Matching Point Sets with respect to the Earth Mover's Distance*

Sergio Cabello† Panos Giannopoulos‡ Christian Knauer§ Günter Rote§

Abstract

The Earth Mover's Distance (EMD) between two weighted point sets (point distributions) is a distance measure commonly used in computer vision for color-based image retrieval and shape matching. It measures the minimum amount of work needed to transform one set into the other one by weight transportation.

We study the following shape matching problem: Given two weighted point sets A and B in the plane, compute a rigid motion of A that minimizes its Earth Mover's Distance to B. No algorithm is known that computes an exact solution to this problem. We present simple FPTASs and polynomial-time $(2+\epsilon)$-approximation algorithms for the minimum Euclidean EMD between A and B under translations and rigid motions.

Keywords: Geometric Optimization, Approximation Algorithms, Shape Matching, Earth Mover's Distance, Weighted Point Sets, Rigid Motions.

1 Introduction

Shape matching is a fundamental problem in computer vision: given two shapes A and B, one wants to determine how closely A resembles B, according to some distance measure between the shapes. In order to measure the similarity of A and B independently of transformations such as translations and/or rotations, one wants to find a transformed version of, say, A that attains the minimum possible distance to B. The problem has received a lot of attention, both in the computer-vision and computational-geometry community; see the surveys by Hagedoorn and Veltkamp [14] and Alt and Guibas [3].

In a typical application such as content-based image retrieval [21], a shape, or pattern in general, is given by a set of feature (curvature, color, etc.) *weighted* points in some metric space, e.g., Euclidean space or CIE-Lab color space [20]. The weight of a point normally represents its significance, that is, the larger the weight, the more important the point for the whole pattern.

*A preliminary version of this paper appeared in the 13th European Symposium on Algorithms, volume 3669 of Lecture Notes in Computer Science, pages 520–531, 2005.

†IMFM, Department of Mathematics, Jadranska 19, SI-1000 Ljubljana, Slovenia, sergio.cabello@imfm.uni-lj.si; partially supported by the European Community Sixth Framework Programme under a Marie Curie Intra-European Fellowship.

‡Corresponding author. Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, D-10099 Berlin, Germany, panos@informatik.hu-berlin.de; research conducted at Institut für Informatik, FU Berlin, Germany, and IICS, Utrecht University, The Netherlands. Partially supported by the Dutch Technology Foundation (STW), project UIF 5055.

§Freie Universität Berlin, Institut für Informatik, Takustraße 9, D-14195 Berlin, Germany, {knauer, rote}@inf.fu-berlin.de.

The Earth Mover's Distance (EMD) is a similarity measure for weighted point sets. It is the discrete version of the well-known *Monge-Kantorovich* mass transportation distance whose potential use for measuring shape similarity was first proposed in an influential paper by Mumford [17]. Since then, the EMD has turned into a popular similarity measure in computer vision with applications in color-based image retrieval [9, 16, 19, 20], shape matching [9, 12, 13] and music score matching [22].

Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two planar weighted point sets with $m \leq n$. A weighted point $a_i \in A$ is defined as $a_i = \{(x_{a_i}, y_{a_i}), w_i\}$ for every $i \in \{1, \ldots, m\}$, where $(x_{a_i}, y_{a_i}) \in \mathbb{R}^2$ and $w_i \in \mathbb{R}^+ \cup \{0\}$ is its weight. A weighted point $b_j \in B$ is defined similarly as $b_j = \{(x_{b_j}, y_{b_j}), u_j\}$ for every $j \in \{1, \ldots, n\}$. Let $W = \sum_{i=1}^{m} w_i$ and $U = \sum_{j=1}^{n} u_j$ be the total weight, or simply weight, of A and B respectively.

Informally, a weighted point a_i can be seen as an amount (supply) of earth or mass, equal to w_i units, positioned at (x_{a_i}, y_{a_i}). Alternatively it can be taken as an empty hole (demand) of w_i units of earth capacity. We assign arbitrarily the role of the supplier to A and that of the receiver/demander to B, setting, in this way, a direction of earth transportation. The Earth Mover's Distance of A to B measures the minimum amount of work needed to fill the holes with earth. A formal definition of the EMD will be given shortly.

We study the following problem: Given two weighted point sets A and B find a rigid motion (isometry) of A that minimizes its Earth Mover's Distance (EMD) to B. Note that we are interested in transformations that change only the position of the points, not their weights. We only consider rigid motions that preserve the orientation (translations and rotations); if reflections are to be allowed, we can solve the problem a second time, for a reflected copy of B. We consider B to be fixed, while A can be translated and/or rotated relative to B. We assume some initial positions for both sets, denoted simply by A and B. We denote by $\mathcal{I}$ the set of all possible rigid motions in the plane, by R_θ a rotation about the origin by angle $\theta \in [0, 2\pi)$, and by $T_{\vec{t}}$ a translation by $\vec{t} \in \mathbb{R}^2$. Any rigid motion $I \in \mathcal{I}$ can be uniquely defined as a translation followed by a rotation, that is, $I = I_{\vec{t},\theta} = R_\theta \circ T_{\vec{t}}$, for some $\theta \in [0, 2\pi)$ and $\vec{t} \in \mathbb{R}^2$. In general, transformed versions of A are denoted by $A(\vec{t}, \theta) = \{a_1(\vec{t}, \theta), \ldots, a_m(\vec{t}, \theta)\}$ for some $I_{\vec{t},\theta} \in \mathcal{I}$. For simplicity, translated only versions of A are denoted by $A(\vec{t}) = \{a_1(\vec{t}), \ldots, a_m(\vec{t})\}$. Similarly, rotated only versions of A are denoted by $A(\theta) = \{a_1(\theta), \ldots, a_m(\theta)\}$.

The EMD between $A(\vec{t}, \theta)$ and B, is a function $\mathrm{EMD} : \mathcal{I} \to \mathbb{R}^+ \cup \{0\}$ defined as

$$\mathrm{EMD}(\vec{t}, \theta) = \min_{F \in \mathcal{F}(A,B)} \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}(\vec{t}, \theta)}{\min\{W, U\}},$$

where $d_{ij}(\vec{t}, \theta)$ is the distance of $a_i(\vec{t}, \theta)$ to b_j, and $F = \{f_{ij}\} \in \mathcal{F}(A, B)$ with $\mathcal{F}(A, B)$ being the set of all feasible flows between A and B defined by the constraints: (i)$f_{ij} \geq 0$ for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$, (ii)$\sum_{j=1}^{n} f_{ij} \leq w_i$ for every $i \in \{1, \ldots, m\}$, (iii)$\sum_{i=1}^{m} f_{ij} \leq u_j$ for every $j \in \{1, \ldots, n\}$, and (iv)$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\{W, U\}$. In case that $\vec{t}$ or θ or both are constant, we simply write $\mathrm{EMD}(\theta)$, $\mathrm{EMD}(\vec{t})$ and EMD respectively. We deal with the Euclidean EMD where d_{ij} is given by the L_2-norm. Our problem can be now stated as follows:

Given two weighted point sets A, B in the plane, compute a rigid motion $I_{\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}}}$ that minimizes $\mathrm{EMD}(\vec{t}, \theta)$.

The problem was first studied by Cohen and Guibas [10] who presented a Flow–Transformation iteration which alternates between finding the optimum flow for a given transformation, and the optimum transformation for a given flow. They showed that this iterative procedure converges, but not necessarily to the global optimum. Computing the EMD for a given transformation is actually the transportation problem, a special minimum cost network flow problem [1] for the solution of which there is a variety of polynomial time algorithms. However, as we discuss later on, the task of finding the optimal transformation for a given flow is not trivial. Cohen and Guibas gave also simple algorithms that compute the optimum translation for the special case where $W = U$ and d_{ij} is the squared Euclidean distance. This case is quite restrictive since, in general, the sets need not have the same weight, and the use of squared Euclidean distance is statistically less robust than Euclidean distance [6].

Observe that the objective function $\mathrm{EMD}(\vec{t}, \theta)$ is not linear in $\vec{t}$ and θ but it is still linear in the flow F. Thus, the minimum EMD occurs at some vertex of the convex polytope $\mathcal{F}(A, B)$. This suggests the following straightforward algorithm: for every vertex $F = \{f_{ij}\}$ of $\mathcal{F}(A, B)$ compute the optimal rigid motion, i.e., the one that minimizes $\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}(\vec{t}, \theta)$. For translations, the latter problem reduces to the *Fermat-Weber* [8, 11] problem, where one wants to find a point that minimizes the sum of weighted distances to a set of given points. No exact solution to this problem is known even in the real RAM model of computation [6]. However, Bose et al. [6] gave a $O(n \log n)$-time $(1+\epsilon)$-approximation algorithm for any fixed dimension. Using their algorithm for every vertex of $\mathcal{F}(A, B)$ gives only a $(1 + \epsilon)$-approximation of the minimum EMD under translations in exponential time.

The EMD is a metric when d_{ij} is a metric and $W = U$ [20]. When $W \neq U$ the EMD inherently performs partial matching since a portion of the weight of the 'heavier' set remains unmatched. The case where $w_i = u_j = 1$ for every $i \in \{1, \dots, m\}$, $j \in \{1, \dots, n\}$ deserves special attention: the integer solutions property of the minimum cost flow problem and the fact that $0 \leq f_{ij} \leq 1$ imply that there is a minimum cost flow from A to B that results in a *partial assignment* between A and B, that is, a perfect matching between A and a subset of B. When $n = m$, the problem is simply referred to as the *assignment* problem.

Results. In this paper, we give simple polynomial-time $(1 + \epsilon)$ and $(2 + \epsilon)$-approximation algorithms for the minimum EMD of two weighted point sets in the plane under translations and rigid motions. The algorithms for translations are given in Section 4 and for rigid motions in Section 6. In the general case where the sets have unequal total weights we compute a $(1+\epsilon)$-approximation in $O((n^3 m/\epsilon^4) \log^2 n)$ time for translations and a $(2+\epsilon)$-approximation in $O((n^4 m^2/\epsilon^4) \log^2 n)$ time for rigid motions. When the sets have equal total weights, the respective running times decrease to $O((n^2/\epsilon^4) \log^2 n)$ and $O((n^3 m/\epsilon^4) \log^2 n)$.

We also show how to compute a $(1 + \epsilon)$-approximation of the minimum cost assignment under translations and rigid motions in $O((n^{3/2}/\epsilon^{7/2}) \log^5 n)$ and $O((n^{7/2}/\epsilon^{9/2}) \log^6 n)$ time respectively. Finally, we give probabilistic $(1 + \epsilon)$-approximations of the minimum cost partial assignment under translations in $O((n^3/\epsilon^4) \log^3 n)$ time and under rigid motions in $O((n^4 m/\epsilon^5) \log^4 n)$ time; both algorithms succeed with high probability.

In Section 3, we give two simple lower bounds on the EMD that are vital to our approximation algorithms. These algorithms need to compute the EMD for a given transformation. Computing the EMD exactly is expensive, and unnecessary since we opt for approximations for our original problem. We begin by showing how to get a $(1 + \epsilon)$-approximation of the EMD in almost quadratic time.

2 Approximating the EMD

Currently, the fastest strongly polynomial-time algorithm for the minimum cost flow problem on a graph $G(V,E)$ is due to Orlin [18], and runs in $O((|E|\log|V|)(|E|+|V|\log|V|))$ time. Several faster but weakly polynomial-time algorithms exist that assume integer edge costs [1] (some even assume integer weights as well). For the transportation problem in the plane, this assumption is very restrictive since the edge costs are given by Euclidean distances. For the latter problem, Atkinson and Vaidya [5] presented a weakly polynomial-time algorithm that assumes integer weights and runs in $O(|V|^{2.5}\log(|V|)\log W)$ time, where W is the largest weight. Since $|V| = m+n$ and $|E| = mn$, Orlin's algorithm runs in $O(m^2n^2\log n + mn^2\log^2 n)$ time while the algorithm by Atkinson and Vaidya runs in $O(n^{2.5}\log n \log W)$ time.

Consider the complete bipartite graph $G(V,E)$ with $V = A\cup B$ and $E = \{(a_i,b_j) : a_i \in A, b_j \in B\}$. Our main idea is to replace $G(V,E)$ by a sparse $(1+\epsilon)$-spanner $G_s(V,E_s)$, i.e., a graph G_s such that the shortest path between any two points in G_s is at most $(1+\epsilon)$ times the Euclidean distance of the points. Note that G_s is not necessarily bipartite. As we will see below, running the algorithm of Orlin on G_s produces an approximate value EMD_s such that $\text{EMD} \le \text{EMD}_s \le (1+\epsilon)\text{EMD}$. For convenience, this simple procedure is referred to as ApxEMD(A,B,ϵ) and given in Figure 1.

ApxEMD(A,B,ϵ):

1. Construct a $(1+\epsilon)$-spanner $G_s(V,E_s)$, $V = A\cup B$, such that $|E_s| = O(n/\epsilon)$.
2. Find a minimum cost flow on G_s using the algorithm by Orlin [18], and report the cost.

Figure 1: Algorithm ApxEMD(A,B,ϵ).

Theorem 1 *Let $A = \{a_1,\ldots,a_m\}$ and $B = \{b_1,\ldots,b_n\}$ be two weighted point sets in the plane with $m \le n$. For any given $\epsilon > 0$, ApxEMD(A,B,ϵ) computes a value EMD_s such that $\text{EMD} \le \text{EMD}_s \le (1+\epsilon)\text{EMD}$ in $O((n^2/\epsilon^2)\log^2 n)$ time.*

Proof: We use Θ-graphs for constructing the spanner $G_s(V,E_s)$ [4]. For any positive angle $\theta \le \pi/4$, the graph $\Theta(V,\theta)$ is a $\left(\frac{1}{\cos\theta-\sin\theta}\right)$-spanner with $O((n+m)/\theta) = O(n/\theta)$ edges that can be constructed in $O(((n+m)/\theta)\log(n+m)) = O((n/\theta)\log n)$ time. Since we want $\frac{1}{\cos\theta-\sin\theta} \le 1+\epsilon$, it suffices to take $\theta = O(\epsilon)$, thus, we can construct the desired $(1+\epsilon)$-spanner $G_s(V,E_s)$ with $O(n/\epsilon)$ edges in $O((n/\epsilon)\log n)$ time.

We proceed by converting G_s into a directed graph as follows: each edge $(a_i,b_j)\in E_s$ is substituted by two directed edges (a_i,b_j) and (b_j,a_i) both with cost d_{ij}. For any pair of vertices a_i, b_j, any shortest path from a_i to b_j in G_s is now directed. Let $\delta(a_i,b_j)$ be such a path and $d(a_i,b_j)$ be its length. Since G_s is not necessarily bipartite, $\delta(a_i,b_j)$ might contain one or more other vertices of A and/or B.

Let $\{f_{ij}\}$ be a minimum cost flow on G. In G_s, we choose to send an amount of f_{ij} from a_i to b_j over $\delta(a_i,b_j)$; see Figure 2 for an illustration. Consider a vertex $v\in V$ that is an intermediate node in $\delta(a_i,b_j)$. Then, f_{ij} enters and leaves v without affecting its total surplus or deficit, that is, the incoming flow minus the outcoming flow. Since $\{f_{ij}\}$ is a feasible flow on G, the flow induced by the above assignment is a feasible flow on G_s. Since

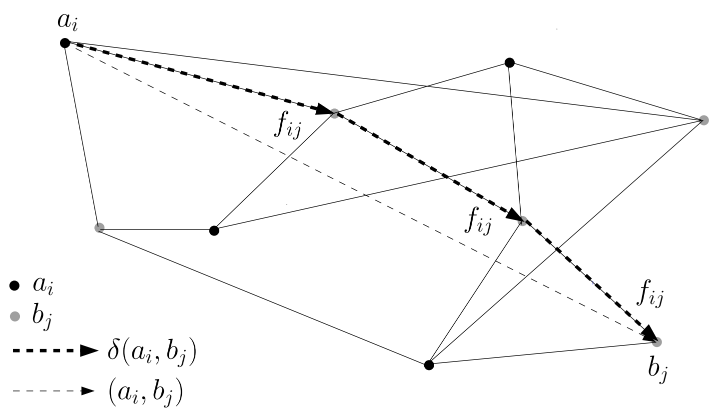

Figure 2: Two point sets $A = \{a_i\}, B = \{b_j\}$, a spanner G_s on $A \cup B$, and a flow f_{ij} sent over $\delta(a_i, b_j)$ in G_s.

$d(a_i, b_j) \leq (1+\epsilon)d_{ij}$ we have

$$\mathrm{EMD}_s \leq \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d(a_i, b_j)}{\min\{W, U\}} \leq \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}(1+\epsilon)d_{ij}}{\min\{W, U\}} = (1+\epsilon)\mathrm{EMD}.$$

Moreover, any minimum cost flow on G_s can be decomposed into flows along paths from supply vertices to demand vertices in G_s and thereby defines some feasible flow on G. Hence, since $d_{ij} \leq d(a_i, b_j)$, we have that $\mathrm{EMD} \leq \mathrm{EMD}_s$.

Regarding the running time, observe that constructing G_s takes $O((n/\epsilon)\log n)$ time. Since $|E_s| = O(n/\epsilon)$, computing a minimum cost flow on G_s takes $O(((n/\epsilon)\log n)(n/\epsilon + n\log n))$ time. In total the algorithm takes $O((n/\epsilon)\log n) + O(((n/\epsilon)\log n)(n/\epsilon + n\log n)) = O((n^2/\epsilon^2)\log^2 n)$ time. $\square$

For the assignment or else minimum cost Euclidean bipartite matching problem in the plane, Varadarajan and Agarwal [23] presented an algorithm that finds a matching with cost at most $(1+\epsilon)$ times the cost of an optimal matching in $O((n/\epsilon)^{3/2}\log^5 n)$ time. We refer to this algorithm as ApxMATCH(A, B, ϵ).

Theorem 2 *[23, Theorem 3.1] Let A and B be two sets of points in the plane with $|A| = |B| = n$. For any given $\epsilon > 0$, a perfect matching between A and B with cost at most $(1+\epsilon)$ times that of an optimal perfect matching can be computed in $O((n/\epsilon)^{3/2}\log^5 n)$ time.*

3 Lower bounds on the EMD

We give two lower bounds on the EMD, that depend on the distance between two points that belong to — or can be easily computed from — A and B. As we will see in the following sections, these lower bounds direct our search for the optimal transformation.

The following simple lower bound comes directly from the definition of the EMD.

Observation 1 *Given two weighted point sets A and B,* $\mathrm{EMD} \geq \min_{i,j} d_{ij}$.

Proof: Let $\{f_{ij}\}$ be an optimal flow between A and B. We have

$$\mathrm{EMD} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}}{\min\{W, U\}} \geq \frac{\min_{i,j} d_{ij} \cdot \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}{\min\{W, U\}} = \min_{i,j} d_{ij},$$

since $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\{W, U\}$. $\square$

The next lower bound is due to Rubner et al. [20], and applies to sets with equal weights. It is based on the notion of the *center of mass* of a weighted point set.

The center of mass $C(A)$ of a planar weighted point set $A = \{(x_{a_i}, y_{a_i}), w_i\}$ for every $i \in \{1, \ldots, m\}$, is defined as

$$C(A) = \frac{\sum_{i=1}^{m} w_i \cdot (x_{a_i}, y_{a_i})}{\sum_{i=1}^{m} w_i}.$$

Theorem 3 *[20] Let A and B be two weighted point sets with equal weights. Then* EMD $\geq d(C(A), C(B))$.

4 Approximation algorithms for translations

Let $\vec{t}_{\text{opt}}$ be an optimal translation. We denote by $\vec{t}_{i\to j}$ the translation which matches a_i and b_j; we call such a translation a *point-to-point* translation. We first prove that there is a point-to-point translation, actually the one that is closest to $\vec{t}_{\text{opt}}$, which gives a 2-approximation of $\text{EMD}(\vec{t}_{\text{opt}})$.

Lemma 1 *Given two weighted point sets A and B,*

$$\text{EMD}(\vec{t}_{\text{opt}}) \leq \min_{i,j} \text{EMD}(\vec{t}_{i\to j}) \leq 2\text{EMD}(\vec{t}_{\text{opt}}).$$

Proof: Clearly, $\text{EMD}(\vec{t}_{\text{opt}}) \leq \min_{i,j} \text{EMD}(\vec{t}_{i\to j})$. Next, consider the optimal position $A(\vec{t}_{\text{opt}})$ of A and an optimal flow $\{f_{ij}\}$ between $A(\vec{t}_{\text{opt}})$ and B. Consider also the distance $d_{ij}(\vec{t}_{\text{opt}})$ for every pair of points $a_i(\vec{t}_{\text{opt}})$, b_j and let $d_{i_0 j_0}(\vec{t}_{\text{opt}})$ be the smallest of all these distances. Assume that we translate $A(\vec{t}_{\text{opt}})$ to the position $A(\vec{t}_{i_0\to j_0})$. Then $d_{i_0 j_0}(\vec{t}_{i_0\to j_0}) = 0$ and, since $d_{i_0 j_0}(\vec{t}_{\text{opt}}) \leq d_{ij}(\vec{t}_{\text{opt}})$, we have that

$$d_{ij}(\vec{t}_{i_0\to j_0}) \leq d_{ij}(\vec{t}_{\text{opt}}) + d_{i_0 j_0}(\vec{t}_{\text{opt}}) \leq 2d_{ij}(\vec{t}_{\text{opt}}),$$

for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$. Hence, we have

$$\begin{aligned}
\min_{i,j} \text{EMD}(\vec{t}_{i\to j}) &\leq \text{EMD}(\vec{t}_{i_0\to j_0}) \\
&\leq \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}(\vec{t}_{i_0\to j_0})}{\min\{W, U\}} \\
&\leq \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} 2d_{ij}(\vec{t}_{\text{opt}})}{\min\{W, U\}} \\
&= 2\text{EMD}(\vec{t}_{\text{opt}}).
\end{aligned}$$

$\square$

Note that observation 1 holds for any translation of A, thus it holds for $\vec{t}_{\text{opt}}$ as well, i.e., $\text{EMD}(\vec{t}_{\text{opt}}) \geq \min_{i,j} d_{ij}(\vec{t}_{\text{opt}})$. This implies that the point-to-point translation which is closest to $\vec{t}_{\text{opt}}$ can be at most $\text{EMD}(\vec{t}_{\text{opt}})$ away from $\vec{t}_{\text{opt}}$. This bound is crucial for the $(1+\epsilon)$-approximation algorithm given in Figure 3. Using a uniform square grid of suitable size we compute the EMD for a limited number of grid translations within a small neighborhood – of size $\text{EMD}(\vec{t}_{\text{opt}})$ – of every point-to-point translation. Note that we do not know $\text{EMD}(\vec{t}_{\text{opt}})$ but we can compute $\min_{i,j} \text{EMD}(\vec{t}_{i\to j})$ which, according to Lemma 1, approximates $\text{EMD}(\vec{t}_{\text{opt}})$ well enough. In order to save time, rather than computing EMD exactly, we will approximate it using the procedure ApxEMD.

> Translation(A, B, ϵ):
>
> 1. Let $\alpha = \min_{i,j} \text{ApxEMD}(A(\vec{t}_{i\to j}), B, 1)$ and let G be a uniform square grid of spacing $\epsilon\alpha/(6\sqrt{2})$.
> 2. For each pair of points $a_i \in A$ and $b_j \in B$ do:
>   (a) Place a disk D of radius α around $\vec{t}_{i\to j}$.
>   (b) For every grid point $\vec{t}_{\text{g}}$ of any cell of G that intersects D compute a value $\widetilde{\text{EMD}}(\vec{t}_{\text{g}}) = \text{ApxEMD}(A(\vec{t}_{\text{g}}), B, \epsilon/3)$.
> 3. Report the grid point $\vec{t}_{\text{apx}}$ that minimizes $\widetilde{\text{EMD}}(\vec{t}_{\text{g}})$.

Figure 3: Algorithm Translation(A, B, ϵ).

Theorem 4 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with $m \le n$. For any given $\epsilon > 0$,* Translation(A, B, ϵ) *computes a translation $\vec{t}_{\text{apx}}$ such that $\text{EMD}(\vec{t}_{\text{apx}}) \le (1+\epsilon)\text{EMD}(\vec{t}_{\text{opt}})$ in $O((n^3 m/\epsilon^4)\log^2 n)$ time.*

Proof: According to Lemma 1

$$\text{EMD}(\vec{t}_{\text{opt}}) \le \min_{i,j} \text{EMD}(\vec{t}_{i\to j}) \le 2\text{EMD}(\vec{t}_{\text{opt}}).$$

From Theorem 1 we have that

$$\text{EMD}(\vec{t}_{i\to j}) \le \text{ApxEMD}(A(\vec{t}_{i\to j}), B, 1) \le 2\text{EMD}(\vec{t}_{i\to j}).$$

Hence, since $\alpha = \min_{i,j} \text{ApxEMD}(A(\vec{t}_{i\to j}), B, 1)$ we have that

$$\text{EMD}(\vec{t}_{\text{opt}}) \le \alpha \le 4\text{EMD}(\vec{t}_{\text{opt}}).$$

Also, according to Observation 1, there is a point-to-point translation $\vec{t}_{i\to j}$ for which $|\vec{t}_{i\to j} - \vec{t}_{\text{opt}}| \le \text{EMD}(\vec{t}_{\text{opt}}) \le \alpha$. Algorithm Translation will, at some stage, consider the α-neighborhood of such a translation, and thus, compute a value $\widetilde{\text{EMD}}(\vec{t}_{\text{g}})$ for the grid point $\vec{t}_{\text{g}}$ that is closest to $\vec{t}_{\text{opt}}$, for which

$$|\vec{t}_{\text{g}} - \vec{t}_{\text{opt}}| \le \sqrt{2(\epsilon\alpha/\sqrt{72})^2}/2 = (\epsilon/3)(\alpha/4) \le (\epsilon/3)\text{EMD}(\vec{t}_{\text{opt}}),$$

and thus $d_{ij}(\vec{t}_{\text{g}}) \le d_{ij}(\vec{t}_{\text{opt}}) + (\epsilon/3)\text{EMD}(\vec{t}_{\text{opt}})$; see Figure 4. Assuming that $\{f_{ij}\}$ is the optimal flow at $\vec{t}_{\text{opt}}$, and similarly to the proof of Lemma 1, we have

$$\begin{aligned}
\text{EMD}(\vec{t}_{\text{g}}) &\le \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}(\vec{t}_{\text{g}})}{\min\{W, U\}} \\
&\le \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} (d_{ij}(\vec{t}_{\text{opt}}) + (\epsilon/3)\text{EMD}(\vec{t}_{\text{opt}}))}{\min\{W, U\}} \\
&= (1+\epsilon/3)\text{EMD}(\vec{t}_{\text{opt}}).
\end{aligned}$$

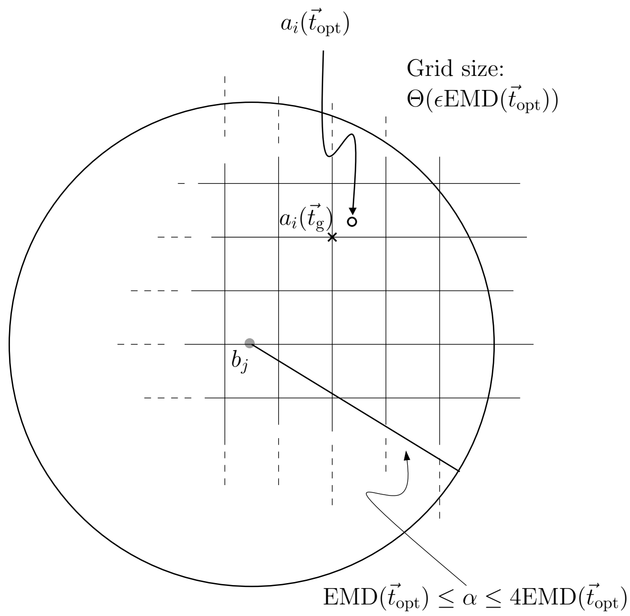

Figure 4: A pair of points a_i, b_j for which $d_{ij}(\vec{t}_{\mathrm{opt}}) \leq \mathrm{EMD}(\vec{t}_{\mathrm{opt}})$, and the grid translation $\vec{t}_{\mathrm{g}}$ of a_i that is closest to $\vec{t}_{\mathrm{opt}}$, for which $|\vec{t}_{\mathrm{g}} - \vec{t}_{\mathrm{opt}}| \leq (\epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$.

From Theorem 1 we have that

$$\mathrm{EMD}(\vec{t}_{\mathrm{g}}) \leq \widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}}) \leq (1+\epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{g}}).$$

Hence, the algorithm reports a translation $\vec{t}_{\mathrm{apx}}$ such that

$$\begin{aligned}
\mathrm{EMD}(\vec{t}_{\mathrm{apx}}) &\leq \widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{apx}}) \\
&\leq \widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}}) \\
&\leq (1+\epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{g}}) \\
&\leq (1+\epsilon/3)(1+\epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}) \\
&\leq (1+\epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}),
\end{aligned}$$

for every $\epsilon \leq 3$. As for the running time, observe that there are nm point-to-point translations, around each of which procedure ApxEMD is run for $O(\alpha^2/(\alpha^2\epsilon^2)) = O(1/\epsilon^2)$ grid points. Hence, the algorithm runs in $O((nm/\epsilon^2)(n^2/\epsilon^2)\log^2 n) = O((n^3m/\epsilon^4)\log^2 n)$ time. □

4.1 Equal weight sets

In this section we consider the case of sets with equal total weights. Let $\vec{t}_{C(A)\to C(B)}$ be the translation that matches the centers of mass $C(A)$ and $C(B)$. As Klein and Veltkamp [15] noted, the lower bound of Theorem 3 implies that the center of mass is a reference point [2] for equal weight sets, resulting in a trivial 2-approximation algorithm: compute $\mathrm{EMD}(\vec{t}_{C(A)\to C(B)})$. The proof is straightforward and very similar to the one of Lemma 1.

Also, according to Theorem 3, $\vec{t}_{\mathrm{opt}}$ is at most $\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$ far away from $\vec{t}_{C(A)\to C(B)}$. Hence, we need to search for $\vec{t}_{\mathrm{opt}}$ only within a small neighborhood of $\vec{t}_{C(A)\to C(B)}$. We modify algorithm TRANSLATION as follows: First, we compute $C(A)$ and $C(B)$. Then, we run APXEMD$(A(\vec{t}_{C(A)\to C(B)}), B, 1)$ and set α to the value returned. Next, we use the same grid size as in TRANSLATION, and run APXEMD$(A(\vec{t}_{\mathrm{g}}), B, \epsilon/3)$ for all the grid points $\vec{t}_{\mathrm{g}}$ which are at most α away from $\vec{t}_{C(A)\to C(B)}$. The minimum over all these approximations gives the desired approximation bound, as it easily follows from arguments very similar to the ones used in the proof of Theorem 4. Note that the total number of grid points to be tested is $O(1/\epsilon^2)$. Hence, we have managed to save an nm term from the time bound of Theorem 4.

Theorem 5 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with equal total weights and $m \le n$. For any given $\epsilon > 0$, a translation $\vec{t}_{\mathrm{apx}}$ such that* $\mathrm{EMD}(\vec{t}_{\mathrm{apx}}) \le (1+\epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$ *can be computed in $O((n^2/\epsilon^4)\log^2 n)$ time.*

For the assignment problem under translations, we can use the above algorithm for equal weight sets, running APXMATCH instead of APXEMD. This reduces the running time further.

Theorem 6 *For any given $\epsilon > 0$, a $(1+\epsilon)$-approximation of the minimum cost assignment under translations can be computed in $O((n^{3/2}/\epsilon^{7/2})\log^5 n)$ time.*

Note that the latter algorithm can be also applied to equal weight sets with bounded integer point weights by replacing each point by as many points as its weight.

4.2 Partial assignment

In Section 3, Observation 1, we saw that, in general, there is at least one pair of points a_i, b_j whose distance is at most EMD. Next, we prove that for the partial assignment case there is a linear number of pairs of points whose distance is at most 2EMD.

Lemma 2 *Given two weighted point sets $A = \{a_1, \ldots, a_m\}$, $B = \{b_1, \ldots, b_n\}$ with $m \le n$ and $w_i = u_j = 1$ for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$, there are at least $m/2$ distances d_{ij} with $d_{ij} \le 2\mathrm{EMD}$.*

Proof: Consider an optimal flow $\{f_{ij}\}$ that results in a partial assignment between A and B. Then there are exactly m flow variables f_{ij} with $f_{ij} = 1$ and $m(n-1)$ variables with zero flow. Since $\min\{W, U\} = m$, we have that $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij} = m\mathrm{EMD}$, where exactly m terms d_{ij} appear in the sum. Since $d_{ij} \ge 0$, it follows that there are at most k out of m distances d_{ij} with $d_{ij} \ge (m/k)\mathrm{EMD}$. Equivalently, there are at least $m-k$ distances d_{ij} with $d_{ij} \le (m/k)\mathrm{EMD}$. We choose $k = m/2$, and the lemma follows. □

Note that algorithm TRANSLATION tests all possible nm pairs of points a_i, b_j in order to find at least one for which $d_{ij}(\vec{t}_{\mathrm{opt}}) \le \mathrm{EMD}(\vec{t}_{\mathrm{opt}})$. Based on the above lemma, we can easily prove that testing an almost linear number of pairs suffices in order to find one for which $d_{ij}(\vec{t}_{\mathrm{opt}}) \le 2\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$ with high probability. Algorithm RANDOMTRANSLATION is given in Figure 5, and it is a straightforward probabilistic version of algorithm TRANSLATION.

Theorem 7 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with $m \le n$ and $w_i = u_j = 1$ for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$. For any given $\epsilon > 0$,* RANDOMTRANSLATION(A, B, ϵ) *computes a translation $\vec{t}_{\mathrm{apx}}$ such that* $\mathrm{EMD}(\vec{t}_{\mathrm{apx}}) \le (1+\epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$ *in $O((n^3/\epsilon^4)\log^3 n)$ time. The algorithm succeeds with probability at least $1 - n^{-1}$.*

RandomTranslation(A, B, ϵ):

1. Repeat $(2/\log e)n \log n$ times:
   (a) Choose a random pair $(a_i, b_j) \in A \times B$.
   (b) Let $\alpha_{ij} = 2\cdot$ApxEMD$(A(\vec{t}_{i\to j}), B, 1)$.
   (c) Let G be a uniform square grid of spacing $\epsilon\alpha_{ij}/(18\sqrt{2})$.
   (d) Place a disk D of radius α_{ij} around $\vec{t}_{i\to j}$.
   (e) For every grid point $\vec{t}_g$ of any cell of G that intersects D compute the value $\widetilde{\mathrm{EMD}}(\vec{t}_g) =$ ApxEMD$(A(\vec{t}_g), B, \epsilon/3)$.
2. Report the grid point $\vec{t}_{\mathrm{apx}}$ that minimizes $\widetilde{\mathrm{EMD}}(\vec{t}_g)$.

Figure 5: Algorithm RandomTranslation(A, B, ϵ).

Proof: According to Lemma 2, there are at least $m/2$ distances $d_{ij}(\vec{t}_{\mathrm{opt}})$ with $d_{ij}(\vec{t}_{\mathrm{opt}}) \leq 2\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$. Since there are in total nm possible distances $d_{ij}(\vec{t}_{\mathrm{opt}})$, we have that

$$\mathbf{Pr}[d_{ij}(\vec{t}_{\mathrm{opt}}) > 2\mathrm{EMD}(\vec{t}_{\mathrm{opt}})] \leq 1 - m/(2nm) = 1 - 1/(2n)$$

for a random pair a_i, b_j. Thus, the probability that K random draws of a pair a_i, b_j will all fail to give a pair for which $d_{ij}(\vec{t}_{\mathrm{opt}}) \leq 2\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$ is at most $(1 - 1/2n)^K \leq e^{-K/2n}$. By choosing $K = (2/\log e)n \log n$ the latter probability is at most $e^{-(\log n)/\log e} = n^{-1}$.

The rest of the proof is almost identical to the proof of Theorem 4. That is, if a pair a_i, b_j for which $d_{ij}(\vec{t}_{\mathrm{opt}}) \leq 2\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$ is tested, then the algorithm will compute a value α_{ij} such that

$$2\mathrm{EMD}(\vec{t}_{\mathrm{opt}}) \leq \alpha_{ij} < 12\mathrm{EMD}(\vec{t}_{\mathrm{opt}}).$$

Moreover, for that pair the algorithm will try a translation $\vec{t}_g$ such that

$$\mathrm{EMD}(\vec{t}_{\mathrm{opt}}) \leq \mathrm{EMD}(\vec{t}_{\mathrm{apx}}) \leq \widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{apx}}) \leq \widetilde{\mathrm{EMD}}(\vec{t}_g) \leq (1+\epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$$

and report $\vec{t}_{\mathrm{apx}}$. The algorithm takes $O(((n \log n)/\epsilon^2)(n/\epsilon)^2 \log^2 n) = O((n^3/\epsilon^4)\log^3 n)$ time, and it fails if and only if all random pairs satisfy $d_{ij} > 2\mathrm{EMD}(\vec{t}_{\mathrm{opt}})$, which happens with probability at most n^{-1}. ◻

Note that, in practice, it may pay off to first carry out Step 1.(b) for *all* randomly chosen pairs and compute the minimum α of the obtained values. The pairs a_i, b_j with $\alpha_{ij} > 6\alpha$ can then be discarded from further consideration.

5 Approximation algorithms for rotations

In this section we give $(2+\epsilon)$ and $(1+\epsilon)$-approximation algorithms for rotations for the general and partial assignment case respectively.

Let $\angle a_i o b_j$ be the angle between the segments $\overline{oa_i}$ and $\overline{ob_j}$ such that $0 \leq \angle a_i o b_j \leq \pi$. Also, let $\theta_{i\to j}$ be the rotation by $\angle a_i o b_j$ that aligns the origin o and points a_i and b_j such

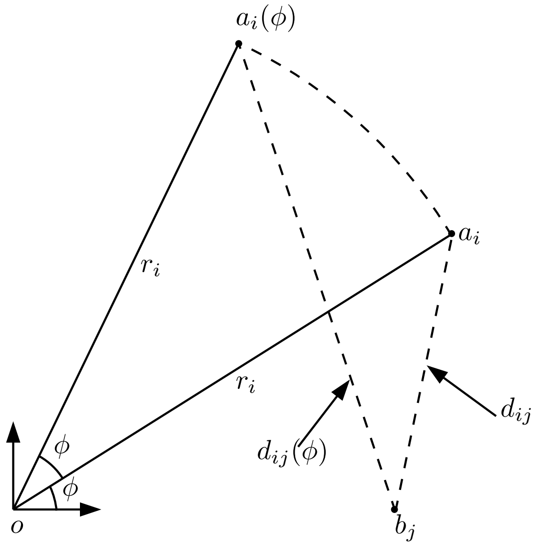

Figure 6: If $\angle a_i o b_j = \phi$ and a_i is rotated about o by ϕ then $d_{ij}(\phi) \leq 2d_{ij}$.

that both a_i and b_j are on the same side of o. Note that this is the rotation that minimizes $d_{ij}(\theta)$; we call such a rotation an *alignment rotation*.

We begin with a simple lemma that we will need later on.

Lemma 3 *Let a_i and b_j be two points in the plane with $\angle a_i o b_j = \phi$. If a_i is rotated by an angle $|\theta| \leq \phi$, then $d_{ij}(\theta) \leq 2d_{ij}$.*

Proof: Note that we are only interested in the rotation of a_i that increases its distance to b_j. We can assume that none of a_i and b_j coincides with the origin. Then, without loss of generality, we assume that $x_{b_j} > 0$, $y_{b_j} = 0$ and $y_{a_i} > 0$. We can assume that $y_{a_i} \neq 0$, since, otherwise, if $x_{a_i} > 0$ then $\phi = 0$ and $d_{ij}(\theta) = d_{ij}$, or if $x_{a_i} < 0$ then $\phi = \pi$ and $d_{ij}(\theta) \leq d_{ij}$.

First, consider the case where $\phi \geq \pi/2$. Then, the smallest possible distance d_{ij} occurs when $\phi = \pi/2$ with $x_{a_i} = 0$ and $d_{ij} = \sqrt{y_{a_i}^2 + x_{b_j}^2}$. The largest possible distance $d_{ij}(\theta)$ occurs when $\angle a_i(\theta) o b_j = \pi$ with $d_{ij}(\theta) = y_{a_i} + x_{b_j}$. Clearly, $d_{ij}(\theta) \leq \sqrt{2} d_{ij}$.

When $\phi < \pi/2$, $d_{ij}(\theta)$ increases with θ hence, since $|\theta| \leq \phi$, it suffices to bound $d_{ij}(\phi)$; see Figure 6 for an illustration. Let $r_i = \sqrt{x_{a_i}^2 + y_{a_i}^2}$ be the rotation radius of a_i. We have

$$d_{ij} = \sqrt{r_i^2 + x_{b_j}^2 - 2x_{b_j} r_i \cos\phi}$$

and

$$d_{ij}(\phi) = \sqrt{r_i^2 + x_{b_j}^2 - 2x_{b_j} r_i \cos(2\phi)}.$$

Then

$$\begin{aligned} 4d_{ij}^2 - d_{ij}^2(\phi) &= 3r_i^2 + 3x_{b_j}^2 + 2x_{b_j} r_i (2\cos^2\phi - 4\cos\phi - 1) \\ &\geq 3(r_i - x_{b_j})^2 \\ &\geq 0, \end{aligned}$$

where in the equality we used that $\cos(2x) = 2\cos^2 x - 1$, and in the first inequality we used that $2(\cos\phi - 1)^2 - 3 \geq -3$. Hence, $d_{ij}(\theta) \leq d_{ij}(\phi) \leq 2d_{ij}$. ◻

Let θ_{opt} be an optimal rotation. Consider the angle $\angle a_i(\theta_{\text{opt}})ob_j$ for every pair of points $a_i(\theta_{\text{opt}})$ and b_j and let $\angle a_{i_0}(\theta_{\text{opt}})ob_{j_0}$ be the smallest of all these angles. Then $\theta_{i_0 \to j_0}$ is the alignment rotation that is *closest* to θ_{opt}. Similarly to Lemma 1, and using Lemma 3, we can now prove that this alignment rotation gives a 2-approximation of $\text{EMD}(\theta_{\text{opt}})$. Hence, we have the following:

Lemma 4 *Given two weighted point sets A and B,*

$$\text{EMD}(\theta_{\text{opt}}) \leq \min_{i,j} \text{EMD}(\theta_{i \to j}) \leq 2\text{EMD}(\theta_{\text{opt}}).$$

Proof: Clearly, $\text{EMD}(\theta_{\text{opt}}) \leq \min_{i,j} \text{EMD}(\theta_{i \to j})$. Consider an optimal position $A(\theta_{\text{opt}})$ of A and an optimal flow $\{f_{ij}\}$ between $A(\theta_{\text{opt}})$ and B. We assume that θ_{opt} is not an alignment rotation, otherwise the lemma holds trivially. Next, consider the angle $\angle a_i(\theta_{\text{opt}})ob_j$ for every pair of points $a_i(\theta_{\text{opt}})$ and b_j, and let $\angle a_{i_0}(\theta_{\text{opt}})ob_{j_0}$ be the smallest of all these angles. Assume that we rotate $A(\theta_{\text{opt}})$ by $\angle a_{i_0}(\theta_{\text{opt}})ob_{j_0}$ to the position $A(\theta_{i_0 \to j_0})$; this is the alignment rotation that is closest to θ_{opt}. Then, $\angle a_{i_0}(\theta_{i_0 \to j_0})ob_{j_0} = 0$ and

$$\angle a_i(\theta_{i_0 \to j_0})ob_j \leq \angle a_i(\theta_{\text{opt}})ob_j + \angle a_{i_0}(\theta_{\text{opt}})ob_{j_0} \leq 2\angle a_i(\theta_{\text{opt}})ob_j,$$

for every $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. According to Lemma 3 we have that $d_{ij}(\theta_{i_0 \to j_0}) \leq 2d_{ij}(\theta_{\text{opt}})$. Concluding,

$$\begin{aligned}
\min_{i,j} \text{EMD}(\theta_{i \to j}) &\leq \text{EMD}(\theta_{i_0 \to j_0}) \\
&\leq \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}(\theta_{i_0 \to j_0})}{\min\{W, U\}} \\
&\leq \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} 2d_{ij}(\theta_{\text{opt}})}{\min\{W, U\}} \\
&= 2\text{EMD}(\theta_{\text{opt}}).
\end{aligned}$$

◻

By approximating $\min_{i,j} \text{EMD}(\theta_{i \to j})$ with $\min_{i,j} \text{ApxEMD}(A(\theta_{i \to j}), B, \epsilon/2)$ we can get a $(2 + \epsilon)$-approximation of $\text{EMD}(\theta_{\text{opt}})$. We call this algorithm $\text{Rotation}(A, B, \epsilon)$. Apart from the cost value, Rotation returns the corresponding rotation $\theta_{i \to j}$ as well.

Theorem 8 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with $m \leq n$. For any given $\epsilon > 0$, $\text{Rotation}(A, B, \epsilon)$ computes a rotation θ_{apx} such that $\text{EMD}(\theta_{\text{apx}}) \leq (2 + \epsilon)\text{EMD}(\theta_{\text{opt}})$ in $O((n^3 m/\epsilon^2) \log^2 n)$ time.*

5.1 Partial assignment.

For the special case where all the weights are one, we can achieve a $(1 + \epsilon)$-approximation as follows. Let $a_1 b_{j_1}, \ldots, a_m b_{j_m}$ be a matching corresponding to an optimal integer flow at an optimal rotation θ_{opt}. Observe that $d_{ij_i}(\theta_{\text{opt}}) \leq m\text{EMD}(\theta_{\text{opt}})$ since $m\text{EMD}(\theta_{\text{opt}}) =$

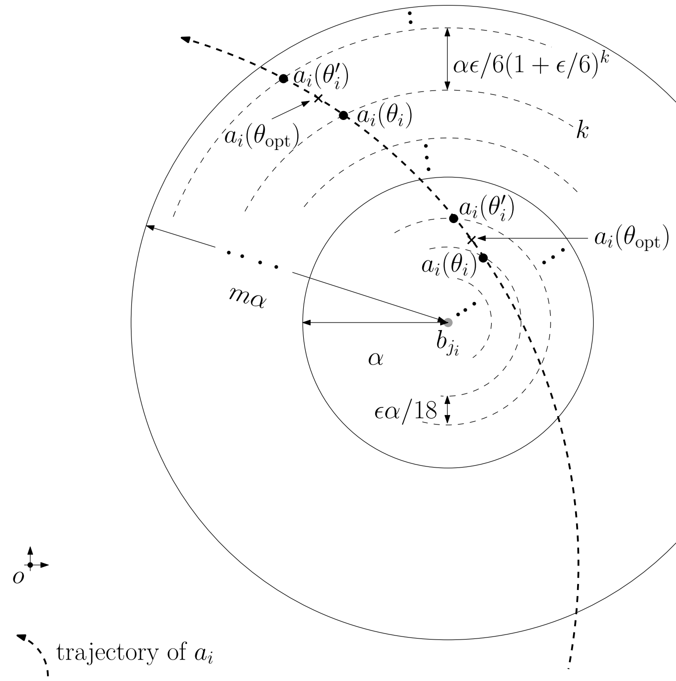

Figure 7: A pair of points a_i, b_{j_i} for which $d_{ij_i}(\theta_{\text{opt}}) \leq m\text{EMD}(\theta_{\text{opt}})$, and two examples of possible positions of $a_i(\theta_{\text{opt}}), a_i(\theta_i)$ and $a_i(\theta_i')$, depending on $d_{ij_i}(\theta_{\text{opt}})$.

$\sum_i d_{ij_i}(\theta_{\text{opt}})$. Thus, in order to find an optimal rotation we only need to consider nm sets of rotations

$$\{\theta \in [0, 2\pi) : d_{ij}(\theta) \leq m\text{EMD}(\theta_{\text{opt}})\},$$

for all i, j. Of course, since we do not know the $\text{EMD}(\theta_{\text{opt}})$, we consider instead the rotations $R_{ij}(\alpha) = \{\theta \in [0, 2\pi) : d_{ij}(\theta) \leq m\alpha\}$, for some value α such that $\text{EMD}(\theta_{\text{opt}}) \leq \alpha \leq 3\text{EMD}(\theta_{\text{opt}})$. Inside each R_{ij} we consider sample rotations Θ_{ij} according to the following. We divide $R_{ij}(\alpha)$ into two parts, $R_{ij}^{<}(\alpha) = \{\theta \in [0, 2\pi) : d_{ij}(\theta) \leq \alpha\}$ and $R_{ij}^{>}(\alpha) = \{\theta \in [0, 2\pi) : \alpha \leq d_{ij}(\theta) \leq m\alpha\}$. Rotations in $R_{ij}^{<}(\alpha)$ are handled by considering the set of distances

$$D_{ij}^{<}(\alpha) = \{k \cdot \epsilon \tfrac{\alpha}{18} \in (0, \alpha) \mid k \in \mathbb{N}\} \cup \{0, \alpha\},$$

which contains $O(1/\epsilon)$ values. Rotations in $R_{ij}^{>}(\alpha)$ are handled by considering the set of distances

$$D_{ij}^{>}(\alpha) = \{\alpha(1 + \epsilon/6)^k \in (\alpha, m\alpha) \mid k \in \mathbb{N}\} \cup \{\alpha, m\alpha\},$$

which contains $O(\log_{1+\epsilon} \frac{m\alpha}{\alpha}) = O(\log_{1+\epsilon} m) = O((1/\epsilon)\log m)$ values. Let $D_{ij}(\alpha) = D_{ij}^{<}(\alpha) \cup D_{ij}^{>}(\alpha)$, and consider the set of angles $\Theta_{ij} = \{\theta_{i\to j}\} \cup \{\theta \in [0, 2\pi) \mid d_{ij}(\theta) \in D_{ij}(\alpha)\}$. Clearly, Θ_{ij} contains $O((1/\epsilon)\log m)$ angles.

Our goal is to prove that the best rotation among $\bigcup_{ij} \Theta_{ij}$ provides a $(1+\epsilon)$-approximation for $\text{EMD}(\theta_{\text{opt}})$. The main idea is that the angles in $R_{ij}^{<}(\alpha)$ take care of distances $d_{ij_i}(\theta_{\text{opt}})$ that are at most α by controlling the absolute error that such pairs produce in the approximation, while the angles in $R_{ij}^{>}(\alpha)$ take care of the distances $d_{ij_i}(\theta_{\text{opt}})$ that are between α and $m\text{EMD}(\theta_{\text{opt}}) \leq m\alpha$ by controlling the relative error that these pairs produce.

A detailed description of the algorithm, referred to as PartRotation, is given in Figure 8. The algorithm shown runs ApxEMD for the general case where $m < n$. When $m = n$, ApxMATCH can be used instead.

PartRotation(A, B, ϵ):

1. Let $\alpha = \min_{i,j}$ ApxEMD$(A(\theta_{i\to j}), B, 1/2)$.
2. For each pair of points $a_i \in A$ and $b_j \in B$ do:
   (a) Compute $D_{ij}(\alpha) = D^{<}_{ij}(\alpha) \cup D^{>}_{ij}(\alpha)$.
   (b) Let $\Theta_{ij} = \{\theta_{i\to j}\} \cup \{\theta \in [0, 2\pi) \mid d_{ij}(\theta) \in D_{ij}(\alpha)\}$.
   (c) For each sample rotation $\theta \in \Theta_{ij}$ compute a value $\widetilde{\mathrm{EMD}}(\theta) =$ApxEMD$(A(\theta), B, \epsilon/3)$.
3. Report the sample rotation θ_{apx} that minimizes $\widetilde{\mathrm{EMD}}(\theta)$.

Figure 8: Algorithm PartRotation(A, B, ϵ).

Theorem 9 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two point sets in the plane with $m \le n$ and $w_i = u_j = 1$ for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$. For any given $\epsilon \in (0, 1)$, PartRotation(A, B, ϵ) computes a rotation θ_{apx} such that $\mathrm{EMD}(\theta_{\mathrm{apx}}) \le (1+\epsilon)\mathrm{EMD}(\theta_{\mathrm{opt}})$ in $O((n^3 m/\epsilon^3)\log^3 n)$ time.*

Proof: First note that $\mathrm{EMD}(\theta_{\mathrm{opt}}) \le \alpha \le 3\mathrm{EMD}(\theta_{\mathrm{opt}})$. Let $a_1 b_{j_1}, \ldots, a_m b_{j_m}$ be a matching corresponding to an optimal integer flow at an optimal rotation θ_{opt}, and consider the sample rotation $\theta_{\mathrm{g}} \in \bigcup_i \Theta_{ij_i}$ that is closest to θ_{opt}. Our objective is to show that $\widetilde{\mathrm{EMD}}(\theta_{\mathrm{g}}) \le (1+\epsilon)\mathrm{EMD}(\theta_{\mathrm{opt}})$. Observe that if $\theta_{\mathrm{opt}} \in \bigcup \Theta_{ij}$ then the approximation holds trivially.

Since $\theta_{i\to j} \in \Theta_{ij}$, $d_{ij}(\theta)$ varies monotonically between two successive values in Θ_{ij}. Consider one pair $a_i b_{j_i}$, and let $\theta_i, \theta_i' \in \Theta_{ij_i}$ be the two closest angles between which θ_{opt} lies, with $d_{ij_i}(\theta_i) < d_{ij_i}(\theta_i')$, see Figure 7. It follows that $d_{ij_i}(\theta_i) \le d_{ij_i}(\theta_{\mathrm{opt}}) \le d_{ij_i}(\theta_i')$ and $d_{ij_i}(\theta_i) \le d_{ij_i}(\theta_{\mathrm{g}}) \le d_{ij_i}(\theta_i')$. If $d_{ij_i}(\theta_{\mathrm{opt}}) < \alpha$, then $\theta_{\mathrm{opt}} \in R^{<}_{ij_i}(\alpha)$, and also $\theta_i, \theta_i' \in R^{<}_{ij_i}(\alpha)$. Since θ_i, θ_i' are contiguous in Θ_{ij_i}, we have $d_{ij_i}(\theta_i') - d_{ij_i}(\theta_i) \le \epsilon\alpha/18$, and therefore

$$d_{ij_i}(\theta_{\mathrm{g}}) - d_{ij_i}(\theta_{\mathrm{opt}}) \le \epsilon\alpha/18 \le \epsilon\mathrm{EMD}(\theta_{\mathrm{opt}})/6.$$

If $d_{ij_i}(\theta_{\mathrm{opt}}) \ge \alpha$, then $\theta_{\mathrm{opt}} \in R^{>}_{ij_i}(\alpha)$, and also $\theta_i, \theta_i' \in R^{>}_{ij_i}(\alpha)$. Again since θ_i, θ_i' are contiguous in Θ_{ij_i}, we have $d_{ij_i}(\theta_i') \le (1+\epsilon/6)d_{ij_i}(\theta_i)$, and therefore $d_{ij_i}(\theta_{\mathrm{g}}) \le (1+\epsilon/6)d_{ij_i}(\theta_{\mathrm{opt}})$.

Then we have

$$\begin{aligned}
\mathrm{EMD}(\theta_{\mathrm{g}}) &\le \frac{\sum_{i=1}^{m} d_{ij_i}(\theta_{\mathrm{g}})}{m} \\
&\le \frac{\sum_{\{i:d_{ij_i}(\theta_{\mathrm{opt}})<\alpha\}} d_{ij_i}(\theta_{\mathrm{g}}) + \sum_{\{i:d_{ij_i}(\theta_{\mathrm{opt}})\ge\alpha\}} d_{ij_i}(\theta_{\mathrm{g}})}{m} \\
&\le \frac{\sum_{\{i:d_{ij_i}(\theta_{\mathrm{opt}})<\alpha\}} (d_{ij_i}(\theta_{\mathrm{opt}}) + \epsilon\mathrm{EMD}(\theta_{\mathrm{opt}})/6)}{m} \\
&\quad + \frac{\sum_{\{i:d_{ij_i}(\theta_{\mathrm{opt}})\ge\alpha\}} (1+\epsilon/6) d_{ij_i}(\theta_{\mathrm{opt}})}{m}
\end{aligned}$$

$$\begin{aligned}
&\leq \frac{\sum_{i=1}^{m} d_{ij_i}(\theta_{\mathrm{opt}})}{m} + \frac{\sum_{\{i: d_{ij_i}(\theta_{\mathrm{opt}})<\alpha\}} \epsilon \mathrm{EMD}(\theta_{\mathrm{opt}})/6}{m} \\
&\quad + \epsilon/6 \frac{\sum_{\{i: d_{ij_i}(\theta_{\mathrm{opt}})\geq\alpha\}} d_{ij_i}(\theta_{\mathrm{opt}})}{m} \\
&\leq \mathrm{EMD}(\theta_{\mathrm{opt}}) + \frac{\sum_{i=1}^{m} \epsilon \mathrm{EMD}(\theta_{\mathrm{opt}})/6}{m} + \epsilon/6 \frac{\sum_{i=1}^{m} d_{ij_i}(\theta_{\mathrm{opt}})}{m} \\
&= \mathrm{EMD}(\theta_{\mathrm{opt}}) + (\epsilon/3)\mathrm{EMD}(\theta_{\mathrm{opt}}),
\end{aligned}$$

and we conclude

$$\begin{aligned}
\mathrm{EMD}(\theta_{\mathrm{apx}}) &\leq \widetilde{\mathrm{EMD}}(\theta_{\mathrm{apx}}) \\
&\leq \widetilde{\mathrm{EMD}}(\theta_{\mathrm{g}}) \\
&\leq (1+\epsilon/3)\mathrm{EMD}(\theta_{\mathrm{g}}) \\
&\leq (1+\epsilon/3)(1+\epsilon/3)\mathrm{EMD}(\theta_{\mathrm{opt}}) \\
&\leq (1+\epsilon)\mathrm{EMD}(\theta_{\mathrm{opt}}).
\end{aligned}$$

As for the running time, note that for each pair of points a_i, b_j we run ApxEMD for $O((1/\epsilon)\log m)$ sample rotations. Hence, PartRotation runs in $O((nm/\epsilon)\log m(n^2/\epsilon^2)\log^2 n)$ $= O((n^3 m/\epsilon^3)\log^3 n)$ time. □

For the assignment problem under rotations, i.e., when $n = m$, we can use the above algorithm, running ApxMATCH instead of ApxEMD, thus, reducing the running time to $O((n^2/\epsilon)\log n(n/\epsilon)^{3/2}\log^5 n) = O((n^{7/2}/\epsilon^{5/2})\log^6 n)$.

Theorem 10 *For any given $\epsilon > 0$, a $(1+\epsilon)$-approximation of the minimum cost assignment under rotations can be computed in $O((n^{7/2}/\epsilon^{5/2})\log^6 n)$ time.*

6 Approximation algorithms for rigid motions

In this section we show how to combine the approximation algorithms for translations with the ones for rotations to get approximation algorithms for rigid motions.

Let $I_{\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}}}$ be an optimal rigid motion. First, we can combine algorithm Rotation with the 2-approximation algorithm for translations in Theorem 8 to get a $(4+\epsilon)$-approximation of the minimum EMD under rigid motions in the following way: for each point-to-point translation $\vec{t}_{i\to j}$, compute a $(2+\epsilon/2)$-approximation of the optimum EMD between $A(\vec{t}_{i\to j})$ and B under rotations about b_j. The minimum over all these approximations gives a $2(2+\epsilon/2)$-approximation of $\mathrm{EMD}(\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}})$; see, for example, the first step of algorithm RigidMotion shown in Figure 9 where a 6-approximation of $\mathrm{EMD}(\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}})$ is computed.

Lemma 5 *For any given $\epsilon > 0$, a $(4+\epsilon)$-approximation of the minimum EMD under rigid motions can be computed in $O((n^4 m^2/\epsilon^2)\log^2 n)$ time.*

Proof: According to Observation 1, there exist two points a_{i_0}, b_{j_0} whose distance at an optimal position of A is at most the minimum EMD under rigid motions. The above algorithm will use, at some stage, b_{j_0} as the center of rotation by translating B appropriately. Of course for this 'new' position of B there is an optimal rigid motion of A, $I_{\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}}}$ for which $d_{i_0,j_0}(\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}}) \leq \mathrm{EMD}(\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}})$ as well.

If A is translated by $\vec{t}_{i_0 \to j_0}$ instead of $\vec{t}_{\rm opt}$, and then rotated by $\theta_{\rm opt}$ we have $d_{ij}(\vec{t}_{i_0\to j_0}, \theta_{\rm opt}) \le d_{ij}(\vec{t}_{\rm opt}, \theta_{\rm opt}) + |\vec{t}_{\rm opt} - \vec{t}_{i_0\to j_0}|$, for every $i \in \{1,\ldots,m\}$, $j \in \{1,\ldots,n\}$. Since $|\vec{t}_{\rm opt} - \vec{t}_{i_0\to j_0}| = d_{i_0,j_0}(\vec{t}_{\rm opt}, \theta_{\rm opt}) \le \mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$ we have that $d_{ij}(\vec{t}_{i_0\to j_0}, \theta_{\rm opt}) \le d_{ij}(\vec{t}_{\rm opt}, \theta_{\rm opt}) + \mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$. Thus, we have

$$\mathrm{EMD}(\vec{t}_{i_0\to j_0}, \theta_{\rm opt}) \le 2\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}).$$

If $\theta^{ij}_{\rm opt}$ is the optimal rotation of $A(\vec{t}_{i\to j})$ about b_j then

$$\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}) \le \mathrm{EMD}(\vec{t}_{i_0\to j_0}, \theta^{i_0 j_0}_{\rm opt}) \le \mathrm{EMD}(\vec{t}_{i_0\to j_0}, \theta_{\rm opt}).$$

Thus,

$$\begin{aligned}
\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}) &\le \min_{i,j} \mathrm{EMD}(\vec{t}_{i\to j}, \theta^{ij}_{\rm opt}) \\
&\le \mathrm{EMD}(\vec{t}_{i_0\to j_0}, \theta^{i_0 j_0}_{\rm opt}) \\
&\le 2\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}).
\end{aligned}$$

From Theorem 8 we also have that

$$\mathrm{EMD}(\vec{t}_{i\to j}, \theta^{ij}_{\rm opt}) \le \textsc{Rotation}(A(\vec{t}_{i\to j}), B, \epsilon/2) \le (2+\epsilon/2)\mathrm{EMD}(\vec{t}_{i\to j}, \theta^{ij}_{\rm opt}).$$

Putting it all together we get

$$\begin{aligned}
\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}) &\le \min_{i,j} \mathrm{EMD}(\vec{t}_{i\to j}, \theta^{ij}_{\rm opt}) \\
&\le \min_{i,j} \textsc{Rotation}(A(\vec{t}_{i\to j}), B, \epsilon/2) \\
&\le (2+\epsilon/2) \min_{i,j} \mathrm{EMD}(\vec{t}_{i\to j}, \theta^{ij}_{\rm opt}) \\
&\le 2(2+\epsilon/2)\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}) \\
&= (4+\epsilon)\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt}).
\end{aligned}$$

Since Rotation is run nm times, the algorithm runs in $O(nm(n^3m/\epsilon^2)\log^2 n) = O((n^4m^2/\epsilon^2)\log^2 n)$ time. □

A $(2+\epsilon)$-approximation algorithm for rigid motions is based on similar ideas. According to Observation 1, there exist two points a_i, b_j whose distance at $I_{\vec{t}_{\rm opt},\theta_{\rm opt}}$ is at most $\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$. We place a grid of suitable size around each $\vec{t}_{i\to j}$. For each grid point $\vec{t}_{\rm g}$ that is at most $\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$ away from $\vec{t}_{i\to j}$ we compute a $(2+\epsilon)$-approximation of the optimum EMD between $A(\vec{t}_{\rm g})$ and B under rotations about b_j. The minimum over all these approximations is within a factor of $(2+\epsilon)$ of $\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$. Since we do not know $\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$, we first compute a 6-approximation of it as shown above. Algorithm $\textsc{RigidMotion}(A, B, \epsilon)$ is shown in Figure 9. For the partial assignment problem, a $(1+\epsilon)$-approximation can be achieved by running PartRotation instead of Rotation.

Theorem 11 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with $m \le n$. For any given $\epsilon > 0$,* $\textsc{RigidMotion}(A, B, \epsilon)$ *computes a rigid motion $I_{\vec{t}_{\rm apx},\theta_{\rm apx}}$ such that $\mathrm{EMD}(\vec{t}_{\rm apx}, \theta_{\rm apx}) \le (2+\epsilon)\mathrm{EMD}(\vec{t}_{\rm opt}, \theta_{\rm opt})$ in $O((n^4m^2/\epsilon^4)\log^2 n)$ time.*

RigidMotion(A, B, ϵ):

1. For each pair of points $a_i \in A$ and $b_j \in B$ do:

   (a) Set the center of rotation, i.e. the origin, to be b_j by translating B appropriately.

   (b) Run Rotation$(A(\vec{t}_{i\to j}), B, 1)$ and let α_{ij} be the cost value returned.

   Let $\alpha = \min_{i,j} \alpha_{ij}$.

2. Let G be a uniform grid of spacing $\alpha\epsilon/(12\sqrt{2})$. For each pair of points $a_i \in A$ and $b_j \in B$ do:

   (a) Set the center of rotation, i.e. the origin, to be b_j by translating B appropriately.

   (b) Place a disk D of radius α around $\vec{t}_{i\to j}$.

   (c) For every grid point $\vec{t}_{\mathrm{g}}$ of any cell of G that intersects D run Rotation$(A(\vec{t}_{\mathrm{g}}), B, \epsilon/3)$. Let $\widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}})$ and $\theta^{\mathrm{g}}_{\mathrm{apx}}$ be the cost value and angle returned respectively.

3. Report the grid point $\vec{t}_{\mathrm{apx}}$ that minimizes $\widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}})$, and the corresponding angle θ_{apx}.

Figure 9: Algorithm RigidMotion(A, B, ϵ).

Proof: The proof is very similar to the proof of Lemma 5. First note that according to that lemma, $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}) \le \alpha \le 6\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$. Consider again a pair of points a_{i_0}, b_{j_0} such that $\vec{t}_{\mathrm{opt}}$ is at most $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ away from $\vec{t}_{i_0\to j_0}$. Since, at some stage, the algorithm will consider b_{j_0} as the center of rotation, we have that $\vec{t}_{\mathrm{opt}} \in D$, where D is a disk of radius α around $\vec{t}_{i_0\to j_0}$. For the grid translation $\vec{t}_{\mathrm{g}}$ that is closest to $\vec{t}_{\mathrm{opt}}$ we have $|\vec{t}_{\mathrm{g}} - \vec{t}_{\mathrm{opt}}| \le (\epsilon/4)(\alpha/6) \le (1/4)\epsilon\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$. Similarly to the proof of Theorem 4 we have that

$$\mathrm{EMD}(\vec{t}_{\mathrm{g}}, \theta_{\mathrm{opt}}) \le (1 + \epsilon/4)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}).$$

If $\theta^{\mathrm{g}}_{\mathrm{opt}}$ is the optimal rotation of $A(\vec{t}_{\mathrm{g}})$ about b_{j_0} then $\mathrm{EMD}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{opt}}) \le \mathrm{EMD}(\vec{t}_{\mathrm{g}}, \theta_{\mathrm{opt}})$. Note that Rotation$(A(\vec{t}_{\mathrm{g}}), B, \epsilon/3)$ returns a cost $\widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}})$ for which

$$\widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}}) \le (2 + \epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{opt}}).$$

Hence, in total we have

$$\begin{aligned}
\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}) &\le \mathrm{EMD}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}) \\
&\le \widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{apx}}) \\
&\le \widetilde{\mathrm{EMD}}(\vec{t}_{\mathrm{g}}) \\
&\le (2 + \epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{g}}, \theta^{\mathrm{g}}_{\mathrm{opt}}) \\
&\le (2 + \epsilon/3)\mathrm{EMD}(\vec{t}_{\mathrm{g}}, \theta_{\mathrm{opt}}) \\
&\le (2 + \epsilon/3)(1 + \epsilon/4)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}) \\
&\le (2 + \epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}}),
\end{aligned}$$

where the last inequality holds for any $\epsilon \le 2$.

Since Rotation runs for $O(nm/\epsilon^2)$ grid translations in total, the algorithm runs in $O((nm/\epsilon^2)(n^3m/\epsilon^2)\log^2 n) = O((n^4m^2/\epsilon^4)\log^2 n)$ time. ◻

6.1 Equal weight sets

As in the case of translations, for equal weight sets we need to search for the optimal translation only around $\vec{t}_{C(A)\to C(B)}$. We set the center of rotation to be $C(B)$. A 6-approximation of $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ can be computed by simply running $\textsc{Rotation}(A(\vec{t}_{C(A)\to C(B)}), B, 1)$. Similarly, we need to run $\textsc{Rotation}(A(\vec{t}_{\mathrm{g}}), B, \epsilon/3)$ only for grid points $\vec{t}_{\mathrm{g}}$ that are close to $\vec{t}_{C(A)\to C(B)}$.

Theorem 12 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with equal total weights and $m \leq n$. For any given $\epsilon > 0$, a rigid motion $I_{\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}}$ such that $\mathrm{EMD}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}) \leq (2+\epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ can be computed in $O((n^3 m/\epsilon^4)\log^2 n)$ time.*

For the assignment problem, instead of using Rotation, we can use the version of PartRotation that runs ApxMATCH to achieve a $(1+\epsilon)$-approximation.

Theorem 13 *For any given $\epsilon > 0$, a $(1+\epsilon)$-approximation of the minimum cost assignment under rigid motions can be computed in $O((n^{7/2}/\epsilon^{9/2})\log^6 n)$ time.*

6.2 Partial assignment

Finally, for the partial assignment problem, a $(1+\epsilon)$-approximation can be computed in $O((nm/\epsilon^2)(n^3 m/\epsilon^3)\log^3 n) = O((n^4 m^2/\epsilon^5)\log^3 n)$ time by algorithm RigidMotion running PartRotation instead of Rotation. Additionally, we can use the same arguments as in the translational case to convert RigidMotion into a randomized algorithm, where its two first steps are executed only for a random selection of $\Theta(n \log n)$ pairs of points. We conclude with the following:

Theorem 14 *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two weighted point sets in the plane with $m \leq n$ and $w_i = u_j = 1$ for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$. For any given $\epsilon > 0$, a rigid motion $I_{\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}}$ such that $\mathrm{EMD}(\vec{t}_{\mathrm{apx}}, \theta_{\mathrm{apx}}) \leq (1+\epsilon)\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ can be computed in $O((n^4 m^2/\epsilon^5)\log^3 n)$ time. The same approximation can also be computed in $O((n^4 m/\epsilon^5)\log^4 n)$ time with success probability at least $1 - n^{-1}$.*

7 Concluding remarks

We have presented polynomial-time $(1+\epsilon)$ and $(2+\epsilon)$-approximation algorithms for the minimum Euclidean EMD under translations and rigid motions.

Note that algorithm ApxEMD in Section 2 can be trivially generalized in higher dimensions: for a d-dimensional point set $A \cup B$, a $(1+\epsilon)$-spanner G_s with $O(\epsilon^{-d+1})$ edges can be computed in $O(n \log n + (n/\epsilon^d)\log(1/\epsilon))$ time [7]. Here, the constants hidden in the notation depend exponentially in the dimension. As before, we can run Orlin's algorithm on G_s and ApxEMD takes $O((n^2/\epsilon^{2(d-1)})\log^2(n/\epsilon))$ time. It is not clear how the approximation algorithm of Varadarajan and Agarwal for the minimum cost bipartite matching in the plane carries on in higher dimensions neither what time bounds are obtained. Also, note that the lower bounds in Section 3 and Lemma 2 hold for any dimension. Hence, for the general EMD in d-dimensional Euclidean space, a $(1+\epsilon)$-approximation of the minimum under translations can be computed in $O((n^3 m/\epsilon^{3d-2})\log^2(n/\epsilon))$ time. Algorithm RandomTranslation generalizes in a similar way.

An open question is whether the $(1 + \epsilon)$-approximation for partial assignment under rotations can be generalized to the general case of arbitrary weights. Another interesting and non-trivial task is to give lower and upper bounds of the complexity of the function $\mathrm{EMD}(\vec{t}, \theta)$, i.e., the total number of its local optima.

Acknowledgments. The authors would like to thank Sariel Har-Peled for helpful discussions.

References

[1] R.K. Ahuja, T.L. Magnati, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, 1993.

[2] O. Aichholzer, H. Alt, and G. Rote. Matching shapes with a reference point. *Internat. J. Comput. Geom. Appl.*, 4:349–363, 1997.

[3] H. Alt and L.J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.R. Sack and J. Urrutia, editors, *Handbook of Comput. Geom.*, pages 121–153. Elsevier Science Publishers B.V. North-Holland, 1999.

[4] S. Arya, D.M. Mount, and M. Smid. Dynamic algorithms for geometric spanners of small diameter: Randomized solutions. *Comput. Geom. Theory Appl.*, 13:91–107, 1999.

[5] D.S. Atkinson and P.M. Vaidya. Using geometry to solve the transportation problem in the plane. *Algorithmica*, 13:442–461, 1995.

[6] P. Bose, A. Maheshwari, and P. Morin. Fast approximations for sums of distances clustering and the Fermat-Weber problem. *Comput. Geom. Theory Appl.*, 24:135–146, 2003.

[7] P.B. Callahan and S.R. Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. 4th Annu. ACM/SIGACT-SIAM Sympos. Discrete Algorithms*, pages 291–300, 1993.

[8] R. Chandrasekaran and A. Tamir. Algebraic optimization: The Fermat-Weber location problem. *Math. Programming*, 46(2):219–224, 1990.

[9] S. Cohen. *Finding Color and Shape Patterns in Images*. PhD thesis, Stanford University, Department of Computer Science, 1999.

[10] S. Cohen and L. Guibas. The earth mover's distance under transformation sets. In *Proc. 7th IEEE Internat. Conf. Computer Vision*, pages 173–187, 1999.

[11] Z. Drezner, K. Klamroth, A. Schöbel, and G. Wesolowsky. The Weber problem. In Z. Drezner and H. Hamacher, editors, *Facility location. Application and theory*, pages 1–36. Springer, 2002.

[12] P. Giannopoulos and R.C. Veltkamp. A pseudo-metric for weighted point sets. In *Proc. 7th European Conf. Computer Vision*, volume 2352 of *Lecture Notes in Computer Science*, pages 715–731. Springer, 2002.

[13] K. Grauman and T. Darell. Fast contour matching using approximate earth mover's distance. In *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 220–227, 2004.

[14] M. Hagedoorn and R.C. Veltkamp. State-of-the-art in shape matching. In M. Lew, editor, *Principles of Visual Information Retrieval*, pages 87–119. Springer, 2001.

[15] O. Klein and R.C. Veltkamp. Approximation algorithms for computing the earth mover's distance under transformations. In *Proc. 16th Internat. Sympos. Algorithms and Comput.*, volume 3827 of *Lecture Notes in Computer Science*, pages 1019–1028. Springer, 2005.

[16] Q. Lv, M. Charikar, and K. Li. Image similarity search with compact data structures. In *Proc. 13th ACM Conf. Information Knowledge Management*, pages 208–217, 2004.

[17] D. Mumford. Mathematical theories of shape: Do they model perception? In *Proc. 1991 Internat. Society for Optical Engineering Sympos., Geometric Methods in Computer Vision*, volume 1570 of *SPIE*, pages 2–10, 1991.

[18] J.B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, 41(2):338–350, 1993.

[19] Y. Rubner, L.J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proc. ARPA Image Understanding Workshop*, pages 661–668, 1997.

[20] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *Internat. J. Computer Vision*, 40(2):99–121, 2000.

[21] M. Tanase and R.C. Veltkamp. A survey of content-based image retrieval systems. In O. Marques and B. Furht, editors, *Content-based image and video retrieval*, pages 47–101. Kluwer Academic Publishers, 2002.

[22] R. Typke, P. Giannopoulos, R.C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. In *Proc. 4th Internat. Sympos. on Music Information Retrieval*, pages 107–114, 2003.

[23] K.R. Varadarajan and P.K. Agarwal. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 805–814, 1999.