### Yuya Higashikawa

School of Social Information Science, University of Hyogo, Kobe, Japan higashikawa@sis.u-hyogo.ac.jp

### Naoki Katoh

School of Social Information Science, University of Hyogo, Kobe, Japan naoki.katoh@gmail.com

### Guohui Lin

Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada guohui@ualberta.ca

### Eiji Miyano

Department of Artificial Intelligence, Kyushu Institute of Technology, Iizuka, Japan miyano@ces.kyutech.ac.jp

### Suguru Tamaki

School of Social Information Science, University of Hyogo, Kobe, Japan tamak@sis.u-hyogo.ac.jp

### Junichi Teruyama

School of Social Information Science, University of Hyogo, Kobe, Japan junichi.teruyama@sis.u-hyogo.ac.jp

### Binhai Zhu

Gianforte School of Computing, Montana State University, Bozeman, MT 59717, USA bhz@montana.edu

### — Abstract

Throughout this paper, a persistence diagram  $\mathcal{P}$  is composed of a set P of planar points (each corresponding to a topological feature) above the line Y = X, as well as the line Y = X itself, i.e.,  $\mathcal{P} = P \cup \{(x, y) | y = x\}$ . Given a set of persistence diagrams  $\mathcal{P}_1, ..., \mathcal{P}_m$ , for the data reduction purpose, one way to summarize their topological features is to compute the center C of them first under the bottleneck distance. Here we mainly focus on the two discrete versions when points in C could be selected with or without replacement from  $P_i$ 's. (We will briefly discuss the continuous case, i.e., points in C are arbitrary, which turns out to be closely related to the 3-dimensional geometric assignment problem). For technical reasons, we first focus on the case when  $|P_i|$ 's are all the same (i.e., all have the same size n), and the problem is to compute a center point set C under the bottleneck matching distance. We show, by a non-trivial reduction from the Planar 3D-Matching problem, that this problem is NP-hard even when m = 3 diagrams are given. This implies that the general center problem for persistence diagrams under the bottleneck distance, when  $P_i$ 's possibly have different sizes, is also NP-hard when  $m \ge 3$ . On the positive side, we show that this problem is polynomially solvable when m = 2 and admits a factor-2 approximation for  $m \ge 3$ . These positive results hold for any  $L_p$  metric when  $P_i$ 's are point sets of the same size, and also hold for the case when  $P_i$ 's have different sizes in the  $L_{\infty}$  metric (i.e., for the Center Persistence Diagram problem). This is the best possible in polynomial time for the Center Persistence Diagram under the bottleneck distance unless P = NP. All these results hold for both of the discrete versions as well as the continuous version; in fact, the NP-hardness and approximation results also hold under the Wasserstein distance for the continuous version.

2012 ACM Subject Classification Theory of computation - Design and analysis of algorithms

**Keywords and phrases** Persistence diagrams, Bottleneck distance, Center persistence diagram, NP-hardness, Approximation algorithms

Acknowledgements This work was done when GL, EM and BZ visited University of Hyogo.

#### 1 Introduction

Computational topology has found a lot of applications in recent years [7]. Among them, persistence diagrams, each being a set of (topological feature) points above and inclusive of the line Y = X in the X-Y plane, have also found various applications, for instance in GIS [1], in neural science [12], in wireless networks [20], and in prostate cancer research [19]. (Such a topological feature point (b, d)in a persistence diagram, which we will simply call a point henceforth, indicates a topological feature which appears at time b and disappears at time d. Hence  $b \leq d$ . In the next section, we will present some technical details.) A consequence is that practitioners gradually have a database of persistence diagrams when processing the input data over certain period of time. It is not uncommon these days that such a database has tens of thousands of persistence diagrams, each with up to several thousands of points. How to process and search these diagrams becomes a new challenge for algorithm designers, especially because the bottleneck distance is typically used to measure the similarity between two persistence diagrams.

In [10] the following problem was studied: given a set of persistence diagrams  $\mathcal{P}_1, ..., \mathcal{P}_m$ , each with size at most n, how to preprocess them so that each has a key  $k_i$  for i = 1, ..., m and for a query persistence diagram Q with key k, an approximate nearest persistence diagram  $\mathcal{P}_i$  can be returned by searching the key k in the data structure for  $k_i$ 's. A hierarchical data structure was built and the keys are basically constructed using snap roundings on a grid with different resolutions. There is a trade-off between the space complexity (i.e., number of keys stored) and the query time. For instance, if one wants an efficient (polylogarithmic) query time, then he/she has to use an exponential space; and with a linear or polynomial space, then he/she needs to spend an exponential query time. Different from traditional problems of searching similar point sets [15], one of the main technical difficulties is to handle points near the line Y = X.

In prostate cancer research, one important part is to determine how the cancer progresses over certain period of time. In [19], a method is to use a persistence diagram for each of the histopathology images (taken over certain period of time). Naturally, for the data collected over some time interval, one could consider packing a corresponding set of persistence diagrams with a center, which could be considered as a *median* persistence diagram summarizing these persistence diagrams. A sequence of such centers over a longer time period would give a rough estimate on how the prostate cancer progresses. This motivates our research. On the other hand, while the traditional center concept (and the corresponding algorithms) has been used for planar point sets (under the Euclidean distance) [23] and on binary strings (under the Hamming distance) [21]; recently we have also seen its applications in more complex objects, like polygonal chains (under the discrete Frechet distance) [17, 2]. In this sense, this paper is also along this line.

Formally, in this paper we consider a way to pack persistence diagrams. Namely, given a set of persistence diagrams  $\mathcal{P}_1, ..., \mathcal{P}_m$ , how to compute a *center* persistence diagram? Here the distance measure used is the traditional bottleneck distance and Wasserstein distance (where we first focus on the former). We first describe the case when all  $\mathcal{P}_i$ 's have the same size n, and later we show how to withdraw this constraint (by slightly increasing the running time of the algorithms). It turns out that the *continuous* case, i.e., when the points in the center can be arbitrary, is very similar to the geometric 3-dimensional assignment problem: Given three points sets  $P_i$  of the same size n and each colored with *color-i* for i = 1..3, divide points in  $P_i$ 's into *n* 3-clusters (or triangles) such that points in each cluster or triangle have different colors, and some geometric quantity (like the maximum area or perimeter of these triangles) is minimized [25, 13]. For our application, we need to investigate discrete versions where points in the center persistence diagram must come from the input diagrams (might be from more than one diagrams). We show that the problem is NP-hard even when m = 3diagrams are given. On the other hand, we show that the problem is polynomially solvable when

m = 2 and the problem admits a 2-approximation for  $m \ge 3$ . At the end, we briefly discuss how to adapt the results to Wasserstein distance for the continuous case. The following table summarizes the main results in this paper.

**Table 1** Results for the Center Persistence Diagram problems under the bottleneck  $(d_B)$  and Wasserstein  $(W_p)$  distances when  $m \ge 3$  diagrams are given.

	Hardness	Inapproximability bound	Approximation factor
$d_B$ , with no replacement	NP-complete	$2-\varepsilon$	2
$d_B$ , with replacement	NP-complete	$2-\varepsilon$	2
$d_B$ , continuous	NP-hard	$2-\varepsilon$	2
$W_p$ , with no replacement	?	?	2
$W_p$ , with replacement	?	?	2
$W_p$ , continuous	NP-hard	?	2

This paper is organized as follows. In Section 2, we give some necessary definitions and we also show, as a warm-up, that the case when m = 2 is polynomially solvable. In Section 3, we prove that the Center Persistence Diagram problem under the bottleneck distance is NP-hard when m = 3 via a non-trivial reduction from the Planar three-dimensional Matching (Planar 3DM) problem. In Section 4, we present the factor-2 approximation algorithm for the problem (when  $m \ge 3$ ). In Section 5, we briefly discuss how to modify the proofs to the continuous Center Persistence Diagram under the Wasserstein distance. In Section 6, we conclude the paper with some open questions.

# 2 Preliminaries

We assume that the readers are familiar with standard terms in algorithms, like approximation algorithms [4], and NP-completeness [11].

## 2.1 Persistence Diagram

Homology is a machinery from algebraic topology which gives the ability to count the number of holes in a k-dimensional simplicial complex. For instance, let X be a simplicial complex, and let the corresponding k-dimensional homology be  $H_k(X)$ , then the dimension of  $H_0(X)$  is the number of path connected components of X and  $H_1(X)$  consists of loops in X, each is a 'hole' in X. It is clear that these numbers are invariant under rigid motions (and almost invariant under small numerical perturbations) on the original data, which is important in many applications. For further details on classical homology theory, the readers are referred to [14], and to [7, 24] for additional information on computational homology. It is well known that the k-dimensional homology of X can be computed in polynomial time [7, 8, 24].

Ignoring the details for topology, the central part of persistent homology is to track the birth and death of the topological features when computing  $H_k(X)$ . These features give a *persistence* diagram (containing the birth and death times of features as pairs (b, d) in the extended plane). See Figure 1 for an example. Note that as a convention, the line Y = X is included in each persistence diagram, where points on the line Y = X provide infinite multiplicity, i.e., a point (t, t) on it could be considered as a dummy feature which is born at time t then immediately dies. Formally, a persistence diagram  $\mathcal{P}$  is composed of a set P of planar points (each corresponding to a topological feature) above the line Y = X, as well as the line Y = X itself, i.e.,  $\mathcal{P} = P \cup \{(x, y) | y = x\}$ . Due to the infinite multiplicity on Y = X, there is always a bijection between  $\mathcal{P}_i$  and  $\mathcal{P}_j$ , even if  $P_i$  and  $P_j$  have different sizes.



**Figure 1** Two persistence diagrams  $\mathcal{P}$  and  $\mathcal{Q}$ , with feature point sets  $P = \{p_1, p_2, p_3\}$  and  $Q = \{q_1, q_2\}$  respectively. A point  $p_1 = (b, d)$  means that it is born at time b and it dies at time d. The projection of  $p_1$  on Y = X gives p'.

Given two persistent diagrams  $\mathcal{P}_i$  and  $\mathcal{P}_j$ , each with O(n) points, the *bottleneck distance* between them is defined as follows:

$$d_B(\mathcal{P}_i, \mathcal{P}_j) = \inf_{\phi} \{ \sup_{x \in \mathcal{P}_i} \|x - \phi(x)\|_{\infty}, \phi : \mathcal{P}_i \to \mathcal{P}_j \text{ is a bijection} \}.$$

Similarly, the *p*-Wasserstein distance is defined as

$$W_p(\mathcal{P}_i, \mathcal{P}_j) = \left(\inf_{\phi} \sum_{x \in \mathcal{P}_i} \|x - \phi(x)\|_{\infty}^p\right)^{1/p}, \phi : \mathcal{P}_i \to \mathcal{P}_j \text{ is a bijection}$$

We refer the readers to [7] for further information regarding persistence diagrams. Our extended results regarding Wasserstein distance will be discussed solely in Section 5, and until then we focus only on the bottleneck distance.

For point sets  $P_1$ ,  $P_2$  of the same size, we will also use  $d_B^p(P_1, P_2)$  to represent their bottleneck matching distance, i.e., let  $\beta$  be a bijection between  $P_1$  and  $P_2$ ,

$$d_B^p(P_1, P_2) = \min_{\beta} \max_{a \in P_1} d_p(a, \beta(a)).$$

Here,  $d_p(-)$  is the distance under the  $L_p$  metric. As we mainly cover the case p = 2, we will use  $d_B(P_i, P_j)$  instead of  $d_B^2(P_i, P_j)$  henceforth. Note that in comparing persistence diagrams, the  $L_{\infty}$  metric is always used. For our hardness constructions, all the valid clusters form either horizontal or vertical segments, hence the distances under  $L_2$  and  $L_{\infty}$  metrics are all equal in our constructions.

While the bottleneck distance between two persistence diagrams is continuous in its original form, it was shown that it can be computed using a discrete method [7], i.e., the traditional geometric bottleneck matching [9], in  $O(n^{1.5} \log n)$  time. In fact, it was shown that the multiplicity property of the line Y = X can be used to compute the bottleneck matching between two diagrams  $\mathcal{P}_1$  and  $\mathcal{P}_2$  more conveniently — regardless of their sizes [7]. This can be done as follows. Let  $P_i$  be the set of feature points in  $\mathcal{P}_i$ . Then project points in  $P_i$  perpendicularly on Y = X to have  $P'_i$  respectively, for i = 1, 2. (See also Figure 1.) It was shown that the bottleneck distance between two diagrams  $\mathcal{P}_1$  and  $\mathcal{P}_2$  is exactly equal to the bottleneck (bipartite) matching distance, in the  $L_{\infty}$  metric, between  $P_1 \cup P'_2$  and  $P_2 \cup P'_1$ . Here the weight or cost of an edge c(u, v), with  $u \in P'_2$  and  $v \in P'_1$ , is set to zero; while  $c(u, v) = ||u - v||_{\infty}$ , if  $u \in P_1$  or  $v \in P_2$ . The *p*-Wasserstein distance can be computed similarly, using a min-sum bipartite matching between  $P_1 \cup P'_2$  and  $P_2 \cup P'_1$ , with all edge costs

raised to  $c^p$ . (Kerber, et al. showed that several steps of the bottleneck matching algorithm can be further simplified [18].) Later, we will extend this construction for more than two diagrams.

## 2.2 **Problem Definition**

Throughout this paper, for two points  $p_1 = (x_1, y_1)$  and  $p_2 = (x_2, y_2)$ , we use  $d_p(p_1, p_2)$  to represent the  $L_p$  distance between  $p_1$  and  $p_2$ , which is  $d_p(p_1, p_2) = (|x_1 - x_2|^p + |y_1 - y_2|^p)^{1/p}$ , for  $p < \infty$ . When  $p = \infty$ ,  $d_{\infty}(p_1, p_2) = ||p_1 - p_2||_{\infty} = \max\{|x_1 - x_2|, |y_1 - y_2|\}$ . We will mainly focus on  $L_2$  and  $L_{\infty}$  metrics, for the former, we simplify it as  $d(p_1, p_2)$ .

# ► Definition 1. The Center Persistence Diagram Problem under the Bottleneck Distance (CPD-B)

**Instance**: A set of m persistence diagrams  $\mathcal{P}_1, ..., \mathcal{P}_m$  with the corresponding feature point sets  $P_1, ..., P_m$  respectively, and a real value r.

**Question**: Is there a persistence diagram Q such that  $\max_i d_B(Q, \mathcal{P}_i) \leq r$ ?

Note that we could have three versions, depending on Q. We mainly focus on the discrete version when the points in Q are selected with no replacement from the multiset  $\bigcup_{i=1..m} P_i$ . It turns out that the other discrete version, i.e., the points in Q are selected with replacement from the set  $\bigcup_{i=1..m} P_i$ , is different from the first version but all the results can be carried over with some simple twist. We will briefly cover the *continuous* case, i.e., when points Q are arbitrary; as we covered earlier in the introduction, when m = 3, this version is very similar to the geometric three-dimensional assignment problem [25, 13].

We will firstly consider two simplified versions of the corresponding problem.

### ▶ Definition 2. The *m*-Bottleneck Matching Without Replacement Problem

**Instance**: A set of m planar point sets  $P_1, ..., P_m$  such that  $|P_1| = \cdots = |P_m| = n$ , and a real value r.

**Question**: Is there a point set Q, with |Q| = n, such that any  $q \in Q$  is selected from the multiset  $\cup_i P_i$  with no replacement and  $\max_i d_B(Q, P_i) \leq r$ ?

### ▶ Definition 3. The *m*-Bottleneck Matching With Replacement Problem

**Instance**: A set of m planar point sets  $P_1, ..., P_m$  such that  $|P_1| = \cdots = |P_m| = n$ , and a real value r.

**Question**: Is there a point set Q, with |Q| = n, such that any  $q \in Q$  is selected from the set  $\cup_i P_i$  with replacement and  $\max_i d_B(Q, P_i) \leq r$ ?

It turns out that these two problems are really to find center points in Q to cover *m*-clusters with an optimal covering radius r, with each cluster being composed of m points, one each from  $P_i$ . For m = 3, this is similar to the geometric three-dimensional assignment problem which aims at finding *m*-clusters with certain criteria [25, 13]. However, the two versions of the problem are slightly different from the geometric three-dimensional assignment problem. The main difference is that in these discrete versions a cluster could be covered by a center point which does not belong to the cluster. See Figure 2 for an example. Also, note that the two versions themselves are slightly different; in fact, their solution values could differ by a factor of 2 (see Figure 3).

Note that we could define a continuous version in which the condition on q is withdrawn and this will be briefly covered at the end of each section. In fact, we focus more on the optimization versions of these problems. We will show that 3-Bottleneck Matching, for both the discrete versions, is NP-hard, immediately implying CPD-B is NP-hard for  $m \ge 3$ . We then present a 2-approximation for the *m*-Bottleneck Matching Problem and later we will show how to make some simple generalization so the 'equal size' condition can be withdrawn for persistence diagrams — this implies that CPD-B



**Figure 2** An example with  $P_1 = \{a, b\}, P_2 = \{c, d\}$  and  $P_3 = \{e, f\}$ , with all the points (except b) on a unit circle and b being the center of the circle. For the 'without replacement' version, the optimal solution is  $Q_1 = \{b, c\}$ , where b covers the 3-cluster  $\{a, d, e\}$ , c covers the 3-cluster  $\{b, c, f\}$  and the optimal covering radius is 1. For the 'with replacement' version, the optimal solution could be the same, but could also be  $\{b, b\}$ .

also admits a 2-approximation for  $m \ge 3$ . We will focus on the 'without replacement' version in our writing, and later we will show how to generalize it to the 'with replacement' version at the end of each section. Henceforth, we will refer to the 'without replacement' version simply as m-Bottleneck Matching unless otherwise specified.

At first, we briefly go over a polynomial time solution for the case when m = 2.



**Figure 3** An example with  $P_1 = \{a, b\}, P_2 = \{c, d\}$  and  $P_3 = \{e, f\}$ , with all the points on a unit line segment and a being the midpoint of the segment. For the 'without replacement' version, the optimal solution is  $Q_1 = \{a, b\}$ , where a covers the 3-cluster  $\{a, c, f\}$ , b covers the 3-cluster  $\{b, d, e\}$  and the optimal covering radius is 1. For the 'with replacement' version, the optimal solution is  $Q_2 = \{a, a\}$ , with the same clusters  $\{a, c, f\}$  and  $\{b, d, e\}$ , and the optimal covering radius being 1/2.

#### 2.3 A Warm-up for m = 2

First, recall that  $|P_1| = |P_2| = n$ . Note that the optimal solution must be the distance between  $p \in P_1$ and  $q \in P_2$ . We first consider the decision version of the problem; namely, given a radius r, find a clustering of 2-points  $\{p_{1i}, p_{2j}\}$ , with  $p_{xy} \in P_x(x = 1, 2)$ , such that the distance between each of them and  $\hat{q}$ , where  $\hat{q}$  is selected with no replacement from the multiset  $P_1 \cup P_2$ , is at most r. (Note that  $\hat{q}$  is not necessarily equal to  $p_{1i}$  or  $p_{2j}$ .) Once having this decision procedure, we could use a binary search to compute the smallest radius such a clustering exists.

Given the radius r, we construct a flow network G = (V, E) as follows: besides the source s and the sink t, there are four layers of nodes. The first layer contains n nodes corresponding to the npoints of  $P_1$ , the second and the third layers both contain 2n nodes corresponding to the 2n points of  $P_1 \cup P_2$ , and the fourth layer contains n nodes corresponding to the n points of  $P_2$ . Each node  $p_{1i}$ in the first layer has a link going out to a node  $p_{\ell i}$  in the second layer if their distance is at most r; for example,  $p_{1i}$  in the first layer surely has a link going out to  $p_{1i}$  in the second layer, due to their distance being 0. Each node  $p_{\ell i}$  in the second layer has only one out-going link to the node  $p_{\ell i}$  in the

third layer. Each node  $p_{\ell i}$  in the third layer has a link going out to a node  $p_{2j}$  in the fourth layer if their distance is at most r. Lastly, the source s has a link going out to every node in the first layer, and every node in the fourth layer has a link going out to the sink t. All the links in the constructed network has unit capacity.

One sees that the path  $s p_{1h} p_{\ell i} p_{\ell i} p_{2j}$  is used to send a unit of flow if and only if 1)  $p_{\ell i}$  is selected into the set Q, and 2)  $p_{\ell i}$  is matched with  $p_{1h}$  ( $p_{2j}$ , respectively) in the bottleneck matching between Q and  $P_1$  ( $P_2$ , respectively). Therefore, the maximum flow has a value n if and only if Q is determined such that the maximum bottleneck matching distance is at most r. Since the constructed network is acyclic, its maximum flow can be computed in  $O(n^3)$  time [22]. A binary search to find the optimal radius leads to a solution running in  $O(n^3 \log n)$  time.

For the 'With Replacement' version, given a radius r, an unweighted bipartite graph between  $P_1$ and  $P_2$  can be first constructed. In the graph there is an edge between  $u \in P_1$  and  $v \in P_2$  if there is a point  $w \in P_1 \cup P_2$  such that a circle with radius r centered w can cover both u and v. Then, the decision problem is to decide whether a perfect matching exists, which can be solved in  $O(n^{2.5})$  time [16]. A binary search for the optimal radius gives a solution which runs in  $O(n^{2.5} \log n)$  time.

For the continuous version, the problem can be solved by computing the geometric bottleneck matching between  $P_1$  and  $P_2$  in  $O(n^{1.5} \log n)$  time [9]. After the matched edges between  $P_1$  and  $P_2$  are identified, the set of midpoints of ll the edges in the matching gives us the solution.

The above algorithm can be generalized for two persistence diagrams, with the distance between two points in the  $L_{\infty}$  metric. In general, the sizes of two persistence diagrams might not be the same, i.e.,  $|P_1|$  might not be the same as  $|P_2|$ . This can be handled easily using the projection method in [7], which has been described in subsection 2.1 and will be generalized for  $m \ge 3$  in Section 4. Hence, we have the following theorem.

**Theorem 4.** The Center Persistence Diagram Problem can be solved in polynomial time, for m = 2 and for all the three versions ('Without Replacement', 'With Replacement' and continuous versions).

In the next section, we will consider the case for m = 3.

# 3 3-Bottleneck Matching is NP-complete

We will first focus on the  $L_2$  metric in this section and at the end of the proof it should be seen that the proof also works for the  $L_{\infty}$  metric. For m = 3, we can color points in  $P_1$ ,  $P_2$  and  $P_3$  in color-1, color-2 and color-3. Then, in this case, the problem is really to find n disks centered at n points from  $P_1 \cup P_2 \cup P_3$ , with smallest radii  $r_i^*$  (i = 1..n) respectively, such that each disk contains exact 3 points of different colors (possibly including the center of the disk); moreover,  $\max_{i=1..n} r_i^*$  is bounded from above by a given value r. We also say that these 3 points form a *cluster*.

It is easily seen that (the decision version of) 3-Bottleneck Matching is in NP. Once the n guessed disks are given, the problem is then a max-flow problem, which can be verified in polynomial time.

We next show that Planar 3-D Matching (Planar 3DM) can be reduced to 3-Bottleneck Matching in polynomial time. The former is a known NP-complete problem [6]. In 3DM, we are given three sets of elements  $E_1, E_2, E_3$  (with  $|E_1| = |E_2| = |E_3| = \gamma$ ) and a set  $\mathcal{T}$  of n triples, where  $T \in \mathcal{T}$ implies that  $T = (a_1, a_2, a_3)$  with  $a_i \in E_i$ . The problem is to decide whether there is a set S of  $\gamma$ triples such that each element in  $E_i$  appears exactly once in (the triples of) S. The Planar 3DM incurs an additional constraint: if we embed elements and triples as points on the plane such that there is an edge between an element a and a triple T iff a appears in T, then the resulting graph is planar.

An example for Planar 3DM is as follows:  $E_1 = \{1, 2\}, E_2 = \{a, b\}, E_3 = \{x, y\}$ , and  $\mathcal{T} = \{(1, a, x), (2, b, x), (2, b, y), (1, b, y)\}$ . The solution is  $S = \{(1, a, x), (2, b, y)\}$ .

Given an instance for Planar 3DM and a corresponding planar graph G with O(n) vertices, we first convert it to a planar graph with degree at most 3. This can be done by replacing a degree-delement node x in G with a path of d nodes  $x_1, ..., x_d$ , each with degree at most 3 and the connection between x and a triple node T is replaced by a connection from  $x_i$  to T for some i (see also Figure 4). We have a resulting planar graph G' = (V(G'), E(G')) with degree at most 3 and with O(n)vertices. Then we construct a rectilinear embedding of G' on a regular rectilinear grid with a unit grid length, where each vertex  $u \in V(G')$  is embedded at a grid point and an edge  $(u, v) \in E(G')$  is embedded as an intersection-free path between u and v on the grid. It is well-known that such an embedding can be computed in  $O(n^2)$  time [26].

Let x be a black node (• in Figure 4) with degree d in G. In the rectilinear embedding of G', the paths from  $x_i$  to  $x_{i+1}$  (i = 1, ..., d - 1) will be the basis of the element gadget for x. (Henceforth, unless otherwise specified, everything we talk about in this reduction is referred to the rectilinear embedding of G'.) We put a copy of  $\bullet$  at each (grid point)  $x_i$  as in Figure 4. (If the path from  $x_i$  to  $x_{i+1}$  is of length greater than one, then we put • at each grid point on the path from  $x_i$  to  $x_{i+1}$ .)

We now put color-2 and color-3 points ( $\Box$  and  $\blacksquare$ ) at 1/3 and 2/3 positions at each grid edge which is contained in some path in an element gadget (in the embedding of G'). These points are put in a way such that it is impossible to use a discrete disk centered at a  $\bullet$  point with radius 1/3 to cover three points with different colors. These patterns are repeated to reach a *triple gadget*, which will be given later. Note that this construction is done similarly for elements y and z, except that the grid points in the element gadgets for y and z are of color-2 ( $\Box$ ) and color-3 ( $\blacksquare$ ) respectively.

**Lemma 5.** In an element gadget for x, exactly one  $x_i$  is covered by a discrete disk of radius 1/3, centered at a (colored) grid point out of the gadget.

**Proof.** Throughout the proof, we refer to Figure 4. Let x be colored by color-1 (e.g.,  $\bullet$ ). In the rectilinear embedding, let the path length between  $x_1$  and  $x_d$  be D. Then, the total number of points on the path from  $x_1$  to  $x_d$ , of colors 1, 2 and 3, is 3D + 1. By the placement of color-2 and color-3 points in the gadget for x, exactly 3D points of them can be covered by D discrete disks of radii 1/3 (centered either at color-2 or color-3 points in the gadget). Therefore, exactly one of  $x_i$  must be covered by a discrete disk centered at a point out of the gadget. 4

When  $x_i$  is covered by a discrete disk of radius 1/3 centered at a point out of the gadget x, we also say that  $x_i$  is *pulled out* of x.



### **Figure 4** The gadget for element *x*.

We now illustrate how to construct a triple gadget T = (x, y, z). It is basically a grid point on which we put three points with different colors. (In Figure 5, we simply use a  $\blacktriangle$  representing such a triple gadget.) The interpretation of T being selected in a solution for Planar 3DM is that the three colored points at  $\blacktriangle$  is covered by a disk of radius zero, centered at one of these three points. When one of these three points at  $\blacktriangle$  is covered by a disk of radius 1/3 centered at some other points (on

the path from one of the elements x, y or z to T), we say that such a point is *pulled out* of the triple gadget T by the corresponding element gadget.

▶ Lemma 6. In a triple gadget for T = (x, y, z), to cover the three points representing T using discrete disks of radii at most 1/3, either all the three points are pulled out of the triple gadget T by the three element gadgets respectively, or none is pulled out. In the latter case, these three points can be covered by a discrete disk of radius zero.

**Proof.** Throughout the proof, we refer to Figure 5. At the triple gadget T, if only one point (say •) is pulled out or two points (say, • and  $\Box$ ) are pulled out, then the remaining points in the triple,  $\Box$  and  $\blacksquare$  or  $\blacksquare$  respectively, could not be properly covered by a discrete disk of radius 1/3 — such a disk would not be able to cover a cluster of exactly three points of distinct colors. Therefore, either all the three points associated with T are pulled out by the three corresponding element gadgets, hence covered by three different discrete disks of radii 1/3; or none of these three points is pulled out. Clearly, in the latter case, these three points associated with T can be covered by a discrete disk of radius zero, as a cluster.

In Figure 5, we show the case when x would not pull any point out of the gadget for T. By Lemma 5, y and z would do the same, leading  $T = \langle x, y, z \rangle$  to be selected in a solution S for Planar 3DM. Similarly, in Figure 6, x would pull a • point out of T. Again, by Lemma 5, y and z would pull  $\Box$  and  $\blacksquare$  points (one each) out of T, which implies that T would not be selected in a solution S for Planar 3DM.



**Figure 5** The triple gadget for  $T = \langle x, y, x \rangle$  (represented as  $\blacktriangle$ , which is really putting three element points on a grid point). In this case the triple  $\langle x, y, z \rangle$  is selected in the final solution (assuming operations are similarly performed on y, z). Exactly one of  $x_i$  (in this case  $x_2$ ) is pulled out of the gadget for the element x.

We hence have the following theorem.

▶ **Theorem 7.** The decision versions of 3-Bottleneck Matching for both the 'Without Replacement' and 'With Replacement' cases are NP-complete, and the decision version of the continuous 3-Bottleneck Matching is NP-hard.

**Proof.** We discuss the 'Without Replacement' discrete case first, and will drop the keyword 'Without Replacement' until the end of the proof. As explained a bit earlier, (the decision version of) 3-Bottleneck Matching is obviously in NP. Moreover, we show in Lemma 5 and Lemma 6 that, given an instance for Planar 3DM with n triples over  $3\gamma$  base elements we can convert it into an instance I of 3Kn points of three colors (Kn points are of color-1, color-2 and color-3 respectively) in polynomial time, where K is related to this polynomial running time. We just formally argue below



**Figure 6** The triple gadget for  $T = \langle x, y, x \rangle$  (represented as  $\blacktriangle$ , which is really putting three element points on a grid point). In this case the triple  $\langle x, y, z \rangle$  would not be selected in the final solution. Note that the black round point in the triple gadget is pulled out by the element x, and the other two points are pulled out similarly by the element y and z.

that Planar 3DM has a solution of  $\gamma$  triples if and only if the converted instance I of 3Kn points can be partitioned into Kn clusters each covered by a discrete disk of radius 1/3; moreover, there are exactly  $\gamma$  such clusters which are covered by discrete disks with radii zero.

'Only if part:' If the Planar 3DM instance has a solution, we have a set S of  $\gamma$  triples which uniquely cover all the  $3\gamma$  elements. Then, at each of the corresponding  $\gamma$  triple gadgets, we use a discrete disk of radius zero to cover the corresponding three points. By Lemma 5, in each element gadget x exactly one point  $x_i$  could be pulled out of the gadget, connecting to these selected triple gadgets. By Lemma 6, the triple gadgets can be covered exactly in two ways. Hence the triples not corresponding to S will be covered in the other way: for  $T = \langle x, y, z \rangle$  not in S, one point of each color will be pulled out of the triple gadget for T.

'If part:' If the converted instance I of 3Kn points can be partitioned into Kn clusters each covered by a discrete disk of radius 1/3 and there are exactly  $\gamma$  clusters whose covering discrete disks have radii zero, then the triples corresponding these clusters of point form a solution to the original Planar 3DM instance. The reason is that, by Lemma 6, the remaining points will be covered by a discrete disk of radius 1/3 (and cannot be further shrunk). Moreover, by Lemma 5, at each element gadget, exactly one point will be fulled out, leading to the corresponding triple gadget being covered by a discrete disk of radius zero — which implies that exactly one element is covered by a selected triple.

It can be seen by now the proof works for the 'With Replacement' discrete version without any change in the proof. For the continuous version, the NP-hardness holds with the same reduction. The reason for the NP-hardness to hold is that the optimal grouping of three points (0,0), (0,1/3), (0,2/3) is to use (0,1/3) as the continuous center, even though itself is still discrete. However, the NP membership does not hold anymore for the continuous case (since a guessed solution involve real numbers). This closes our proof.

Note that in the above proof, if Planar 3DM does not have a solution, then we need to use discrete disks of radii at least 2/3 to have a valid solution for 3-Bottleneck Matching. This implies that finding a factor- $(2 - \varepsilon)$  approximation for (the optimization version of) 3-Bottleneck Matching remains NP-hard.

► **Corollary 8.** It is NP-hard to approximate (the optimization version of) 3-Bottleneck Matching within a factor  $2 - \varepsilon$ , for some  $\varepsilon > 0$  and for all the three versions.

We comment that the NP-hardness proofs in [13, 25] also use a reduction from Planar 3DM; however, those proofs are only for the  $L_2$  metric. Here, it is clear that our reduction also works for the  $L_{\infty}$  metric without any modification — this is due to that all clusters in our construction are either horizontal or vertical, therefore the distances within a cluster would be the same under  $L_2$  and  $L_{\infty}$ . With respect to the CPD-B problem, points in color-*i*, *i* = 1, 2, 3, are the basis for us to construct a persistence diagram. To handle the line Y = X in a persistence diagram, let the diameter of the (union of the) three constructed point sets of different colors be  $\hat{D}$ , we then translate these points as a whole set rigidly such that all the points are at least  $2\hat{D}$  distance away from Y = X. We then have three persistence diagrams. (The translation is to neutralize the infinite multiplicity of Y = X, i.e., to enforce that all points on Y = X can be ignored when computing the bottleneck distance between the corresponding persistence diagrams.) Hence, we have the following corollary.

▶ **Corollary 9.** It is NP-hard to approximate (the optimization version of) Center Persistence Diagram problem under the bottleneck distance for  $m \ge 3$  within a factor  $2 - \varepsilon$ , for some  $\varepsilon > 0$  and for all the three versions.

In the next section, we present tight approximation algorithms for the above problems.

# 4 A Tight Approximation

## 4.1 Approximation for *m*-Bottleneck Matching

We first present a simple Algorithm 1 for *m*-Bottleneck Matching as follows. Recall that in the *m*-Bottleneck Matching problem we are given *m* sets of planar points  $P_1, ..., P_m$ , all with the same size *n*. Without of generality, let the points in  $P_i$  be colored with color-*i*.

- **1.** Pick any color, say, color-1.
- **2.** Compute the bottleneck matching  $M_{1,i}$  between  $P_1$  and  $P_i$  for i = 2, ..., m.
- **3.** For the m 1 edges  $(p_{j_1}^1, p_{j_i}^i) \in M_{1,i}$  for i = 2, ...m, where  $p_y^x \in P_x$  for x = 1, ..., m, form a cluster  $\{p_{j_1}^1, p_{j_2}^2, ..., p_{j_m}^m\}$  with  $p_{j_1}^1$  as its center.

We comment that the algorithm itself is similar to the one given for m = 3 in [13], which has a different objective function (i.e., minimizing the maximum perimeter of clusters). We show next that Algorithm 1 is a factor-2 approximation for *m*-Bottleneck Matching. Surprisingly, the main tool here is the triangle inequality of a distance measure. Note that we can not only handle for any given  $m \ge 3$ , we also need some twist in the proof a bit later for the three versions of the Center Persistence Diagram problem, where the diagrams could have different sizes.

▶ **Theorem 10.** Algorithm 1 is a polynomial time factor-2 approximation for m-Bottleneck Matching for all the three versions (i.e., 'Without Replacement', 'With Replacement' and continuous versions).

**Proof.** One clearly sees that the running time of Algorithm 1 is  $O(mn^{1.5} \log n)$ .

(1) We discuss the 'Without Replacement' first. In an optimal solution for *m*-Bottleneck Matching with its radius OPT, let  $\{q_1, q_2, ..., q_m\}$  denote a cluster with its discrete center  $\hat{q}$ , where  $q_i$  is in color-*i*, for i = 1, ..., m.

From OPT  $\geq \max\{d(\hat{q}, q_1), d(\hat{q}, q_2), ..., d(\hat{q}, q_m)\}$  and the triangle inequality, we have  $d(q_1, q_j) \leq d(q_1, \hat{q}) + d(q_j, \hat{q}) \leq 2 \cdot \text{OPT}$  for j = 2, ..., m. This implies a matching between  $P_1$  and  $P_i$  with radius at most  $2 \cdot \text{OPT}$ , for i = 2, ..., m.

Let APP denote the maximum radius between the m-1 bottleneck matchings computed in Algorithm 1; then the radius of the produced solution is APP, and we have APP  $\leq d(q_1, q_j) \leq 2 \cdot \text{OPT}$ 

for j = 2, ..., m. That is, Algorithm 1 is a polynomial time factor-2 approximation for *m*-Bottleneck Matching (for the 'Without Replacement' version).

(2) We next discuss the 'With Replacement' case. Note that we never need to change the algorithm. In this case, first note that the points in the center C are selected from  $P_i$ 's with replacement. Then, given an optimal solution for this case we notice that some points in  $P_i$ 's can be selected multiple times (i.e., more than once) in C. Let q be such a point in C. If q covers a cluster including itself, then we leave that cluster alone; otherwise, pick any cluster covered by q and also leave q and that cluster alone. Then, anytime when q covers  $\{p_{j_1}^1, p_{j_2}^2, ..., p_{j_m}^m\}$  once more with  $q \notin \{p_{j_1}^1, p_{j_2}^2, ..., p_{j_m}^m\}$ , we switch the center for this cluster to  $p_{j_1}^1$  (i.e., the point with color-1). Clearly, we have

$$d(p_{j_1}^1, p_{j_i}^i) \le d(p_{j_1}^1, q) + d(q, p_{j_i}^i) \le 2 \cdot \text{OPT},$$

for i = 2, ..., m. Then, combined with the other ('Without Replacement') case covered in part (1), we can conclude that Algorithm 1 provides a 2-approximation for the 'With Replacement' case as well when  $m \ge 3$ . In fact, the example in Figure 3 shows a simple matching lower bound of factor 2.

(3) The proof for the continuous case would be almost identical as in part (1); in fact, we just need to define  $\hat{q}$  as an arbitrary point covering the given cluster  $\{q_1, q_2, ..., q_m\}$ . And the remaining arguments would be the same.

# 4.2 Generalization to the Center Persistence Diagram Problem under the Bottleneck Distance

First of all, note that the above approximation algorithm works for *m*-Bottleneck Matching when the metric is  $L_{\infty}$ . Hence, obviously it works for the case when the input is a set of *m* persistence diagrams (all having the same size), whose (feature) points are all far away from Y = X, and the distance measure is the bottleneck distance. (Recall that, when computing the bottleneck distance between two persistence diagrams using a projection method, we always use the  $L_{\infty}$  metric to measure the distance between two points.)

We next show how to generalize the factor-2 approximation algorithm for *m*-Bottleneck Matching to the Center Persistence Diagram problem, first for m = 3. Note that we are given *m* persistence diagrams  $\mathcal{P}_1, \mathcal{P}_2, ...,$  and  $\mathcal{P}_m$ , with the corresponding non-diagonal point sets being  $P_1, P_2, ...,$  and  $P_m$  respectively. Here the sizes of  $P_i$ 's could be different and we assume that the points in  $P_i$  are of color-*i* for i = 1, ..., m.

Given a point  $p \in P_i$ , let  $\tau(p)$  be the (perpendicular) projection of p on the line Y = X. Consequently, let  $\tau(P_i)$  be the projected points of  $P_i$  on Y = X, i.e.,

$$\tau(P_i) = \{\tau(p) | p \in P_i\}.$$

When m = 2, i.e., when we are only given  $\mathcal{P}_1$  and  $\mathcal{P}_2$ , not necessarily of the same size, it was shown by Edelsbrunner and Harer that  $d_B(\mathcal{P}_1, \mathcal{P}_2) = d_B^{\infty}(P_1 \cup \tau(P_2), P_2 \cup \tau(P_1))$  [7]. (Note that  $|P_1 \cup \tau(P_2)| = |P_2 \cup \tau(P_1)|$ .) We next generalize this result. For  $i \in M = \{1, 2, ..., m\}$ , let  $M(-i) = \{1, 2, ..., i - 1, i + 1, ..., m\}$ . We have the following lemma.

▶ Lemma 11. Let  $\{i, j\} \subseteq M = \{1, 2, ..., m\}$ . Let  $\tau_i(P_k)$  be the projected points of  $P_k$  on Y = X such that these projected points all have color-*i*, with  $k \in M, i \neq k$ . Then,

$$d_B(\mathcal{P}_i, \mathcal{P}_j) = d_B^{\infty}(P_i \bigcup_{k \in M(-i)} \tau_i(P_k), P_j \bigcup_{k \in M(-j)} \tau_j(P_k)).$$

**Proof.** First, notice that, in terms of sizes, we have

$$|P_i \bigcup_{k \in M(-i)} \tau_i(P_k)| = |P_j \bigcup_{k \in M(-j)} \tau_j(P_k)| = \sum_{l=1}^m |P_l|.$$

Following [7], we have

$$d_B(\mathcal{P}_i, \mathcal{P}_j) = d_B^{\infty}(P_i \cup \tau_i(P_j), P_j \cup \tau_j(P_i)).$$

Note that

$$(P_i \bigcup_{k \in M(-i)} \tau_i(P_k)) - (P_i \cup \tau_i(P_j)) = (\bigcup_{k \in M(-i)} \tau_i(P_k)) - \tau_i(P_j)$$

and

$$(P_j \bigcup_{k \in M(-j)} \tau_j(P_k)) - (P_j \cup \tau_j(P_i)) = (\bigcup_{k \in M(-j)} \tau_j(P_k)) - \tau_j(P_i)$$

are really two sets of identical points with color-*i* and color-*j* on Y = X respectively. By the definition of infinite multiplicity property of a persistence diagram, adding these (identical) points on Y = X would not change the bottleneck matching distance between point sets  $P_i \cup \tau_i(P_j)$  and  $P_j \cup \tau_j(P_i)$ . Consequently,

$$d_B^{\infty}(P_i \cup \tau_i(P_j), P_j \cup \tau_j(P_i)) = d_B^{\infty}(P_i \bigcup_{k \in M(-i)} \tau_i(P_k), P_j \bigcup_{k \in M(-j)} \tau_j(P_k)).$$

Therefore, we have

$$d_B(\mathcal{P}_i, \mathcal{P}_j) = d_B^{\infty}(P_i \bigcup_{k \in M(-i)} \tau_i(P_k), P_j \bigcup_{k \in M(-j)} \tau_j(P_k)).$$

The implication of the above lemma is that the approximation algorithm in the previous subsection can be used to compute the approximate center of m persistence diagrams. The algorithm can be generalized by simply projecting each point of color-i, say  $p \in P_i$ , on Y = X to have m-1 projection points with every color k, where  $k \in M(-i)$ . Then we have m augmented sets  $P''_i$ , i = 1, ..., m, of distinct colors, but with the same size  $\sum_{l=1..m} |P_l|$ . Finally, we simply run Algorithm 1 over  $\{P''_1, P''_2, ..., P''_m\}$ , with the distance in the  $L_{\infty}$  metric, to have a factor-2 approximation. We leave out the details for the analysis as at this point all we need is the triangle inequality of the  $L_{\infty}$  metric.

**Theorem 12.** There is a polynomial time factor-2 approximation for the Center Persistence Diagram problem under the bottleneck distance with m input diagrams for all the three versions (i.e., 'Without Replacement', 'With Replacement' and continuous versions).

**Proof.** The analysis of the approximation factor is identical with Theorem 10. However, when *m* is part of the input, each of the augmented point set  $P''_i$ , i = 1..., m, has a size  $\sum_{l=1..m} |P_l| = O(mn)$ . Therefore, the running time of the algorithm increases to  $O((mn)^{1.5} \log(mn))$ , which is, nonetheless, still polynomial.

It is interesting to raise the question whether these results still hold if the *p*-Wasserstein distance is used, which we depict in the next section. As we will see there, different from under the bottleneck distance, a lot of questions still remain open.

### 5 Center Persistence Diagram under the Wasserstein Distance

# ► Definition 13. The Center Persistence Diagram Problem under the *p*-Wasserstein Distance (CPD-W)

**Instance**: A set of m persistence diagrams  $\mathcal{P}_1, ..., \mathcal{P}_m$  with the corresponding feature point sets  $P_1, ..., P_m$  respectively, and a real value r.

**Question**: Is there a persistence diagram Q such that  $\max_i W_p(Q, \mathcal{P}_i) \leq r$ ?

Note that, similar to CPD-B, we could have three versions depending on Q: (1) selected with no replacement from the multiset  $\bigcup_{i=1..m} P_i$ , (2) selected with replacement from the set  $\bigcup_{i=1..m} P_i$ , and (3) arbitrarily selected. We call the first two versions *discrete* and the third case *continuous*. Here we will deal with the continuous case as it is still unknown how to deal with the discrete cases yet.

Given two planar point sets P and Q with size n, they naturally form a complete bipartite graph  $\langle P, Q \rangle$ . Let c(P, Q) be the *weight* or *total cost* of the minimum weight matching in the bipartite graph  $\langle P, Q \rangle$ , where the weight or cost of an edge (p, q) is defines as  $c(p, q) = ||p - q||_2$ , for  $p \in P, q \in Q$ .

### ▶ Definition 14. The *m*-BottleneckSum Matching Problem

**Instance**: A set of m planar point sets  $P_1, ..., P_m$  such that  $|P_1| = \cdots = |P_m| = n$ , and a real value r.

**Question**: Is there a point set Q, with |Q| = n, such that any  $q \in Q$  is arbitrarily selected and  $\max_i c(Q, P_i) \leq r$ ?

The 'With Replacement' and 'With No Replacement' discrete cases can be defined similarly as in Section 2. But we only focus on this continuous version here.

### 5.1 3-BottleneckSum Matching is NP-hard

In this subsection, we first prove that 3-BottleneckSum Matching is NP-hard. The crux to modify the proof in Theorem 6 is that the objective function is to minimize the maximum summation of distances, therefore in Figure 6 around the triple gadget T the sum of distances from Q to  $P_i$ 's might be different. In fact, with the three clusters (a, b, w), (c, d, u) and (e, f, v) alone, the sum of distances from the corresponding centers, i.e., b, c and e, to the points in different colors in these clusters would already be all different.





The reduction is still from Planar 3DM. The main change is, at each grid edge with endpoints in color-*i*, we put two points with different colors at the midpoint of that edge. See Figure 7. Then, the optimal (continuous) center for a cluster of three points in different colors would be the midpoint of these three points with different colors, i.e., with minimum radius 1/4. In Figure 7, the centers marked as 'X' are for the clusters (a, b, w), (c, d, u) and (e, f, v) respectively. Note that each of the centers has the same distance to the three points in distinct colors in the corresponding cluster. We thus have the following theorem.

### ▶ Theorem 15. The continuous version of 3-BottleneckSum Matching is NP-hard.

**Proof.** The reduction is similar to that in Theorem 7 and can be done in  $O(n^2)$  time. Let  $P_i$  be the set of points used in color-*i* in the construction, for i = 1..3 and with  $|P_1| = |P_2| = |P_3|$ . Then we could have exactly  $|P_1|$  clusters. We claim that  $P_1, P_2$  and  $P_3$  admit a point set Q incurring a 3-BottleneckSum matching with a cost of  $(|P_1| - \gamma)/4$  if and only if the Planar 3DM instance has a YES solution. We leave out the argument details as they are almost identical to those in Theorem 7.

We show in the next subsection how Theorem 16 can be extended to the continuous Center Persistence Diagram problem under the *p*-Wasserstein distance.

# 5.2 Continuous Center Persistence Diagram under Wasserstein Distance is NP-hard

▶ Corollary 16. The (continuous) Center Persistence Diagram problem under p-Wasserstein distance with  $m \ge 3$  input persistence diagrams is NP-hard.

**Proof.** Our reduction is exactly the same as in Theorem 16. We then move the constructed points at least  $2\hat{D}$  distance away from Y = X as in Corollary 9, where  $\hat{D}$  is the diameter of all the constructed points on the rectilinear grid.

Let  $P_i$  be the set of points used in color-*i* in the construction, for i = 1..3 and with  $|P_1| = |P_2| = |P_3|$ . As all our  $|P_1|$  clusters form either a horizontal or vertical interval with length 1/2, the  $L_{\infty}$  distance would be the same as under  $L_2$ , i.e., each cluster would contribute a value  $(1/4)^p$  toward computing the *p*-Wasserstein distance between Q and  $\mathcal{P}_i$  — using the midpoint of the interval as the corresponding center. Therefore, we claim that  $P_1, P_2$  and  $P_3$  admit a center persistence diagram Q with a maximum *p*-Wasserstein distance  $\frac{(|P_1|-\gamma)^{1/p}}{4}$  to all  $P_i$ 's if and only if the Planar 3DM instance has a YES solution. We again leave out the arguments. This concludes the proof.

Note that, for  $p < \infty$ , the  $2 - \varepsilon$  inapproximability bound does not hold anymore as in Corollaries 8 and 9. Moreover, the proof of Corollary 16 does not hold for the discrete versions of CPD-W. Further research is needed along this line. On the other hand, we comment that the 2-approximation algorithm still works as the *p*-Wasserstein distance fulfills the triangle inequality.

# 6 Concluding Remarks

In this paper, we study systematically the Center Persistence Diagram problem under both the bottleneck and *p*-Wasserstein distances. Under the bottleneck distance, the results are tight as we have a  $2 - \varepsilon$  inapproximability lower bound and a 2-approximation algorithm (in fact, for all the three versions). Under the *p*-Wasserstein distance, unfortunately, we only have the NP-hardness for the continuous version and a 2-approximation, how to reduce the gap poses an interesting open problem. In fact, a similar question of obtaining some APX-hardness result was posed in [5] already, although the (min-sum) objective function there is slightly different. For the discrete cases under the *p*-Wasserstein distance, it is not even known whether the problems are NP-hard.

### — References

M. Ahmed, B. Fasy and C. Wenk. Local persistent homology based distance between maps. In Proc. 22nd ACM SIGSPATIAL International Conference on Advances in GIS (SIGSPATIAL'14), pages 43-52, 2014.

- 2 K. Buchin, A. Driemel, J. Gudmundsson, M. Horton, I. Kostitsyna, M. Loffler and M. Struijs. Approximating (k,l)-center clustering for curves. In *Proc. 30th ACM-SIAM Symp. on Discrete Algorithms (SODA'19)*, pages 2922-2938, 2019.
- 3 R. Burkard, M. Dell'Amico and S. Martello. Assignment Problems. SIAM, 2009.
- 4 T. Cormen, C. Leiserson, R. Rivest, C. Stein. *Introduction to Algorithms*, second edition, MIT Press, 2001.
- **5** A. Custic, B. Klinz and G. Woeginger. Geometric versions of the three-dimensional assignment problem under general norms. *Discrete Optimization*, 18:38-55, 2015.
- 6 Martin Dyer and Alan Frieze. Planar 3DM is NP-complete. J. Algorithms, 7:174-184, 1986.
- 7 H. Edelsbrunner and J. Harer. Computational Topology: An Introduction. American Mathematical Soc., 2010.
- 8 H. Edelsbrunner, D. Letscher and A. Zomorodian. Topological persistence and simplification. *Disc. and Comp. Geom.*, 28:511-513, 2002.
- **9** A. Efrat, A. Itai and M. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1-28, 2001.
- 10 B. Fasy, X. He, Z. Liu, S. Micka, D. Millman and B. Zhu. Approximate nearest neighbors in the space of persistence diagrams. *CoRR abs/1812.11257*, Dec, 2018.
- M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- 12 C. Giusti, E. Pastalkova, C. Curto and V. Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455-13460, 2015.
- **13** D. Goossens, S. Polyakovskiy, F. Spieksma and G. Woeginger. The approximability of three-dimensional assignment problems with bottleneck objective. *Optim. Lett.*, 4:7-16, 2010.
- 14 A. Hatcher. Algebraic Topology. *Camb. Univ. Press*, 2001.
- **15** P. Heffernan and S. Schirra. Approximate decision algorithms for point set congruence. *Computational Geometry: Theory and Applications*, 4(3):137-156, 1994.
- 16 J. Hopcroft and R. Karp. An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225-231, 1973.
- 17 P. Indyk. Approximate nearest neighbor algorithms for Frechet distance via product metrics. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry (SoCG'02)*, pp. 102-106, 2002.
- 18 M. Kerber, D. Morozov and A. Nigmetov. Geometry helps to compare persistence diagrams. In Proceedings of the 18th Workshop on Algorithm Engineering and Experiments (ALENEX'16), pp. 103-112, SIAM, 2016.
- 19 P. Lawson, J. Schupbach, B. Fasy and J. Sheppard. Persistent homology for the automatic classification of prostate cancer aggressiveness in histopathology images. In *Proc. Medical Imaging: Digital Pathology* 2019, paper 109560G, 2019.
- **20** N-K. Le, P. Martins, L. Decreusefond and A. Vergne. Simplicial homology based energy saving algorithms for wireless networks. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 166-172, 2015.
- 21 M. Li, B. Ma and L. Wang. On the closest string and substring problems. J. ACM, 49(2):157-171, 2002.
- 22 V.M. Malhotra, M.P. Kumar and S.N. Maheshwari. An  $O(|V|^3)$  algorithm for finding maximum flows in networks. *Info. Process. Lett.*, 7(6):277-278, 1978.
- 23 S. Masuyama, T. Ibaraki and T. Hasegawa. The computational complexity of the m-center problems. *Transac. of IEICE.*, E64(2):77-64, 1981.
- 24 K. Mischaikow, T. Kaczynski and M. Mrozek. *Computational Homology*, Applied Mathematical Sciences, 157, Springer, 2004.
- **25** F. Spieksma and G. Woeginger. Geometric three-dimensional assignment problems. *European J. of Operation Research*, 91:611-618, 1996.
- 26 L.G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Computers*, 30(2):135-140, 1981.