Algorithmica (2017) 78:741–770
DOI 10.1007/s00453-016-0173-4

FPTAS for Minimizing the Earth Mover's Distance Under Rigid Transformations and Related Problems

Hu Ding[1] · **Jinhui Xu**[2]

Received: 7 July 2015 / Accepted: 3 June 2016 / Published online: 10 June 2016

Abstract In this paper, we consider the problem (denoted as EMDRT) of minimizing the earth mover's distance between two sets of weighted points A and B in $\mathbb{R}^d$ under rigid transformation. EMDRT is an important problem in both theory and applications and has received considerable attentions in recent years. Previous research on this problem has resulted in only constant factor approximations and it has been an open problem for a long time to achieve PTAS solution. In this paper, we present the first FPTAS algorithm for EMDRT. Our algorithm runs roughly in $O((nm)^{d+2}(\log nm)^{2d})$ time (which is close to a lower bound on any PTAS for this problem), where n and m are the sizes of A and B, respectively. Our result is based on several new techniques, such as the *Sequential Orthogonal Decomposition* and *Optimum Guided Base*, and can be extended to several related problems, such as the problem of earth mover's distance under similarity transformation and the alignment problem, to achieve FPTAS for each of them.

This research was supported in part by NSF under Grants CCF-1422324, IIS- 1422591, IIS-1115220, and CNS-1547167. A preliminary version of this paper has appeared in the 21st European Symposium on Algorithms (ESA 2013).

✉ Hu Ding
huding@msu.edu

Jinhui Xu
jinhui@buffalo.edu

[1] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA

[2] Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, USA

Keywords Earth mover's distance · Approximation Algorithms · Rigid transformation · Geometric matching

1 Introduction

In this paper, we study the problem (denoted as EMDRT) of minimizing the earth mover's distance between two sets of weighted points A and B (with size n and m, respectively) in $\mathbb{R}^d$ under rigid transformation. In EMDRT, each point in A and B is associated with a nonnegative weight, and the objective is to determine the best rigid transformation $\mathcal{T}$ for B so that the earth mover's distance (EMD) between A and $\mathcal{T}(B)$ is minimized, where EMD measures the minimum transportation cost between the two point sets. EMDRT is an important problem in both theory and applications. In theory, it is a natural generalization of the Euclidean bipartite matching problem (i.e., from one-to-one matching to many-to-many matching) and is a powerful model for a number of other matching or partial matching problems. For instance, if all points in A and B have unit weight, EMDRT becomes a one-to-one matching problem. If all points in A have unit weight and all points in B have infinity weight, EMDRT becomes a many-to-one matching problem (*i.e.,* the Hausdorff distance matching problem).

In applications, EMDRT has connections to many EMD (or its variants such as *Proportional Transportation Distance (PTD)*) based problems in computer vision and pattern recognition[12,21,24,28,30], and can be used to solve the challenging alignment problem for rigid objects and compute their similarities. Besides the commonly studied computer vision problems in 2 or 3D space, EMDRT also finds some interesting applications in general d-dimensional space. One such example is the alignment problem of biological networks (such as protein-protein interaction network or gene regulatory network). Given two biological networks, finding their alignment is a fundamental problem in bioinformatics for detecting the evolution and mutation of species [18]. Since the networks are usually represented by graphs, directly computing their alignment could be very costly and of high time complexity (e.g., reduction to graph isomorphism). One way to solve this problem is to first make use of the fact that biological networks can often be embedded into Euclidean space (due to their intrinsic nature), and then convert the alignment problem from graph domain to geometry domain. Since embedding preserves the pairwise distances of the vertices, each biological network becomes a rigid structure in the Euclidean space. Furthermore, since the matching between networks is not necessarily one-to-one matching, finding their alignment can thus be reduced to minimizing the EMD of two point-sets under rigid transformation, i.e., EMDRT.

A number of results exist for EMDRT and its related problems. Cabello et al. [16] presented several approximation results in $\mathbb{R}^2$ space; particularly, they gave a $(2+\epsilon)$-approximation solution for the 2D EMDRT problem, and a $(1+\epsilon)$-approximation solution for a special case in which only translation is allowed. Later, Klein and Veltkamp [27] introduced a few improved results by using *reference point*, and achieved an $O(2^{d-1})$-approximation for EMDRT in $\mathbb{R}^d$ space. It has been an open problem for quite some time to achieve PTAS for this problem [16]. For static points

(i.e., without transformation), several approximation algorithms for computing EMD were presented in [4–6,16,26,29].

For the related *alignment* (also called *geometric matching*) problem, there is a long and rich history [1–3,7–11,13,14,17,19,22,23,25]. Some early results on this problem can be found in the survey paper by Alt and Guibas [2]. Points in the alignment problem can be matched in an either one-to-one or many-to-one (i.e., Hausdorff distance) manner. For one-to-one matching, Agarwal and Phillips [9] presented a PTAS for $2D$ point-sets using the total squared Euclidean distance between matched points as the cost function. Benkert et al. [11] gave a PTAS for minimizing the bottleneck (i.e., the maximum distance between the matched points) of the $2D$ case. For many-to-one matching, Goodrich et al. [23] obtained constant approximation solutions in 2 and $3D$ spaces, and showed practicality of their algorithms. Later, Gavrilov et al. [22] and Cardoze et al. [17] achieved pseudo-PTAS for $2D$ and fixed dimensional spaces, respectively, with running time depending on the spread ratio of the point-sets. There are also a number heuristic algorithms for the alignment problem. The most popular technique in practice is probably the *Iterative Closest Point (ICP)* algorithm [10], which finds a proper transformation in a step-by-step fashion so that its objective value converges to a local minimum. Ezra et al. [20] showed that the number of iterations of ICP is bounded by $O(m^d n^d)$. Furthermore, if the correspondences (i.e., to-be-matched point pairs) between the two point-sets are known in advance, the alignment problem can be solved optimally by some linear algebra techniques [7].

1.1 Our Results

In this paper, we present the first FPTAS algorithm for EMDRT in any fixed dimensional space. Our result is based on a few new techniques, such as *Sequential Orthogonal Decomposition (SOD)* and *Optimum-Guided-Base (OGB)*. SOD decomposes a rigid transformation into a sequence of primitive operations which enables us to accurately analyze how the transportation flow changes in a step-by-step fashion during the whole process of rigid transformation and therefore have a better estimation on the quality of solution[1]. OGB enables us to use some information of the unknown optimal solution to select some critical points which partially define the rigid transformation. We show that although OGB cannot be explicitly implemented, its result can actually be implicitly obtained. A major advantage of OGB is that it can help us to significantly reduce the search space. Consequently, our FPTAS runs in roughly $O((nm)^{d+2}(\log nm)^{2d})$ time, which is close to (i.e., matches the order of magnitude of the degree of) the lower bound $\max\{m, n\}^{\Omega(d)}$ on the running time of any PTAS for EMDRT [15].

Our techniques for EMDRT can be extended to several related problems, such as the problem of minimizing EMD under similarity transformation (*i.e.*, rigid transformation plus scaling) and the alignment problem, and achieve an FPTAS for each of them. For the alignment problem, we consider three different cost metrics, the l_∞ sense metric

[1] Note that [17,23] also analyze the rigid transformations in a step-by-step fashion, but their methods are only for unweighted point-sets.

and l_1 sense metric in the one-to-one matching case, and the Hausdorff distance metric in the many-to-one matching case. Our result for the Hausdorff distance metric is an FPTAS, while existing results [17,22] are only pseudo-PTAS whose running times depend on the spread ratios.

The rest of the paper is organized as follows. Section 2 introduces a few definitions. Section 3 gives an overview of our approach for achieving FPTAS for the EMDRT problem. In Sect. 4, we discuss our key technique, *Sequential Orthogonal Decomposition (SOD)*, and its properties. In Sect. 5, we present our first FPTAS for the EMDRT problem. Section 6 shows that our FPTAS can be further improved so that its running time is close to the lower bound. Section 7 extends our techniques to several other related problems.

2 Preliminaries

This section introduces several definitions to be used throughout the paper.

Definition 1 *(Rigid Transformation)* Let P be a set of points in $\mathbb{R}^d$. A rigid transformation $\mathcal{T}(P)$ of P is a transformation (i.e., rotation, translation, reflection, or their combination) which preserves the pairwise distances of points in P. After a rigid transformation, the new set of points $\mathcal{T}(P)$ is called an image of P.

Definition 2 (*Earth Mover's Distance (EMD)*) Let $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ be two sets of weighted points in $\mathbb{R}^d$ with nonnegative weights α_i and β_j for each $p_i \in A$ and $q_j \in B$ respectively, and W^A and W^B be their respective total weights. The earth mover's distance between A and B is

$$EMD(A, B) = \frac{\min_F \sum_{i=1}^{n} \sum_{j=1}^{m} h_{ij} ||p_i - q_j||}{\min\{W^A, W^B\}}, \tag{1}$$

where $H = \{h_{ij}\}$ is a feasible flow satisfying the following conditions.

1. $h_{ij} \geq 0$, for any $1 \leq i \leq n$ and $1 \leq j \leq m$;
2. $\sum_{j=1}^{m} h_{ij} \leq \alpha_i$ for any $1 \leq i \leq n$;
3. $\sum_{i=1}^{n} h_{ij} \leq \beta_j$ for any $1 \leq j \leq m$;
4. $\sum_{i=1}^{n} \sum_{j=1}^{m} h_{ij} = \min\left\{W^A, W^B\right\}$.

Definition 3 (*Earth Mover's Distance Under Rigid Transformation (EMDRT)*) Given two weighted point sets A and B in $\mathbb{R}^d$, the problem of the earth mover's distance between A and B under rigid transformation is to determine a rigid transformation $\mathcal{T}$ for B so as to minimize the earth mover's distance $EMD(A, \mathcal{T}(B))$.

Orientation and Reflection For simplicity, we do not consider reflection in the rigid transformation, as it can be captured by performing our algorithm twice, one for the original point-set B and the other for its mirror image.

3 Overview of Our Approach

From Definition 3, we know that our main task for achieving an FPTAS is to find a good approximation of the optimal rigid transformation $\mathcal{T}_{opt}$ for B. By Lemma 1, we will know that this is equivalent to identifying d points R (called *reference system*) from A and another d points (called *base*) from B and determining a rigid transformation $\mathcal{T}$ to map points in the base to the neighborhoods of the points in R.

Our approach consists of two main steps: (1) Design an approximation algorithm to compute an upper bound of the optimal objective value, and (2) use the upper bound to derive an FPTAS. In both steps, directly searching for $\mathcal{T}$ in the rigid transformation space could be very costly. Our idea is to introduce a new technique called *Sequential Orthogonal Decomposition (SOD)*, which decomposes a rigid transformation into a sequence of d primitive operations (i.e., a translation and $d-1$ one-dimensional rotations). One important property of SOD is that its outcome is independent of the initial position of B and depends only on the choice of R and the base[2]. This enables us to assume that B is initially located at $\mathcal{T}_{opt}(B)$. Another important property of SOD is that it allows us to analyze, in a step by step fashion, how the transportation flow changes when B moves from $\mathcal{T}_{opt}(B)$ to $SOD(B)$. This gives us an accurate estimation on the quality of the solution.

The quality of the rigid transformation $\mathcal{T}$ determined by SOD depends on the reference system R and the base. To find a good R, we first build a grid in the neighborhood of each point in A and then consider R as all possible d-tuple point-set where each point comes from the neighbor grid of one individual point in A. To find a good base, a key problem is that a small error in the rotation could cause some point in B to move a long distance and therefore introduce a large error (see Fig. 1). To avoid this problem, we select the base as a set of d ordered points in B which are as "dispersed" as possible (since other choices cause larger error). We consider two types of dispersions. Type 1 dispersion is based on the weighted distance (with weight β_j) between each point q_j and the flat spanned by all points appearing before q_j in the sorted order of the base (see *Algorithm Base-Selection* in Sect. 5.1). Type 2 dispersion considers not only the weighted distance, but also the distance between each q_j and all points in A in the optimal solution (see *Optimum-Guided-Base (OGB)* in Sect. 6). We show that although OGB cannot be explicitly implemented (as it depends on the optimal solution), it can actually be implicitly obtained. A major advantage of the second type of dispersion is that it enables us to significantly improve the running time.

4 Sequential Orthogonal Decomposition

To solve the EMDRT problem, we first introduce the sequential orthogonal decomposition which will be used as a key technique in our algorithms. We start with the following lemma on rigid structure [3].

[2] We just need to determine which point belongs to the base from B, rather than specify the position of each point.

[3] This lemma might have been studied in some other papers or presented in other forms; for self-completeness, we present our proof here.

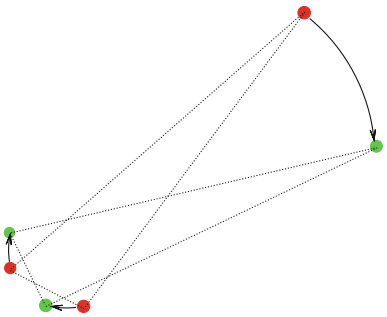

Fig. 1 A small rotation of a non-dispersed base (i.e., the two red points at the *left bottom*) causes a large error for a faraway point at the *right top*

Lemma 1 *For any rigid structure in $\mathbb{R}^d$ with $m \geq d$ vertices (or points), its position is completely fixed if the locations of any d vertices that are not contained in any $(d-2)$-dimensional flat* [4]*, are fixed.*

Proof Let P be any rigid structure in $\mathbb{R}^d$ with vertices $\{p_1, p_2, \ldots, p_m\}$. Without loss of generality, we assume that $\{p_1, p_2, \ldots, p_d\}$ are the d fixed vertices that are not contained in any $(d-2)$-dimensional flat. In other words, $span\{p_1, \ldots, p_d\}$ is a $(d-1)$-dimensional flat.

Consider the process of fixing the positions of the d vertices one by one. In the first step, we fix the position of p_1. Then for any other vertex p_i of P, the feasible region of its location is constrained on a d-dimensional sphere centered at p_1 and with radius $||p_i - p_1||$. If we continue this process, in the j-th step, we have already fixed the positions of $\{p_1, \ldots, p_j\}$. Let $\mathcal{F}_j = span\{p_1, \ldots, p_j\}$ be the flat spanned by the first j vertices, and $\tilde{p}_i$ be the orthogonal projection of p_i on $\mathcal{F}_j$ for any $i > j$. Then the feasible region of the location of p_i is constrained on a $(d-j+1)$-dimensional sphere centered at $\tilde{p}_i$ and with radius $||p_i - \tilde{p}_i||$. Thus, in the d-th step, the positions of $\{p_1, \ldots, p_d\}$ have all been fixed and any vertex p_i, $i > d$, is constrained on a 1-dimensional sphere, *i.e.*, the two endpoints of a segment. Since reflection is not allowed in our rigid transformation, only one endpoint is feasible. Thus, each vertex p_i is fixed, and the lemma is true. □

Sequential Orthogonal Decomposition (SOD) Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points in $\mathbb{R}^d$ with $n \geq d$, and $R = \{r_1, \ldots, r_d\}$ be another set of fixed points (i.e., whose locations will not change), called *reference system*, which spans a $(d-1)$-dimensional flat in $\mathbb{R}^d$. Then for any injective mapping f from $\{1, \ldots, d\}$ to $\{1, \ldots, n\}$, if $\{p_{f(1)}, \ldots, p_{f(d)}\}$ span a $(d-1)$-dimensional flat, we define a sequential orthogonal decomposition, $SOD(P, R, f)$, as follows (before touching the details of *SOD* in general d-dimensional space, the reader is referred to Fig. 2 which illustrates a toy example in 3D).

1. In step 1, perform a translation on P such that $p_{f(1)}$ coincides with r_1. Let p_i^1 be $p_i \in P$ in its new position, and P^1 be the new image of P.
2. In the j-th step for $2 \leq j \leq d$

[4] In $\mathbb{R}^d$, a j-dimensional flat for any $j < d$ is a j-dimensional subspace translated by some vector.

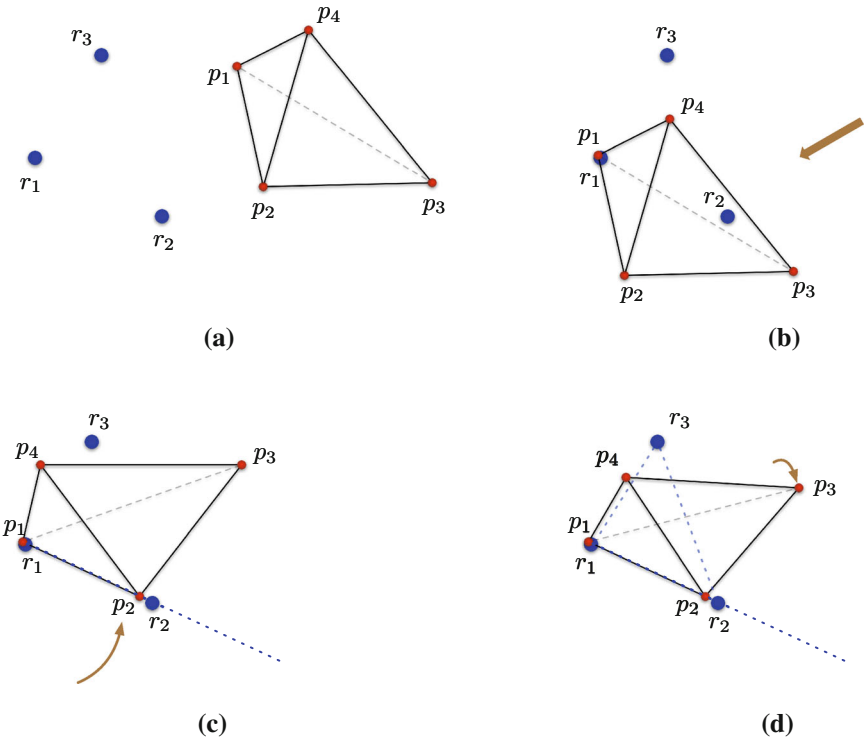

Fig. 2 An illustrative example of SOD in 3D, where $n = 4$ and $f(i) = i$ for $i = 1, 2, 3$. **a** Is the initial positions of the points; the first step **b** is a translation of P to coincide r_1 with p_1; **c** is a rotation of P about r_1 such that r_1, r_2, and p_2 are collinear (the *rotation axis* is the *line* orthogonal to the two dimensional flat spanned by $r_2 - r_1$ and $p_2 - p_1$ before the rotation. See the description of SOD for details); **d** is a rotation of P about the line $\overline{r_1 r_2}$ such that p_3 falls onto the two dimensional flat spanned by r_1, r_2, and r_3

(a) Let $\mathcal{H}_{j-1}$ be the flat spanning $\{r_1, \ldots, r_{j-1}\}$, and $\tilde{p}^{j-1}_{f(j)}$ and $\tilde{r}_j$ be the orthogonal projections of $p^{j-1}_{f(j)}$ and r_j on $\mathcal{H}_{j-1}$, respectively.

(b) Let Δ_j be the 2-dimensional subspace determined by the two vectors, $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$ and $r_j - \tilde{r}_j$, and $\Delta_j^{\perp}$ be the $(d-2)$-dimensional flat orthogonal to Δ_j and containing $\tilde{p}^{j-1}_{f(j)}$.

(c) Perform a one-dimensional rotation $\mathcal{T}$ on P^{j-1} about $\Delta_j^{\perp}$ such that the vector $\mathcal{T}(p^{j-1}_{f(j)}) - \tilde{p}^{j-1}_{f(j)}$ is parallel to the vector $r_j - \tilde{r}_j$ (see Fig. 3).

(d) Let p^j_i denote p_i in its new position, and P^j denote the new image of P.

We first prove the feasibility of SOD.

Theorem 1 (SOD feasibility) *In Step j ($2 \le j \le d$) of SOD, there exists a one-dimensional rotation $\mathcal{T}$ on P^{j-1} about $\Delta_j^{\perp}$ such that the vector $\mathcal{T}\left(p^{j-1}_{f(j)}\right) - \tilde{p}^{j-1}_{f(j)}$ is parallel to the vector $r_j - \tilde{r}_j$.*

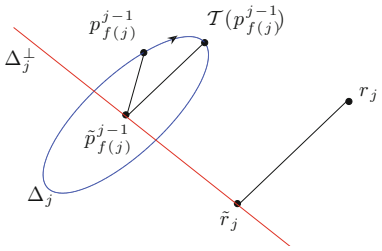

Fig. 3 One-dimensional rotation in SOD

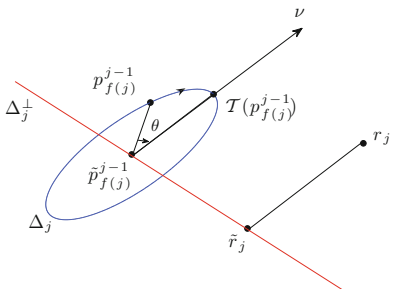

Fig. 4 An illustration for Theorem 1

Proof Firstly, since both $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$ and $r_j - \tilde{r}_j$ are perpendicular to $\mathcal{H}_{j-1}$, we know that $\mathcal{H}_{j-1} \subset \Delta_j^{\perp}$. Thus, for any one-dimensional rotation about $\Delta_j^{\perp}$, the whole $\mathcal{H}_{j-1}$ stays invariant. Since $\tilde{p}^{j-1}_{f(j)} \in \mathcal{H}_{j-1}$, we have $\mathcal{T}(\tilde{p}^{j-1}_{f(j)}) = \tilde{p}^{j-1}_{f(j)}$.

Let ν be the ray starting from $\tilde{p}^{j-1}_{f(j)}$ and parallel to $r_j - \tilde{r}_j$, and θ be the angle between $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$ and ν (see Fig. 4). Consider the one-dimensional rotation about $\Delta_j^{\perp}$ with angle θ. Since both ν and $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$ are perpendicular to $\Delta_j^{\perp}$, $\mathcal{T}(p^{j-1}_{f(j)})$ must locate on the two-dimensional flat determined by ν and $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$. Moreover, since $\mathcal{T}(\tilde{p}^{j-1}_{f(j)}) = \tilde{p}^{j-1}_{f(j)}$ and the rotation angle is θ, vector $\mathcal{T}(p^{j-1}_{f(j)}) - \tilde{p}^{j-1}_{f(j)}$ must be collinear with ν. In other words, $\mathcal{T}(p^{j-1}_{f(j)}) - \tilde{p}^{j-1}_{f(j)}$ is parallel to the vector $r_j - \tilde{r}_j$. □

4.1 Some Properties of SOD

In this section we reveal several important properties of SOD, which will be used to analyze the proposed algorithms for EMDRT. For convenience, we still use the ray ν constructed in the proof of Theorem 1.

Lemma 2 *Let $\mathcal{I}(P)$ be any image of P (i.e., $\mathcal{I}(P)$ is the new P after a rigid transformation). Then the outputs of SOD(P, R, f) and SOD$(\mathcal{I}(P), R, f)$ are the same.*

To prove Lemma 2, we need the following result.

Lemma 3 *In the j-th step of SOD, $p^{j}_{f(j)}$ locates on $\mathcal{H}_j$.*

Proof Notice that $p^j_{f(j)}$ is the new position of $p_{f(j)}$ after performing the rotation in the j-th step, and $\mathcal{H}_j = span\{r_1, \ldots, r_j\} = span\{\mathcal{H}_{j-1}, r_j - \tilde{r}_j\}$. From the proof of Theorem 1, we know that $p^j_{f(j)}$ locates on the ray ν. Furthermore, since ν is parallel to $r_j - \tilde{r}_j$ and emits from $\tilde{p}^{j-1}_{f(j)}$ (which locates on $\mathcal{H}_{j-1}$), we know that ν locates on $\mathcal{H}_j$. Thus, $p^j_{f(j)}$ also locates on $\mathcal{H}_j$. □

With the above lemma, we now prove Lemma 2.

Proof (**of Lemma** 2) Let $\mathcal{I}(p_i)$ and $\mathcal{I}(p_i)^j$ denote the corresponding points of p_i and p^j_i respectively in $\mathcal{I}(P)$. From Lemma 3, we know that $p^j_{f(j)}$ locates on $\mathcal{H}_j$, $1 \le j \le d$. Note that for any $j < l \le d$, we have $\mathcal{H}_j \subset \mathcal{H}_{l-1} \subset \Delta^{\perp}_l$. Thus, the position of $p_{f(j)}$ does not change after the j-th step (*i.e,* always located at $p^j_{f(j)}$). Thus, we just need to show that $p^j_{f(j)} = \mathcal{I}(p_{f(j)})^j$ for each $1 \le j \le d$, and then use Lemma 1 to complete the proof. Below we prove $p^j_{f(j)} = \mathcal{I}(p_{f(j)})^j$ by mathematics induction on j.

Base case For $j = 1$, it is easy to know that $p^1_{f(1)} = \mathcal{I}(p_{f(1)})^1 = r_1$.

Induction step Assume that $p^l_{f(l)} = \mathcal{I}(p_{f(l)})^l$ for any $l \le j-1$ and $j \ge 2$. Recall that in the SOD procedure, we denote the images of P and $\mathcal{I}(P)$ as P^j and $\mathcal{I}(P)^j$ respectively in the j-th step. Thus, $\mathcal{I}(P)^j$ is also an image of P^j under some rigid transformation $\mathcal{T}_{\mathcal{R}}$. From the induction hypothesis, we know that all $p^l_{f(l)}$'s stay invariant under the rigid transformation $\mathcal{T}_{\mathcal{R}}$. By Lemma 3, we know that $\mathcal{H}_{j-1} = span\{p^1_{f(1)}, \ldots, p^{j-1}_{f(j-1)}\}$. Thus, the whole $\mathcal{H}_{j-1}$ also stay invariant under rigid transformation $\mathcal{T}_{\mathcal{R}}$. Meanwhile, since $\tilde{p}^{j-1}_{f(j)}$ locates on $\mathcal{H}_{j-1}$, $\tilde{p}^{j-1}_{f(j)}$ stays invariant as well under rigid transformation $\mathcal{T}_{\mathcal{R}}$, i.e., $\tilde{p}^{j-1}_{f(j)} = \mathcal{T}_{\mathcal{R}}(\tilde{p}^{j-1}_{f(j)})$, where $\mathcal{T}_{\mathcal{R}}(\tilde{p}^{j-1}_{f(j)})$ is the projection of $\mathcal{I}(p_{f(j)})^{j-1}$ on $\mathcal{H}_{j-1}$. From the proof of Theorem 1, we know that both $p^j_{f(j)}$ and $\mathcal{I}(p_{f(j)})^j$ locate on ray ν. Since $\tilde{p}^{j-1}_{f(j)} = \mathcal{T}_{\mathcal{R}}(\tilde{p}^{j-1}_{f(j)})$ and from the fact that rigid transformation preserves pairwise distance, i.e., $||p^j_{f(j)} - \tilde{p}^{j-1}_{f(j)}|| = ||\mathcal{I}(p_{f(j)})^j - \mathcal{T}_{\mathcal{R}}(\tilde{p}^{j-1}_{f(j)})||$, we know that $p^j_{f(j)} = \mathcal{I}(p_{f(j)})^j$. This completes the induction step, and the lemma is true. □

Lemma 4 *For* $2 \le j \le d$, $||p^j_{f(j)} - p^{j-1}_{f(j)}|| \le 2||p^{j-1}_{f(j)} - r_j||$.

Proof Firstly, from triangle inequality, we know that

$$||p^j_{f(j)} - p^{j-1}_{f(j)}|| \le ||p^j_{f(j)} - r_j|| + ||p^{j-1}_{f(j)} - r_j||. \tag{2}$$

Next, we prove that $||p^j_{f(j)} - r_j|| \le ||p^{j-1}_{f(j)} - r_j||$. For simplicity, we let $x = ||\tilde{p}^{j-1}_{f(j)} - \tilde{r}_j||$, $y = ||r_j - \tilde{r}_j||$, and $z = ||p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}||$. Since both $\tilde{p}^{j-1}_{f(j)}$ and $\tilde{r}_j$ locate on $\mathcal{H}_{j-1}$, we know that vector $\tilde{p}^{j-1}_{f(j)} - \tilde{r}_j$ is perpendicular to both $r_j - \tilde{r}_j$ and $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$. Recall that θ is the angle between $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$ and ν in the

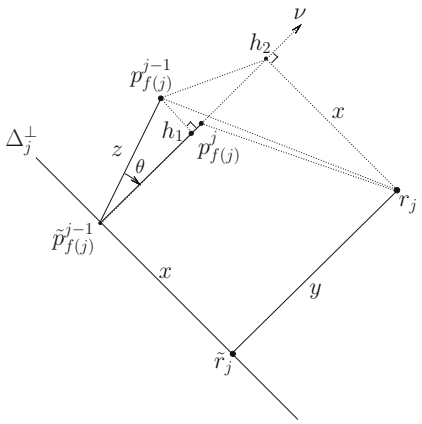

Fig. 5 An illustration for Lemma 4

proof of Theorem 1 (see Fig. 5). Let h_1 and h_2 denote the projections of $p^{j-1}_{f(j)}$ and r_j on ray ν, respectively. Because $r_j - \tilde{r}_j$ and $h_2 - \tilde{p}^{j-1}_{f(j)}$ are parallel with each other, and both $\tilde{r}_j - \tilde{p}^{j-1}_{f(j)}$ and $r_j - h_2$ are perpendicular to $r_j - \tilde{r}_j$ and $h_2 - \tilde{p}^{j-1}_{f(j)}$, we know that the four points $\{\tilde{r}_j, r_j, h_2, \tilde{p}^{j-1}_{f(j)}\}$ form a rectangle. Meanwhile, vector $r_j - h_2$ is perpendicular to the two-dimensional flat determined by ν and $p^{j-1}_{f(j)} - \tilde{p}^{j-1}_{f(j)}$. Thus, we have

$$\begin{aligned} ||p^{j-1}_{f(j)} - r_j||^2 &= ||h_2 - r_j||^2 + ||p^{j-1}_{f(j)} - h_2||^2 \\ &= ||h_2 - r_j||^2 + ||p^{j-1}_{f(j)} - h_1||^2 + ||h_1 - h_2||^2 \\ &= x^2 + (z\sin\theta)^2 + (y - z\cos\theta)^2 \\ &= x^2 + y^2 + z^2 - 2yz\cos\theta, \end{aligned} \tag{3}$$

$$\begin{aligned} ||p^{j}_{f(j)} - r_j||^2 &= ||h_2 - r_j||^2 + ||p^{j}_{f(j)} - h_2||^2 \\ &= x^2 + (y - z)^2 \\ &= x^2 + y^2 + z^2 - 2yz. \end{aligned} \tag{4}$$

Comparing (3) and (4), we know that $||p^{j}_{f(j)} - r_j|| \le ||p^{j-1}_{f(j)} - r_j||$. Combining this and (2), we have $||p^{j}_{f(j)} - p^{j-1}_{f(j)}|| \le 2||p^{j-1}_{f(j)} - r_j||$. □

As for the running time of SOD, we know that there are d steps in the process, and each step involves computing the projection of one point to the corresponding flat, which costs $O(d^3)$ time. Thus, we have the following lemma.

Lemma 5 *SOD can be performed in $O(|P|d^4)$ time.*

5 FPTAS For EMDRT

In this section, we present an FPTAS for EMDRT. Our algorithm first applies SOD to obtain an upper bound on the optimal objective value of EMDRT, and then use

it to determine a proximity region (called grid-ball) for each point in A containing its possible match in B. By searching a grid in each such grid-ball, we show that an FPTAS can be attained for EMDRT.

To solve EMDRT, a basic problem is to determine the earth mover's distance (EMD) between two sets of fixed points without considering any rigid transformation. In [16], Cabello et al. introduced a $(1+\epsilon)$-approximation algorithm for computing EMD in a plane, and generalized it to any d-dimensional space. Below is a lemma proved in [16].

Lemma 6 ([16]) *Given two weighted point sets A and B in $\mathbb{R}^d$ and a small $\epsilon > 0$, there exists an algorithm which outputs a $(1+\epsilon)$-approximation of EMD between A and B in $O((n^2/\epsilon^{2(d-1)})\log^2(n/\epsilon))$ time, where $n = \max\{|A|, |B|\}$.*

5.1 Upper Bound

For the upper bound, we notice that although [27] provides an $O(2^{d-1})$-approximate solution, it cannot be used as an upper bound as it assumes that the two input point sets have equal total weight, which may not be the case in our problem. To obtain an upper bound for EMDRT, we need the following definition.

Definition 4 (*Bottleneck*) Let $H=\{h_{ij}\}$ be any feasible flow between A and B in Definition 2. Then the Bottleneck of H is defined as $\mathcal{BN}(A, B, H) = \max_{i,j}\{\frac{h_{ij}||p_i - q_j||}{\min\{W^A, W^B\}}\}$.

From the above definition, we immediately have the following lemma.

Lemma 7 *For any feasible flow H between A and B,*

$$\frac{1}{nm}\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} h_{ij}||p_i - q_j||}{\min\{W^A, W^B\}} \le \mathcal{BN}(A, B, H) \le \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} h_{ij}||p_i - q_j||}{\min\{W^A, W^B\}}.$$

Our main idea for obtaining an upper bound is to first identify a good base from B, then enumerate all possible subsets of A with cardinality d as the reference system R, and finally use SOD to obtain a rigid transformation for B. The criterium for selecting the base is to make them as "dispersed" as possible, where the dispersiveness is measured by the weighted distance between each point $q_j \in B$ and the flat spanned by all determined base points.

Algorithm Upper-Bound-for-EMDRT
Input Two weighted point sets $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ in $\mathbb{R}^d$ with weight $\alpha_i \ge 0$ and $\beta_j \ge 0$ for p_i and q_j respectively, and $W^A \ge W^B$.
Output An upper bound on $\min_{\mathcal{T}} EMD(A, \mathcal{T}(B))$.

1. Call Base-Selection on B, and let $\{q_{b(1)}, \ldots, q_{b(d)}\}$ be the output base.
2. Enumerate all d-point tuples from $[A]^d = A \times \cdots \times A$. For each tuple $R = \{p_{i(1)}, \ldots, p_{i(d)}\}$, Do
   (a) Define the mapping such that $f(i(j)) = b(j)$ for each $1 \le j \le d$.
      (a) Define the mapping such that $f(i(j)) = b(j)$ for each $1 \le j \le d$.

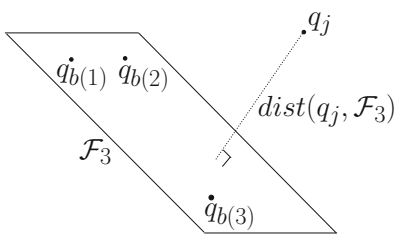

Fig. 6 An illustration for Algorithm base-selection (for $l = 3$)

(b) Emulate the execution of the procedure $SOD(B, R, f)$, and stop at the final step or at the l-th step if $p_{i(l)}$ locates on the flat $span\{p_{i(1)}, \ldots, p_{i(l-1)}\}$ [5]. Then compute the $(1+\epsilon)$-approximation of EMD between A and the output the image of B by using the algorithm in Lemma 6.

3. Output the image of B which has the minimum EMD to A among all images of B corresponding to the tuples of $[A]^d$.

Algorithm Base-Selection
Input A weighted point set $B = \{q_1, \ldots, q_m\}$ in $\mathbb{R}^d$ with nonnegative weight β_j for each q_j.
Output A base which is an ordered subset $\{q_{b(1)}, \ldots, q_{b(d)}\}$ of points in B.

1. Select the point with largest weight from B, and denote it as $q_{b(1)}$. Let $l = 1$, and repeat the following steps until $l = d$.
   (a) Let $\mathcal{F}_l$ be the flat spanned by $\{q_{b(1)}, \ldots, q_{b(l)}\}$. See Fig. 6.
   (b) Find the point realizing $\max\{\beta_j \cdot dist(q_j, \mathcal{F}_l) \mid 1 \le j \le m\}$, and denote it as $q_{b(l+1)}$, where $dist(q_j, \mathcal{F}_l)$ is the distance between q_j and $\mathcal{F}_l$.
   (c) Let $l = l + 1$.
2. Output $\{q_{b(1)}, \ldots, q_{b(d)}\}$.

Theorem 2 *The algorithm of Upper-Bound-for-EMDRT yields in $O(n^{d+2}(\log n)^2 md^4)$ time an upper bound Π which is a $((1+\epsilon)nm(n+1)(2n+1)^{d-1})$-approximation of the optimal objective value.*

To prove Theorem 2, we need the following lemma. Let $\mathcal{T}_{opt}$ be an optimal rigid transformation (i.e., the one realizing the value of $\min_{\mathcal{T}} EMD(A, \mathcal{T}(B))$), and $\overline{H} = \{\overline{h}_{ij}\}$ be the corresponding optimal flow. Since the algorithm enumerates all d-point tuples in $[A]^d$, we just need to focus on the tuple $\{p_{i(1)}, \ldots, p_{i(d)}\}$ which has $\overline{h}_{i(j)b(j)} = \max\{\overline{h}_{ib(j)} \mid 1 \le i \le n\}$ for each $1 \le j \le d$. With a slight abuse of notation, we use $R = \{p_{i(1)}, \ldots, p_{i(d)}\}$ to denote the tuple which satisfies this requirement.

Lemma 8 *Let $\mathcal{I}(B)$ be the final output image of B by $SOD(B, R, f)$ in the above algorithm. Then $\mathcal{BN}(A, \mathcal{I}(B), \overline{H}) \le (n+1)(2n+1)^{d-1}\mathcal{BN}(A, \mathcal{T}_{opt}(B), \overline{H})$.*

[5] Actually, another reason causing stop is that $q_{b(l)}$ locates on the flat $span\{q_{b(1)}, \ldots, q_{b(l-1)}\}$ for some $2 \le l \le d$. If that happens, recalling the Algorithm Base-Selection, we know that the whole B would locate on the $(l-2)$-dimensional flat. In this case we can assume that $SOD(B, R, f)$ does not stop in the following $d - l + 1$ steps, but the corresponding $d - l + 1$ rotation angles are all zeros.

Proof We analyze how the Bottleneck changes from $\mathcal{BN}(A, \mathcal{T}_{opt}(B), \overline{H})$ to $\mathcal{BN}(A, \mathcal{I}(B), \overline{H})$ in a step-by-step fashion. Note that the original position of B may not be the same as $\mathcal{T}_{opt}(B)$. By Lemma 2, we know that the final result of $SOD(B, R, f)$ is always the same. Thus, we can assume that the original position of B is at $\mathcal{T}_{opt}(B)$ if the emulation in Step 2(b) finishes all steps of $SOD(B, R, f)$. Furthermore, if the emulation stops at the l-th step for some $l \leq d$, in this case, by the proof of Lemma 2 we know that the first $l-1$ points $\{q_{b(1)}, \ldots, q_{b(l-1)}\}$ always stay at their respective same positions even if B does not locate at $\mathcal{T}_{opt}(B)$. Below we will show that it does not affect us from bounding the Bottleneck, even if it happens.

Let $u_0 = \mathcal{BN}(A, \mathcal{T}_{opt}(B), \overline{H})$ be the Bottleneck at the beginning, and u_j be the Bottleneck in the j-th step of the *SOD* procedure. Then in the first step, we have the following for any $1 \leq t \leq n$ and $1 \leq l \leq m$.

$$\begin{aligned} \overline{h}_{tl}||p_t - q_l^1|| &\leq \overline{h}_{tl}(||p_t - q_l|| + ||q_l - q_l^1||) \\ &\leq \overline{h}_{tl}||p_t - q_l|| + \beta_{b(1)}||q_l - q_l^1|| \\ &= \overline{h}_{tl}||p_t - q_l|| + n \times \frac{\beta_{b(1)}}{n}||q_{b(1)} - p_{i(1)}||, \end{aligned} \tag{5}$$

where q_l^1 is the new position of q_l after the first step, the second inequality follows from the fact that $\beta_{b(1)}$ is the largest weight (see Step 1 of the Base-Selection algorithm), and the equality follows from the translation performed in the first step of SOD (i.e., $||q_l - q_l^1|| = ||q_{b(1)} - p_{i(1)}||$). Since $\overline{h}_{i(j)b(j)} = \max\{\overline{h}_{ib(j)} \mid 1 \leq i \leq n\}$, we know that $\overline{h}_{i(1)b(1)} \geq \frac{\beta_{b(1)}}{n}$. Combining this and (5), we have $\overline{h}_{tl}||p_t - q_l^1|| \leq \overline{h}_{tl}||p_t - q_l|| + n\overline{h}_{i(1)b(1)}||q_{b(1)} - p_{i(1)}||$. Thus by Definition 4, we have $u_1 \leq (1+n)u_0$.

In the j-th step ($j \geq 2$), there are two cases: (1) $p_{i(j)}$ does not locate on the flat determined by $\{p_{i(1)}, \ldots, p_{i(j-1)}\}$ and (2) $p_{i(j)}$ locates on the flat. For case (1), by Lemma 4, we know that

$$||q_{b(j)}^j - q_{b(j)}^{j-1}|| \leq 2||q_{b(j)}^{j-1} - p_{i(j)}||, \tag{6}$$

where $q_{b(j)}^{j-1}$ and $q_{b(j)}^j$ are point $q_{b(j)}$ at its new locations in the $(j-1)$-th and j-th steps respectively. The following claim is a key to case (1).

Claim 1 *In case (1), for any* $1 \leq t \leq n$ *and* $1 \leq l \leq m$,

$$\overline{h}_{tl}||p_t - q_l^j|| \leq \overline{h}_{tl}||p_t - q_l^{j-1}|| + 2n\overline{h}_{i(j)b(j)}||q_{b(j)}^{j-1} - p_{i(j)}||.$$

Proof Note that when we perform the one-dimensional rotation in the j-th step of *SOD*, the loci of $\{q_l \mid 1 \leq l \leq m\}$ are a set of co-centered circles with radius $dist(q_l^{j-1}, \Delta_j^{\perp})$ for each l. Let θ be the angle rotated in the one-dimensional rotation. Then, since $dist(q_l^{j-1}, \Delta_j^{\perp}) = dist(q_l^j, \Delta_j^{\perp})$ (see Fig. 7), we have

$$||q_l^j - q_l^{j-1}|| = 2dist\left(q_l^{j-1}, \Delta_j^{\perp}\right) \sin\frac{\theta}{2}. \tag{7}$$

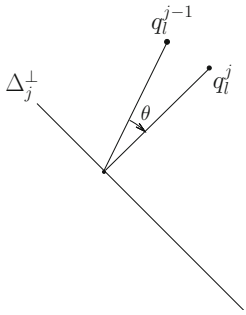

Fig. 7 An illustration for Claim 1

From the Base-Selection algorithm, we know that

$$\beta_l \cdot dist\left(q_l^{j-1}, \Delta_j^\perp\right) \leq \beta_{b(j)} \cdot dist(q_{b(j)}^{j-1}, \Delta_j^\perp). \tag{8}$$

Thus, combining (6), (7) and (8), we have

$$\begin{aligned} ||q_l^j - q_l^{j-1}|| &= 2dist\left(q_l^{j-1}, \Delta_j^\perp\right) \sin\frac{\theta}{2} \\ &\leq 2\frac{\beta_{b(j)}}{\beta_l} dist\left(q_{b(j)}^{j-1}, \Delta_j^\perp\right) \sin\frac{\theta}{2} \\ &= \frac{\beta_{b(j)}}{\beta_l} ||q_{b(j)}^j - q_{b(j)}^{j-1}|| \\ &\leq 2\frac{\beta_{b(j)}}{\beta_l} ||q_{b(j)}^{j-1} - p_{i(j)}||, \end{aligned} \tag{9}$$

where the first inequality follows from (8), the second equality follows from (7) (replacing l by $b(j)$), and the second inequality follows from (6). By the assumption that $\overline{h}_{i(j)b(j)} = \max\{\overline{h}_{ib(j)} \mid 1 \leq i \leq n\}$, we get $\overline{h}_{i(j)b(j)} \geq \frac{\beta_{b(j)}}{n}$. Thus, from (9) we have

$$\begin{aligned} \overline{h}_{tl}||p_t - q_l^j|| &\leq \overline{h}_{tl}\left(||p_t - q_l^{j-1}|| + ||q_l^j - q_l^{j-1}||\right) \\ &\leq \overline{h}_{tl}||p_t - q_l^{j-1}|| + \beta_l||q_l^j - q_l^{j-1}|| \\ &\leq \overline{h}_{tl}||p_t - q_l^{j-1}|| + 2\beta_{b(j)}||q_{b(j)}^{j-1} - p_{i(j)}|| \\ &\leq \overline{h}_{tl}||p_t - q_l^{j-1}|| + 2n\overline{h}_{i(j)b(j)}||q_{b(j)}^{j-1} - p_{i(j)}||, \end{aligned} \tag{10}$$

where the first inequality follows from triangle inequality, the second follows from the fact that $\overline{h}_{tl} \leq \beta_l$, the third follows from (9), and the forth follows from the fact that $\overline{h}_{i(j)b(j)} \geq \frac{\beta_{b(j)}}{n}$. This completes the proof of Claim 1. □

Now we resume to proving case (1). From Definition 4, we know that both $\overline{h}_{tl}||p_t - q_l^{j-1}||$ and $\overline{h}_{i(j)b(j)}||q_{b(j)}^{j-1} - p_{i(j)}||$ are no more than $u_{j-1}W^B$. Thus, we have $u_j \leq (1+2n)u_{j-1}$ for case (1).

For case (2), we know that $p_{i(j)}$ locates on the flat spanning $\{p_{i(1)}, \ldots, p_{i(j-1)}\}$. This means that the emulation in Step 2(b) stops at the j-th step of SOD. Let $\tilde{q}_{b(j)}^{j-1}$ be the projection of $q_{b(j)}^{j-1}$ on the flat $span\{p_{i(1)}, \ldots, p_{i(j-1)}\}$. As we mentioned earlier, in this case the first $j-1$ points $\{q_{b(1)}, \ldots, q_{b(j-1)}\}$ always stay at their respective same positions even if B does not originally located at $\mathcal{T}_{opt}(B)$. It is easy to see that $span\{p_{i(1)}, \ldots, p_{i(j-1)}\} = span\{q_{b(1)}^{j-1}, \ldots, q_{b(j-1)}^{j-1}\}$. Thus, $q_{b(j)}^{j-1}$ is constrained on the $(d-j+2)$-dimensional sphere, which is centered at $\tilde{q}_{b(j)}^{j-1}$ and with radius $dist(q_{b(j)}^{j-1}, \Delta_j^{\perp}) = ||q_{b(j)}^{j-1} - \tilde{q}_{b(j)}^{j-1}||$. By triangle inequality, we have

$$\begin{aligned} ||q_{b(j)}^{j} - q_{b(j)}^{j-1}|| &\le ||q_{b(j)}^{j} - \tilde{q}_{b(j)}^{j-1}|| + ||q_{b(j)}^{j-1} - \tilde{q}_{b(j)}^{j-1}|| \\ &= 2||q_{b(j)}^{j-1} - \tilde{q}_{b(j)}^{j-1}|| \\ &\le 2||q_{b(j)}^{j-1} - p_{i(j)}||, \end{aligned} \tag{11}$$

where the second inequality follows from the facts that $p_{i(j)}$ locates on $\Delta_j^{\perp}$ and $\tilde{q}_{b(j)}^{j-1}$ is the projection of $q_{b(j)}^{j-1}$ on $\Delta_j^{\perp}$, which imply that $||q_{b(j)}^{j-1} - \tilde{q}_{b(j)}^{j-1}|| \le ||q_{b(j)}^{j-1} - p_{i(j)}||$. Since (11) is equivalent to (6) in case (1), by a similar argument, we know that Claim 1 also holds for case (2). This means that $u_j \le (1+2n)u_{j-1}$.

In summary, we have $u_j \le (1+n)(1+2n)^{j-1}u_0$ for either case. This implies that $\mathcal{BN}(A, \mathcal{I}(B), \overline{H}) \le (n+1)(2n+1)^{d-1}\mathcal{BN}(A, \mathcal{T}_{opt}(B), \overline{H})$. □

With the above lemma, we now prove Theorem 2.

Proof (**of Theorem** 2) Let $\Pi_{opt} = EMD(A, \mathcal{T}_{opt}(B))$, and Π be the objective value returned by the algorithm. Obviously, $\Pi_{opt} \le \Pi$. By Lemma 7, we know

$$\Pi \le nm\mathcal{BN}(A, \mathcal{I}(B), \overline{H}), \tag{12}$$

$$\mathcal{BN}(A, \mathcal{T}_{opt}(B), \overline{H}) \le \Pi_{opt}. \tag{13}$$

By Lemma 8, we immediately have $\Pi \le nm(n+1)(2n+1)^{d-1}\Pi_{opt}$. Since the algorithm in Lemma 6 only yields a $(1+\epsilon)$-approximation of *EMD*, we have an additional $(1+\epsilon)$ factor in the approximation ratio.

Running time It is easy to see that the Base-Selection algorithm takes $O(md^4)$ time. Since we enumerate all n^d tuples in $[A]^d$ and each tuple corresponds to a call to the procedure of *SOD* which costs $O(md^4)$ time (by Lemma 5), the total running time is thus $O(n^{d+2}(\log n)^2 md^4)$ (including the time by the algorithm in Lemma 6), where the hidden constant depends on ϵ and d. □

5.2 The FPTAS Algorithm

The upper bound determined in last section enables us to achieve a $(1+\epsilon)$-approximation for EMDRT in the following way. We first draw d balls (called *Grid-Ball*) centered at the d points of A respectively, and then build a grid inside

each ball. For each d-grid-point tuple (one from each grid-ball), emulate the *SOD* procedure on B. The purpose of using grid points is for determining more "accurate" rigid transformation for B. To implement this approach, we need to resolve two major issues: (1) What is the radius of each grid-ball and (2) what is the density of the grids. Below we discuss our ideas on each of them.

Lemma 9 *If $W^A \geq W^B$ in EMDRT, then for any $1 \leq j \leq m$,*

$$\min_{1\leq i\leq n} ||p_i - \mathcal{T}_{opt}(q_j)|| \leq \frac{nW^B}{\beta_j} EMD(A, \mathcal{T}_{opt}(B)). \tag{14}$$

Proof Let $\overline{H} = \{\overline{h}_{ij}\}$ be the flow realizing the distance of $EMD(A, \mathcal{T}_{opt}(B))$ (i.e., $\overline{H}$ is the optimal flow). From Definition 2, we know that for any $1 \leq i \leq n$ and $1 \leq j \leq m$,

$$||p_i - \mathcal{T}_{opt}(q_j)|| \leq \frac{W^B}{\overline{h}_{ij}} EMD(A, \mathcal{T}_{opt}(B)). \tag{15}$$

By pigeonhole principle, we know that $\max_{1\leq i\leq n} \overline{h}_{ij} \geq \frac{\beta_j}{n}$. Combining this and (15), we have (14). □

Lemma 9 indicates that for each q_j, there exists some p_i whose grid-ball contains $\mathcal{T}_{opt}(q_j)$ if its radius is set to be $\frac{nW^B}{\beta_j} EMD(A, \mathcal{T}_{opt}(B))$. However, since $EMD(A, \mathcal{T}_{opt}(B))$ is unknown, we cannot directly use $\frac{nW^B}{\beta_j} EMD(A, \mathcal{T}_{opt}(B))$ as the radius. Since the algorithm of Upper-Bound-for-EMDRT yields an upper bound of $EMD(A, \mathcal{T}_{opt}(B))$, we can use it to approximate the radius. The following lemma suggests a way to determine the density of the grid.

Lemma 10 *If $W^A \geq W^B$ and $\mathcal{T}$ is a rigid transformation for B such that for each $1 \leq j \leq m$, at least one of the following two conditions hold,*

1. $||\mathcal{T}(q_j) - \mathcal{T}_{opt}(q_j)|| \leq c\epsilon \min_{1\leq i\leq n} ||p_i - \mathcal{T}_{opt}(q_j)||$;
2. $||\mathcal{T}(q_j) - \mathcal{T}_{opt}(q_j)|| \leq c\epsilon \frac{W^B}{m\beta_j} EMD(A, \mathcal{T}_{opt}(B))$,

where $c > 0$ is a constant, then $EMD(A, \mathcal{T}(B)) \leq (1 + 2c\epsilon) EMD(A, \mathcal{T}_{opt}(B))$.

Proof By Definition 2 and triangle inequality, we have

$$EMD(A, \mathcal{T}(B)) \leq EMD(A, \mathcal{T}_{opt}(B)) + \frac{1}{W^B} \sum_{j=1}^{m} \beta_j ||\mathcal{T}(q_j) - \mathcal{T}_{opt}(q_j)||. \tag{16}$$

Consider the two conditions in the lemma. Firstly, we assume that Condition 1 holds for all $1 \leq j \leq m$ (i.e., $||\mathcal{T}(q_j) - \mathcal{T}_{opt}(q_j)|| \leq c\epsilon \min_{1\leq i\leq n} ||p_i - \mathcal{T}_{opt}(q_j)||$). Then we know that

$$\begin{aligned}\beta_j||\mathcal{T}(q_j)-\mathcal{T}_{opt}(q_j)|| &\le c\epsilon\beta_j \min_{1\le i\le n}||p_i-\mathcal{T}_{opt}(q_j)||\\ &= c\epsilon\sum_{i=1}^{n}\overline{h}_{ij}\min_{1\le i\le n}||p_i-\mathcal{T}_{opt}(q_j)||\\ &\le c\epsilon\sum_{i=1}^{n}\overline{h}_{ij}||p_i-\mathcal{T}_{opt}(q_j)||, \qquad (17)\end{aligned}$$

where the first equality follows from the fact that $\beta_j=\sum_{i=1}^{n}\overline{h}_{ij}$. Summing both sides of (17) over j, we have

$$\sum_{j=1}^{m}\beta_j||\mathcal{T}(q_j)-\mathcal{T}_{opt}(q_j)||\le c\epsilon\sum_{j=1}^{m}\sum_{i=1}^{n}\overline{h}_{ij}||p_i-\mathcal{T}_{opt}(q_j)||=c\epsilon W^B EMD(A,\mathcal{T}_{opt}(B)).$$

Combining the above inequality and (16), we complete the proof for this case.

Secondly, we assume that Condition 2 holds for all $1\le j\le m$ (i.e., $||\mathcal{T}(q_j)-\mathcal{T}_{opt}(q_j)||\le c\epsilon\frac{W^B}{m\beta_j}EMD(A,\mathcal{T}_{opt}(B))$). Then we have

$$\sum_{j=1}^{m}\beta_j||\mathcal{T}(q_j)-\mathcal{T}_{opt}(q_j)||\le c\epsilon\sum_{j=1}^{m}\beta_j\frac{W^B}{m\beta_j}EMD(A,\mathcal{T}_{opt}(B))=c\epsilon W^B EMD(A,\mathcal{T}_{opt}(B)).$$

Thus, by (16), we know that the lemma holds for the second case as well.

Finally, we consider the case where different j may satisfy different conditions.

In this case, we can use the same strategy to prove the result. The only difference is that we need to add a factor of 2 (i.e., $\sum_{j=1}^{m}\beta_j||\mathcal{T}(q_j)-\mathcal{T}_{opt}(q_j)||\le 2c\epsilon W^B EMD(A,\mathcal{T}_{opt}(B))$). Thus the lemma holds for all cases. □

Algorithm FPTAS-for-EMDRT

Input Two weighted point sets $A=\{p_1,\ldots,p_n\}$ and $B=\{q_1,\ldots,q_m\}$ in $\mathbb{R}^d$ with weight $\alpha_i\ge 0$ and $\beta_j\ge 0$ for p_i and q_j respectively, and $W^A\ge W^B$; a small number $\epsilon>0$.

Output A rigid transformation $\mathcal{T}$ for B with $EMD(A,\mathcal{T}(B))\le(1+\epsilon)EMD(A,\mathcal{T}_{opt}(B))$.

1. Call Upper-Bound-for-EMDRT in Sect. 5.1, and let Π be the yielded upper bound. Construct the set $\Gamma=\{\frac{\Pi}{2^t}\mid t=0,1,\ldots,2d\log(\max\{n,m\})\}$.
2. Call Base-Selection on B in Sect. 5.1, and let $\{q_{b(1)},\ldots,q_{b(d)}\}$ be the base.
3. Enumerate all d-point tuples from $[A]^d=A\times\cdots\times A$. For each tuple $R=\{p_{i(1)},\ldots,p_{i(d)}\}$, do
   (a) Call Grid-Construction algorithm, and let $\{G_1,\ldots,G_d\}$ be the output.
   (b) For each tuple $\{g_1,\ldots,g_d\}\in G_1\times G_2\times\cdots\times G_d$, emulate SOD on B, $\{g_1,\ldots,g_d\}$, and the mapping f, where $f(j)=b(j)$. Compute the EMD between A and the output image of B.
4. Output the image of B among all the obtained images which has the minimum EMD to A.

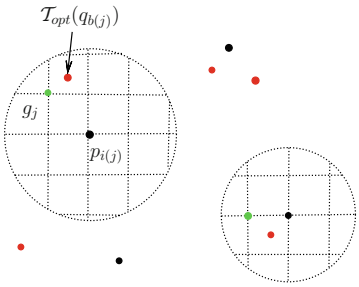

Fig. 8 *Black points* are A, *red points* are $\mathcal{T}_{opt}(B)$, and *green points* are grid points (Color figure online)

Algorithm Grid-Construction
Input Γ, $\{q_{b(1)}, \ldots, q_{b(d)}\}$ and $R = \{p_{i(1)}, \ldots, p_{i(d)}\}$.
Output Grids $\{G_1, \ldots, G_d\}$

1. For each $1 \leq j \leq d$, do:
   (a) Build the set of radius candidates $\{\frac{nW^B}{\beta_{b(j)}}\gamma \mid \gamma \in \Gamma\}$.
   (b) For each radius candidate r, construct a grid-ball centered at $p_{i(j)}$ and with radius r, and build a grid inside the ball with grid length $\frac{r\epsilon}{8 \cdot 3^d nm\sqrt{d}}$.
   (c) Let G_j denote the union of the grids inside all the grid-balls.
2. Output $\{G_1, \ldots, G_d\}$.

Theorem 3 *The above FPTAS-for-EMDRT algorithm yields a* $(1+\epsilon)$*-approxim- ation for the EMDRT problem in* $O((nm)^{O(d^2)}(\sqrt{d}/\epsilon)^{d^2}3^{d^3}d^4)$ *time.*

Sketch of the proof From Lemma 9 and 10, we know that for each $\mathcal{T}_{opt}(q_{b(j)})$, there is one grid point close to it. We denote these d grid points as $\{g_1, \ldots, g_d\}$ (see Fig. 8). We construct an **implicit** set of points $B' = \{q'_1, \ldots, q'_m\}$ called *'relayer'*, where $q'_{b(j)} = g_j$ for $1 \leq j \leq d$, and $q'_l = \mathcal{T}_{opt}(q_l)$ for any $l \notin \{b(j) \mid 1 \leq j \leq d\}$, and assign each q'_l a weight β_l. B' is used as a bridge to show the quality of solution from the above algorithm. Particularly, we first prove that $EMD(A, B')$ is close to the optimal objective value, since B' is close to $\mathcal{T}_{opt}(B)$. Then, we show that $\mathcal{I}(B)$ is close to B', where $\mathcal{I}(B)$ is the output from the execution of SOD on B and $\{g_1, \ldots, g_d\}$. Finally, we view B' as the one which relays all flow from $\mathcal{I}(B)$ to A; this implies that $EMD(A, \mathcal{I}(B))$ is also close to the optimal objective value.

Proof (**of Theorem** 3) Firstly, by Theorem 2 and Step 1 of the FPTAS-for-EMDRT algorithm, we know that there must exist $\gamma_0 \in \Gamma$ such that

$$EMD(A, \mathcal{T}_{opt}(B)) \leq \gamma_0 \leq 2EMD(A, \mathcal{T}_{opt}(B)). \tag{18}$$

Since the algorithm enumerates all d-tuples from $[A]^d$, we can focus on the tuple $\{p_{i(1)}, \ldots, p_{i(d)}\}$ which has $||p_{i(j)} - \mathcal{T}_{opt}(q_{b(j)})|| = \min_{1 \leq i \leq n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||$ for each $1 \leq j \leq d$. We know that if we let $r_j = \frac{nW^B}{\beta_{b(j)}}\gamma_0$ for all $1 \leq j \leq d$, then the grid length satisfies the following inequality,

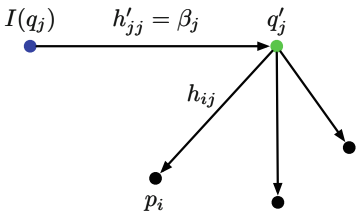

Fig. 9 An example showing the flow from $\mathcal{I}(q_j)$ to p_i passing through the q'_j

$$\frac{r_j \epsilon}{8 \cdot 3^d nm\sqrt{d}} \leq \frac{\epsilon W^B}{4 \cdot 3^d m\sqrt{d}\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B)). \tag{19}$$

From Lemma 9, we know that $\mathcal{T}_{opt}(q_{b(j)})$ locates inside the grid-ball centered at $p_{i(j)}$ and with radius r_j. Thus, there exists one grid point $g_j \in G_j$ such that

$$||g_j - \mathcal{T}_{opt}(q_{b(j)})|| \leq c\epsilon \frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B)), \tag{20}$$

where $c = 1/(4 \cdot 3^d)$ (see Fig. 8). Now, we construct an implicit point set $B' = \{q'_1, \ldots, q'_m\}$ as the *relayer* point set, where $q'_{b(j)} = g_j$ for $1 \leq j \leq d$, and $q'_l = \mathcal{T}_{opt}(q_l)$ for any $l \notin \{b(j) \mid 1 \leq j \leq d\}$, and assign each q'_l a weight β_l. From Lemma 10 and inequality (20), we have

$$EMD(A, B') \leq (1 + 2c\epsilon) EMD(A, \mathcal{T}_{opt}(B)). \tag{21}$$

This means that $EMD(A, B')$ is close to the optimal objective value.

Consider executing the SOD procedure on B and $\{g_1, \ldots, g_d\}$ (i.e., $\{q'_{b(1)}, \ldots, q'_{b(d)}\}$), and let $\mathcal{I}(B)$ be the output image of B. We define the flow $H' = \{h'_{ij}\}$ from $\mathcal{T}_{opt}(B)$ to B' as $h'_{jj} = \beta_j$ for each $1 \leq j \leq m$ and $h'_{ij} = 0$ for all $i \neq j$. Note that from (20), we have

$$\mathcal{BN}(\mathcal{T}_{opt}(B), B', H') \leq \frac{c\epsilon}{m} EMD(A, \mathcal{T}_{opt}(B)). \tag{22}$$

Using a similar idea in the proof of Lemma 8, we have

$$\mathcal{BN}(\mathcal{I}(B), B', H') \leq 2 \cdot 3^{d-1} \mathcal{BN}(\mathcal{T}_{opt}(B), B', H') \leq \frac{\epsilon}{2m} EMD(A, \mathcal{T}_{opt}(B)). \tag{23}$$

The only difference from Lemma 8 is that we replace the factor of $(n+1)(2n+1)^{d-1}$ in Lemma 8 by $2 \cdot 3^{d-1}$ (due to the difference in the flow H').

In the above construction, B' can be viewed as a point set which **relays** all the flow from $\mathcal{I}(B)$ to A. More specifically, any flow from $\mathcal{I}(q_j)$ to p_i can be thought as a flow arriving q'_j first, and then flowing to p_i (see Fig. 9). From (21), (23), and triangle inequality, we have

$$EMD(A, \mathcal{I}(B)) \leq EMD(A, B') + \sum_{j=1}^{m} \mathcal{BN}(\mathcal{I}(B), B', H') \leq (1+\epsilon) EMD(A, \mathcal{T}_{opt}(B)).$$

This means that $\mathcal{I}(B)$ is the desired image of B which yields a $(1+\epsilon)$-approxim- ation for EMDRT. Since the algorithm enumerates all tuples in $[A]^d$ and $G_1 \times G_2 \times \cdots \times G_d$, $\mathcal{I}(B)$ must be one of the output images.

Running time The most time consuming step is enumerating all tuples in $G_1 \times G_2 \times \cdots \times G_d$. Since each $|G_j| = O(d \log(\max\{n, m\})(\frac{8 \cdot 3^d nm\sqrt{d}}{\epsilon})^d)$, the total running time is $O((nm)^{O(d^2)}(\sqrt{d}/\epsilon)^{d^2} 3^{d^3} d^4)$ (including the time for *SOD*). □

6 FPTAS with Improved Running Time

The FPTAS algorithm given in Sect. 5.2 can be further improved in its running time. To achieve this, we notice that the most time-consuming part of the algorithm is for examining all combinations of the grid points inside the d grid-balls. To speed up the computation, it is desirable to reduce the size of the grids. For this purpose, we first observe that the contribution of a point $q_j \in B$ to the optimal objective value is $\sum_{i=1}^{n} h_{ij} ||p_i - \mathcal{T}_{opt}(q_j)|| / \min\{W^A, W^B\}$, where $\sum_{i=1}^{n} h_{ij}$ can be viewed as the weight β_j of q_j (by Definition 2) and $||p_i - \mathcal{T}_{opt}(q_j)||$ is the distance between $p_i \in A$ and the position of q_j in the optimal solution. This means that, to reduce the error caused by the rigid transformation, we need to consider both β_j and $||p_i - \mathcal{T}_{opt}(q_j)||$ when selecting the base. The Base-Selection Algorithm in Sect. 5.1 maximizes the dispersion of the base using only the weight (i.e., type 1 dispersion; see Sect. 3). Thus, a better way for maximizing the dispersion of the base is to consider both factors (i.e., type 2 dispersion). This motivates us to consider the following base selection algorithm.

Algorithm Optimum-Guided-Base (OGB)
Input A weighted point set $B = \{q_1, \ldots, q_m\}$ in $\mathbb{R}^d$ with weight $\beta_j \geq 0$ for q_j.
Output A base which is an ordered subset of points in B.

1. For each $1 \leq j \leq m$, define a value

$$v_j = \max \left\{ \min_{1 \leq i \leq n} ||p_i - \mathcal{T}_{opt}(q_j)||, \frac{W^B}{m\beta_j} EMD(A, \mathcal{T}_{opt}(B)) \right\}.$$

2. Select the point in B with the minimum v_j value, and denote it as $q_{b(1)}$.
3. Set $l = 1$, and repeat the following steps until $l = d$.
   (a) Let $\mathcal{F}_l$ the flat spanning $\{q_{b(1)}, \ldots, q_{b(l)}\}$.
   (b) Find the point in B realizing the value of

$$\max \left\{ \frac{1}{v_j} dist(q_j, \mathcal{F}_l) \mid 1 \leq j \leq m \right\},$$

and denote it as $q_{b(l+1)}$, where $dist(q_j, \mathcal{F}_l)$ is the distance between q_j and $\mathcal{F}_l$.

(c) Let $l = l + 1$.

4. Output $\{q_{b(1)}, \ldots, q_{b(d)}\}$.

Note: The above OGB algorithm differs from the Base-Selection algorithm mainly on the use of v_j value for selecting the base. Since v_j depends on the optimal solution, OGB cannot be directly implemented. To resolve this issue, we can enumerate all d-tuples in $[B]^d$, and find the one corresponding to the output of OGB. Thus, for ease of analysis, we can assume that OGB is available.

With OGB, we can now discuss our improved FPTAS algorithm. Since most part of the improved algorithm is similar to the original FPTAS algorithm given in Sect. 5.2, below we just list the main differences.

1. Replace the Base-Selection algorithm by the OGB algorithm.
2. In the Grid-Construction algorithm, replace the set of radius candidates by $\{\frac{nW^B}{2^t\beta_{b(j)}}\gamma \mid \gamma \in \Gamma, t = 1, 2, \ldots, \log(nm)\}$ for each $1 \le j \le d$, and change the grid length to $\frac{r\epsilon}{8\cdot 3^d\sqrt{d}}$.

Before analyzing the above algorithm, we first consider the following two essential questions.

Why is the grid size reduced? To see why the above algorithm is an improved FPTAS, we first analyze the grid size. Similar to the original FPTAS, we can build a grid-ball centered at $p_{i(j)}$ and with radius

$$v_{b(j)} = \max\left\{\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||, \frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B))\right\}. \tag{24}$$

By Lemma 10, we know that we can choose the grid length to be

$$\begin{aligned} &O\left(\frac{\epsilon}{\sqrt{d}} \max\left\{\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||, \frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B))\right\}\right) \\ &= O\left(\frac{\epsilon}{\sqrt{d}} v_{b(j)}\right), \end{aligned} \tag{25}$$

This means that the ratio between the radius and grid length is $O(\frac{\sqrt{d}}{\epsilon})$, which is independent of m and n, while the ratio in the first FPTAS (in Sect. 5.2) is $O(\frac{\sqrt{d}}{\epsilon}nm)$ (see Step 1(b) in **Algorithm Grid-Construction**). Thus, the grid size in each grid-ball is bounded by a constant depending only on d and ϵ, and consequently the running time can be significantly reduced.

Why take the maximum of the two terms as the radius? In the above improved algorithm, a natural question is "Why do we take the maximum of $\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||$ and $\frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B))$, rather than just one of them, as the radius of the grid-balls?" To understand the rationale behind this, we first consider what happens if only one of them is used as the radius. If $\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)}||$ is the radius, an immediate difficulty is how to estimate the value of $\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||$, since it might be extremely small. Also, if $\frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B))$ is

the radius, the grid-ball may not even contain $\mathcal{T}_{opt}(q_{b(j)})$ and therefore it is difficult to estimate the error related to $q_{b(j)}$. This indicates that using either term as the radius leads to different problem. Surprisingly, if we simply take the maximum of the two terms, $\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||$ and $\frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B))$, as the radius, both difficulties can be rather easily overcome. This is because with $\max\{\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_{b(j)})||, \frac{W^B}{m\beta_{b(j)}} EMD(A, \mathcal{T}_{opt}(B))\}$ as the radius, we can determine its approximate value by guessing $O(\log mn)$ times, using the upper bound (of the optimal objective value) from Sect. 5.1 (note that there is no need to obtain the exact value; some constant approximation will be sufficient for us to reduce the grid size). Also with this radius, we can always guarantee that the grid ball contains $\mathcal{T}_{opt}(q_{b(j)})$ (i.e., ensured by the first term). Thus, it is necessary to use the maximum of the two terms as the radius.

With the above understanding, we can show the following theorem using a similar argument given in the proof of Theorem 3.

Theorem 4 *The improved FPTAS algorithm yields a* $(1+\epsilon)$*-approximation for the EMDRT problem in* $O((nm)^{d+2}(\log(nm))^{2d}(\sqrt{d}/\epsilon)^{d^2}3^{d^3}d^4)$ *time.*

Proof Firstly, from Lemma 9 and the definition of v_j we know that there must exist one r_j in the set of radius candidates such that $v_j \le r_j \le 2v_j$. By a similar argument in the proof of Theorem 3, we know that there exists one grid point $g_j \in G_j$ such that

$$||g_j - \mathcal{T}_{opt}(q_{b(j)})|| \le c\epsilon v_j, \tag{26}$$

where $c = 1/(4 \cdot 3^d)$. Following the same strategy in the proof of Theorem 3, we construct an implicit point set $B' = \{q'_1, \ldots, q'_m\}$, where $q'_{b(j)} = g_j$ for $1 \le j \le d$, and $q'_l = \mathcal{T}_{opt}(q_l)$ for any $l \notin \{b(j) \mid 1 \le j \le d\}$, and assign each q'_l a weight β_l. Consider the execution of SOD on B with $\{g_1, \ldots, g_d\}$ as the reference system. In the l-th step of *SOD*, we let $u^l_j = \frac{||q^l_j - q'_j||}{c\epsilon v_j}$ for each $1 \le j \le m$, where q^l_j is q_j at its new position in the l-th step. When $l = 0$, we know that

$$u^0_j \le 1 \tag{27}$$

by (26). For other values of l, we have the following claim.

Claim 2 *For any* $1 \le j \le m$, $u^1_j \le 2$ *and* $u^l_j \le u^{l-1}_j + 2u^{l-1}_{b(l)}$ *for all* $l \ge 2$.

Proof Firstly, when translating $\mathcal{T}_{opt}(B)$ to make $q^1_{b(1)}$ coincident with $q'_{b(1)}$, we have

$$\begin{aligned} u^1_j &= \frac{||q^1_j - q'_j||}{c\epsilon v_j} \\ &\le \frac{||q^1_j - q^0_j|| + ||q^0_j - q'_j||}{c\epsilon v_j} \\ &= \frac{||q^1_j - q^0_j|| + ||q^0_{b(1)} - q'_{b(1)}||}{c\epsilon v_j} \end{aligned}$$

$$
\begin{aligned}
&= u_j^0 + \frac{v_{b(1)}}{v_j} u_{b(1)}^0 \\
&\le u_j^0 + u_{b(1)}^0 \\
&\le 2, \qquad (28)
\end{aligned}
$$

where the second inequality follows from the fact that $v_{b(1)} = \min_{1 \le j \le m} v_j$, and the last inequality follows from (27).

Secondly, in the l-th step for $l \ge 2$, from OGB we know that $q_{b(l)}$ realizes the value of $\max\{\frac{1}{v_j} dist(q_j, \mathcal{F}_{l-1}) \mid 1 \le j \le m\}$. Thus for each $1 \le j \le m$, we have

$$
\begin{aligned}
\left\| |q_j^l - q_j^{l-1}| \right\| &= \frac{dist(q_j, \mathcal{F}_{l-1})}{dist(q_{b(l)}, \mathcal{F}_{l-1})} \left\| |q_{b(l)}^l - q_{b(l)}^{l-1}| \right\| \\
&\le \frac{v_j}{v_{b(l)}} \left\| |q_{b(l)}^l - q_{b(l)}^{l-1}| \right\|. \qquad (29)
\end{aligned}
$$

Moreover, from Lemma 4, we have

$$
\begin{aligned}
||q_{b(l)}^l - q_{b(l)}^{l-1}|| &\le 2||q_{b(l)}^{l-1} - q'_{b(l)}|| \\
&= 2c\epsilon v_{b(l)} u_{b(l)}^{l-1}. \qquad (30)
\end{aligned}
$$

Combining (29) and (30), we have

$$
\left\| |q_j^l - q_j^{l-1}| \right\| \le 2c\epsilon v_j u_{b(l)}^{l-1}. \qquad (31)
$$

By triangle inequality and (31), we have

$$
\begin{aligned}
u_j^l &= \frac{||q_j^l - q'_j||}{c\epsilon v_j} \\
&\le \frac{||q_j^l - q_j^{l-1}|| + ||q_j^{l-1} - q'_j||}{c\epsilon v_j} \le u_j^{l-1} + 2u_{b(l)}^{l-1}. \qquad (32)
\end{aligned}
$$

Thus Claim 2 is true. □

From Claim 2 and (27), we can know that $u_j^d \le 2 \cdot 3^{d-1}$ for each $1 \le j \le m$ (by simple calculation and mathematical induction). Together with triangle inequality and (26) (note that $g_j = q'_{b(j)}$), we have $||q_j^d - \mathcal{T}_{opt}(q_j)|| \le (1 + 2 \cdot 3^{d-1}) c\epsilon v_j$. Using Lemma 10, we obtain the desired FPTAS solution.

Running time Comparing to the previous FPTAS algorithm, there are two major differences which affect the running time. One is that we have to enumerate all m^d tuples in $[B]^d$ to implement OGB. The other is that $|G_j|$ is reduced to $O(d(\log(nm))^2(\frac{8 \cdot 3^d \sqrt{d}}{\epsilon})^d)$. Thus, the total time is reduced to $O((nm)^{d+2} (\log(nm))^{2d} (\sqrt{d}/\epsilon)^{d^2} 3^{d^3} d^4)$. □

Lower bound on running time Cabello et al. [15] showed that given two point sets A and B in $\mathbb{R}^d$ with size $|A| = m \le |B| = n$, it is not only NP-hard, but also W[1]-hard

(with d as a parameter) to determine whether A is congruent with a subset of B, and cannot be solved in $O(mn^{o(d)})$-time unless $SNP \subset DTIME(2^{o(n)})$. Since the subset congruent problem can be reduced to our EMDRT problem (i.e., let each $\alpha_i = \beta_j = 1$, and $EMD(A, \mathcal{T}_{opt}(B)) = 0$), this means that obtaining an FPTAS for EMDRT costs at least $mn^{\Omega(d)}$-time (this is because the optimal objective value of the congruent problem is 0 and a $(1+\epsilon)$-approximation of the objective value implies an exact solution to the congruent problem). As our algorithm takes $O((nm)^{d+2}(\log(nm))^{2d})$-time (if ignore the factor involving only d and ϵ), this suggests that it almost reaches the limit.

7 Extensions to Some Related Problems

In this section, we show that our techniques for EMDRT can be extended to several related problems and achieve FPTAS for each of them.

7.1 EMD Under Similarity Transformation

In this section, we consider the problem of minimizing EMD of two weighted point sets under Similarity Transformation (EMDST). We start with the definition of similarity transformation.

Definition 5 (*Similarity Transformation*) Similarity transformation in $\mathbb{R}^d$ includes rigid transformation, scaling, or their combination.

Similar to SOD, we have the following decomposition for similarity transformation.

Orthogonal Similarity Transformation Decomposition (OSTD) OSTD consists all steps of SOD and an additional step (after Step 1; i.e., the translation step making $p_{f(1)}$ coincident with r_1) which scales P about r_1 by some factor to be determined later.

As shown in Lemma 2, the outcome of SOD is independent of the initial position of the point set P. Due to the additional scaling step, this nice property seemingly no longer holds for OSTD. However, as it will be shown later, this property is actually still true for OSTD for some carefully chosen scaling factor. Below, we first present an algorithm similar to the Upper-Bound on EMDRT algorithm in Sect. 5.1, and determine the scaling factor in one of its main steps.

Algorithm Upper-Bound on EMDST The main body of the algorithm is almost the same as Upper-Bound on EMDRT (in Sect. 5.1); the only difference is that SOD is replaced by OSTD in Step 2(b). The scaling factor of OSTD is determined as follows. Let $\mathcal{T}_{opt}$ be an optimal similarity transformation, $\mathcal{T}_{opt}(q_{b(2)})^1$ be $q_{b(2)}$ in its new position after B is translated to make $q_{b(1)}$ coincident with $p_{i(1)}$, and p_{i_*} be the point in A closest to $\mathcal{T}_{opt}(q_{b(2)})^1$ (note that p_{i_*} is not necessary in $R = \{p_{i(1)}, \ldots, p_{i(d)}\}$). Since p_{i_*} depends on the unknown optimal transformation $\mathcal{T}_{opt}$, it cannot be found directly; however, it is possible to obtain it implicitly by the same strategy used in the Optimum-Guided-Base algorithm in Sect. 6. With p_{i_*}, we can then choose the scaling factor of OSTD as such a value that

$$||q^s_{b(2)} - q^s_{b(1)}|| = ||p_{i_*} - p_{i(1)}||,$$

where q^s_j is q_j in its new position after scaling about $p_{i(1)}$ for each $1 \le j \le m$ (note that $q^s_{b(1)} = q^1_{b(1)}$ since $q^1_{b(1)}$ coincides with $p_{i(1)}$).

The following lemma is the counterpart of Lemma 2 for OSTD.

Lemma 11 *In the algorithm of Upper-Bound on EMDST, if the reference system* $R = \{p_{i(1)}, \ldots, p_{i(d)}\}$ *and the mapping function* f *are fixed, the output of* $OSTD(B, R, f)$ *is always the same as that of* $OSTD(\mathcal{I}(B), R, f)$ *for any similarity transformation* $\mathcal{I}$.

Proof To prove this lemma, our strategy is to first show that the outputs are congruent with each other [6], and then demonstrate that they are exactly the same.

Claim 3 $OSTD(B, R, f)$ *and* $OSTD(\mathcal{I}(B), R, f)$ *are congruent with each other.*

Proof Since all the steps of $OSTD$, except for the scaling step, are rigid transformations, we only need to show that B and $\mathcal{I}(B)$ are congruent with each other after the scaling step. To prove this, we first introduce the following basic fact.

Fact *Let* $Q = \{q_1, \ldots, q_m\}$ *be a point set, and* $\mathcal{T}(Q)$ *be its image transformed by some similarity transformation* $\mathcal{T}$. Q *is congruent with* $\mathcal{T}(Q)$ *if and only if there exists a pair of points* q_j *and* $q_{j'}$ *in* Q *satisfying the condition* $||q_j - q_{j'}|| = ||\mathcal{T}(q_j) - \mathcal{T}(q_{j'})||$.

With this fact, we now show the correctness of Claim 3. For any chosen $\mathcal{T}_{opt}$, we know that both $\mathcal{T}_{opt}(B)$ and $\mathcal{T}_{opt}(q_{b(2)})^1$ are fixed. Thus, the index i_* is also fixed. This means that the length between $q_{b(2)}$ and $q_{b(1)}$ is always scaled to be the fixed value of $||p_{i_*} - p_{i(1)}||$. By the above Fact, we know that B^s and $\mathcal{I}(B)^s$ are congruent with each other, where B^s and $\mathcal{I}(B)^s$ are the new B and $\mathcal{I}(B)$, respectively, after the scaling. This completes the proof for the claim. □

Since $OSTD(B, R, f)$ and $OSTD(\mathcal{I}(B), R, f)$ are always congruent with each other after the scaling step and the following $d - 1$ steps of OSTD are the same as SOD, using the same idea in the proof of Lemma 2, we can show that the final outputs are exactly the same. □

The lemma below on the running time of OSTD trivially follows from Lemma 5 and the definition of OSTD.

Lemma 12 *OSTD can be performed in* $O(|P|d^4)$ *time.*

We now present the main result for the upper bound algorithm of EMDST.

Theorem 5 *The algorithm of Upper-Bound on EMDST yields in* $O(n^{d+3}(\log n)^2 md^4)$ *time an upper bound* Π *which is a* $2(1+\epsilon)(nm+1)nm(2n+1)^{d-1}$*-approximation of the optimal objective value.*

[6] Two point sets are congruent if one can be transformed to completely coincide with the other by some rigid transformation.

Proof By Lemma 11, we can assume that the initial position of B is at $\mathcal{T}_{opt}(B)$. This indicates that we can use the same idea in the proof of Theorem 2 to analyze the change of the bottleneck between A and B in a step-by-step fashion, and obtain the approximation ratio of $EMD(A, \mathcal{T}_{opt}(B))$. The only place that we need to pay attention is the scaling step. We have the following claim first.

Claim 4 *Let d_j be the distance traveled by q_j, $1 \leq j \leq m$, when performing the scaling step in OSTD. Then $\beta_j d_j \leq \beta_{b(2)} d_{b(2)}$.*

Proof First, we have the following two inequalities.

$$\beta_j ||q_j - q_{b(1)}|| \leq \beta_{b(2)} ||q_{b(2)} - q_{b(1)}||, \tag{33}$$

$$\frac{d_{b(2)}}{d_j} = \frac{||q_{b(2)} - q_{b(1)}||}{||q_j - q_{b(1)}||}, \tag{34}$$

where the first inequality follows from the fact that $b(2) = \arg_j \max\{\beta_j ||q_j - q_{b(1)}|| \mid 1 \leq j \leq m\}$, and the second inequality follows from the property of the scaling. Combining (33) and (34), we have the inequality in Claim 4.

Let B^1 and B^s be the images of B after the first step and the scaling step of OSTD. By Claim 4 and triangle inequality, we have

$$EMD(A, B^s) \leq EMD(A, B^1) + \frac{1}{W^B} m \beta_{b(2)} d_{b(2)}. \tag{35}$$

Also, we have the following inequality for $d_{b(2)}$.

Claim 5 $d_{b(2)} \leq ||q^1_{b(2)} - p_i||$ *for any* $1 \leq i \leq n$.

Proof First, since $||q^s_{b(2)} - q^s_{b(1)}|| = ||p_{i_*} - p_{i(1)}||$ and $q^s_{b(1)} = q^1_{b(1)} = p_{i(1)}$ (see Fig. 10), we know that

$$\begin{aligned} d_{b(2)} &= ||q^s_{b(2)} - q^1_{b(2)}|| \\ &= |\, ||q^s_{b(2)} - q^1_{b(1)}|| - ||q^1_{b(2)} - q^1_{b(1)}||\, | \\ &= |\, ||p_{i_*} - p_{i(1)}|| - ||q^1_{b(2)} - p_{i(1)}||\, |, \end{aligned} \tag{36}$$

where the second equality follows from the fact that the three points $q^1_{b(1)}$, $q^1_{b(2)}$ and $q^s_{b(2)}$ are collinear (see Fig. 10). By triangle inequality, we have the following inequality.

$$||q^1_{b(2)} - p_{i_*}|| \geq |\, ||p_{i_*} - p_{i(1)}|| - ||q^1_{b(2)} - p_{i(1)}||\, | \,. \tag{37}$$

Plugging (37) into (36), we have $d_{b(2)} \leq ||q^1_{b(2)} - p_{i_*}||$. Meanwhile, since p_{i_*} is the closest point to $q^1_{b(2)}$ in A, we know that $d_{b(2)} \leq ||q^1_{b(2)} - p_i||$ for any $1 \leq i \leq n$. □

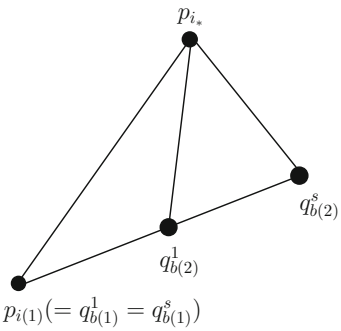

Fig. 10 An illustration for Claim 5

From Claim 5, Definition 4, and Lemma 7, we know that

$$\begin{aligned}\beta_{b(2)}d_{b(2)} &= \sum_{i=1}^{n} h_{ib(2)}d_{b(2)} \\ &\leq \sum_{i=1}^{n} h_{ib(2)}||q^1_{b(2)} - p_i|| \\ &\leq nW^B\mathcal{BN}(A, B^1, H) \\ &\leq nW^B EMD(A, B^1), \end{aligned} \tag{38}$$

where H is the flow realizing $EMD(A, B^1)$. Combining (35) and (38), we get

$$EMD(A, B^s) \leq (1 + mn)EMD(A, B^1). \tag{39}$$

It means that due to the scaling step, we have an extra factor $(1 + mn)$ in the approximation ratio, comparing to that in Theorem 2.

As for the running time, comparing to Theorem 2, there is an additional factor of n which is for guessing p_{i_*} (for determining the scaling factor). This completes the proof. □

Using the above upper bound and techniques similar to the ones in Sects. 5 and 6, we have an FPTAS algorithm for EMDST. The following theorem summarizes the result for EMDST. Similar to Theorem 5, there is an additional factor of n which is for guessing p_{i_*}.

Theorem 6 *There exists a* $(1+\epsilon)$*-approximation algorithm for the EMDST problem which runs in* $O(n^{d+3}m^{d+2}(\log(nm))^{2(d+1)}(\sqrt{d}/\epsilon)^{(d+1)^2}3^{(d+1)^3}d^4)$ *time.*

7.2 Alignment problem

In this section, we show that the *Alignment* problem is a natural extension of EMDRT, and can be solved in a similar fashion.

Definition 6 (*One-to-One Matching*) Let $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ be two sets of points in $\mathbb{R}^d$ with $n \leq m$. A one-to-one matching between A and B is an injective mapping f from $\{1, \ldots, n\}$ to $\{1, \ldots, m\}$ which matches p_i to $q_{f(i)}$ for $1 \leq i \leq n$. The cost of the matching can be either in l_∞ or in l_1 sense, and is defined as $M_\infty(A, B) = \max\{||p_i - q_{f(i)}|| \mid 1 \leq i \leq n\}$ and $M_1(A, B) = \sum_{i=1}^{n} ||p_i - q_{f(i)}||$ respectively.

Definition 7 (*(Directed) Hausdorff Distance*) Let $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ be two sets of points in $\mathbb{R}^d$ with $n \leq m$. The (directed) Hausdorff Distance from A to B is defined as $HD(A, B) = \max_{1 \leq i \leq n} \min\{||p_i - q_j|| \mid 1 \leq j \leq m\}$.

Note We only consider the directed Hausdorff distance from A to B, as the undirected Hausdorff distance can be obtained by computing the directed Hausdorff distance twice, one from A to B and the other from B to A.

Definition 8 (*Alignment*) Let $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ be two sets of points in $\mathbb{R}^d$. The alignment of A and B is to find a rigid transformation $\mathcal{T}$ so as to minimize the cost between A and $\mathcal{T}(B)$, where the cost metric is M_∞, M_1 or HD.

From the above definitions, we immediately have the following lemma.

Lemma 13 *Let $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ be two sets of weighted points in $\mathbb{R}^d$ with $n \leq m$. If all point in A and B have unit weight, then minimizing the EMD between A and B under rigid transformation is equivalent to determining the alignment of A and B using cost metric M_1.*

This means that we can use our FPTAS algorithm to directly solve the alignment problem with cost metric M_1.

Theorem 7 *There exists a $(1+\epsilon)$-approximation algorithm for the alignment problem with cost metrics M_1 which runs in $O((nm)^{d+2}(\log(nm))^{2d}(\sqrt{d}/\epsilon)^{d^2}3^{d^3}d^4)$ time.*

The following lemma shows the connection between the alignment problem and the Bottleneck problem and directly follows from the definitions.

Lemma 14 *Let $A = \{p_1, \ldots, p_n\}$ and $B = \{q_1, \ldots, q_m\}$ be two sets of weighted points in $\mathbb{R}^d$ with $n \leq m$.*

1. *If all points in A and B have unit weight, then minimizing the Bottleneck $\mathcal{BN}$ between A and B under rigid transformation is equivalent to determining the alignment of A and B using cost metric M_∞;*
2. *if each p_i has unit weight and each q_j has infinity weight, then minimizing the Bottleneck $\mathcal{BN}$ between A and B under rigid transformation is equivalent to determining the alignment of A and B using cost metric HD.*

Note In the above lemma, A or B having infinity weight will not cause issue in our FPTAS algorithm. This is because, by Definition 2, we know that the cost function (1) only allows $\min\{W_A, W_B\}$ amount of flow transporting from A to B.

Although we do not present an FPTAS algorithm for the Bottleneck problem $\mathcal{BN}$ in Sect. 5, we can use techniques similar to those in the FPTAS algorithm for EMDRT to achieve an FPTAS for $\mathcal{BN}$.

Minimizing the Bottleneck First of all, from Lemma 8, we know that the algorithm given in Sect. 5.1 yields a $(n+1)(2n+1)^{d-1}$-approximation for the Bottleneck problem, which can be used to build grids and achieve an FPTAS for the Bottleneck problem in a way similar to the algorithm in Sect. 5.2. Note that to build the grids, we still need to determine the radius and grid length of the grid-balls. For the radius, it is easy to achieve a result similar to Lemma 9 (i.e., $\min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_j)|| \le \frac{nW^B}{\beta_j}\mathcal{BN}_{opt}$, where $\mathcal{BN}_{opt}$ is the optimal value, and $\mathcal{T}_{opt}$ is the corresponding optimal rigid transformation). For grid length, we also have a result similar to Lemma 10 (i.e., $||\mathcal{T}(q_j) - \mathcal{T}_{opt}(q_j)||$ is no more than $c\epsilon \min_{1\le i\le n} ||p_i - \mathcal{T}_{opt}(q_j)||$ or $c\epsilon\frac{W^B}{\beta_j}\mathcal{BN}_{opt}$). Thus, it is easy to see that we can use the same algorithm in Sect. 5.2 (with slight changes on the radius and density of the grids) to achieve an FPTAS for the Bottleneck problem. This means that we have FPTAS for the alignment problem using either M_∞ or HD as the cost metric.

Theorem 8 *There exists a $(1+\epsilon)$-approximation algorithm for the alignment problem using either M_∞ or HD as the cost metrics which runs in $O((nm)^{d+2}(\log(nm))^{2d}(\sqrt{d}/\epsilon)^{d^2}3^{d^3}d^4)$ time.*

References

1. Alt, H., Behrends, B., Blomer, J.: Approximate matching of polygonal shapes (Extended Abstract). In: Proceedings of the 7th ACM Symposium on Computational Geometry (SoCG'91), pp. 186–193 (1991)
2. Alt, H., Guibas, L.: Discrete geometric shapes: matching, interpolation, and approximation. In: Sack, J.-R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 121–153. Elsevier, Amsterdam (1999)
3. Alt, H., Mehlhorn, K., Wagener, H., Welzl, E.: Congruence, similarity, and symmetries of geometric objects. Discrete Comput. Geom. **3**, 237–256 (1988)
4. Andoni, A., Indyk, P., Krauthgamer, R.: Earth mover's distance over high-dimensional spaces. In: Proccedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA'08), pp. 343–352 (2008)
5. Andoni, A., Do Ba, K., Indyk, P., Woodruff, D.P.: Efficient sketches for earth mover's distance, with applications. In: Proccedings 50th IEEE Symposium on Foundations of Computer Science (FOCS'09), pp. 324–330 (2009)
6. Andoni, A., Onak, K., Nikolov, A., Yaroslavtsev, G.: Parallel Algorithms for Geometric Graph Problems. In: Proccedings of the 46th Symposium on Theory of Computing Conference (STOC'14), pp. 574–583 (2014)
7. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intell **9**(5), 698–700 (1987)
8. Arkin, E.M., Kedem, K., Mitchell, J.S.B., Sprinzak, J., Werman, M.: Matching points into pairwise-disjoint noise regions: combinatorial bounds and algorithms. INFORMS J. Comput. **4**(4), 375–386 (1992)
9. Agarwal, P.K., Phillips, J.M.: On bipartite matching under the RMS distance. In: Proccedings of the 18th Canadian Conference on Computational Geometry (CCCG'06) (2006)
10. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)

11. Benkert, M., Gudmundsson, J., Merrick, D., Wolle, T.: Approximate one-to-one point pattern matching. J. Discrete Algorithms **15**, 1–15 (2012)
12. Cohen, S.: Finding color and shape patterns in images. PhD thesis, Stanford University, Department of Compute Science (1999)
13. Chew, L.P., Dor, D., Efrat, A., Kedem, K.: Geometric pattern matching in d-dimensional space. In: Proccedings of the 3rd European Symposium on Algorithms (ESA'95), pp. 264–279 (1995)
14. Chew, L.P., Goodrich, M.T., Huttenlocher, D.P., Kedem, K., Kleinberg, J.M., Kravets, D.: Geometric pattern matching under euclidean motion. Comput. Geom. **7**, 113–124 (1997)
15. Cabello, S., Giannopoulos, P., Knauer, C.: On the parameterized complexity of d-dimensional point set pattern matching. Inf. Process. Lett. **105**(2), 73–77 (2008)
16. Cabello, S., Giannopoulos, P., Knauer, C., Rote, G.: Matching point sets with respect to the earth mover's distance. Comput. Geom.: Theory Appl. **39**(2), 118–133 (2008)
17. Cardoze, D.E., Schulman, L.J.: Pattern matching for spatial point sets. In: Proccedings of the 39th IEEE Symposium on Foundations of Computer Science (FOCS'98), pp. 156–165 (1998)
18. Clark, C., Kalita, J.: A comparison of algorithms for the pairwise alignment of biological networks. Bioinformatics **30**(16), 2351–2359 (2014)
19. Efrat, A., Itai, A.: Improvements on bottleneck matching and related problems using geometry. In: Proccedings of the 12th ACM Symposium on Computational Geometry (SoCG'96), pp. 301–310 (1996)
20. Ezra, E., Sharir, M., Efrat, A.: On the performance of the ICP algorithm. Comput. Geom. **41**(1–2), 77–93 (2008)
21. Graumann, K., Darell, T.: Fast contour matching using approximate earth mover's distance. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04), pp. 220–227 (2004)
22. Gavrilov, M., Indyk, P., Motwani, R., Venkatasubramanian, S.: Combinatorial and experimental methods for approximate point pattern matching. Algorithmica **38**(1), 59–90 (2004)
23. Goodrich, M.T., Mitchell, J.S.B., Orletsky, M.W.: Approximate geometric pattern matching under rigid motions. IEEE Trans. Pattern Anal. Mach. Intell. **21**(4), 371–379 (1999)
24. Giannopoulos, P., Veltkamp, R.: A pseudo-metric for weighted point sets. In: Proccedings of 7th European Conference Computer Vision (ECCV'02), pp. 715–731 (2002)
25. Huttenlocher, D.P., Kedem, K., Kleinberg, J.M.: On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane. In: Proccedings of the 8th ACM Symposium on Computational Geometry (SoCG'92), pp. 110–119 (1992)
26. Indyk, P.: A near linear time constant factor approximation for Euclidean bichromatic matching (cost). In: Proccedings of the 8th ACM-SIAM Symposium on Discrete Algorithms (SODA'07), pp. 39–42 (2007)
27. Klein, O., Veltkamp, R.C.: Approximation algorithms for computing the earth mover's distance under transformations. In: Proccedings of the 16th International Symposium on Algorithms and Computation (ISAAC'05), pp. 1019–1028 (2005)
28. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis. **40**(2), 99–121 (2000)
29. Sharathkumar, R., Agarwal, P. K.: Algorithms for the transportation problem in geometric settings. In: Proccedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA '12), pp. 306–317 (2012)
30. Typke, R., Giannopoulos, P., Veltkamp, R.C., Wierking, F., Oostrum, R.: Using transportation distances for measuring melodic similarity. In: Proccedings of the 4th International Conference Music Information Retrieval, pp. 107–114 (2003)