_n) \in \mathbb{Q}^n$ . We denote the set of all univariate rational time series of length  $n$  by  $\mathcal{T}_n$ . Furthermore,  $\mathcal{T} = \bigcup_{n \in \mathbb{N}} \mathcal{T}_n$  denotes the set of all univariate rational time series of finite length. For every  $n \in \mathbb{N}$ , let  $[n]$  denote the set  $\{1, \dots, n\}$ .

The next definition is fundamental for our central computational problem.

**Definition 1.** A warping path of order  $m \times n$  is a sequence  $p = (p_1, \dots, p_L)$  with  $p_\ell \in [m] \times [n]$  for all  $\ell \in [L]$  such that

- i)  $p_1 = (1, 1)$ ,
- ii)  $p_L = (m, n)$ , and
- iii)  $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$  for all  $\ell \in [L - 1]$ .

Note that  $\max\{m, n\} \leq L \leq m + n$ . We denote the set of all warping paths of order  $m \times n$  by  $\mathcal{P}_{m,n}$ . For two time series  $x = (x_1, \dots, x_m)$  and  $y = (y_1, \dots, y_n)$ , a warping path  $p = (p_1, \dots, p_L) \in \mathcal{P}_{m,n}$  defines an alignment of  $x$  and  $y$ : each pair  $p_\ell = (i_\ell, j_\ell)$  of  $p$  aligns element  $x_{i_\ell}$  with  $y_{j_\ell}$ . The cost  $C_p(x, y)$  for aligning  $x$  and  $y$  along warping path  $p$  is defined as  $C_p(x, y) = \sum_{\ell=1}^L (x_{i_\ell} - y_{j_\ell})^2$ . The *dtw-distance* between  $x$  and  $y$  is defined as

$$\text{dtw}(x, y) := \min_{p \in \mathcal{P}_{m,n}} \left\{ \sqrt{C_p(x, y)} \right\}.$$

A warping path  $p$  with  $C_p(x, y) = (\text{dtw}(x, y))^2$  is called an *optimal* warping path for  $x$  and  $y$ .

We remark that  $\text{dtw}(x, y)$  for two time series  $x, y$  of length  $n$  can be computed in subquadratic time  $O(n^2 \log \log \log n / \log \log n)$  [12]. The existence of a strongly subquadratic-time (that is,  $O(n^{2-\epsilon})$  for some  $\epsilon > 0$ ) algorithm is considered unlikely [5].

Our central computational problem is defined as follows.

**WEIGHTED DTW-MEAN**

**Input:** Sample  $\mathcal{X} = (x^{(1)}, \dots, x^{(k)})$  of  $k$  univariate rational time series and rational nonnegative weights  $w_1, \dots, w_k$ .

**Task:** A univariate rational time series  $z$  that minimizes the weighted Fréchet function  $F_w(z)$ .

The special case of uniform weights (that is,  $w_i = \frac{1}{k}$  for all  $i \in [k]$ ) is called DTW-MEAN. It is known that a weighted mean always exists [17, Remark 2.13] (however, it is not necessarily unique). Note that for rational inputs, every weighted mean is also rational (see Lemma 5), that is, WEIGHTED DTW-MEAN always has a solution.

## 3 Problematic Statements in the Literature

In this section we discuss misleading and wrong claims and errors in the literature.

### 3.1 NP-Hardness

The DTW-MEAN problem is often related to the STEINER STRING (STS) problem [2, 10, 15, 18, 23–25, 27]. A *Steiner string* [14] (or *Steiner sequence*) for a set  $S$  of strings is a string  $t$  that minimizes  $\sum_{s \in S} D(t, s)$ , where  $D$  is a distance measure between two strings often assumed to fulfill the triangle inequality (e.g., the weighted edit distance). Computing a Steiner string is equivalent to solving the MULTIPLE SEQUENCE ALIGNMENT (MSA) problem [14]. Both, STS and MSA, are known to be NP-hard for several distance measures even on binary alphabets [4, 20]. Interestingly, as we show in Section 4, WEIGHTED DTW-MEAN is solvable in polynomial time for the space of binary time series.

Several papers mention the NP-hardness results for MSA and STS in the context of DTW-MEAN [10, 15, 18, 23], thereby suggesting that DTW-MEAN is also NP-hard. However, it is not clear (and not shown in the literature) how to reduce from MSA (or STS) to DTW-MEAN since the involved distance measures are significantly different. For example, the dtw-distance lacks the metric property of the edit distance. Very recently, devising an intricate reduction from the MULTICOLORED CLIQUE problem, Bulteau et al. [6] showed that DTW-MEAN is NP-hard, W[1]-hard with respect to the sample size  $k$ , and not solvable in time  $\rho(k) \cdot n^{o(k)}$  for any function  $\rho$ , where  $n$  is the maximum length of any sample time series (unless the *Exponential Time Hypothesis* fails).

### 3.2 Computation of Exact Solutions

Two exponential-time algorithms were proposed to exactly solve DTW-MEAN. The first approach is based on multiple (sequence) alignment in bioinformatics [14] and the second is a brute-force method [25]. We show that neither algorithm is guaranteed to return an optimal solution for every DTW-MEAN instance.

**Multiple Alignment.** It is claimed that DTW-MEAN can be solved by averaging a (global) multiple alignment of the  $k$  input series [15, 24, 25, 27]. A multiple alignment of  $k$  time series in the context of dynamic time warping is described as computing a  $k$ -dimensional warping path in a  $k$ -dimensional matrix (a precise formal definition is not given). Concerning the running time, it is claimed that computing a multiple alignment requires  $\Theta(n^k)$  time [24, 25] ( $O(n^k)$  time [27]), where  $n$  is the maximum length of an input time series. Neither the upper bound of  $O(n^k)$  nor the lower bound of  $\Omega(n^k)$  on the running time are formally proven.

Given a multiple alignment, it is claimed that averaging the  $k$  resulting aligned time series column-wise yields a mean [24, Definition 4]. We show that this is not correct even for two time series. Note that for two time series, a multiple alignment is simply obtained by an optimal warping path. However, the column-wise average of two aligned time series obtained from an optimal warping path is not always an optimal solution for a DTW-MEAN instance as the following example shows.

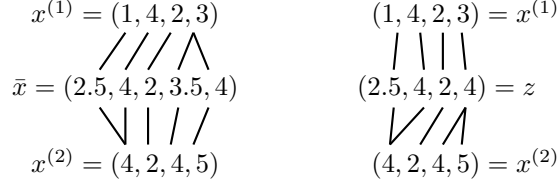


Figure 1: Illustration of Example 1. Shown are two time series  $x^{(1)}$  and  $x^{(2)}$ , as well as the time series  $\bar{x}$  obtained by the multiple alignment approach and a mean  $z$ . Lines indicate optimal alignments between the time series. Note that  $\text{dtw}(x^{(1)}, \bar{x})^2 = \text{dtw}(x^{(2)}, \bar{x})^2 = 3.5$  whereas  $\text{dtw}(x^{(1)}, z)^2 = \text{dtw}(x^{(2)}, z)^2 = 3.25$ .

**Example 1.** Let  $x^{(1)} = (1, 4, 2, 3)$  and  $x^{(2)} = (4, 2, 4, 5)$ . Using an exhaustive search, we obtain the unique optimal warping path

$$p = ((1, 1), (2, 1), (3, 2), (4, 3), (4, 4))$$

of length five. The two corresponding aligned length-5 time series are

$$\begin{aligned} x_p^{(1)} &= (1, 4, 2, 3, 3), \\ x_p^{(2)} &= (4, 4, 2, 4, 5). \end{aligned}$$

The arithmetic mean of these time series is  $\bar{x} = (2.5, 4, 2, 3.5, 4)$ . However, for  $z = (2.5, 4, 2, 4)$ , we have  $F(z) = 6.5 < 7 = F(\bar{x})$ , which shows that  $\bar{x}$  is not a mean (see Figure 1).

In Example 1, the time series  $\bar{x}$  is also not an optimal choice among all time series of length five since also  $z' = (2.5, 4, 4, 2, 4)$  satisfies  $F(z') = 6.5 < F(\bar{x})$ . In fact, by computer-based exhaustive search we found that no warping path  $p \in \mathcal{P}_{4,4}$  yields a mean for  $x^{(1)}$  and  $x^{(2)}$  by averaging the aligned time series  $x_p^{(1)}$  and  $x_p^{(2)}$ . We conclude that a multiple alignment as defined by Petitjean and Gançarski [24, Definition 5] that shall produce an averaged time series that minimizes the Fréchet function does not exist in general. Example 1 implies that incremental pairwise averaging strategies such as NLAFF [13] or PSA [21] are based on a wrong mean computation for two time series.

We finish with another erroneous example from the literature for three time series [24, Figure 2].

**Example 2.** For the three time series

$$\begin{aligned} x^{(1)} &= (1, 10, 0, 0, 4), \\ x^{(2)} &= (0, 2, 10, 0, 0), \\ x^{(3)} &= (0, 0, 0, 10, 0), \end{aligned}$$

the multiple alignment is given as

$$\begin{aligned}x^{(1)'} &= (1, 1, 1, 10, 0, 0, 4), \\x^{(2)'} &= (0, 0, 2, 10, 0, 0, 0), \\x^{(3)'} &= (0, 0, 0, 10, 0, 0, 0),\end{aligned}$$

yielding the arithmetic mean  $\bar{x} = (\frac{1}{3}, \frac{1}{3}, 1, 10, 0, 0, \frac{4}{3})$  with  $F(\bar{x}) = 14/3 \geq 4.66$ . However, for  $z = (\frac{1}{4}, 1, 10, 0, \frac{4}{3})$ , it holds  $F(z) \leq 4.48 < F(\bar{x})$ .

**Brute-Force Algorithm.** Another approach to solve DTW-MEAN is based on a brute-force algorithm [25] working as follows: Suppose an optimal mean is of length  $m$ . Consider for each input time series a partition into  $m$  consecutive non-empty parts and align the  $i$ -th element in the mean with all elements in the  $i$ -th part of each time series. It is claimed that a mean can be found by trying out all possible partitions into  $m$  consecutive non-empty parts for each input time series.

This approach is problematic since not all possible solutions are considered (as two elements in the mean can be aligned with the same element of an input time series). Example 1 depicts this problem.

## 4 Polynomial-time Solvability for Binary Data

By restricting the values in the time series (input and mean) to be binary (0 or 1), we arrive at the special case BINARY DTW-MEAN.

BINARY DTW-MEAN

**Input:** Sample  $\mathcal{X} = (x^{(1)}, \dots, x^{(k)})$  of  $k$  time series with elements in  $\{0, 1\}$  and rational nonnegative weights  $w_1, \dots, w_k$ .

**Task:** Find a time series  $z \in \{0, 1\}^*$  that minimizes  $F_w(z)$ .

We prove that BINARY DTW-MEAN is polynomial-time solvable.

**Theorem 1.** BINARY DTW-MEAN for  $k$  input time series is solvable with  $O(kn^3)$  arithmetic operations, where  $n$  is the maximum length of any input time series.

To show polynomial-time solvability of BINARY DTW-MEAN, we first prove some preliminary results about the dtw-distance of binary time series and properties of a binary mean. We start with the following general definition.

**Definition 2.** A time series  $x = (x_1, \dots, x_n)$  is condensed if no two consecutive elements are equal, that is,  $x_i \neq x_{i+1}$  holds for all  $i \in [n-1]$ . We denote the condensation of a time series  $x$  by  $\tilde{x}$  and define it to be the time series obtained by repeatedly removing one of two equal consecutive elements in  $x$  until the remaining series is condensed.

The following proposition implies that a mean can always be assumed to be condensed. Note that this holds for arbitrary time series (not only for the binary case) but not for the constrained mean.

**Proposition 1.** *Let  $x$  be a time series and let  $\tilde{x}$  denote its condensation. Then, for every time series  $y$ , it holds that  $\text{dtw}(\tilde{x}, y) \leq \text{dtw}(x, y)$ .*

*Proof.* Let  $y$  have length  $m$  and assume that  $x = (x_1, \dots, x_n)$  is not condensed. Then,  $x_i = x_{i+1}$  holds for some  $i \in [n-1]$ . Let  $p = ((i_1, j_1), \dots, (i_L, j_L))$  be an optimal warping path for  $x$  and  $y$ . Now, consider the time series  $x' = (x_1, \dots, x_i, x_{i+2}, \dots, x_n)$  that is obtained from  $x$  by deleting the element  $x_{i+1}$ . We construct a warping path  $p'$  for  $x'$  and  $y$  such that  $C_{p'}(x', y) = C_p(x, y)$ . To this end, let  $p_a = (i_a, j_a)$ ,  $2 \leq a \leq L$ , be the first index pair in  $p$  where  $i_a = i+1$  (hence,  $i_{a-1} = i$ ). Now, we consider two cases.

If  $j_a = j_{a-1} + 1$ , then we define the order- $((n-1) \times m)$  warping path

$$p' := ((i_1, j_1), \dots, (i_{a-1}, j_{a-1}), (i_a - 1, j_a), \dots, (i_L - 1, j_L))$$

of length  $L$ . Thus, each element of  $y$  that was aligned to  $x_{i+1}$  in  $p$  is now aligned to  $x_i$  instead. To check that  $p'$  is a valid warping path, note first that  $(i_1, j_1) = (1, 1)$  and  $(i_L - 1, j_L) = (n-1, m)$  holds since  $p$  is a warping path. Also, it holds

$$\begin{aligned} \forall 1 \leq \ell \leq a-2: (i_{\ell+1}, j_{\ell+1}) - (i_\ell, j_\ell) &\in \{(1, 0), (0, 1), (1, 1)\}, \\ (i_a - 1, j_a) - (i_{a-1}, j_{a-1}) &= (0, 1), \\ \forall a \leq \ell \leq L-1: (i_{\ell+1} - 1, j_{\ell+1}) - (i_\ell - 1, j_\ell) &\in \{(1, 0), (0, 1), (1, 1)\}. \end{aligned}$$

The cost of  $p'$  is

$$\begin{aligned} C_{p'}(x', y) &= \sum_{\ell=1}^{a-1} (x'_{i_\ell} - y_{j_\ell})^2 + \sum_{\ell=a}^L (x'_{(i_\ell-1)} - y_{j_\ell})^2 \\ &= \sum_{\ell=1}^{a-1} (x_{i_\ell} - y_{j_\ell})^2 + \sum_{\ell=a}^L (x_{i_\ell} - y_{j_\ell})^2 = C_p(x, y). \end{aligned}$$

Otherwise, if  $j_a = j_{a-1}$ , then we define the warping path

$$p' := ((i_1, j_1), \dots, (i_{a-1}, j_{a-1}), (i_{a+1} - 1, j_{a+1}), \dots, (i_L - 1, j_L))$$

of length  $L-1$ . Again,  $(i_1, j_1) = (1, 1)$  and  $(i_L - 1, j_L) = (n-1, m)$  holds since  $p$  is a warping path. Clearly, also

$$\begin{aligned} \forall 1 \leq \ell \leq a-2: (i_{\ell+1}, j_{\ell+1}) - (i_\ell, j_\ell) &\in \{(1, 0), (0, 1), (1, 1)\} \text{ and} \\ \forall a+1 \leq \ell \leq L-1: (i_{\ell+1} - 1, j_{\ell+1}) - (i_\ell - 1, j_\ell) &\in \{(1, 0), (0, 1), (1, 1)\} \end{aligned}$$

holds. Finally, we have  $(i_{a+1} - 1, j_{a+1}) - (i_{a-1}, j_{a-1}) \in \{(1, 0), (0, 1), (1, 1)\}$  since  $i_{a+1} - 1 - i_{a-1} = i_{a+1} - i_a$  holds and also  $j_{a+1} - j_{a-1} = j_{a+1} - j_a$  holds.

Thus,  $p'$  is a valid warping path and its cost is

$$\begin{aligned} C_{p'}(x', y) &= \sum_{\ell=1}^{a-1} (x'_{i_\ell} - y_{j_\ell})^2 + \sum_{\ell=a+1}^L (x'_{(i_\ell-1)} - y_{j_\ell})^2 \\ &= \sum_{\ell=1}^{a-1} (x_{i_\ell} - y_{j_\ell})^2 + \sum_{\ell=a+1}^L (x_{i_\ell} - y_{j_\ell})^2 \\ &= C_p(x, y) - (x_{i_a} - y_{j_a})^2. \end{aligned}$$

Since in both cases above, the cost does not increase, we obtain

$$\text{dtw}(x', y) \leq C_{p'}(x', y) \leq C_p(x, y) = \text{dtw}(x, y).$$

Repeating this argument until  $x'$  is condensed finishes the proof.  $\square$

Proposition 1 implies that we can assume a mean to be condensed. Next, we want to prove an upper bound on the length of a binary mean. To this end, we analyze the dtw-distances of binary time series. Note that a binary condensed time series is fully determined by its first element and its length. We use this property to give a closed expression for the dtw-distance of two condensed binary time series.

**Lemma 1.** *Let  $x = (x_1, \dots, x_n)$ ,  $y = (y_1, \dots, y_m) \in \{0, 1\}^*$  be two condensed binary time series with  $n \geq m$ . Then, it holds*

$$\text{dtw}(x, y)^2 = \begin{cases} \lceil (n-m)/2 \rceil, & x_1 = y_1 \\ 2, & x_1 \neq y_1 \wedge n = m \\ 1 + \lfloor (n-m)/2 \rfloor, & x_1 \neq y_1 \wedge n > m \end{cases}$$

*Proof.* We prove the statement by first giving a warping path that has the claimed cost and second proving that every warping path has at least the claimed cost.

“ $\leq$ ”: We show that there exists a warping path  $p$  between  $x$  and  $y$  that has the claimed cost. The warping path  $p$  is defined as follows:

If  $x_1 = y_1$ , then we have  $x_i = y_i$  for all  $i \in [m]$  (since  $x$  and  $y$  are condensed and binary) and we set

$$p := ((1, 1), (2, 2), \dots, (m, m), (m+1, m), \dots, (n, m)).$$

This warping path has cost  $C_p(x, y) = \sum_{i=m+1}^n |x_i - y_m| = \lceil (n-m)/2 \rceil$ .

If  $x_1 \neq y_1$ , then we have  $x_i = y_{i-1}$  for all  $2 \leq i \leq m$ . Thus, for  $n = m$ , the warping path

$$p := ((1, 1), (2, 1), (3, 2), \dots, (n, m-1), (n, m))$$



has cost  $C_p(x, y) = |x_1 - y_1| + |x_n - y_m| = 2$ . Finally, for  $n > m$ , the warping path  $p := ((1, 1), (2, 1), (3, 2), \dots, (m+1, m), (m+2, m), \dots, (n, m))$  yields cost

$$\begin{aligned} C_p(x, y) &= |x_1 - y_1| + \sum_{i=2}^{m+1} |x_i - y_{i-1}| + \sum_{i=m+2}^n |x_i - y_m| \\ &= 1 + \sum_{i=m+2}^n |x_i - y_m| = 1 + \lceil (n - m - 1)/2 \rceil = 1 + \lfloor (n - m)/2 \rfloor. \end{aligned}$$

“ $\geq$ ”: We show that every warping path has at least the cost claimed above. Consider an optimal warping path  $p = (p_1, \dots, p_L)$  for  $x$  and  $y$  and note that there are at least  $n - m$  different indices  $\ell \in [L - 1]$  such that  $p_{\ell+1} - p_\ell = (1, 0)$  since  $n \geq m$ . For every such pairs  $p_{\ell+1} = (i_{\ell+1}, j_{\ell+1})$ ,  $p_\ell = (i_\ell, j_\ell)$  with  $j_{\ell+1} = j_\ell$ , we have

$$|x_{i_{\ell+1}} - y_{j_{\ell+1}}| + |x_{i_\ell} - y_{j_\ell}| = |x_{i_{\ell+1}} - y_{j_\ell}| + |x_{i_\ell} - y_{j_\ell}| = 1,$$

since  $x_{j_{\ell+1}} \neq x_{j_\ell}$  (recall that  $x$  is condensed and binary). Hence, at least for every second such index  $\ell$  (starting from the first one) a cost of 1 is induced. Hence,  $\text{dtw}(x, y)^2 \geq \lceil (n - m)/2 \rceil$ . If  $x_1 = y_1$ , then this lower bound matches the claimed cost.

If  $x_1 \neq y_1$  and  $n = m$ , then also  $x_n \neq y_m$  and hence the cost is at least 2.

If  $x_1 \neq y_1$  and  $n > m$ , then we can assume that  $p_2 = (2, 1)$ . To see this, note that for  $p_2 \neq (2, 1)$  the cost is at least  $1 + \lceil (n - m)/2 \rceil$  by the above argument. If  $p_2 = (2, 1)$ , then the subpath  $(p_2, p_3, \dots, p_L)$  of  $p$  is an optimal warping path between  $(x_2, \dots, x_n)$  and  $y$ , where  $x_2 = y_1$  and  $n - 1 \geq m$ . As we have already shown above, this path has cost  $\lceil (n - 1 - m)/2 \rceil$ . Hence, in this case  $p$  has cost  $1 + \lceil (n - 1 - m)/2 \rceil = 1 + \lfloor (n - m)/2 \rfloor$ . Thus, we can assume that  $p_2 = (2, 1)$  in which case the cost of  $p$  matches the claimed cost of the lemma. This finishes the proof.  $\square$

Note that according to Lemma 1, for a fixed condensed binary time series  $y$  of length  $m$ , the value  $\text{dtw}(x, y)^2$  is monotonically increasing in the length of  $x$  for all condensed binary time series  $x$  of length  $n \geq m + 1$ . We use this property later in the proof of Lemma 3 where we derive an upper bound on the length of a binary mean. In order to prove Lemma 3, we also need the following lemma concerning the dtw-distances between condensed and non-condensed time series.

**Lemma 2.** *Let  $x = (x_1, \dots, x_n)$  be a condensed binary time series and let  $y = (y_1, \dots, y_m) \in \{0, 1\}^*$  with  $n \geq m$ . Then, for the condensation  $\tilde{y}$  of  $y$  it holds  $\text{dtw}(x, y)^2 = \text{dtw}(x, \tilde{y})^2$ .*

*Proof.* Assume that  $y$  is not condensed. Then,  $y$  consists of  $\ell \in [m]$  blocks, where a block is a maximal subsequence of consecutive 0's or consecutive 1's in  $y$ . Let  $m_1, \dots, m_\ell$  denote the lengths of these blocks where  $m_1 + \dots + m_\ell = m$ . Note also that  $\tilde{y}$  has length  $\ell$  with  $\ell < m \leq n$ . We define a warping path  $p$

between  $x$  and  $y$  such that  $C_p(x, y) = \text{dtw}(x, \tilde{y})^2$ . Note that, by Lemma 1, we have

$$\text{dtw}(x, \tilde{y})^2 = \begin{cases} \lceil (n - \ell)/2 \rceil, & x_1 = \tilde{y}_1 \\ 1 + \lfloor (n - \ell)/2 \rfloor, & x_1 \neq \tilde{y}_1 \end{cases}.$$

If  $x_1 = y_1$ , then we set  $p := ((1, 1), \dots, (1, m_1), (2, m_1 + 1), \dots, (2, m_1 + m_2), \dots, (\ell, m), (\ell + 1, m), \dots, (n, m))$  and obtain cost  $C_p(x, y) = \sum_{i=\ell+1}^n |x_i - y_m| = \lceil (n - \ell)/2 \rceil$ .

If  $x_1 \neq y_1$ , then we set  $p := ((1, 1), (2, 1), \dots, (2, m_1), (3, m_1 + 1), \dots, (3, m_1 + m_2), \dots, (\ell + 1, m), (\ell + 2, m), \dots, (n, m))$  and obtain cost

$$C_p(x, y) = 1 + \sum_{i=\ell+2}^n |x_i - y_m| = 1 + \lfloor (n - \ell)/2 \rfloor.$$

□

We now have all ingredients to show that there always exists a binary mean of length at most one larger than the maximum length of any input time series.

**Lemma 3.** *For binary input time series  $x^{(1)}, \dots, x^{(k)} \in \{0, 1\}^*$  of maximum length  $n$ , there exists a binary mean  $z \in \{0, 1\}^*$  of length at most  $n + 1$ .*

*Proof.* Assume that  $z = (z_1, \dots, z_m) \in \{0, 1\}^*$  is a mean of length  $m > n + 1$ . By Proposition 1, we can assume that  $z$  is condensed, that is,  $z_i \neq z_{i+1}$  for all  $i \in [m - 1]$ . We claim that  $z' := (z_1, \dots, z_{n+1})$  is also a mean. We prove this claim by showing that  $\text{dtw}(z', x^{(i)})^2 \leq \text{dtw}(z, x^{(i)})^2$  holds for all  $i \in [k]$ . By Lemmas 1 and 2, we have  $\text{dtw}(z', x^{(i)})^2 = \text{dtw}(z', \tilde{x}^{(i)})^2 \leq \text{dtw}(z, \tilde{x}^{(i)})^2 = \text{dtw}(z, x^{(i)})^2$ , where the inequality follows from Lemma 1 since  $z'$  is of length  $n + 1 < m$  and the dtw-distance is monotonically increasing. □

Having established that a binary mean can always be assumed to be condensed and of bounded length, we now show that it can be found in polynomial time.

*Proof of Theorem 1.* By Proposition 1 and Lemma 3, we can assume the desired mean  $z \in \{0, 1\}^*$  to be a condensed series of length at most  $n + 1$ . Thus, there are at most  $2n + 2$  many possible candidates for  $z$ . For each candidate  $z$ , we can compute the value  $F_w(z)$  with  $O(kn^2)$  arithmetic operations and select the one with the smallest value. Overall, this yields  $O(kn^3)$  arithmetic operations. □

## 5 An Exact Algorithm Solving DTW-Mean

We develop a nontrivial exponential-time algorithm solving WEIGHTED DTW-MEAN exactly. The key is to observe a certain structure of a mean and the corresponding alignments to the input time series. To this end, we define redundant elements in a mean. Note that this concept was already used by Jain and Schultz [17, Theorem 2.7] in order to prove the existence of a mean of bounded length (though [17, Definition 3.20] is slightly different).

**Definition 3.** Let  $x^{(1)}, \dots, x^{(k)}$  and  $z = (z_1, \dots, z_m)$  be time series and let  $p^{(j)}$ ,  $j \in [k]$ , denote an optimal warping path between  $x^{(j)}$  and  $z$ . We call an element  $z_i$  of  $z$  redundant if in every time series  $x^{(j)}$ ,  $j \in [k]$ , there exists an element that is aligned with  $z_i$  and with another element of  $z$  by  $p^{(j)}$ .

The next lemma states that there always exists a mean without redundant elements (similarly to [17, Theorem 2.7]).

**Lemma 4.** There exist a mean  $z$  for time series  $x^{(1)}, \dots, x^{(k)}$  and optimal warping paths  $p^{(j)}$  between  $z$  and  $x^{(j)}$  for each  $j \in [k]$  such that  $z$  contains no redundant element.

*Proof.* Let  $z$  be a mean of  $x^{(1)}, \dots, x^{(k)}$  with optimal warping paths  $p^{(j)}$ ,  $j \in [k]$ , such that the element  $z_i$  is redundant (recall that a mean  $z$  always exists). We show that there also exists a mean  $z'$  and optimal warping paths  $p^{(j)'}$  such that no element in  $z'$  is redundant.

Assume first that there exists a  $j \in [k]$  such that the element  $z_i$  is aligned by  $p^{(j)}$  with at least one element  $x_\ell^{(j)}$  in  $x^{(j)}$  that is not aligned with any other element in  $z$ . Let  $L$  denote the length of  $p^{(j)}$ . Then,  $p^{(j)}$  is of the form

$$p^{(j)} = (p_1, \dots, (i-1, \ell_{t-1}), (i, \ell_t), \dots, (i, \ell_{t+\alpha}), (i+1, \ell_{t+\alpha+1}), \dots, p_L)$$

for some  $\ell_t \leq \ell \leq \ell_{t+\alpha}$  and  $\alpha \geq 1$ . Since  $z_i$  is redundant, it follows that  $\ell_{t-1} = \ell_t$  or  $\ell_{t+\alpha} = \ell_{t+\alpha+1}$  holds. If  $\ell_{t-1} = \ell_t$ , then we remove the pair  $(i, \ell_t)$  from  $p^{(j)}$ . Also, if  $\ell_{t+\alpha} = \ell_{t+\alpha+1}$ , then we remove the pair  $(i, \ell_{t+\alpha})$  from  $p^{(j)}$ . Note that this yields a warping path  $p^{(j)'}$  between  $z$  and  $x^{(j)}$  since even if we removed both pairs  $(i, \ell_t)$  and  $(i, \ell_{t+\alpha})$ , then we know by assumption that there still exists the pair  $(i, \ell)$  with  $\ell_t < \ell < \ell_{t+\alpha}$  in  $p^{(j)'}$  since  $z_i$  is aligned with  $x_\ell^{(j)}$  which is not aligned with another element in  $z$ . Since we only removed pairs from  $p^{(j)}$ , it holds  $C_{p^{(j)'}}(z, x^{(j)}) \leq C_{p^{(j)}}(z, x^{(j)})$ . Moreover,  $z_i$  is not redundant anymore.

Now, assume that for all  $j \in [k]$ ,  $z_i$  is aligned only with elements in  $x^{(j)}$  which are also aligned with another element of  $z$  by  $p^{(j)}$  (that is,  $z_i$  is redundant according to [17, Definition 3.20]). Let  $z'$  denote the time series obtained by deleting the element  $z_i$  from  $z$ . Jain and Schultz [17, Proof of Theorem 2.7] showed that in this case  $\text{dtw}(z', x^{(j)}) \leq \text{dtw}(z, x^{(j)})$  holds for all  $j \in [k]$ .

Under both assumptions above, we reduced the number of redundant elements. Hence, we can repeat the above arguments until we obtain a mean  $z'$  without redundant elements.  $\square$

Lemma 4 allows us to devise a dynamic program computing a mean without redundant elements. We compute a mean by testing all possibilities to align the last mean element to elements from the input time series while recursively adding an optimal solution for the remaining non-aligned elements in the input time series (see Figure 2). We use the assumption that the mean does not contain redundant elements for this recursive approach.

Before describing our dynamic program, we prove the following lemma concerning the optimal value of a mean element for given alignments.

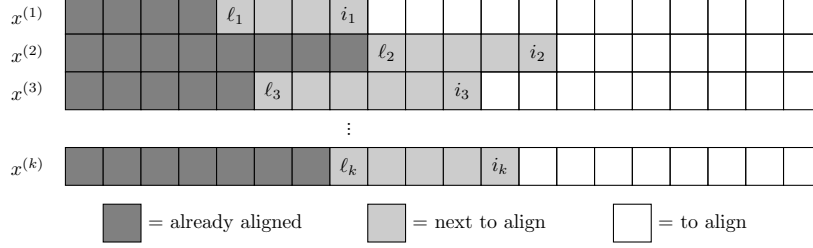


Figure 2: Illustration of the dynamic program computing a mean  $(z_1, \dots, z_q)$  for the subseries  $(x_1^{(1)}, \dots, x_{i_1}^{(1)}), \dots, (x_1^{(k)}, \dots, x_{i_k}^{(k)})$  (white elements are not considered at the current iteration). The light gray elements are aligned to the last mean element  $z_q$ . The remaining mean elements  $(z_1, \dots, z_{q-1})$  form an optimal mean for the dark gray subseries.

**Lemma 5.** Let  $x^{(1)}, \dots, x^{(k)}$  be time series and let  $w_1, \dots, w_k$  be nonnegative weights. Let  $n_j$  denote the length of  $x^{(j)}$ ,  $j \in [k]$ . Further, let  $p^{(j)}$  be a warping path of order  $m \times n_j$  for  $m \in \mathbb{N}$  and let  $z = \arg \min_{x \in \mathcal{T}_m} \sum_{j=1}^k w_j C_{p^{(j)}}(x, x^{(j)})$ . For  $i \in [m]$ , let  $x_{\ell_{ij}}^{(j)}, \dots, x_{h_{ij}}^{(j)}$  denote the elements of  $x^{(j)}$  that are aligned with element  $z_i$  by  $p^{(j)}$ . Then,

$$z_i = \frac{\sum_{j=1}^k w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_t^{(j)}}{\sum_{j=1}^k w_j (h_{ij} - \ell_{ij} + 1)}.$$

*Proof.* By the assumption of the lemma, we have

$$\begin{aligned} z &= \arg \min_{x \in \mathcal{T}_m} \sum_{j=1}^k w_j C_{p^{(j)}}(x, x^{(j)}) = \arg \min_{x \in \mathcal{T}_m} \sum_{j=1}^k w_j \sum_{i=1}^m \sum_{t=\ell_{ij}}^{h_{ij}} (x_i - x_t^{(j)})^2 \\ &= \arg \min_{x \in \mathcal{T}_m} \sum_{i=1}^m \sum_{j=1}^k w_j \sum_{t=\ell_{ij}}^{h_{ij}} (x_i - x_t^{(j)})^2. \end{aligned}$$

Hence, for each  $i \in [m]$ , it holds

$$z_i = \arg \min_{\mu \in \mathbb{Q}} \sum_{j=1}^k w_j \sum_{t=\ell_{ij}}^{h_{ij}} (\mu - x_t^{(j)})^2.$$

Note that the above sum is a convex function in  $\mu$  (since all weights are non-

negative). Setting the first derivative with respect to  $\mu$  equal to zero yields

$$\begin{aligned}
& \sum_{j=1}^k \left( 2w_j(h_{ij} - \ell_{ij} + 1)z_i - 2w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_t^{(j)} \right) = 0 \\
\Leftrightarrow & \quad 2z_i \sum_{j=1}^k (w_j(h_{ij} - \ell_{ij} + 1)) = 2 \sum_{j=1}^k w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_t^{(j)} \\
\Leftrightarrow & \quad z_i = \frac{\sum_{j=1}^k w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_t^{(j)}}{\sum_{j=1}^k w_j (h_{ij} - \ell_{ij} + 1)}
\end{aligned}$$

□

We now prove our main theorem.

**Theorem 2.** WEIGHTED DTW-MEAN for  $k$  input time series is solvable with  $O(n^{2k+1}2^k k)$  arithmetical operations, where  $n$  is the maximum length of any input time series.

*Proof.* Assume for simplicity that all time series have length  $n$  (the general case can be solved analogously). We find a mean using a dynamic programming approach. Let  $C$  be a  $k$ -dimensional table, where for all  $(i_1, \dots, i_k) \in [n]^k$ , we define

$$C[i_1, \dots, i_k] = \min_{z \in \mathcal{T}} \left( \sum_{j=1}^k w_j \left( \text{dtw}(z, (x_1^{(j)}, \dots, x_{i_j}^{(j)})) \right)^2 \right),$$

that is,  $C[i_1, \dots, i_k]$  is the value  $F_w(z)$  of the weighted Fréchet function of a mean  $z$  for the subseries  $(x_1^{(1)}, \dots, x_{i_1}^{(1)}), \dots, (x_1^{(k)}, \dots, x_{i_k}^{(k)})$ . Clearly,  $C[n, \dots, n]$  is the optimal value  $F_w(z)$  of a mean for the input instance.

For  $i_1 = i_2 = \dots = i_k = 1$ , a mean  $z$  clearly contains just one element and each optimal warping path between  $z$  and  $(x_1^{(j)})$  trivially equals  $((1, 1))$ . By Lemma 5, we initialize

$$C[1, \dots, 1] = \sum_{j=1}^k w_j (x_1^{(j)} - \mu)^2, \quad \mu = \sum_{j=1}^k w_j x_1^{(j)}.$$

Hence, the corresponding mean is  $z = (\mu)$ .

For the case that  $i_j > 1$  holds for at least one  $j \in [k]$ , consider a mean  $z$  for  $(x_1^{(1)}, \dots, x_{i_1}^{(1)}), \dots, (x_1^{(k)}, \dots, x_{i_k}^{(k)})$ . By Lemma 4, we can assume that there exist optimal warping paths  $p^{(j)}$  between  $z$  and  $(x_1^{(j)}, \dots, x_{i_j}^{(j)})$  such that  $z$  contains no redundant elements. Let  $z_q$  be the last element of  $z$ . Then, for each  $j \in [k]$ ,  $z_q$  is aligned by  $p^{(j)}$  with some elements  $x_{\ell_j}^{(j)}, \dots, x_{i_j}^{(j)}$  for  $\ell_j \in [i_j]$ . By Lemma 5, it follows that

$$z_q = \frac{\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} x_t^{(j)}}{\sum_{j=1}^k w_j (i_j - \ell_j + 1)}.$$

Hence, the contribution of  $z_q$  to  $F_w(z)$  is  $\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} (x_t^{(j)} - z_q)^2$ .

Now, assume that there exists another element  $z_{q-1}$  in  $z$ . Clearly, for each  $j \in [k]$ ,  $z_{q-1}$  is aligned only with elements of indices up to  $\ell_j$  since otherwise the warping path conditions are violated. Hence,  $F_w(z)$  can be obtained recursively from a mean of the subseries  $(x_1^{(1)}, \dots, x_{\ell_1}^{(1)}), \dots, (x_1^{(k)}, \dots, x_{\ell_k}^{(k)})$ . Recall, however, that we assumed  $z$  not to contain any redundant element. It follows that  $z_{q-1}$  cannot be aligned with  $x_{\ell_j}^{(j)}$  for all  $j \in [k]$  since  $z_q$  is already aligned with each  $x_{\ell_j}^{(j)}$ . Hence, we add the minimum value  $C[\ell'_1, \dots, \ell'_k]$  over all  $\ell'_j \in \{\max(1, \ell_j - 1), \ell_j\}$ , where  $\ell'_j = \ell_j - 1$  holds for at least one  $j \in [k]$ .

We arrive at the following recursion:

$$C[i_1, \dots, i_k] = \min\{c^*(\ell_1, \dots, \ell_k) + \sigma(\ell_1, \dots, \ell_k) \mid \ell_1 \in [i_1], \dots, \ell_k \in [i_k]\},$$

where

$$\sigma(\ell_1, \dots, \ell_k) := \sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} (x_t^{(j)} - \mu)^2, \quad \mu := \frac{\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} x_t^{(j)}}{\sum_{j=1}^k w_j (i_j - \ell_j + 1)},$$

and

$$c^*(\ell_1, \dots, \ell_k) := \min\{C[\ell'_1, \dots, \ell'_k] \mid \ell'_j \in \{\max(1, \ell_j - 1), \ell_j\}, \sum_{j=1}^k (\ell_j - \ell'_j) > 0\}$$

if  $\ell_j > 1$  holds for some  $j \in [k]$ , and  $c^*(1, \dots, 1) := 0$ .

In order to compute  $C[i_1, \dots, i_k]$ , a minimum is computed over all possible choices  $\ell_j \in [i_j]$ ,  $j \in [k]$ . For each choice  $\ell_1, \dots, \ell_k$ , the value  $\mu$  corresponds to the last element of a mean and  $\sigma(\ell_1, \dots, \ell_k)$  is the induced cost of aligning this element with  $x_{\ell_j}^{(j)}, \dots, x_{i_j}^{(j)}$  for each  $j \in [k]$ . The value  $c^*(\ell_1, \dots, \ell_k)$  yields the value  $F_w(z')$  of a mean  $z'$  for the remaining subseries  $(x_1^{(j)}, \dots, x_{\ell'_j}^{(j)}), j \in [k]$ , over all  $\ell'_j \in \{\max(1, \ell_j - 1), \ell_j\}$  such that  $\sum_{j=1}^k (\ell_j - \ell'_j) > 0$ , which implies that  $\ell'_j = \ell_j - 1$  holds for at least one  $j \in [k]$ . This condition guarantees that we only find alignments which do not yield redundant elements in the mean (which we can assume by Lemma 4). Note that  $\ell_j = 1$  implies that  $\ell'_j = 1$  (since index 0 does not exist in  $x^{(j)}$ ).

The dynamic programming table  $C$  can be filled iteratively along the dimensions starting from  $C[1, \dots, 1]$ . The overall number of entries is  $n^k$ . For each table entry, the minimum of a set containing  $O(n^k)$  elements is computed. Computing an element requires the computation of  $\sigma(\ell_1, \dots, \ell_k)$  which can be done with  $O(kn)$  arithmetical operations plus the computation of  $c^*(\ell_1, \dots, \ell_k)$  which is the minimum of a set of size at most  $2^k$  whose elements can be obtained by constant-time table look-ups. Thus,  $C$  can be filled using  $O(n^k \cdot n^k \cdot 2^k \cdot kn)$  arithmetical operations. A mean can be obtained by storing the values  $\mu$  for which the minimum in the above recursion is attained (Algorithm 1 contains the pseudocode).  $\square$

---

**Algorithm 1:** Exact Dynamic Program (EDP) for WEIGHTED DTW-MEAN

---

**Input:** Time series  $x^{(1)}, \dots, x^{(k)}$  of length  $n$  and weights  $w_1, \dots, w_k > 0$ .  
**Output:** Mean  $z$  and  $F_w(z)$ .  
Initialize  $C$  //  $k$ -dimensional DP table storing  $F_w$ -values  
Initialize  $Z$  //  $k$ -dimensional table storing means  
**foreach**  $(i_1, \dots, i_k) \in [n]^k$  **do** // fill tables iteratively  
     $C[i_1, \dots, i_k] := \infty$   
     $Z[i_1, \dots, i_k] := ()$   
    **foreach**  $(\ell_1, \dots, \ell_k) \in [i_1] \times \dots \times [i_k]$  **do** // compute  $C[i_1, \dots, i_k]$   
         $\mu := (\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} x_t^{(j)}) / (\sum_{j=1}^k w_j (i_j - \ell_j + 1))$   
         $\sigma := \sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} (x_t^{(j)} - \mu)^2$   
         $c^* := \infty$   
         $z := ()$   
        **if**  $\ell_1 = \ell_2 = \dots = \ell_k = 1$  **then**  
             $c^* := 0$   
        **else** // compute  $c^*(\ell_1, \dots, \ell_k)$  based on table look-ups  
            **foreach**  $(\ell'_1, \dots, \ell'_k) \in \{\ell_1 - 1, \ell_1\} \times \dots \times \{\ell_k - 1, \ell_k\}$  **do**  
                **if**  $\forall j \in [k] : \ell'_j \geq 1$  **and**  $\exists j \in [k] : \ell'_j < \ell_j$  **then**  
                    **if**  $C[\ell'_1, \dots, \ell'_k] < c^*$  **then**  
                         $c^* := C[\ell'_1, \dots, \ell'_k]$   
                         $z := Z[\ell'_1, \dots, \ell'_k]$   
        **if**  $c^* + \sigma < C[i_1, \dots, i_k]$  **then** // update mean and  $F_w$ -value  
             $C[i_1, \dots, i_k] := c^* + \sigma$   
             $Z[i_1, \dots, i_k] := \text{append}(z, \mu)$   
**return**  $(Z[n, \dots, n], C[n, \dots, n])$

---

We close with some remarks on the above result.

- The dynamic program also allows to compute all (non-redundant) means by storing all possible values for which the minimum in the recursion is attained.
- It is possible to incorporate a fixed length  $q$  into the dynamic program (by adding another dimension to the table  $C$ ) such that it outputs only optimal solutions of length  $q$ .<sup>1</sup> The running time increases by a factor of  $q^2$ .
- The dynamic program can easily be extended to multivariate time series with elements in  $\mathbb{Q}^d$  and a cost function  $C_p(x, y) := \sum_{\ell=1}^L \|x_{i_\ell} - y_{j_\ell}\|_2^2$  (with a running time increase by a factor of  $d$ ).

---

<sup>1</sup>Source code available at <http://www.akt.tu-berlin.de/menue/software/>.

Data Set	#	$n$	Type
ItalyPowerDemand	1096	24	SENSOR
SyntheticControl	600	60	SIMULATED
SonyAIBORobotSurface2	980	65	SENSOR
SonyAIBORobotSurface1	621	70	SENSOR
ProximalPhalanxTW	605	80	IMAGE
ProximalPhalanxOutlineCorrect	891	80	IMAGE
ProximalPhalanxOutlineAgeGroup	605	80	IMAGE
PhalangesOutlinesCorrect	2658	80	IMAGE
MiddlePhalanxTW	553	80	IMAGE
MiddlePhalanxOutlineCorrect	891	80	IMAGE
MiddlePhalanxOutlineAgeGroup	554	80	IMAGE
DistalPhalanxTW	539	80	IMAGE
DistalPhalanxOutlineCorrect	876	80	IMAGE
DistalPhalanxOutlineAgeGroup	539	80	IMAGE
TwoLeadECG	1162	82	ECG
MoteStrain	1272	84	SENSOR
ECG200	200	96	ECG
MedicalImages	1141	99	IMAGE
TwoPatterns	5000	128	SIMULATED
SwedishLeaf	1125	128	IMAGE
CBF	930	128	SIMULATED
FacesUCR	2250	131	IMAGE
FaceAll	2250	131	IMAGE
ECGFiveDays	884	136	ECG
ECG5000	5000	140	ECG
Plane	210	144	SENSOR
GunPoint	200	150	MOTION

Table 1: List of 27 UCR time series data sets. Columns # and  $n$  show the number and length of time series, respectively. The last column refers to the respective application domains (e.g. ECG stands for electrocardiography).

## 6 Experiments

The goal of this section is twofold: First, we empirically study properties of a mean that are relevant for mean-based applications in data mining as well as for devising heuristics for mean computation. Second, we assess the performance of state-of-the-art heuristics.

### 6.1 Test Data

We used data sets derived from random walks and the 27 UCR data sets [7] listed in Table 1. Due to the running time of Algorithm 1, only data sets with short time series were considered. Random walks were used to conduct controlled experiments within the same problem domain in order to investigate mean properties and to assess the performance of heuristics more objectively



under different conditions.

A random walk  $x = (x_1, \dots, x_n)$  is of the form

$$\begin{aligned} x_1 &= \varepsilon_1, \\ x_i &= x_{i-1} + \varepsilon_i \quad \text{for all } 2 \leq i \leq n, \end{aligned}$$

where the  $\varepsilon_i$  are random numbers drawn from the normal distribution  $N(0, 1)$ . For the UCR data sets we merged the training and test sets. Time series within each UCR data set have the same length  $n$ . We restricted the experiments to data sets whose time series have length  $n \leq 150$ .

Unless otherwise stated, we generated samples according to the following procedures:

- $\mathcal{S}_{\text{rw}}$ : For every  $n \in \{10, 20, \dots, 100\}$ , we generated 1,000 pairs of random walks of length  $n$  giving a total of 10,000 samples of size  $k = 2$ .
- $\mathcal{S}_{\text{rw}}^k$ : For every  $k \in \{2, \dots, 6\}$ , we generated 1,000 samples consisting of  $k$  random walks of length  $n = 6$  giving a total of 5,000 samples.
- $\mathcal{S}_{\text{ucr}}$ : For every UCR data set, we randomly sampled 1,000 different pairs of time series giving a total of 27,000 samples of size  $k = 2$ .

## 6.2 Mean Properties

The first set of experiments studies mean properties relevant for devising heuristics and mean-based applications. Due to the running time of algorithm 1, the majority of experiments focused on properties of condensed means. The analysis of properties of arbitrary means are confined to a subset of tiny scale problems.

### 6.2.1 Uniqueness

Non-unique means can cause problems in theory and practice. For example, proving that sample means are consistent estimators of population means becomes more complicated for non-unique means. In addition, non-unique means can introduce undesired ambiguities into mean-based applications. For example, the performance of  $k$ -means clustering in DTW spaces [15, 26, 27, 33] does not only depend on the choice of initial means but also on the choice of re-computed means during optimization. Consequently, the extent of these difficulties depends—at least in principle—on the prevalence of non-unique means. In the following, we investigate the prevalence of non-unique condensed (and non-condensed) means.

**Setup:** We applied Algorithm 1 to all samples of type  $\mathcal{S}_{\text{rw}}$ ,  $\mathcal{S}_{\text{rw}}^k$ , and  $\mathcal{S}_{\text{ucr}}$ . The percentage  $P_{\text{ucm}}$  of unique condensed means as well as the average and maximum number of condensed means, denoted by  $\bar{\alpha}$  and  $\alpha^*$ , were recorded. For samples consisting of pairs of time series of length  $n \leq 40$ , we tested the existence of non-condensed means. These samples are of type  $\mathcal{S}_{\text{rw}}$  with  $k = 2$  and  $n \in \{10, 20, 30, 40\}$  and the 1,000 samples from the *ItalyPowerDemand* data set.

Condensed means				
data set	$P_{\text{ucm}}$	$\bar{\alpha}$	$\alpha^*$	$\bar{\delta}$
$S_{\text{rw}}$	100.0	1.0 ( $\pm 0.00$ )	1.0	-30.3 ( $\pm 25.79$ )
$S_{\text{rw}}^k$	100.0	1.0 ( $\pm 0.00$ )	1.0	-38.5 ( $\pm 18.91$ )
ItalyPowerDemand	99.4	1.0 ( $\pm 0.08$ )	2.0	-5.8 ( $\pm 11.29$ )
synthetic_control	100.0	1.0 ( $\pm 0.00$ )	1.0	-8.3 ( $\pm 16.87$ )
SonyAIBORobotSurface	43.9	2.2 ( $\pm 2.01$ )	16.0	-13.9 ( $\pm 5.53$ )
SonyAIBORobotSurfaceII	48.2	2.1 ( $\pm 2.03$ )	16.0	-12.9 ( $\pm 7.15$ )
ProximalPhalanxTW	80.2	1.2 ( $\pm 0.40$ )	2.0	-16.0 ( $\pm 13.79$ )
ProximalPhalanxOutlineCorrect	69.5	1.3 ( $\pm 0.46$ )	2.0	-19.9 ( $\pm 13.48$ )
ProximalPhalanxOutlineAgeGroup	78.6	1.2 ( $\pm 0.41$ )	2.0	-17.5 ( $\pm 13.15$ )
PhalangesOutlinesCorrect	82.5	1.2 ( $\pm 0.38$ )	2.0	-17.3 ( $\pm 13.24$ )
MiddlePhalanxTW	82.8	1.2 ( $\pm 0.38$ )	2.0	-15.6 ( $\pm 13.67$ )
MiddlePhalanxOutlineCorrect	88.3	1.1 ( $\pm 0.32$ )	2.0	-13.8 ( $\pm 13.76$ )
MiddlePhalanxOutlineAgeGroup	85.2	1.1 ( $\pm 0.36$ )	2.0	-14.7 ( $\pm 13.64$ )
DistalPhalanxTW	74.5	1.3 ( $\pm 0.44$ )	2.0	-17.9 ( $\pm 13.20$ )
DistalPhalanxOutlineCorrect	81.7	1.2 ( $\pm 0.39$ )	2.0	-16.4 ( $\pm 13.46$ )
DistalPhalanxOutlineAgeGroup	76.6	1.2 ( $\pm 0.42$ )	2.0	-17.6 ( $\pm 13.21$ )
TwoLeadECG	94.7	1.1 ( $\pm 0.26$ )	4.0	-9.3 ( $\pm 8.07$ )
MoteStrain	100.0	1.0 ( $\pm 0.00$ )	1.0	-30.9 ( $\pm 16.55$ )
ECG200	100.0	1.0 ( $\pm 0.00$ )	1.0	-8.8 ( $\pm 9.59$ )
MedicalImages	100.0	1.0 ( $\pm 0.00$ )	1.0	-32.1 ( $\pm 14.34$ )
Two_Patterns	0.5	7.9 ( $\pm 4.66$ )	16.0	-27.3 ( $\pm 6.40$ )
SwedishLeaf	100.0	1.0 ( $\pm 0.00$ )	1.0	0.9 ( $\pm 6.13$ )
CBF	100.0	1.0 ( $\pm 0.00$ )	1.0	2.8 ( $\pm 6.82$ )
FacesUCR	99.7	1.0 ( $\pm 0.05$ )	2.0	-8.2 ( $\pm 5.85$ )
FaceAll	99.8	1.0 ( $\pm 0.04$ )	2.0	-8.1 ( $\pm 5.72$ )
ECGFiveDays	97.4	1.0 ( $\pm 0.16$ )	2.0	-22.3 ( $\pm 13.87$ )
ECG5000	100.0	1.0 ( $\pm 0.00$ )	1.0	-21.5 ( $\pm 16.04$ )
Plane	100.0	1.0 ( $\pm 0.00$ )	1.0	-2.3 ( $\pm 4.68$ )
Gun_Point	100.0	1.0 ( $\pm 0.00$ )	1.0	-49.1 ( $\pm 14.02$ )
Non-condensed means				
data set	$P_{\text{ncm}}$			
$S_{\text{rw}}$ with $10 \leq n \leq 40$	0.3			
ItalyPowerDemand	2.1			
Notation				
$P_{\text{ucm}}$	percentage of unique condensed means			
$P_{\text{ncm}}$	percentage of non-condensed means			
$\bar{\alpha}$	average number of condensed means			
$\alpha^*$	maximum number of condensed means			
$\bar{\delta}$	average length-deviation			

Table 2: The table on the top shows properties on uniqueness and length of condensed means. Numbers in parentheses show standard deviations. The table below shows the percentage of non-condensed means for a restricted set of samples. The bottom table describes the notation used as column identifiers.

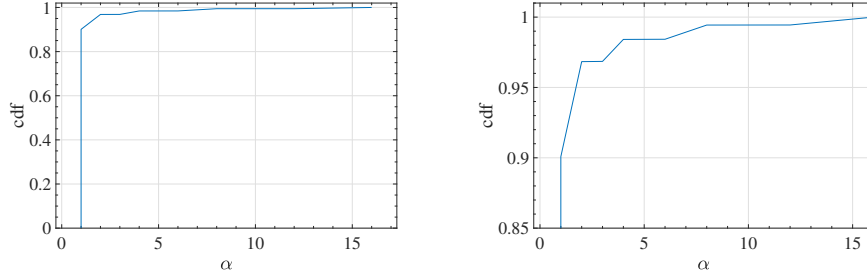


Figure 3: Estimated cumulative distribution function (cdf) of the number  $\alpha$  of condensed means over all 42,000 samples. The right plot shows an excerpt of the left plot with the cdf restricted to the interval  $[0.85, 1]$ .

**Results and Discussion:** Table 2 summarizes the results. Figure 3 shows the estimated cumulative distribution function of the number  $\alpha$  of condensed means over all 42,000 samples. We made the following observations:

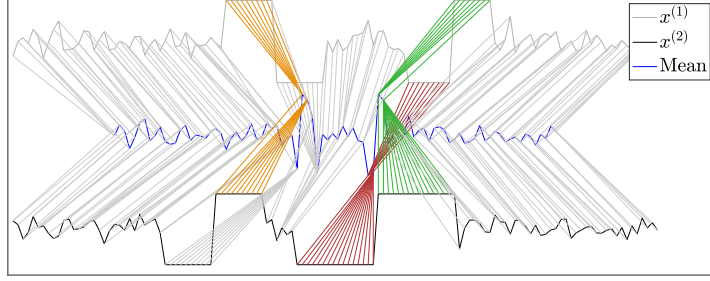
1. A condensed mean is unique for the majority of samples (see column  $P_{\text{ucm}}$  of Table 2 and Figure 3).
2. Non-condensed means occur exceptionally, that is, in less than 2.5% of all samples with short time series (see column  $P_{\text{ncm}}$  of Table 2).
3. The average and maximum number of condensed means is between one and two for all but four UCR data sets (see columns  $\bar{\alpha}$  and  $\alpha^*$  of Table 2).

These findings indicate that unique (condensed) means are more likely than non-unique means. The implications of the proposed observations are twofold: First, mean-based methods such as  $k$ -means clustering are less likely to be prone to problems caused by ambiguities. Second, the observations give rise to the hypothesis that a (condensed) mean of a sample is unique *almost everywhere* in a measure-theoretic sense. If this hypothesis is true, then the aforementioned problems caused by non-unique means are hopefully negligible in practice (we remark that optimal warping paths are unique almost everywhere [16] which might indicate that the above hypothesis holds).

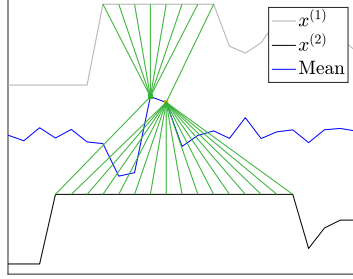
A notable exception is the (simulated) *Two\_Patterns* data set with only 0.5% of samples with a unique condensed mean. Figure 4 shows two typical time series of this data set and describes why their condensed mean is not uniquely determined. From the description it follows that warping of two plateaus can cause non-uniqueness. Such plateaus are common in the *Two\_Patterns* data set which explains the low percentage of unique condensed means.

### 6.2.2 Length

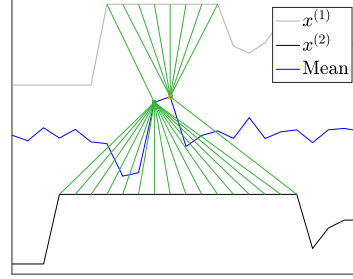
For a given sample, the current state-of-the-art heuristics [8, 15, 25, 30] approximate a mean by approximately solving a constrained DTW-MEAN problem.



(a)



(b)



(c)

Figure 4: Two time series  $x^{(1)}$  and  $x^{(2)}$  from the *Two\_Patterns* data set with eight different condensed means. Plot (a) shows an example of a mean and corresponding optimal warping paths between the mean and both sample time series. The three colored warpings show regions that cause non-uniqueness. Plots (b) and (c) show a detailed view of the green warpings. In (b) the mean of (a) is depicted. In (c) another mean is shown. In (b) and (c) both plateaus of  $x^{(1)}$  and  $x^{(2)}$  are aligned with only two points in the mean; one point is aligned with all but one points of the plateau of  $x^{(1)}$ , the other point is aligned with all but one points of the plateau of  $x^{(2)}$ . The only difference in (b) and (c) is the order of the two mean points. Each pair of warped plateaus provides two possibilities to construct a mean. Consequently, the three warped pairs of plateaus in (a) induce  $2^3 = 8$  condensed means.

The constrained problem restricts the set of feasible solutions to the subset  $\mathcal{T}_m$  of time series of length  $m$ . The parameter  $m$  is typically chosen within the range of the lengths of the sample time series. The two main questions are whether the subset  $\mathcal{T}_m$  contains a mean at all and what the best possible approximation we can achieve is when constraining the solution set to  $\mathcal{T}_m$ ? We start with the first question.

**Setup:** We applied Algorithm 1 to all samples of type  $\mathcal{S}_{\text{rw}}$ ,  $\mathcal{S}_{\text{rw}}^k$ , and  $\mathcal{S}_{\text{ucr}}$ . For every sample  $\mathcal{X}$ , the set of all condensed means was computed and the lengths

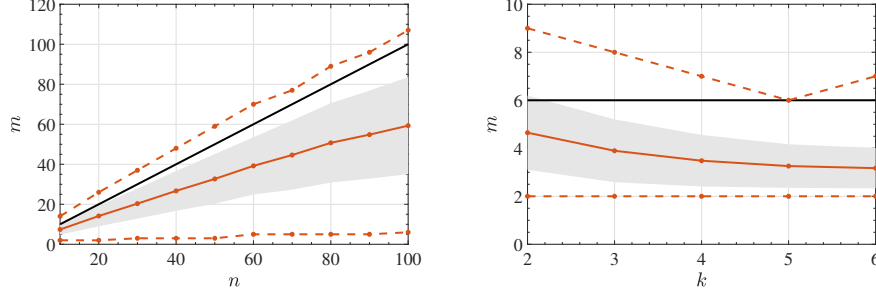


Figure 5: Average length  $m$  of condensed means (red lines) as functions of the length of sample time series  $n$  (left) and sample size  $k$  (right). The gray shaded areas depict the standard deviations and the dashed red lines the minimum and maximum length of condensed means. The black lines show the length of the sample time series.

were recorded. We computed the *length-deviation*

$$\delta_{\mathcal{X}} := 100 \cdot \frac{m_{\text{cm}} - n}{n},$$

where  $n$  is the length of the sample time series and  $m_{\text{cm}}$  is the average length of all condensed means of  $\mathcal{X}$ . Negative (positive) values of  $\delta_{\mathcal{X}}$  mean that  $m_{\text{cm}}$  is  $|\delta_{\mathcal{X}}|$  percent smaller (larger) than  $n$ .

**Results and Discussion:** Table 2 and Figure 5 summarize the results. To discuss the results, we consider the *constrained Fréchet variation*

$$F_m^* = \min_{z \in \mathcal{T}_m} F(z)$$

for every length  $m \in \mathbb{N}$ .

The first observation to be made is that different condensed means of the same sample have the same length for all 42,000 samples. Recall that non-condensed means occur only rarely (see column  $P_{\text{ncm}}$  of Table 2). Thus, means of a sample  $\mathcal{X}$  are typically condensed with identical length  $m_{\text{cm}}$ . Under these conditions, the function  $F_m^*$  has a unique minimum at  $m_{\text{cm}}$ . Thus, for a majority of samples, the solution space  $\mathcal{T}_m$  does not contain a mean for any  $m \neq m_{\text{cm}}$ . This gives rise to the question of how the length  $m_{\text{cm}}$  of condensed means is related to the length  $n$  of the sample time series. We observed the following:

1. The average length of condensed means is (substantially) less than the length of the sample time series for all but two UCR data sets (see column  $\bar{\delta}$  of Table 2).
2. The average length  $m_{\text{cm}}$  of condensed means increases with increasing length  $n$  of random walks (see Figure 5 (left)). The best linear fit in a squared error sense has slope 0.58 indicating that  $m_{\text{cm}}$  increases slower than  $n$ . No correlation between average length-deviations and length of sample time series could be observed across different UCR data sets.

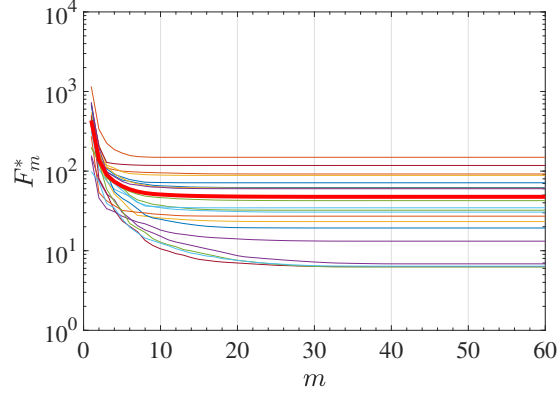


Figure 6: Constrained Fréchet variations  $F_m^*$  as functions of length  $m \in \{1, \dots, 60\}$  shown in lin-log scale. The functions  $F_m^*$  were computed for 1,000 samples consisting of two random walks of length  $n = 40$ . The plot depicts 20 randomly selected functions  $F_m^*$ . The highlighted red line is the average function  $\bar{F}_m^*$  over all 1,000 functions.






3. The average length  $m_{\text{cm}}$  of condensed means decreases with increasing sample size  $k$  of random walks (see Figure 5 (right)).

The above observations have the following implications: Most state-of-the-art heuristics were tested on sample time series with identical length  $n$  using  $m = n$  as predefined mean length [8, 25, 30]. The empirical findings suggest that the length of a condensed mean is substantially shorter than the standard setting  $m = n$ . Consequently, in most cases, the solution space  $\mathcal{T}_n$  of state-of-the-art methods does not contain a mean. Thus, setting the mean length equal to the length of the input time series introduces a *structural* error that can not be overcome by any solver of the corresponding constrained DTW-MEAN problem. The results suggest to consider mean lengths which are smaller than  $n$ . Note that in this case the computation time of mean-algorithms should decrease.

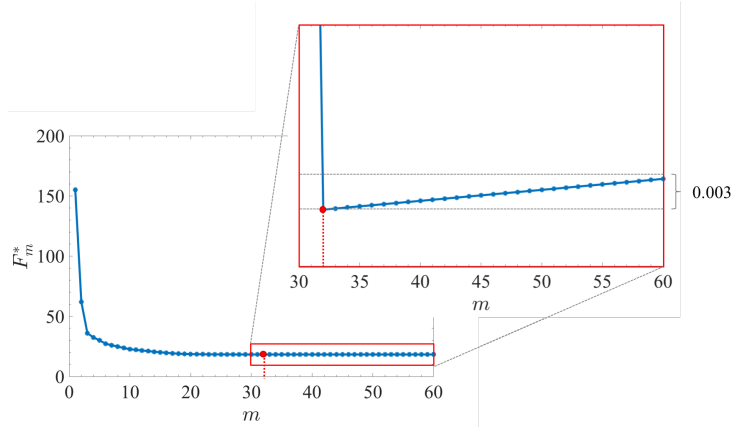
We now address the second main question, that is, how the choice of a mean length  $m$  affects the structural error  $\varepsilon_m := F_m^* - F_*$ , where  $F_m^* = \min_{z \in \mathcal{T}_m} F(z)$  is the constrained and  $F_* = \min_{z \in \mathcal{T}} F(z)$  is the unconstrained Fréchet variation. To estimate the structural error, we study the constrained Fréchet variation  $F_m^*$  as a function of  $m$ .

**Setup:** For every  $n \in \{5, 10, 15, \dots, 40\}$  we generated 1,000 pairs of random walks. We applied Algorithm 1 to all samples and computed the constrained Fréchet variations  $F_m^*$  for all  $m \in \{1, \dots, 60\}$ .

**Results and Discussion:** Figures 6 and 7 depict the main characteristics of the functions  $F_m^*$ . As before, condensed means of the same sample always had identical length  $m_{\text{cm}}$ . We use this result for our discussion on the shape

	$\mathcal{D}_1$	$\mathcal{D}_2$	
<b>monotonous</b>	96.0 %		76.6 %
			21.6 %
<b>constant</b>			1.2 %
<b>non-monotonous</b>	4.0 %		0.6 %
			

(a) Characteristic behavior of  $F_m^*$  on  $\mathcal{D}_1$  and  $\mathcal{D}_2$ .



(b) Typical shape of  $F_m^*$ .

Figure 7: Plot (a) sketches the main characteristics of all observed shapes of  $F_m^*$  on  $\mathcal{D}_1$  and  $\mathcal{D}_2$  together with their percentage frequency. Sketches exaggerate bumps and slopes to better highlight the main features. Plot (b) shows a typical example of the most common shape of  $F_m^*$  for a sample of two random walks of length  $n = 40$ . The function  $F_m^*$  strictly decreases on  $\mathcal{D}_1$  until it reaches its global minimum at  $m_{\text{cm}} = 32$  as highlighted by the red circle. Then the function  $F_m^*$  strictly increases on  $\mathcal{D}_2$ . The best linear fit of  $F_m^*$  on  $\mathcal{D}_2$  has a very small slope of approximately  $10^{-5}$ .

of  $F_m^*$  by decomposing its domain  $\mathcal{D} = \{1, \dots, 60\}$  into two sub-domains  $\mathcal{D}_1 = \{1, \dots, m_{\text{cm}}\}$  and  $\mathcal{D}_2 = \{m_{\text{cm}}, \dots, 60\}$ .

We observed the following typical behavior: The function  $F_m^*$  first rapidly decreases on  $\mathcal{D}_1$  until it reaches its global minimum at  $m_{\text{cm}}$  and then increases on  $\mathcal{D}_2$  with a small slope. The shape of  $F_m^*$  roughly resembles an exponential decay on  $\mathcal{D}_1$  followed by a linear tail on  $\mathcal{D}_2$ . Figure 7 (a) categorizes all observed shapes of  $F_m^*$  on  $\mathcal{D}_1$  and  $\mathcal{D}_2$ . The most common shape of  $F_m^*$  is strictly monotonous on  $\mathcal{D}_1$  (96.0%) and  $\mathcal{D}_2$  (98.2 %). Figure 7 (b) presents a typical

n	$a_{\text{med}}$	$a_{\text{avg}}$	$a_{\text{std}}$	$a_{\text{max}}$	$E_{\text{med}}$	$E_{\text{avg}}$	$E_{\text{std}}$	$E_{\text{max}}$
5	0.0034	0.12	0.36	3.78	0.423	4.43	9.59	67.40
10	0.0008	0.05	0.25	3.37	0.057	2.13	8.50	119.42
15	0.0005	0.05	0.24	2.80	0.028	1.49	7.30	92.65
20	0.0002	0.03	0.14	1.83	0.012	0.69	3.36	40.22
25	0.0002	0.02	0.14	3.15	0.008	0.47	2.77	37.70
30	0.0001	0.01	0.10	1.63	0.004	0.37	2.42	40.37
35	0.0001	0.01	0.06	1.35	0.003	0.20	2.22	52.44
40	0.0001	0.01	0.12	2.16	0.002	0.28	2.24	51.20

Table 3: Slopes  $a$  of best linear fits of the second part of  $F_m^*$  and error percentages  $E$  of the structural error. Subscripts med, avg, std, max refer to the median, average, standard deviation, and maximum, respectively, of 1,000 pairs of random walks of length  $n$ .

example of  $F_m^*$ . Non-monotonous or constant parts in a curve occur only rarely. These findings indicate that the function  $F_m^*$  is strictly convex in most cases. This implies that the further away  $m$  is from the global minimum  $m_{\text{cm}}$ , the larger the structural error  $\varepsilon_m$  is.

Table 3 summarizes the slopes of the best linear fits of  $F_m^*$  on  $\mathcal{D}_2$  and the error percentages  $E = 100 \cdot (F_n^* - F_*)/F_*$  of the constrained Fréchet variation at parameter  $m = n$ , where  $n$  is the length of the sample time series. The median slopes on  $\mathcal{D}_2$  are low  $a_{\text{med}} \approx 0.0034$  for  $n = 5$  and tend to further decrease with increasing length  $n$  to  $a_{\text{med}} \approx 0.0001$  for  $n = 40$ . As a consequence of the low median slopes, the median error percentages decline from  $E_{\text{med}} \approx 0.42$  to less than  $E_{\text{med}} \approx 0.01$  for increasing  $n$ . These findings indicate that the common practice of setting the parameter  $m$  of the constrained DTW-MEAN problem to the length  $n$  of the sample time series results in a low structural error on median. Due to the low median slopes on  $\mathcal{D}_2$ , solving the constrained DTW-MEAN problem instead of the unconstrained DTW-MEAN problem can in principle give good approximations for values  $m > m_{\text{cm}}$ . The average and maximum values for the slopes and error percentages, however, show that there are outliers that result in large structural error between 37% and 120%.

### 6.2.3 Shape

One problem that is often associated with dynamic time warping is shape averaging. Niennattrakul and Ratanamahatana [21] proposed a shape averaging algorithm that uses dynamic time warping, but differs from minimizing the Fréchet function. Sun et al. [34] claimed that their approach of minimizing the Fréchet function preserves shape characteristics. Cuturi and Blondel [8] criticized that solutions found by DBA may have a shape that is not representative for the sample time series.

It is well-known that dynamic time warping tends to warp similar shapes (subsequences) of two time series onto each other. Hence, at first glance one



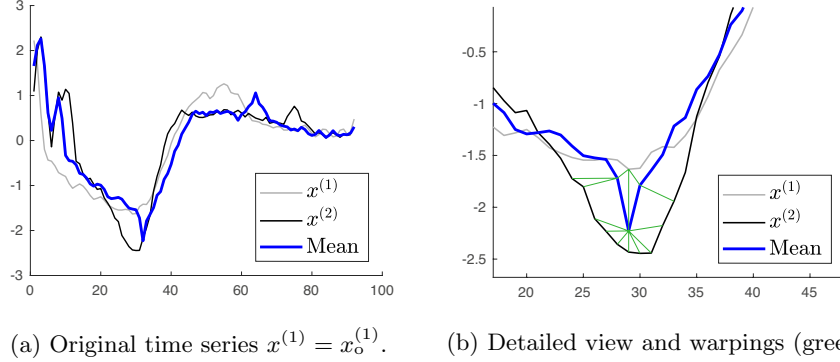


Figure 8: Mean with kink for two time series  $x^{(1)}$  and  $x^{(2)}$ . Plot (b) shifts the mean along the time axis for clarification.

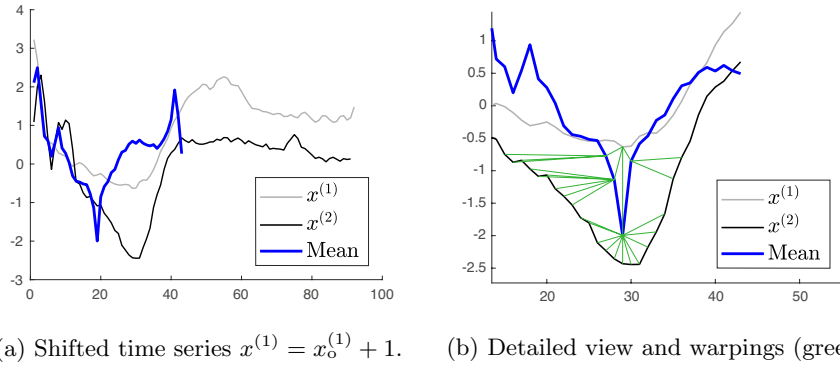


Figure 9: Mean for the two time series from Figure 8, but  $x^{(1)}$  is shifted along the  $y$ -axis. Plot (b) shifts the mean along the time axis for clarification.

may expect that a mean preserves shapes. However, the commonly and here considered definition of a mean is statistically motivated. We clarify in this section that a mean in DTW spaces does not preserve shapes in general.

**Setup:** We selected two sample time series from the UCR data set *ECG200*. Both time series were normalized with zero mean and standard deviation one. We applied Algorithm 1 and plotted the sample time series and their mean.

**Results and Discussion:** Figures 8 and 9 show a mean of the two sample time series  $x^{(1)}$  and  $x^{(2)}$ . In Figure 8 (a) the mean has a small kink around time index 30. This kink was also observed in the solution found by DBA in a similar experiment [8]. Cuturi and Blondel [8] criticized that this kink is not present in any of the sample time series. We now can conclude that presence of kinks is a

peculiarity of means in DTW spaces, rather than a property of the heuristics. Figure 8 (b) illustrates the warpings near the kink. The kink can be explained as follows: Both sample time series have broad valleys but on a different scale. In this case, the lowest point of the upper time series  $x^{(1)}$  is warped through mean elements onto multiple points of the lower time series  $x^{(2)}$ , which results in a kink in the mean.

To strengthen this effect, we shifted  $x^{(1)}$  upwards in Figure 9. As expected, the kink becomes larger, because more points of  $x^{(2)}$  are warped onto a single point of  $x^{(1)}$  through mean elements. Moreover, the length of the mean reduces significantly. We conclude that a mean is not necessarily a suitable representative of the sample time series in terms of shape. Normalization of the sample time series may help to reduce this effect.

#### 6.2.4 Summary

The main findings of our first series of experiments are:

1. A mean of a sample is usually condensed, unique and shorter than the sample time series.
2. Choosing a mean length that is larger than the optimal value induces only a small structural error in most cases.
3. A mean does not necessarily preserve shapes of the sample time series.

The first two points show that in practice one might hope not to encounter critical problems by non-unique means or by selecting a wrong mean length. The third point suggests that smoother shapes of a mean require different objective functions to be minimized or different distance measures.

### 6.3 Performance of Heuristics

The goal of this series of experiments is to assess the performance of state-of-the-art heuristics in terms of minimizing the Frchet variation.

#### 6.3.1 Experimental Setup

**Algorithms:** Table 4 lists the mean-algorithms considered in this experiment.

We implemented EDP and MAL (for two time series) in Java. For SDTW and BSG we used the python implementation provided by Blondel [3]. For DBA and SSG we used the implementation in Java [31].

**Setup:** All six algorithms were applied to 27,000 samples of type  $\mathcal{S}_{\text{ucr}}$  and to 10,000 samples of type  $\mathcal{S}_{\text{rw}}$ . For the 5,000 samples of type  $\mathcal{S}_{\text{rw}}^k$  all mean algorithms but MAL were applied.

EDP and MAL required no parameter optimization. For the other algorithms, the following settings were used: We optimized the algorithms on every sample using different parameter configurations and reported the best result.

Algorithm	Acr.	Ref.
Exact dynamic programming	EDP	Alg. 1
Multiple alignment	MAL	[15, 24]
DTW barycenter averaging	DBA	[25]
Soft-dtw	SDTW	[8]
Batch subgradient	BSG	[8, 30]
Stochastic subgradient	SSG	[30]

Table 4: List of algorithms compared in our experiments.

Every configuration was composed of an initialization method and an optional parameter selection. As initialization we used (i) the arithmetic mean of the sample, (ii) a random time series from the sample, and (iii) a random time series drawn from a normal distribution with zero mean and standard deviation one. DBA and BSG required no additional parameter selection. For SDTW, we selected the best smoothing parameter  $\gamma \in \{0.001, 0.01, 0.1, 1\}$  and for SSG the best initial step size  $\eta_0 \in \{0.25, 0.2, 0.15, 0.1\}$ . The step size of SSG was linearly decreased from  $\eta_0$  to  $\eta_0/10$  within the first 100 epochs and then remained constant for the remaining 100 epochs (one epoch is a full pass through the sample). Thus, for DBA and BSG, we picked the best result from the three initialization methods for each sample, and for SDTW and SSG we picked the best result from twelve parameter configurations (three initializations  $\times$  four parameter values). The length of the mean was set beforehand to the length of the sample time series (time series of a sample always have identical length). For every sample and every parameter configuration, all algorithms terminated after 200 epochs at the latest. The tolerance parameter of SDTW and BSG was set to  $\varepsilon = 10^{-6}$ .

**Performance Metric:** We consider error percentages to assess the solution quality. The error percentage of algorithm  $A$  for sample  $\mathcal{S}$  is defined by

$$E = 100 \cdot (F_A - F_*)/F_*,$$

where  $F_A$  is the solution obtained by algorithm  $A$  and  $F_* = \min_{z \in \mathcal{T}} F(z)$  is the optimal solution obtained by EDP.

### 6.3.2 Results and Discussion

Figures 10 and 11 depict the performance profiles of the algorithms and quantitative summaries of their error percentages for samples of type  $\mathcal{S}_{\text{ucr}}$  and  $\mathcal{S}_{\text{rw}}$ , respectively.<sup>2</sup> The performance profiles exhibit the same pattern in general but differ by some shifts in the curves which is due to different data domains. (Appendix B presents the results in more detail.)

<sup>2</sup>Appendix A describes performance profiles in more detail.

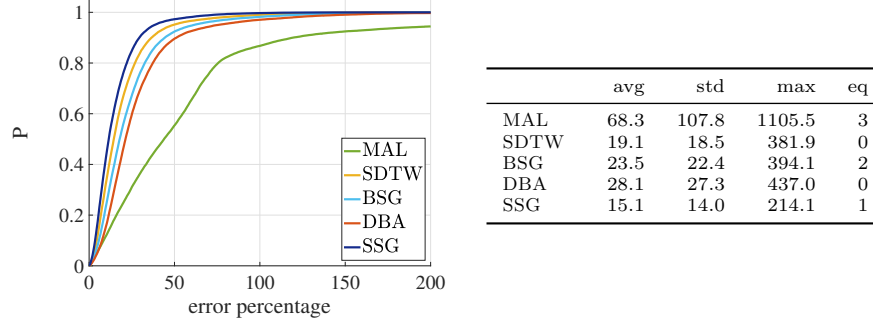


Figure 10: Results for  $\mathcal{S}_{\text{ucr}}$ -samples. **Left:** Performance profiles of all heuristics (a larger area under the curve indicates better performance). The performance profile of the exact algorithm EDP is the constant line  $P = 1$  and therefore not highlighted. The other performance profiles are truncated at 200% error for the sake of presentation. A point  $(E_A, P_A)$  on the curve of an algorithm  $A$  states that solutions obtained by  $A$  deviate by at most  $E_A$  percent from the optimal solution with probability  $P_A$  (estimated over 27,000 different samples). **Right:** Average error percentage (avg), standard deviation (std), and maximum error percentage (max) of the heuristics. The last column shows how often an optimal solution was found by the heuristic.

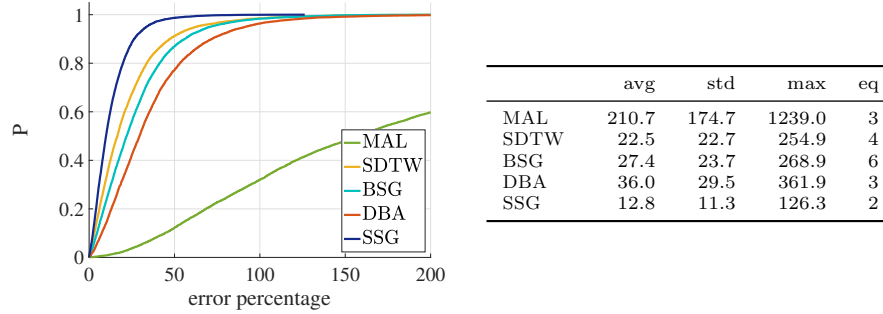


Figure 11: Results for  $\mathcal{S}_{\text{rw}}$ -samples. **Left:** Performance profiles of all heuristics. **Right:** Average error percentage (avg), standard deviation (std), and maximum error percentage (max) of the heuristics. The last column shows how often an optimal solution was found by the heuristic.

**Performance of Multiple Alignment (MAL).** Section 3.2 refutes the claim that DTW-MEAN can be solved by a multiple alignment approach. The remaining question is to which extent MAL fails to solve DTW-MEAN.

Figures 10 and 11 show that MAL exactly solved DTW-MEAN in only three out of 27,000  $\mathcal{S}_{\text{ucr}}$ -samples and another three out of 10,000  $\mathcal{S}_{\text{rw}}$ -samples. The average error percentages of MAL on  $\mathcal{S}_{\text{ucr}}$  and  $\mathcal{S}_{\text{rw}}$  are 68.3 and 210.7, respectively. The error percentage of MAL is larger than 50% for more than 40% of all  $\mathcal{S}_{\text{ucr}}$ -samples and for roughly 90% of all  $\mathcal{S}_{\text{rw}}$ -samples. These observations

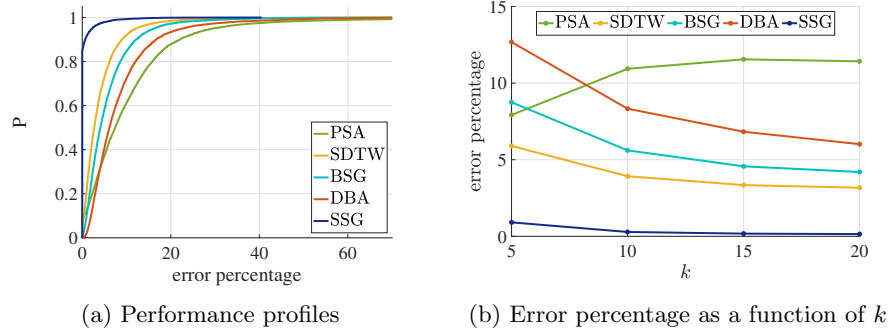


Figure 12: Results on samples  $\mathcal{S}_{\text{ucr}}^k$ . Plot (a) shows the performance profiles and plot (b) the average error percentages as a function of the sample size  $k$ .

suggest that MAL is far from being optimal or competitive. This may explain why algorithms based on pairwise mean computation via MAL such as NLAFF [13] and PSA [21] are not competitive [25, 32].

**Performance of State-of-the-Art Heuristics.** We discuss the performance of SDTW, BSG, DBA, and SSG. The best and most robust method is the stochastic subgradient method (SSG) with 15.1% and 12.8% average error percentage on  $\mathcal{S}_{\text{ucr}}$ - and  $\mathcal{S}_{\text{rw}}$ -samples, respectively. Standard deviation and maximum error percentages of SSG are lowest among all other heuristics. Nevertheless, the solution quality of all heuristics is rather poor leaving much space for improvements.

**Performance of Progressive Alignment (PSA).** Progressive alignment methods combine pairwise averages beginning with the two most similar time series. In each step, two time series are averaged and replaced by their weighted average. The weights correspond to the number of sample time series involved in creating the time series. The currently best performing progressive alignment method for time series averaging is called prioritized shape averaging (PSA) [21]. This method applies a variant of MAL for pairwise averaging. Empirical results suggest that PSA is not competitive to DBA [25, 32]. We evaluate the performance of a modified variant of PSA using EDP for pairwise averaging.

As test data, we used samples of type  $\mathcal{S}_{\text{ucr}}^k$ : For every UCR data set and every sample size  $k \in \{5, 10, 15, 20\}$ , we randomly selected 100 samples of  $k$  time series giving a total of 10,800 samples. We applied the modified PSA algorithm and the algorithms SDTW, BSG, DBA, and SSG on all 10,800 samples (the setup is the same as described in Section 6.3.1).

Figure 12 summarizes the results. The performance profiles exhibit similar patterns for  $\mathcal{S}_{\text{ucr}}^k$ -samples as those obtained for  $\mathcal{S}_{\text{ucr}}$ -samples (Figure 10). PSA is the worst performing heuristic. Figure 12 (b) shows that the average error percentage of PSA increases with increasing sample size  $k$ , whereas the average

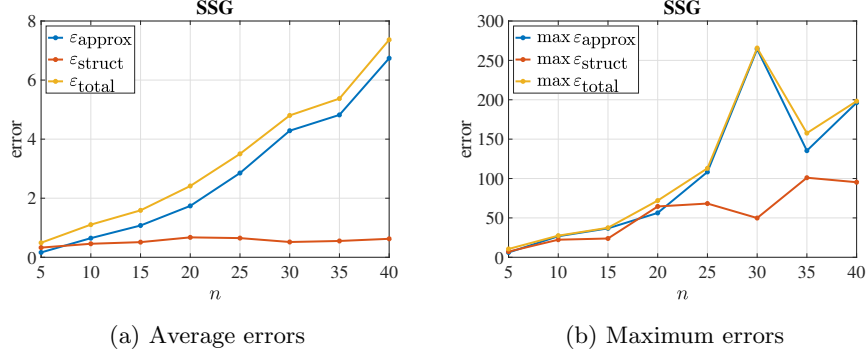


Figure 13: Error decomposition of SSG on pairs of random walks with  $5 \leq n \leq 40$ . Both plots show the total error  $\varepsilon_{\text{total}}$  (yellow), the approximation error  $\varepsilon_{\text{approx}}$  (blue), and the structural error  $\varepsilon_{\text{struct}}$  (red). Plot (a) averages and plot (b) maximizes the errors over 1,000 samples at each  $n$ .

error percentage of all other heuristics exhibit the opposite trend. These results show that the weak performance of progressive alignment methods is not explained by suboptimal pairwise mean computation.

**Error Decomposition.** The previous results revealed a poor solution quality of state-of-the-art heuristics. In Section 6.2.2, we have seen that choosing a fixed mean length  $m$  introduces a certain structural error. The error  $\varepsilon$  of an algorithm  $A$  can thus be written as

$$\varepsilon = F_A - F_* = \underbrace{(F_A - F_m^*)}_{\text{approximation error}} + \underbrace{(F_m^* - F_*)}_{\text{structural error } \varepsilon_m},$$

where  $F_A$  is the value returned by algorithm  $A$ ,  $F_* = \min_{z \in \mathcal{T}} F(z)$  is the Fréchet variation, and  $F_m^* = \min_{z \in \mathcal{T}_m} F(z)$  is the constrained Fréchet variation. The approximation error  $F_A - F_m^*$  represents the inability of algorithm  $A$  to solve the constrained DTW-MEAN problem.

We conducted some experiments to assess the contribution of the approximation and the structural error to the total error of the considered heuristics. For every length  $n \in \{5, 10, 15, \dots, 40\}$ , we generated 1,000 samples of pairs of random walks of length  $n$  giving a total of 8,000 samples. We applied SDTW, BSG, DBA, and SSG on all samples using the same setting as described in Section 6.3.1.

Figure 13 depicts the results of the best performing heuristic (SSG). The other three heuristics exhibited the same behavior with identical structural error but different approximation errors. We made the following observations:

1. The total error is dominated by the approximation error (for larger  $n$ ).
2. Both total and approximation error increase with increasing length  $n$ .

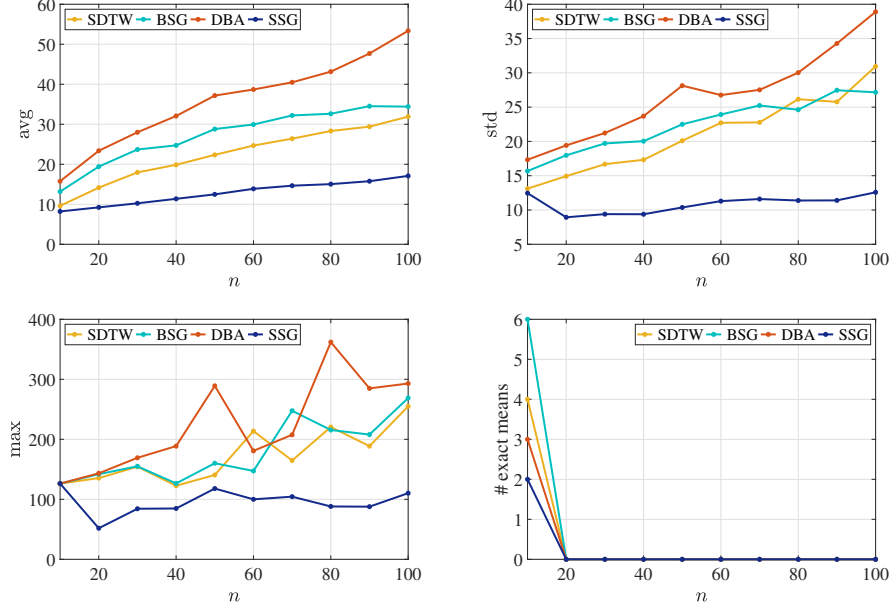


Figure 14: Results on samples of type  $\mathcal{S}_{\text{rw}}$  consisting of  $k = 2$  random walks of length  $n$ . Average error percentage (avg), standard deviation (std), maximum error percentage (max) and number of exact solutions are plotted as a function of the length  $n$ .

3. The structural error is independent of the length  $n$ .
4. In exceptional cases the structural error can be large.

These observations indicate that the total error is mainly caused by the approximation error. In contrast, the structural error introduced by fixing the mean length becomes negligible for longer sample time series and is large only in very few cases (as concluded in Section 6.2.4). The results suggest that devising heuristics should mainly focus on improving the approximation error.

**Effect of Sample Time Series Length  $n$ .** We investigated how the performance of SDTW, BSG, DBA, and SSG is related to the length  $n$  of the sample time series for random walk samples of type  $\mathcal{S}_{\text{rw}}$ .

Figure 14 summarizes the results. The trend is that the performance of all heuristics with respect to solution quality (avg) and robustness (std, max) deteriorates with increasing length  $n$ . Since we observed before that the structural error is independent of  $n$  and likely to be negligible, we conclude that the heuristics perform worse in solving the constrained DTW-MEAN problem for larger  $n$ . No state-of-the-art method found an optimal solution for length  $n \geq 20$ .

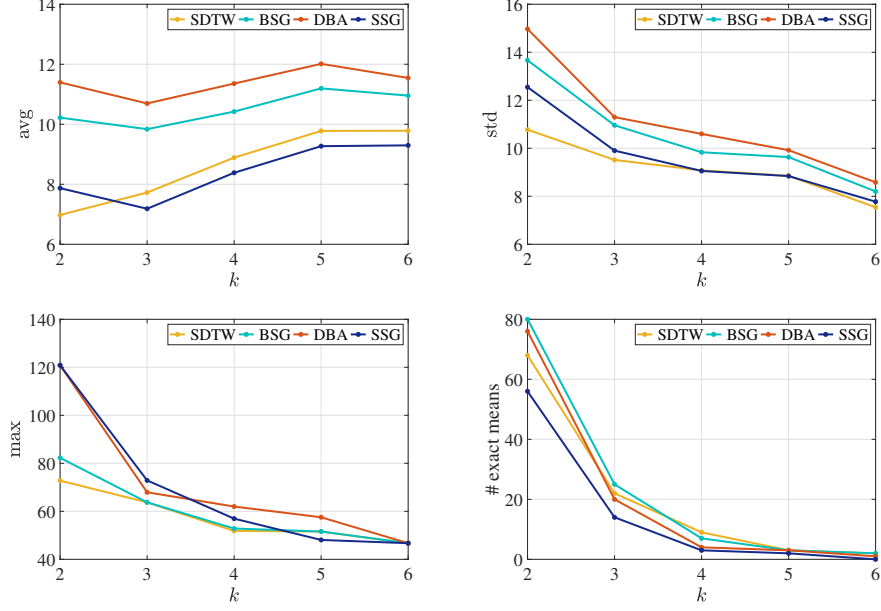


Figure 15: Results on samples  $\mathcal{S}_{\text{rw}}^k$  each of which consists of  $k$  random walks. Average error percentage (avg), standard deviation (std), maximum error percentage (max) and number of exact solutions are plotted as a function of the sample size  $k$ .

**Effect of Sample Size  $k$ .** We analyzed how the performance of SDTW, BSG, DBA, and SSG depends on the sample size  $k$  for samples of type  $\mathcal{S}_{\text{rw}}^k$ .

Figure 15 shows the results. We observed no clear trend with respect to average error-percentage and a decline of the number of exact solutions found by all heuristics. In contrast, robustness (that is, standard deviation and maximum error percentage) of all four heuristics substantially improved with increasing  $k$ .

### 6.3.3 Summary

The main findings of our second series of experiments are:

1. State-of-the-art heuristics do not perform well on average due to a large approximation error.
2. MAL is not competitive to state-of-the-art heuristics (for  $k = 2$ ).
3. PSA using exact weighted means is not competitive to state-of-the-art heuristics.

Concerning running times, our exact dynamic program is clearly slower than state-of-the-art heuristics (for example, our implementation required 66 seconds on average to compute a mean of two time series of length 100, which is slower by a factor of  $10^5$  compared to DBA).



## 7 Conclusion

We developed an exact exponential-time algorithm for WEIGHTED DTW-MEAN and conducted extensive experiments (on small samples containing up to 6 time series of different lengths up to 150) to investigate characteristics of an exact mean and to benchmark existing state-of-the-art heuristics in terms of their solution quality. On the positive side, we found that a mean is often unique and that fixing the length of a mean in advance does usually not affect the solution quality much. On the negative side, we showed that basically all heuristics perform poorly in the worst case. These findings urge for devising better algorithms to solve DTW-MEAN. Our empirical observations (e.g. concerning the typical mean length) might prove useful for this. As a side result, we also developed a polynomial-time algorithm for binary time series.

We conclude with some challenges for future research. From an algorithmic point of view it is interesting to investigate whether one can extend the polynomial-time solvability of BINARY DTW-MEAN to larger “alphabet” sizes; already the case of alphabet size three is open. Another open question is whether WEIGHTED DTW-MEAN is polynomial-time solvable for time series of constant lengths (maybe it is even fixed-parameter tractable with respect to the maximum length). Finally, we wonder whether there are other practically relevant restrictions of DTW-MEAN that make the problem more tractable, for example, fixing the length of a mean. Another example, also motivated from a practical point of view, is to compute means with a given size of time windows (also known as the Sakoe-Chiba band [29]). On a high level, a time window constrains the warping path to only select tuples within a specified range. On an applied side, our experimental results strongly motivate the search for further improved, more “robust” heuristics.

**Acknowledgements.** This work was supported by the Deutsche Forschungsgemeinschaft under grants JA 2109/4-1 and NI 369/13-2, and by a Feodor Lynen return fellowship of the Alexander von Humboldt Foundation. The work on the theoretical part of this paper started at the research retreat of the Algorithmics and Computational Complexity group, TU Berlin, held at Boiensdorf, Baltic Sea, April 2017, with MB, TF, VF, and RN participating.

## References

- [1] W. H. Abdulla, D. Chow, and G. Sin. Cross-words reference template for dtw-based speech recognition systems. In *Proceeding of the IEEE Conference on Convergent Technologies for the Asia-Pacific Region (TENCON 2003)*, volume 4, pages 1576–1579. IEEE, 2003. 1
- [2] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time-series clustering – A decade review. *Information Systems*, 53:16–38, 2015. 4

- [3] M. Blondel. Python implementation of soft-DTW. <https://github.com/mblondel/soft-dtw>, 2017. 26
- [4] P. Bonizzoni and G. Della Vedova. The complexity of multiple sequence alignment with SP-score that is a metric. *Theoretical Computer Science*, 259(1–2):63–79, 2001. 4
- [5] K. Bringmann and M. Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS '15)*, pages 79–97. IEEE, 2015. 3
- [6] L. Bulteau, V. Froese, and R. Niedermeier. Hardness of consensus problems for circular strings and time series averaging. *CoRR*, abs/1804.02854, 2018. 2, 4
- [7] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The UCR time series classification archive, 2015. URL [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/). 16
- [8] M. Cuturi and M. Blondel. Soft-DTW: a differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning (ICML '17)*, volume 70 of *Proceedings of Machine Learning Research*, pages 894–903. PMLR, 2017. 2, 19, 22, 24, 25, 27
- [9] E. Dolan and J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. 37
- [10] F. Fahiman, J. C. Bezdek, S. M. Erfani, C. Leckie, and M. Palaniswami. Fuzzy c-Shape: A new algorithm for clustering finite time series waveforms. In *Proceeding of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2017. 4
- [11] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'Institut Henri Poincaré*, pages 215–310, 1948. 2
- [12] O. Gold and M. Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP '17)*, volume 80 of *LIPIcs*, pages 25:1–25:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. 3
- [13] L. Gupta, D. L. Molfese, R. Tammana, and P. G. Simos. Nonlinear alignment and averaging for estimating the evoked potential. *IEEE Transactions on Biomedical Engineering*, 43(4):348–356, 1996. 5, 29
- [14] D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997. 4

- [15] V. Hautamaki, P. Nykanen, and P. Franti. Time-series clustering by approximate prototypes. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pages 1–4. IEEE, 2008. 1, 2, 4, 17, 19, 27
- [16] B. Jain and D. Schultz. Optimal warping paths are unique for almost every pair of time series. *arXiv:1705.05681*, 2017. 19
- [17] B. Jain and D. Schultz. A reduction theorem for the sample mean in dynamic time warping spaces. *CoRR*, abs/1610.04460v3, 2018. 3, 10, 11
- [18] P.-F. Marteau. Times series averaging and denoising from a probabilistic perspective on time-elastic kernels. *CoRR*, abs/1611.09194, 2016. 4
- [19] M. Morel, C. Achard, R. Kulpa, and S. Dubuisson. Time-series averaging using constrained dynamic time warping with tolerance. *Pattern Recognition*, 74:77–89, 2018. 1
- [20] F. Nicolas and E. Rivals. Hardness results for the center and median string problems under the weighted and unweighted edit distances. *Journal of Discrete Algorithms*, 3(2):390–415, 2005. 4
- [21] V. Niennattrakul and C. A. Ratanamahatana. Shape averaging under time warping. In *Proceedings of the 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '09)*, volume 02, pages 626–629, 2009. 5, 24, 29
- [22] T. Oates, L. Firoiu, and P. R. Cohen. Clustering time series with hidden Markov models and dynamic time warping. In *Proceedings of the IJCAI'99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21. Citeseer, 1999. 1
- [23] J. Paparrizos and L. Gravano. Fast and accurate time-series clustering. *ACM Transactions on Database Systems*, 42(2):8:1–8:49, 2017. 4
- [24] F. Petitjean and P. Gançarski. Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment. *Theoretical Computer Science*, 414(1):76–91, 2012. 2, 4, 5, 27
- [25] F. Petitjean, A. Ketterlin, and P. Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011. 2, 4, 6, 19, 22, 27, 29
- [26] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *Proceeding of the 2014 IEEE International Conference on Data Mining (ICDM'14)*, pages 470–479. IEEE, 2014. 17

- [27] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowledge and Information Systems*, 47(1):1–26, 2016. 1, 4, 17
- [28] L. Rabiner and J. Wilpon. Considerations in applying clustering techniques to speaker-independent word recognition. *The Journal of the Acoustical Society of America*, 66(3):663–673, 1979. 1
- [29] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. 33
- [30] D. Schultz and B. Jain. Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces. *Pattern Recognition*, 74 (Supplement C):340–358, 2018. 2, 19, 22, 27
- [31] D. Schultz and B. J. Jain. Sample mean algorithms for averaging in dynamic time warping spaces, 2016. URL <https://doi.org/10.5281/zenodo.216233>. 26
- [32] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier. Progressive and iterative approaches for time series averaging. In *Proceedings of the 1st International Conference on Advanced Analytics and Learning on Temporal Data-Volume 1425*, pages 111–117. CEUR-WS.org, 2015. 29
- [33] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier. Generalized  $k$ -means-based clustering for temporal data under weighted and kernel time warp. *Pattern Recognition Letters*, 75:63–69, 2016. 1, 2, 17
- [34] T. Sun, H. Liu, H. Yu, and C. P. Chen. Degree-pruning dynamic programming approaches to central time series minimizing dynamic time warping distance. *IEEE Transactions on Cybernetics*, 47(7):1719–1729, 2017. 24

## A Performance Profiles

To compare the performance of the mean algorithms, we used a slight variation of the performance profiles proposed by Dolan and Moré [9]. A performance profile is a cumulative distribution function for a performance metric. Here, the chosen performance metric is the error percentage from the exact solution.

To define a performance profile, we assume that  $\mathbb{A}$  is a set of mean algorithms and  $\mathbb{S}$  is a set of samples each of which consists of  $k$  time series. For each sample  $\mathcal{X} \in \mathbb{X}$  and each mean algorithm  $A \in \mathbb{A}$ , we define  $E_{A,\mathcal{S}}$  as the error percentage obtained by applying algorithm  $A$  on sample  $\mathcal{S}$ . The performance profile of algorithm  $A$  over all samples  $\mathcal{S} \in \mathbb{S}$  is the empirical cumulative distribution function defined by

$$P_A(\tau) = \frac{1}{|\mathbb{S}|} |\{\mathcal{S} \in \mathbb{S} : E_{A,\mathcal{S}} \leq \tau\}|$$

for all  $\tau \geq 0$ . Thus,  $P_A(\tau)$  is the estimated probability that the error percentage of algorithm  $A$  is at most  $\tau$ . The value  $P_A(0)$  is the estimated probability that algorithm  $A$  finds an exact solution.

## B Results

UCR data set	MAL	SDTW	BSG	DBA	SSG
ItalyPowerDemand	26.1	13.8	17.6	19.2	<b>10.5</b>
synthetic_control	54.7	22.1	26.5	29.4	<b>17.9</b>
SonyAIBORobotSurface	33.6	20.6	26.0	28.9	<b>17.2</b>
SonyAIBORobotSurfaceII	28.4	20.0	23.8	26.7	<b>17.8</b>
ProximalPhalanxTW	35.2	11.8	15.3	17.1	<b>10.6</b>
ProximalPhalanxOutlineCorrect	35.1	13.4	16.9	18.4	<b>11.9</b>
ProximalPhalanxOutlineAgeGroup	36.9	11.7	15.3	17.4	<b>10.8</b>
PhalangesOutlinesCorrect	49.7	15.2	19.5	22.3	<b>13.2</b>
MiddlePhalanxTW	36.8	12.5	14.8	17.3	<b>10.0</b>
MiddlePhalanxOutlineCorrect	38.7	13.3	16.2	18.4	<b>10.8</b>
MiddlePhalanxOutlineAgeGroup	37.0	12.8	15.1	17.6	<b>9.9</b>
DistalPhalanxTW	41.9	12.0	15.3	18.6	<b>10.5</b>
DistalPhalanxOutlineCorrect	48.4	13.4	17.4	21.2	<b>11.6</b>
DistalPhalanxOutlineAgeGroup	43.6	12.4	15.5	18.6	<b>10.6</b>
TwoLeadECG	44.4	12.0	15.3	17.7	<b>9.6</b>
MoteStrain	79.7	17.5	23.3	29.2	<b>12.6</b>
ECG200	51.4	20.5	23.5	26.5	<b>14.2</b>
MedicalImages	84.6	16.8	21.0	27.6	<b>10.6</b>
Two_Patterns	184.0	73.5	82.8	92.8	<b>56.4</b>
SwedishLeaf	56.2	18.0	23.2	30.6	<b>13.4</b>
CBF	28.5	31.2	33.8	36.9	<b>26.4</b>
FacesUCR	38.4	22.5	27.6	30.2	<b>18.0</b>
FaceAll	37.7	22.6	27.6	30.2	<b>18.3</b>
ECGFiveDays	41.7	11.0	13.0	15.9	<b>8.5</b>
ECG5000	77.8	15.3	18.0	22.4	<b>9.9</b>
Plane	83.1	13.6	18.6	30.0	<b>9.7</b>
Gun_Point	489.8	36.2	52.0	77.0	<b>27.4</b>
<b>Total</b>	<b>68.3</b>	<b>19.1</b>	<b>23.5</b>	<b>28.1</b>	<b>15.1</b>

Table 5: Average error percentage of the five heuristics on samples  $\mathcal{S}_{\text{ucr}}$  of size  $k = 2$  grouped by UCR data set. Averages were taken over 1,000 samples randomly drawn from each data set.

$n$	MAL	SDTW	BSG	DBA	SSG
10	64.3	9.6	13.2	15.8	<b>8.2</b>
20	107.4	14.2	19.4	23.4	<b>9.2</b>
30	152.1	18.0	23.7	28.0	<b>10.2</b>
40	179.1	19.9	24.7	32.1	<b>11.4</b>
50	206.5	22.4	28.8	37.2	<b>12.5</b>
60	231.2	24.7	29.9	38.7	<b>13.9</b>
70	253.2	26.4	32.2	40.5	<b>14.6</b>
80	269.7	28.3	32.6	43.2	<b>15.0</b>
90	303.9	29.4	34.5	47.7	<b>15.8</b>
100	339.8	31.9	34.4	53.4	<b>17.1</b>
<b>Total</b>	<b>210.7</b>	<b>22.5</b>	<b>27.4</b>	<b>36.0</b>	<b>12.8</b>

Table 6: Average error percentage of the five heuristics on  $\mathcal{S}_{\text{rw}}$ -samples of size  $k = 2$  grouped by length  $n$  of random walks. Averages were taken over 1,000 samples for each length  $n$ .

		avg	std	max	eq
<hr/>					
$k = 2$					
	SDTW	7.0	10.8	72.8	68.0
	BSG	10.2	13.7	82.3	80.0
	DBA	11.4	15.0	120.8	76.0
	SSG	7.9	12.5	120.8	56.0
<hr/>					
$k = 3$					
	SDTW	7.7	9.5	63.7	22.0
	BSG	9.8	11.0	63.7	25.0
	DBA	10.7	11.3	67.9	20.0
	SSG	7.2	9.9	72.9	14.0
<hr/>					
$k = 4$					
	SDTW	8.9	9.1	51.8	9.0
	BSG	10.4	9.8	52.8	7.0
	DBA	11.4	10.6	62.0	4.0
	SSG	8.4	9.1	57.0	3.0
<hr/>					
$k = 5$					
	SDTW	9.8	8.9	51.6	3.0
	BSG	11.2	9.6	51.6	3.0
	DBA	12.0	9.9	57.5	3.0
	SSG	9.3	8.8	48.1	2.0
<hr/>					
$k = 6$					
	SDTW	9.8	7.5	46.7	2.0
	BSG	11.0	8.2	46.7	2.0
	DBA	11.5	8.6	46.7	1.0
	SSG	9.3	7.8	46.7	0.0
<hr/>					

Table 7: Average error percentage of the five heuristics on  $\mathcal{S}_{\text{rw}}^k$ -samples of varying sample size  $k$ . Length of random walks was  $n = 6$ . Averages were taken over 1,000 samples for each sample size  $k$ .