A Sub-Quadratic Algorithm for Bipartite Matching of Planar Points with Bounded Integer Coordinates*

R. Sharathkumar
Department of Computer Science
Stanford University
sharathk@stanford.edu

ABSTRACT

Let $A, B \in [\Delta]^2$, $|A| = |B| = n$, be point sets where each point has a positive integer coordinate bounded by Δ. For an arbitrary small constant $\delta > 0$, we design an algorithm to compute a minimum-cost Euclidean bipartite matching of A, B in $O(n^{3/2+\delta}\log(n\Delta))$ time; all previous exact algorithms for the Euclidean bipartite matching, even when the point sets have bounded integer coordinates take $\Omega(n^2)$ time.

First, we compute in $O(n^{3/2+\delta}\log n\Delta)$ time, a candidate set $\mathcal{E} \subseteq A \times B$ such that $M^* \subseteq \mathcal{E}$ and the graph $G(A \cup B, \mathcal{E})$ is planar; here M^* is a minimum-cost matching of A and B. Next, we compute M^* from this weighted bipartite planar graph in $O(n^{3/2}\log n)$ using the algorithm described in [6].

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Nonnumerical Algorithms and Problems—*geometrical problems and computations*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms*

Keywords

Euclidean bipartite matching

1. INTRODUCTION

For $A, B \in [\Delta]^2$, let $G(A, B) = G(A \cup B, A \times B)$[1], be a weighted bipartite graph with $|A| = |B| = n$. Let the Euclidean distance $\|ab\|$ be the cost of an edge $(a, b) \in A \times B$. A *matching* M in G is a set of vertex-disjoint edges. The *cost* of M denoted by $\mathtt{w}(M)$ is the sum of cost of its edges, i.e.,

$$\mathtt{w}(M) = \sum_{(a,b)\in M} \|ab\|.$$

M is a *perfect matching* if $|M| = n$. We call a perfect matching on $G(A, B)$ *optimal* and denote it by $M^*(A, B)$ (or M^* when A and B is obvious from the context), if its cost is minimum among all perfect matchings of G. In this paper we present an algorithm to compute M^* in $O(n^{3/2+\delta}\log(n\Delta))$ time; here $\delta > 0$ is an arbitrarily small constant[2]. Throughout this paper, we use δ to represent this constant.

Previous work. Optimal matching on weighted bipartite graphs with n vertices and m edges can be computed using the Hungarian algorithm in $O(mn)$ time [7]. If a perfect matching does not exist, it computes a maximum cardinality minimum-cost matching. For weighted graphs, if the edge costs are positive integers bounded by $n^{O(1)}$, Gabow and Tarjan [4] show that an optimal matching can be computed in $O(m\sqrt{n}\log n)$ time.

For the case where $A, B \subset \mathbb{R}^2$ and an L_p-norm, Vaidya presents an algorithm to compute optimal matching in $O(n^{2.5})$ time [10]. Agarwal *et al.* [1] improve the running time to $O(n^{2+\delta})$. For L_1- and L_∞-norms, Vaidya [10] shows that the running time of his algorithm can be improved to $n^2\log^{O(d)} n$. It is an open question whether there is a sub-quadratic algorithm that computes an optimal Euclidean bipartite matching for planar point sets.

There are algorithms that compute an ε-approximate matching – a matching whose cost is within a factor $(1+\varepsilon)$ of the optimal – in sub-quadratic time. After a sequence of sub-quadratic approximation algorithms [2, 5, 11], Sharathkumar and Agarwal presented an algorithm to compute an ε-approximate matching in near-linear time [9].

Sharathkumar and Agarwal [8] present an alternate approximation algorithm to compute, for any cost function $\mathtt{d}(\cdot,\cdot)$, a matching within an additive error $\varepsilon > 0$ of the optimal matching in time $O(n^{3/2}\Phi(n)\log(n\Delta/\varepsilon))$; here $\Phi(n)$ is the query and update time of a weighted nearest-neighbor data structure under $\mathtt{d}(\cdot,\cdot)$. Let M be a smallest-cost perfect matching among all perfect matchings with a cost (strictly) greater than $\mathtt{w}(M^*)$, i.e., $M = \mathrm{argmin}_{M',\mathtt{w}(M')\neq\mathtt{w}(M^*)}\,\mathtt{w}(M')$. The algorithm of [8] returns an optimal matching when $\varepsilon = \mathtt{w}(M) - \mathtt{w}(M^*)$. For the L_1 and L_∞-norms, the cost of every edge is an integer and hence the value of ε is at least 1. Therefore, this algorithm returns an optimal matching in $O(n^{3/2}\mathrm{poly}(\log n)\log(n\Delta))$ time. For the Euclidean norm (resp. L_p-norm), however, the expression for ε, i.e., $\mathtt{w}(M) - \mathtt{w}(M^*)$ is the difference between the sum of n square

*This work is supported by NSF under grants CCF-1011228, CCF-1161480 and ARO grant 1139681-1-TXBEE

[1]For simplicity in exposition, we assume that there are no duplicate points in $A \cup B$.

[2]The constant of proportionality hidden in the big-O notation depends on δ and tends to ∞ as δ tends to 0.

SoCG'13, June 17–20, 2013, Rio de Janeiro, Brazil.

roots (resp. p^{th}-root) of integers bounded by $2\Delta^2$, i.e.,

$$\varepsilon < \sum_{i=1}^{n} \sqrt{a_i} - \sum_{j=1}^{n} \sqrt{b_j} \qquad a_i, b_j \in [1 : 2\Delta^2].$$

Determining the upper and lower bounds on the smallest value of ε – also known as the *sum of square roots problem* – is an important open question in computational geometry. In particular, it is known that the value of ε can be at least $\Delta^{-2^{2k+1}}$ (See Lemma 1). Using this value of ε, we obtain only an exponential upper bound on the running time of the algorithm [8] for computing optimal matching.

LEMMA 1. [3] *The minimum non-zero difference between two sum of square roots of integers*

$$\left| \sum_{i=1}^{k} \sqrt{a_i} - \sum_{i=1}^{k} \sqrt{b_i} \right|,$$

where a_i, b_i are integers no larger than Δ^2 is at least $\Delta^{-2^{2k+1}}$

Our results. In this paper, we present an algorithm to compute optimal Euclidean bipartite matching of A, B in $O(n^{3/2+\delta} \log(n\Delta))$ time. To the best of our knowledge, this is the first sub-quadratic algorithm for computing Euclidean optimal matching even when points are from a bounded integer grid. Our algorithm extends to L_p-norm, for any integer $1 < p < \infty$. However, we restrict the description of our algorithm to the Euclidean norm.

Our approach uses the algorithm of [8] to first compute an approximate matching by solving an approximate linear program in $O(n^{3/2+\delta} \log(n\Delta))$ time. Next, exploiting the geometry of the point set and using the dual weights on points returned by the algorithm of [8], our algorithm computes a small candidate set $\mathcal{E} \subset A \times B$ in $O(n^{1+\delta})$ time. The candidate set contains an optimal matching M^*, i.e., $M^* \subseteq \mathcal{E}$ and the graph induced by it, $\mathrm{G}(A \cup B, \mathcal{E})$ is planar. Finally, we use the algorithm of Lipton and Tarjan [6] to compute M^* from the planar graph $\mathrm{G}(A \cup B, \mathcal{E})$ in $O(n^{3/2} \log n)$ time.

The rest of the paper is structured as follows. In Section 2, we describe an approximate linear program for matching. In Section 3, we present new properties of this approximate linear program. Exploiting these properties, in Section 4, we present an algorithm to compute a candidate set with the desired properties. This leads to the main result of our paper (Theorem 1). We conclude in Section 5.

2. PRELIMINARIES

In this section, we introduce terminology useful in describing our algorithm.

For $v \in A \cup B$, let $y(v)$ be its *dual weight*. A matching M and a set of dual weights are dual feasible if for every edge $(a, b) \in A \times B$:

$$y(a) + y(b) \le \mathtt{d}(a, b), \tag{1}$$

$$y(a) + y(b) = \mathtt{d}(a, b), \quad \text{if } (a, b) \in M. \tag{2}$$

It is well-known that any dual feasible perfect matching is optimal. See [7] for details.

For a scaling parameter $\theta > 0$, let *θ-scaled graph*, $\mathrm{G}_\theta(A, B)$, be the graph identical to $\mathrm{G}(A, B)$, except that the cost of every edge $(u, v) \in A \times B$ is $\mathtt{d}_\theta(u, v) = \left\lceil \frac{\|uv\|}{\theta} \right\rceil$. Given a graph $\mathrm{G}_\theta(A, B)$, a 1-*feasible matching* consists of a matching M and set of dual weights $y(\cdot)$ such that for all edges between $(u, v) \in A \times B$ we have

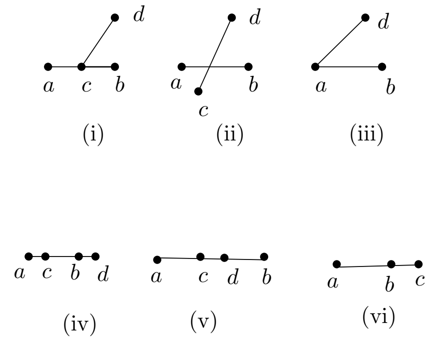

Figure 1. (a, b) and (c, d) intersect in (i) and (ii). In (iii) (a, b),(a, d) do not intersect. In (iv) and (v), (a, b) and (c, d) overlap. In (vi), (a, b) and (a, c) overlap.

$$y(u) + y(v) \le \mathtt{d}_\theta(u, v) + 1, \tag{3}$$

$$y(u) + y(v) = \mathtt{d}_\theta(u, v) \quad \text{for } (u, v) \in M. \tag{4}$$

The above conditions with the +1 removed from (3) are identical to (1) and (2). A 1-*optimal matching* is a perfect matching that is 1-feasible.

For $\varepsilon > 0$ and $\theta = \varepsilon/3n$, the cost of a 1-optimal matching M is within an additive error ε from the optimal and such a matching and the corresponding set of dual weights can be computed in time $O(n^{3/2+\delta} \log(n\Delta/\varepsilon))$(See [8] for details), i.e.,

$$\mathtt{w}(M) \le \mathtt{w}(M^*) + \varepsilon.$$

For any two points a, b, we denote the line segment joining a and b by $\overline{ab}$. Let $\mathrm{int}(\overline{ab})$ be the interior of the line segment $\overline{ab}$, i.e., $\overline{ab} \setminus \{a, b\}$. e is the *point of intersection* of (a, b) and (c, d) if $e \in \overline{ab} \cap \overline{cd}$ and $e \in \mathrm{int}(\overline{ab}) \cup \mathrm{int}(\overline{cd})$. When a, b, c and d are non-collinear, we refer to (a, b) and (c, d) as *intersecting edges*. When a, b, c and d are collinear, we refer to (a, b) and (c, d) as *overlapping edges*. See Figure 1.

3. PROPERTIES OF 1-OPTIMAL MATCHING

In this section, we present new properties of a 1-optimal matching M and the corresponding set of dual weights of $\mathrm{G}_\theta(A, B)$. Throughout this paper, we set θ to be $\frac{1}{n\Delta^{33}}$. For any point $v \in A \cup B$, let *scaled dual weight* of v is

$$\tilde{y}(v) = \theta \cdot y(v) = \frac{y(v)}{n\Delta^{33}}.$$

From (3) and (4), we obtain the following inequalities:

$$\begin{aligned} \tilde{y}(u) + \tilde{y}(v) &= \frac{1}{n\Delta^{33}}(y(u) + y(v)) \\ &\le \frac{1}{n\Delta^{33}}(\lceil (n\Delta^{33})\|uv\| \rceil + 1) \\ &\le \|uv\| + \frac{2}{n\Delta^{33}}, \end{aligned} \tag{5}$$

$$\begin{aligned} \tilde{y}(u) + \tilde{y}(v) &= \frac{1}{n\Delta^{33}}(y(u) + y(v)) \\ &= \frac{1}{n\Delta^{33}} \lceil (n\Delta^{33})\|uv\| \rceil \\ &\ge \|uv\| \quad \text{for } (u, v) \in M. \end{aligned} \tag{6}$$

Every edge of any optimal matching M^* satisfies the property described in the following lemma.

LEMMA 2. *Let $A, B \subset [\Delta]^2$ and let M^* be any optimal matching of A, B. Let M be a 1-optimal matching on $\mathrm{G}_\theta(A, B)$, for $\theta = \frac{1}{n\Delta^{33}}$, and $\tilde{y}(\cdot)$ be the set of scaled dual weights. For any edge $(a, b) \in M^*$,*

$$\tilde{y}(a) + \tilde{y}(b) > \|ab\| - \frac{2}{\Delta^{33}}. \tag{7}$$

PROOF. Suppose there is an edge $(a, b) \in M^*$ such that $\tilde{y}(a) + \tilde{y}(b) \le \|ab\| - \frac{2}{\Delta^{33}}$.

$$\begin{aligned} \mathtt{w}(M) &\le \sum_{(a',b')\in M} (\tilde{y}(a') + \tilde{y}(b')) = \sum_{u\in A\cup B} \tilde{y}(u) \\ &\le \Big(\sum_{u\in(A\cup B)\setminus\{a,b\}} \tilde{y}(u)\Big) + \|ab\| - \frac{2}{\Delta^{33}} \\ &\le \Big(\sum_{(a',b')\in M^*\setminus(a,b)} (\tilde{y}(a') + \tilde{y}(b'))\Big) + \|ab\| - \frac{2}{\Delta^{33}}. \end{aligned}$$

Since, $M^* \setminus (a, b)$ has $n - 1$ edges each of which satisfy (5). Therefore, we have

$$\begin{aligned} \mathtt{w}(M) &\le \mathtt{w}(M^* \setminus (a, b)) + \frac{2(n-1)}{n\Delta^{33}} + \|ab\| - \frac{2}{\Delta^{33}} \\ &< \mathtt{w}(M^*). \end{aligned}$$

This contradicts our assumption that M^* is a minimum-cost matching. □

We refer to every edge that satisfies (7) as an *eligible* edge. Let $E \subseteq A \times B$ be the set of all eligible edges. In the following lemmas, we show that no two eligible edges intersect.

LEMMA 3. *For any point set $P \subset [\Delta]^2$, let $a, b, c, d \in P$ be four non-collinear points. Suppose segments $\overline{ab}$ and $\overline{cd}$ intersect. Then,*

$$\|ad\| + \|bc\| + 1/\Delta^{32} < \|ab\| + \|cd\|. \tag{8}$$

PROOF. Let $e \in \overline{ab} \cap \overline{cd}$, i.e., e be the point of intersection of the segments. In Figure 2(ii), three of the four points are collinear. In this case, b and e are the same point. Since a, b, c and d are non-collinear, either $\{b, e, c\}$ or $\{a, e, d\}$ are non-collinear. For three non-collinear points under Euclidean distance, triangular inequality holds with a strict inequality. [3] Therefore, we have

$$\|ab\| + \|cd\| = \|be\| + \|ec\| + \|de\| + \|ea\| > \|bc\| + \|ad\|.$$

All points have integer coordinates and therefore all the four distances, i.e., $\|ab\|, \|cd\|, \|bc\|$ and $\|ad\|$ are square roots of integers. From Lemma 1 we have, $\|ab\| + \|cd\| - \|bc\| - \|ad\| > 1/\Delta^{32}$. □

LEMMA 4. *Let $\theta = \frac{1}{n\Delta^{33}}$. For a 1-optimal matching M and a set of dual weights $y(\cdot)$ on $\mathrm{G}_\theta(A, B)$, let for any $u \in A\cup B$, $\tilde{y}(u) = y(u)\cdot\theta$. For four non-collinear points $a, c \in A$ and $b, d \in B$, suppose (a, b) and (c, d) are eligible edges then (a, b) and (c, d) do not intersect.*

[3]This claim holds for all norms except L_1 and L_∞-norms.

PROOF. Suppose (a, b) and (c, d) are eligible edges that intersect. From (7) and (8), we have

$$\begin{aligned} \tilde{y}(a) + \tilde{y}(b) + \tilde{y}(c) + \tilde{y}(d) &> \|ab\| + \|cd\| - \frac{4}{\Delta^{33}} \\ &> \|ad\| + \|bc\| - \frac{4}{\Delta^{33}} + \frac{1}{\Delta^{32}} \\ &> \|ad\| + \|bc\| + \frac{1}{2\Delta^{32}}. \end{aligned} \tag{9}$$

From (5), we have $\tilde{y}(a) + \tilde{y}(d) \le \|ad\| + 2/(n\Delta^{33})$ and $\tilde{y}(b) + \tilde{y}(c) \le \|bc\| + 2/(n\Delta^{33})$. Adding the two, we get

$$\tilde{y}(a) + \tilde{y}(b) + \tilde{y}(c) + \tilde{y}(d) \le \|ad\| + \|bc\| + 4/(n\Delta^{33}),$$

which contradicts (9). □

Suppose, no three points in $A \cup B$ are collinear, then Lemma 4 implies that $\mathrm{G}(A \cup B, E)$ is a planar graph. However, when points are collinear, the example of Figure 3 shows that there can be $\Omega(n^2)$ eligible edges. In the next section, we describe a procedure to compute a small candidate set of edges $\mathcal{E}$. The candidate set is such that $M^* \subseteq \mathcal{E}$ and $\mathrm{G}(A \cup B, \mathcal{E})$ is planar.

4. COMPUTING CANDIDATE SET

In this section, we present an algorithm to compute a candidate set $\mathcal{E}$ in time $O(n^{3/2+\delta}\log(n\Delta))$. First, our algorithm identifies subsets of collinear points that have eligible edges between them. Next, our algorithm computes the candidate set by choosing a small number of eligible edges from each of these subsets. We show that the eligible set constructed by our algorithm is such that $M^* \subseteq \mathcal{E}$ and $\mathrm{G}(A \cup B, \mathcal{E})$ is planar. Recollect that M is a 1-optimal matching, $\tilde{y}(\cdot)$ is the set of scaled dual weights and E is the set of eligible edges.

A graph $\mathrm{G}(A \cup B, \mathbb{E})$ is a *planar straight-line graph* if for any two edges $(a, b), (c, d) \in \mathbb{E}$, $\mathrm{int}(\overline{ab}) \cap \mathrm{int}(\overline{cd}) = \emptyset$. Let $P \subset (A\cup B)\times(A\cup B)$. Suppose $P = \{(p_1, q_1), \ldots, (p_k, q_k)\}$. For $1 \le i \le k$, let A_i (resp. B_i) be the subset of points of A (resp. B) that lie in the interior of the line segment joining p_i and q_i, i.e.,

$$\begin{aligned} A_i &= A \cap \mathrm{int}(\overline{p_iq_i}), \\ B_i &= B \cap \mathrm{int}(\overline{p_iq_i}) \end{aligned} \tag{10}$$

Let $E_i \subseteq E$ be the set of eligible edges that are contained in $\overline{p_iq_i}$ (See Figure 4),i.e.,

$$E_i = \{(a, b) \mid (a, b) \in E, \overline{ab} \subseteq \overline{p_i, q_i}\}. \tag{11}$$

$P = \{(p_1, q_1), \ldots, (p_k, q_k)\}$ is an *eligible edge decomposition* if it satisfies the following properties:

(P1) $E = \bigcup_{i=1}^{k} E_i$.

(P2) $\mathrm{G}(A \cup B, P)$ is a planar straight-line graph.

In Section 4.1, we give an algorithm to compute an eligible edge decomposition, i.e., a set P that satisfies (P1) and (P2). For each $(p_i, q_i) \in P$, the algorithm also returns A_i and B_i as defined in (10). For any $(p_i, q_i) \in P$ consider the sets A_i and B_i. In Lemma 5 and Lemma 6, we present properties satisfied by A_i and B_i that are useful in describing our algorithm.

LEMMA 5. *For any eligible edge $(u, v) \in E$, if $u \in A_i \cup B_i$ then $v \in A_i \cup B_i \cup \{p_i, q_i\}$.*

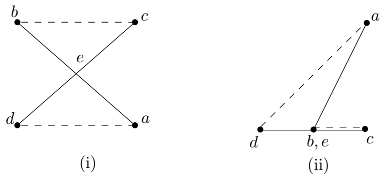

Figure 2. Proof of Lemma 3.

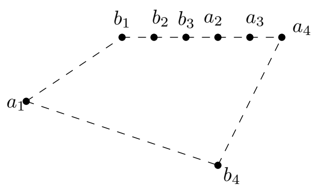

Figure 4. $P = \{(a_1, b_1), (b_1, a_4), (a_4, b_4), (b_4, a_1)\}$. Let $E = \{(\{b_1, b_2, b_3\} \times \{a_2, a_3, a_4\}) \cup \{b_1, a_1\} \cup \{a_1, b_4\} \cup \{b_4, a_4\}$. For (b_1, a_4), $B_i = \{b_2, b_3\}$ and $A_i = \{a_2, a_3\}$, $E_i = (A_i \cup \{a_4\}) \times (B_i \cup \{b_1\})$

Proof. $u \in A_i \cup B_i$, therefore $u \in \overline{p_i q_i}$ Let $V = A_i \cup B_i \cup \{p_i, q_i\}$. Suppose $v \notin V$, then $v \notin \overline{p_i q_i}$ and $(u, v) \notin E_i$. Let $j \neq i$ be such that $(u, v) \in E_j$ (From (P1), we know that such an E_j exists). By definition, $\overline{uv} \subseteq \overline{p_j q_j}$. This implies $u \in \overline{p_i q_i} \cap \overline{p_j q_j}$ and $u \in \text{int}(\overline{p_i q_i})$ contradicting (P2). □

Lemma 6. *Suppose $|B_i| \leq |A_i|$ Then,*

(i) $|A_i| \leq |B_i| + 2$,

(ii) if $|A_i| = |B_i| + 1$ then either $p_i \in B$ or $q_i \in B$,

(iii) if $|A_i| = |B_i| + 2$ then $p_i \in B$ and $q_i \in B$.

Proof. For every point $a \in A_i$, let b be such that $(a, b) \in M$

(i) (a, b) is eligible, therefore from Lemma 5, $b \in B_i \cup \{p_i, q_i\}$ implying $|B_i \cup \{p_i, q_i\}| \geq |A_i|$ or $|B_i| + 2 \geq |A_i|$.

(ii) Since both $p_i, q_i \in A$, for $a \in A_i$, we have $b \in B_i$ (from Lemma 5). This implies $|B_i| \geq |A_i|$ contradicting $|A_i| = |B_i| + 1$.

(iii) Since $p_i \in A$, for $a \in A_i$ we have $b \in B_i \cup \{q_i\}$ (from Lemma 5). This implies $|B_i| \geq |A_i| + 1$ contradicting our assumption that $|A_i| = |B_i| + 2$. □

A symmetric version of Lemma 6 holds for the case where $|A_i| \leq |B_i|$. Now, we describe our algorithm for computing the candidate set $\mathcal{E}$.

Algorithm. First, using the algorithm of [8], we compute a 1-optimal matching M of $G_\theta(A, B)$ for $\theta = 1/(n\Delta^{33})$. Next, using the algorithm presented in Section 4.1, we compute an eligible edge decomposition P. This algorithm also returns sets A_i and B_i for each $(p_i, q_i) \in P$.

Initialize $\mathcal{E}$ to $\emptyset$. Let $P = \{(p_1, q_1), \ldots, (p_k, q_k)\}$ be the eligible edge decomposition returned by the algorithm in Section 4.1. Recollect the notation $M^*(A, B)$ is an optimal matching on $G(A \cup B, A \times B)$. For every $(p_i, q_i) \in P$, consider the sets A_i and B_i. Suppose $|B_i| \leq |A_i|$. From Lemma 6(i), $|A_i| \leq |B_i| + 2$ (A symmetric algorithm along with a symmetric version of Lemma 6 is used when $|A_i| \leq |B_i| \leq |A_i| + 2$).

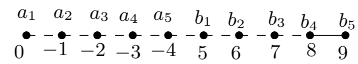

Figure 3. $A = \{a_1, \ldots, a_5\}, B = \{b_1, \ldots, b_5\}$. Every point is a unit distance from its neighbor. For the dual weights shown every pair (a_i, b_j) is eligible.

Case 1 $|B_i| = |A_i|$. If $p_i \in A, q_i \in B$, then

$$\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i) \cup M^*(A_i \cup \{p_i\}, B_i \cup \{q_i\}).$$

If $p_i, q_i \in A$ or if $p_i, q_i \in B$, then

$$\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i).$$

Case 2 $|B_i| + 1 = |A_i|$. If $p_i \in A, q_i \in B$, then

$$\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i \cup \{q_i\}).$$

If $p_i, q_i \in B$, then

$$\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i \cup \{q_i\}) \cup M^*(A_i, B_i \cup \{p_i\}).$$

From Lemma 6(ii) both p_i and q_i cannot be in A.

Case 3 $|B_i| = |A_i| + 2$. If $p_i, q_i \in B$, then

$$\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i \cup \{p_i, q_i\}).$$

From Lemma 6(iii), neither $p_i \in A$ nor $q_i \in A$.

Running time. For $\theta = 1/(n\Delta^{33})$, computing 1-optimal matching M takes $O(n^{3/2+\delta} \log(n\Delta))$ time [8]. Using the algorithm of Section 4.1, $P = \{(p_1, q_1), \ldots, (p_k, q_k)\}$ and for every $(p_i, q_i) \in P$, A_i and B_i can be computed in $O(n^{1+\delta})$ time. $G(A \cup B, P)$ is a planar straight-line graph and therefore $|P| = O(n)$.

Let $n_i = \max(|A_i|, |B_i|)$. Note that the points in $A_i \cup B_i$ are collinear. For every A_i, B_i, our procedure computes at most two optimal matchings of collinear point sets, each of size $O(n_i)$. An optimal matching for collinear points can be computed in $O(n_i \log n_i)$ time – this algorithm is identical to the $O(n \log n)$ algorithm for computing optimal matching for 1-dimensional point sets. For any $i \neq j$, $A_i \cap A_j = \emptyset$ and $B_i \cap B_j = \emptyset$(from (P2)) and therefore $\sum_i n_i = n$. Computing candidate set from P takes $O(\sum_i (n_i \log n_i) + |P|) = O(n \log n)$ time. Total time taken by the procedure is dominated by the time taken to compute M which is $O(n^{3/2+\delta} \log(n\Delta))$.

Correctness. Lemma 7 and Lemma 8 together show that the candidate set has the desired properties.

Lemma 7. *For $A, B \subseteq [\Delta]^2$, there is an optimal matching M^* in $G(A, B)$ such that the candidate set $\mathcal{E}$ generated by our algorithm contain M^*, i.e.,*

$$M^* \subseteq \mathcal{E}.$$

Proof. Fix an optimal matching M^*. Given candidate set $\mathcal{E}$, we modify M^* without increasing its cost such that all edges of M^* are included in $\mathcal{E}$. For any A_i, B_i, let

$$M_i^* = \bigcup_{(u,v) \in M^* \cap E_i} (u, v)$$

$$A'_i = \bigcup_{a \in A, (a,b) \in M^*_i} a, \qquad B'_i = \bigcup_{b \in B, (a,b) \in M^*_i} b.$$

$|A'_i| = |B'_i|$ and from Lemma 5, $A_i \cup B_i \subseteq A'_i \cup B'_i \subseteq A_i \cup B_i \cup \{p_i, q_i\}$. Let us assume $|B_i| \leq |A_i| \leq |B_i| + 2$. (A symmetric argument holds when $|A_i| \leq |B_i| \leq |A_i| + 2$).

Case 1 $|B_i| = |A_i|$. If $p_i \in A, q_i \in B$, then either $A'_i = A_i$, $B'_i = B_i$ or $A'_i = A_i \cup \{p_i\}$, $B'_i = B_i \cup \{q_i\}$. If $p_i, q_i \in A$ or $p_i, q_i \in B$, then $A'_i = A_i$ and $B'_i = B_i$.

Case 2 $|B_i| + 1 = |A_i|$. If $p_i \in A, q_i \in B$, then $A'_i = A_i, B'_i = B_i \cup \{q_i\}$. If $p_i, q_i \in B$, then $A'_i = A_i$ and either $B'_i = B_i \cup \{p_i\}$ or $B'_i = B_i \cup \{q_i\}$. From Lemma 6(ii), both p_i and q_i cannot be in A.

Case 3 $|B_i| + 2 = |A_i|$. If $p_i, q_i \in B$, then $A'_i = A_i$ and $B'_i = B_i \cup \{p_i, q_i\}$. From Lemma 6(iii), $p_i \notin A$ and $q_i \notin A$

For all the cases mentioned above, our algorithm adds the optimal matching $M^*(A'_i, B'_i)$ to the candidate set $\mathcal{E}$. Modifying $M^* \leftarrow (M^* \setminus M^*_i) \cup M^*(A'_i, B'_i)$ does not increase the cost of M^* since $\mathtt{w}(M^*_i) \geq \mathtt{w}(M^*(A'_i, B'_i))$. Furthermore, this modification ensures that $M^* \cap E_i \subseteq \mathcal{E}$. Repeating this for all i, we get $M^* \cap \bigcup_i E_i \subseteq M^* \cap E \subseteq \mathcal{E}$ (From (P1)). From Lemma 3, we know that $M^* \subseteq E$ and therefore, it follows that $M^* \subseteq \mathcal{E}$. $\square$

LEMMA 8. *Let $\mathcal{E}$ be the candidate set generated by our algorithm. Then $\mathrm{G}(A \cup B, \mathcal{E})$ is a planar graph.*

PROOF. From (P2), we know that the graph $\mathrm{G}(A \cup B, P)$ has a planar embedding where each edge (p_i, q_i) is drawn as the line segment $\overline{p_i q_i}$.

$\mathcal{E}$ can have multiple connected components. We show that every connected component of $\mathrm{G}(A \cup B, \mathcal{E})$ is a planar graph by describing a planar embedding of candidate set $\mathcal{E}$ as follows. For A_i, B_i, let us assume $|B_i| \leq |A_i| \leq |B_i| + 2$. (A symmetric argument holds when $|A_i| \leq |B_i| \leq |A_i| + 2$).

Case 1 $|A_i| = |B_i|$. If $p_i \in A, q_i \in B$, the algorithm adds $M^*(A_i, B_i) \cup M^*(A_i \cup \{p_i\}, B_i \cup \{q_i\})$ to $\mathcal{E}$. Apart from a set of vertex-disjoint edges and cycles – each of which is a planar component of $\mathrm{G}(A \cup B, \mathcal{E})$, $M^*(A_i, B_i) \cup M^*(A_i \cup \{p_i\}, B_i \cup \{q_i\})$ also consists of an alternating path K from p_i to q_i. Let $K = \langle p_i, e_1, e_2, \ldots, q_i \rangle$. Since every vertex on K has degree 2, K is homeomorphic to the edge (p_i, q_i). Therefore, K can be drawn as shown in Figure 5(i) so that planarity is not violated. If $p_i, q_i \in A$ or $p_i, q_i \in B$, then we add $M^*(A_i, B_i)$. Every edge of $M^*(A_i, B_i)$ is a vertex-disjoint edge each of which is a planar component of $\mathrm{G}(A \cup B, \mathcal{E})$.

Case 2 $|A_i| = |B_i| + 1$. If $p_i \in A, q_i \in B$, then $\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i \cup \{q_i\})$. Apart from vertex-disjoint edges, there is an edge in the candidate set, say (q_i, e') incident on q_i. e' has degree 1 and therefore can be drawn without violating planarity. See Figure 5(ii). If $p_i, q_i \in B$, then $\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i \cup \{q_i\}) \cup M^*(A_i, B_i \cup \{p_i\})$. Apart from a set of vertex-disjoint edges and cycles – each of which is a planar component – there are two vertex-disjoint alternating paths, one each from p_i and q_i. Let $K_1 = \langle p_i, e_1, \ldots, e_s \rangle$ and $K_2 = \langle q_i, f_1, \ldots, f_t \rangle$. K_1 and K_2 can be drawn as shown in Figure 5(iii).

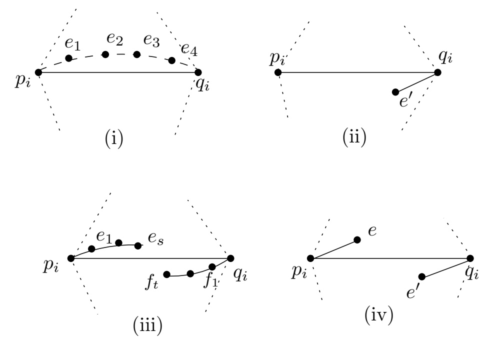

Figure 5. Proof of Lemma 8.

Case 3 $|A_i| = |B_i| + 2$. If $p_i, q_i \in B$, then $\mathcal{E} \leftarrow \mathcal{E} \cup M^*(A_i, B_i \cup \{p_i, q_i\})$. Apart from a set of vertex-disjoint edges and cycles – each of which is a planar component – there are at most two edges, say (p_i, e) and (q_i, e'). Both e and e' have degree 1 and therefore the edges can be drawn without violating planarity. See Figure 5(iv). $\square$

Lipton and Tarjan [6] present an algorithm to compute optimal matching in a weighted bipartite planar graph in $O(n^{3/2} \log n)$ time. Using their algorithm, we can compute M^* from $\mathrm{G}(A \cup B, \mathcal{E})$ leading to the following:

THEOREM 1. *For $A, B \subseteq [\Delta]^2$, there is an algorithm that, for an arbitrarily small constant $\delta > 0$, computes the minimum-cost Euclidean bipartite matching in $\mathrm{G}(A, B)$ in $O(n^{3/2+\delta} \log(n\Delta))$ time.*

4.1 Computing eligible edge decomposition

In this section, given a 1-optimal matching M of $\mathrm{G}_\theta(A, B)$ for $\theta = 1/(n\Delta^{33})$, we present an algorithm to compute an eligible edge decomposition $P = \{(p_1, q_1), \ldots, (p_k, q_k)\}$ that satisfies (P1) and (P2). For every $(p_i, q_i) \in P$, our algorithm also returns sets A_i, B_i as defined in (10).

For $u \in A \cup B$ and a line ℓ, u is *active* with respect to ℓ if there is an eligible edge $(u, v) \in E$ with $\overline{uv} \subset \ell$. Let $\mathcal{P}$ be point-line pairs such that $(p, \ell) \in \mathcal{P}$ if and only if p is active with respect to ℓ. First, we present a procedure to compute $\mathcal{P}$.

Computing $\mathcal{P}$. Let $M = \{(a_1, b_1) \ldots (a_n, b_n)\}$ be the 1-optimal matching. We index points of A and B such that points with the same index are matched to each other in M. Consider a decomposition of M into sets $\{M_1, \ldots, M_t\}$ such that two edges $(a_i, b_i), (a_h, b_h) \in M$ belong to the same partition, say M_j, if and only if a_i, b_i, a_h and b_h are collinear. Let $A^i = \{a_j \mid a_j \in A, (a_j, b_j) \in M_i\}$ and $B^i = \{b_j \mid b_j \in B, (a_j, b_j) \in M_i\}$. Any point $u \in A \cup B$ is *internal* if u is active only with respect to a single line ℓ, i.e., all eligible edges incident on u lie on line ℓ. u is *external* if for u is active with respect to two distinct lines ℓ and ℓ'. In Figure 4, b_2, b_3, a_2, and a_3 are internal points and b_1, a_1, b_4 and a_4 are external points. For any line ℓ, we direct ℓ from $x = -\infty$ to $x = \infty$. If the line is parallel to y-axis, we direct it from $y = -\infty$ to $y = +\infty$.

Initially $\mathcal{P} = \emptyset$. We present our procedure for points in A.

Consider any $a_i \in A^j$. We describe our procedure for a_i. Let ℓ_j be the line containing (a_i, b_i). We add (a_i, ℓ_j) to $\mathcal{P}$. Initialize $B' = B \setminus B^j$. Compute

$$b_r = \operatorname*{argmin}_{b \in B \setminus B^j} (\|a_i b\| - \tilde{y}(a_i) - \tilde{y}(b)). \tag{12}$$

Suppose $(a_i, b_r) \in E$. Let ℓ' be the line containing $\overline{a_i b_r}$. We add (a_i, ℓ') to $\mathcal{P}$, set $B' \leftarrow B' \setminus \{b_r\}$ and repeat the procedure. When $(a_i, b_r) \notin E$, we set $B' \leftarrow B \setminus B^j$ and terminate the procedure. We execute this procedure for all points in A^j and then repeat this procedure for every A^j. This completes the description of our algorithm to compute point-line pairs of $\mathcal{P}$ of points in A. A symmetric procedure computes all the point-line pairs for points in B.

Lemma 9 and Lemma 10 are used in the analysis of our procedure.

LEMMA 9. *Suppose* $(u, v) \in E$. *For every external point* p, $p \notin \mathrm{int}(\overline{uv})$.

PROOF. Suppose $p \in \mathrm{int}(\overline{uv})$. Since p is external, p is active with respect to at least two distinct lines ℓ and ℓ'. Without loss of generality, let us assume $\overline{uv} \not\subset \ell$. Let $(p, q) \in E$ be such that $(p, q) \subset \ell$. Then, (u, v) and (p, q) intersect contradicting Lemma 4. □

LEMMA 10. *There are atmost $O(n)$ eligible edges generated by our procedure.*

PROOF. First, we show that at least one of the end points of every eligible edge generated by our procedure is an external point. Suppose $a_i \in A^j$ is an internal point. Let us assume there is an edge, (a_i, b_h), such that b_h is internal. By construction, $b_h \in B \setminus B^j$ and therefore, b_h, a_h and a_i are not collinear. This implies that b_h is an external point contradicting our assumption.

Next, we bound the total number of eligible edges with at least one external end point. For any external point a_i, we first show that the total number of eligible edges (a_i, b_h) where b_h is an internal point is $O(n)$. We claim that b_h can have at most two eligible edges to external points. For the sake of contradiction, let b_h have eligible edges to $a_i, a_{i'}, a_{i''}$ where $a_i, a_{i'}, a_{i''}$ are external points. b_h is an internal point and hence $a_i, a_{i'}, a_{i''}$ and b_h are collinear. Therefore, at least one point, say $a_{i'}$, is contained in (b_h, a_i) or $(b_h, a_{i''})$. $a_{i'}$ is an external point contained in the interior of an eligible edge contradicting Lemma 9. Therefore, there are at most $2n$ eligible edges with exactly one external end-point.

Next, we bound the number of eligible edges with two external end-points. Let (a_i, b_h) and $(a_{i'}, b_{h'})$ be two such eligible edges where $a_i, a_{i'}, b_h$ and $b_{h'}$ are external points. From Lemma 4, (a_i, b_h) and $(a_{i'}, b_{h'})$ do not intersect. From Lemma 9, (a_i, b_h) and $(a_{i'}, b_{h'})$ do not overlap. Therefore, the set of eligible edges with two external end-points induce a planar straight-line graph. The number of eligible edges with two external end points is $O(n)$. □

For each eligible edge, our algorithm adds one point-line pair in $\mathcal{P}$ resulting in the following corollary.

COROLLARY 1. $|\mathcal{P}| = O(n)$.

Let $L = \{\ell \mid u \in A \cup B, (u, \ell) \in \mathcal{P}\}$. For any line $\ell \in L$, let $E_\ell \subseteq E$ be the set of eligible edges whose both end-points are on ℓ. Lemma 11 shows that the set of E_ℓ decompose the eligible set of edges.

LEMMA 11. $\bigcup_{\ell \in L} E_\ell = E$.

PROOF. Suppose (u, v) is an eligible edge and $(u, v) \subset \ell$. If both u and v are internal, our procedure adds (u, ℓ) to $\mathcal{P}$. If either u or v, say u, is external. Our procedure computes all eligible edges incident on u, including (u, v), and adds the point-line pair – (u, ℓ) in the case when (u, v) is the eligible edge – to $\mathcal{P}$. In both cases, $\ell \in L$ and therefore, $(u, v) \in \bigcup_{\ell \in L} E_\ell$. □

Using a straight-forward algorithm, we compute the decomposition of M in $O(n \log n)$ time. We maintain a dynamic weighted nearest-neighbor data structure on points of B with every point $v \in B$ having a weight $\tilde{y}(v)$. While processing points of A^j, we first delete B^j from the data structure. We compute (12) by querying for the weighted nearest neighbor of a_i in the data structure. If (b_r, a_i) is eligible, we delete b_r from the data structure and repeat the procedure. Otherwise, we terminate the procedure for a_i. We repeat the procedure for all points $a_i \in A^j$. Finally, when all points of A^j are processed, we add points of B^j to the data structure. Let η_{ij} be the number of eligible edges incident on a_i. The procedure for a_i executes $\eta_{ij} + 1$ queries and updates. Let $\sum_{a_i \in A^j} \eta_{ij} = \eta_j$ be the total number of eligible edges between A^j and $B \setminus B^j$. For the set A^j, there are at most $O(\eta_j + |A^j| + |B^j|)$ queries and updates to the dynamic weighted nearest neighbor data structure by our procedure. Each query and update takes $O(n^\delta)$ time [1]. Therefore the total time taken over all A^j is $O(\sum_j (\eta_j) + |A| + |B|)$. Since, from Lemma 10, $\sum_j \eta_j = O(n)$, the total time taken by our procedure is $O(n^{1+\delta})$.

Algorithm to compute P. Using $\mathcal{P}$, we present a procedure to compute an eligible edge decomposition P. For any line $\ell \in L$, let Let $O_\ell = \{e \mid e$ is external point and $(e, \ell) \in \mathcal{P}\}$ be the set of external points that lie on line ℓ; We index points in $O_\ell = \{e_1, \ldots, e_t\}$ in the order in which they appear as we move along the direction assigned to ℓ. Let A_ℓ and B_ℓ be the internal points of A and B that lie on line ℓ.

Initially $P = \emptyset$. For line $\ell \in L$, let $\{A_\ell^1, \ldots, A_\ell^{(t+1)}\}$ (resp. $\{B_\ell^1, \ldots, B_\ell^{(t+1)}\}$) be a partition of A_ℓ (resp. B_ℓ)– every point in A_ℓ^j (resp. B_ℓ^j) is an internal point of A (resp. B) that lies on the line segment $\overline{e_{j-1}e_j}$ (A_ℓ^1 and B_ℓ^1 lie between $x = -\infty$ and e_1; A_ℓ^{t+1} and B_ℓ^{t+1} lie between e_t and $x = \infty$). Let E_ℓ^j be the set of eligible edges whose end-points are in $A_\ell^j \cup B_\ell^j \cup \{e_j, e_{j+1}\}$.

Consider

$$F_\ell^i = \bigcup_{(a,b) \in E_\ell^i} \overline{ab} \tag{13}$$

Suppose F_ℓ^i is a set of r disjoint intervals whose end points are $\{(p_1, q_1) \ldots (p_r, q_r)\}$. We set $P \leftarrow P \cup \{(p_1, q_1), \ldots, (p_r, q_r)\}$. We repeat this for every i and every line $\ell \in L$. This completes the description of our algorithm for computing an eligible edge decomposition P. For every interval (p_i, q_i) in P, we can generate the corresponding A_i, B_i by simply scanning $\mathcal{P}$.

The only non-trivial step in our procedure is computing (13). Consider a weighted bipartite graph $\mathrm{G}(A_\ell^i, B_\ell^i)$, where the cost of an edge (a, b) is $\|ab\| - \tilde{y}(a) - \tilde{y}(b)$. We compute the minimum spanning tree T of $\mathrm{G}(A_\ell^i, B_\ell^i)$. Next, we remove all edges of T that are ineligible. The resulting graph is a forest $F = \{T_1, \ldots, T_s\}$. We merge two trees in $T_i, T_j \in F$, if the intervals corresponding to the union of the edges in T_i and T_j overlap. This set of intervals – one each for every tree in F – corresponds to the set of intervals of (13).

Minimum spanning tree of a complete bipartite graph on collinear points takes $O(n \log n)$ time. Detecting and merging trees whose boundaries overlap can be done using dynamic binary search tree data structure in a straight-forward way in $O(n \log n)$ time.

Proof of (P1): For any eligible edge $(u, v) \in E$, from Lemma 11, there is a line $\ell \in L$, such that $(u, v) \in E_\ell$. We claim that there is an i such that $(u, v) \in E_\ell^i$. For the sake of contradiction, let us assume there is an edge $(u, v) \in E_\ell$, such that for some i, $u \in A_\ell^i \cup B_\ell^i$ and $v \notin A_\ell^i \cup B_\ell^i \cup \{e_i, e_{i+1}\}$. Since all edges of E_ℓ are collinear by construction, there is an external point, either e_i or e_{i+1} that lies between u and v. This contradicts Lemma 9. Therefore, $u, v \in E_\ell^i$. By construction, u, v are in the same interval of F_ℓ^i, say (p_j, q_j). Since $(p_j, q_j) \in P$, and $\overline{uv} \subseteq \overline{p_j q_j}$, we have $(u, v) \in E_i$. This is true for every edge $(u, v) \in E$ and therefore $\bigcup_i E_i = E$.

Proof of (P2): Suppose there are two pairs $(p, q), (p', q') \in P$ such that (p, q) intersects with (p', q'). Let e be the point of intersection and let $e \in \text{int}(p_i, q_i)$. Suppose $e \notin A \cup B$, then from Lemma 12, we have $e \in \text{int}(\overline{uv}) \cup \text{int}(\overline{u'v'})$ for $(u, v), (u', v') \in E$ and $\overline{uv} \subseteq \overline{pq}$ and $\overline{u'v'} \subseteq \overline{p'q'}$. (u, v) and (u', v') intersect contradicting Lemma 4. Suppose $e \in A \cup B$ and $e \notin \text{int}(a, b)$ for all $(a, b) \in E$. Then, by construction, e is an external point – there is one eligible edge (e, v) such that $\overline{ev} \subseteq \overline{pq}$ and one eligible edge (e, v') such that $\overline{ev'} \subseteq \overline{p'q'}$. By construction, every point on the line segment joining p and q are internal contradicting e being an external point.

Now we argue that (p, q) does not overlap with (p', q'). Without loss of generality, let us assume that $q \in \overline{p'q'}$. If q is an external point, then P will contain (p', q) and (q, q') and not (p', q') reaching a contradiction. Therefore q should be an internal point. Let p, q, p' and q' lie between the same two external points, say e_i, e_{i+1}. Since the algorithm that constructs P combines all edges of E_ℓ^i that overlap, we will have p, q, p' and q' belong to the same interval, say $(p'', q'') \in P$. Thus, we reach a contradiction of our assumption that (p, q) (p', q') are in P.

LEMMA 12. *Let P be an eligible edge decomposition generated by our algorithm. For any $(p_i, q_i) \in P$, suppose $e \in \overline{p_i q_i}$ then there is an eligible edge $(u, v) \in E_i$ such that $e \in \overline{uv}$.*

PROOF. By construction, each (p_i, q_i) is an interval in the union of eligible edges of E_ℓ^j for some j, ℓ. Therefore, there is an eligible edge $(u, v) \in E$ such that $e \in \overline{uv}$. □

5. CONCLUSION

We present an algorithm to compute a minimum-cost Euclidean bipartite matching of $A, B \subseteq [\Delta]^2$ in $O(n^{3/2+\delta} \log(n\Delta))$ time. First, using the algorithm of [8], our algorithm computes an approximately optimal matching. This is done by optimally solving an approximate linear program. Using the geometric properties of the dual weights computed for this approximate linear program, our algorithm computes a small candidate set $\mathcal{E}$ such that there is a minimum-cost matching $M^* \subseteq \mathcal{E}$ and $\mathrm{G}(A \cup B, \mathcal{E})$ is a planar graph. Then, using the algorithm presented in [6], our algorithm computes an optimal matching in $\mathrm{G}(A \cup B, \mathcal{E})$.

We conclude by stating the following open problems.

- Is there a sub-quadratic algorithm to compute a minimum-cost Euclidean bipartite matching of sets $A, B \subset \mathbb{R}^2$?
- Is there an algorithm to compute minimum-cost Euclidean bipartite matching of point sets $A, B \subseteq [\Delta]^d$ in $O(dn^{2.5} \log(n\Delta))$ time?

6. REFERENCES

[1] P. K. Agarwal, A. Efrat, and M. Sharir, Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications, *SIAM J. Comput.*, 29 (1999), 39–50.

[2] P. K. Agarwal and K. R. Varadarajan, A near-linear constant-factor approximation for euclidean bipartite matching?, *Proc. 12th Annual Sympos. Comput. Geom.*, 2004, pp. 247–252.

[3] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra, A strong and easily computable separation bound for arithmetic expressions involving radicals, *Algorithmica*, 27 (2000), 87–99.

[4] H. N. Gabow and R. Tarjan, Faster scaling algorithms for network problems, *SIAM J. Comput.*, 18 (1989), 1013–1036.

[5] P. Indyk, A near linear time constant factor approximation for euclidean bichromatic matching (cost), *Proc. 18th Annual ACM/SIAM Sympos. on Discrete Algorithms*, 2007, pp. 39–42.

[6] R. J. Lipton and R. E. Tarjan, Applications of a planar separator theorem, *SIAM J. Comput.*, 9 (1980), 615–627.

[7] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Inc., 1982.

[8] R. Sharathkumar and P. K. Agarwal, Algorithms for transportation problem in geometric settings, *Proc. 23rd Annual ACM/SIAM Sympos. on Discrete Algorithms*, 2012, pp. 306–317.

[9] R. Sharathkumar and P. K. Agarwal, A near-linear time approximation algorithm for geometric bipartite matching., *Proc. 44th Annual ACM Annual Sympos. on Theory of Comput.*, 2012, pp. 385–394.

[10] P. M. Vaidya, Geometry helps in matching, *SIAM J. Comput.*, 18 (1989), 1201–1225.

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