Parallel Complexity of Geometric Bipartite Matching

Sujoy Bhore
Department of Computer Science & Engineering, Indian Institute of Technology Bombay, India

Sarfaraz Equbal
Department of Computer Science & Engineering, Indian Institute of Technology Bombay, India

Rohit Gurjar
Department of Computer Science & Engineering, Indian Institute of Technology Bombay, India

Abstract

In this work, we study the parallel complexity of the geometric minimum-weight bipartite perfect matching (GWBPM) problem in $\mathbb{R}^2$. Here our graph is the complete bipartite graph G on two sets of points A and B in $\mathbb{R}^2$ ($|A| = |B| = n$) and the weight of each edge $(a, b) \in A \times B$ is the ℓ_p distance (for some integer $p \geq 2$) between the corresponding points, i.e., $||a - b||_p$. The objective is to find a minimum weight perfect matching of $A \cup B$. In their seminal work, Mulmuley, Vazirani, and Vazirani (STOC 1987) showed that the weighted perfect matching problem on general bipartite graphs is in RNC. Almost three decades later, Fenner, Gurjar, and Thierauf (STOC 2016) showed that the problem is in Quasi-NC. Both of these results work only when the weights are of $O(\log n)$ bits. It is a long-standing open question to show the problem to be in NC.

First, we show that in a geometric bipartite graph under the ℓ_p metric for any $p \geq 2$, unless we take $\Omega(n)$ bits of approximation for weights, we cannot distinguish the minimum-weight perfect matching from other perfect matchings. This means that we cannot hope for an MVV-like NC/RNC algorithm for solving GWBPM exactly (even when vertex coordinates are small integers).

Next, we give an NC algorithm (assuming vertex coordinates are small integers) that solves GWBPM up to $1/\text{poly}(n)$ additive error, under the l_p metric for any p ≥ 2.

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization; Theory of computation → Computational geometry

Keywords and phrases Parallel algorithms, Geometric matching, Derandomization, Isolation Lemma

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.12

Funding *Rohit Gurjar*: supported by SERB MATRICS grant MTR/2022/001009.

1 Introduction

The perfect matching problem is one of the well-studied problems in Complexity theory, especially, in the context of derandomization and parallelization. Given a graph $G = (V, E)$, the problem asks, whether the graph contains a matching that matches every vertex of G. Due to Edmonds [12], the problem is known to be solvable in polynomial time. However, the parallel complexity of the problem has not been completely resolved till today. In 1979, Lovász [19] showed that perfect matching can be solved by efficient randomized parallel algorithms, i.e., the problem is in RNC. Hence, the main question, with respect to its parallel complexity, is whether this randomness is necessary, i.e., whether the problem is in NC[1].

[1] The class NC represents the problems that have efficient parallel algorithms, i.e., they have uniform circuits of polynomial size and polylog depth

44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024).
Editors: Siddharth Barman and Sławomir Lasota; Article No. 12; pp. 12:1–12:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The search version of the problem asks to explicitly construct a perfect matching in a graph if one exists. Note that in the parallel setting, there is no obvious reduction from search to decision. This version is also known to be in RNC [18, 21]. The Mulmuley-Vazirani-Vazirani (MVV) algorithm [21], in fact, also works for the weighted version of the problem, where there is a polynomially bounded weight assignment given on the edges of the graph.

The MVV algorithm [21] introduced the celebrated *Isolation lemma*. A weight assignment is called *isolating* for a graph G, if the minimum weight perfect matching in G is unique if one exists. Mulmuley, Vazirani, and Vazirani [21] showed that given an isolating weight assignment with polynomially bounded integer weights for a graph G, a perfect matching in G can be constructed in NC. The only place where they use randomization is to get an isolating weight assignment. Their Isolation lemma states that a random weight assignment is isolating!

Derandomizing the Isolation lemma means to construct such a weight assignment deterministically in NC. A line of work derandomized the Isolation Lemma for special families of graphs, e.g., planar bipartite graphs [11, 27], strongly chordal graphs [10], graphs with a small number of perfect matchings [16]. In 2016, Fenner, Gurjar, and Thierauf [14] showed that the bipartite perfect matching problem is in quasi-NC, by an almost complete derandomization of the Isolation Lemma. Later, Svensson and Tarnawski [26] showed that the problem in general graphs is also in Quasi-NC. Subsequently, Anari and Vazirani [5] gave an NC algorithm for finding a perfect matching in general planar graphs. All of these algorithms work for the weighted version (poly-bounded) of the problem as well.

What remains a challenging open question is to find an NC algorithm for any versions (decision/search/weighted) of the perfect matching problem, even for bipartite graphs. Inspired by the positive results on planar bipartite graphs, we investigate the weighted version of the perfect matching problem in the geometric setting (2 dimensional).

Geometric Bipartite Matching

Let A and B be two point sets in $\mathbb{R}^2$ of size n each. Consider the **complete** bipartite graph $G(A, B, E)$ with the following cost function on the edges: for any edge $e = (a, b)$, define $\mathcal{C}(e) = ||a - b||_p$, where $|| \cdot ||_p$ denotes the ℓ_p norm for some integer $p \geq 2$. In other words, we consider the ℓ_p distance between the endpoints as the cost of an edge. The cost of a perfect matching M is the sum of its edge costs $\mathcal{C}(M) = \Sigma_{e \in M} \mathcal{C}(e)$. The Geometric Minimum-Weight Bipartite Perfect Matching (GWBPM) problem is to find $M_{opt} = \text{argmin}_{|M|=n} \mathcal{C}(M)$, that is, the optimal perfect matching with respect to function $\mathcal{C}$. GWBPM is a fundamental problem in Computational Geometry and has been studied extensively over the years. See Section 1.2 for an overview of the results. In this work, we focus on the parallel complexity of the GWBPM problem, and ask the following question –

▶ **Question 1.** *Is GWBPM in NC?*

Optimization problems in computational geometry are usually studied in real arithmetic computational model, where comparing two distances or sums of distances is assumed to be a unit cost operation. However, in the bit complexity model, it is not clear if distances, which can be irrational numbers (under ℓ_p metric for $p \geq 2$), can be efficiently added or compared. In fact, the problem of comparing two sums of square roots (or other pth roots) is not known to be in P (see, for example, [22, 1]). See [13] for some recent progress on the sum of square roots problem.

Our path towards showing an NC algorithm for GWBPM naturally goes via the MVV algorithm. Recall that the MVV algorithm works only when the given weights/costs are polynomially bounded integers, because in intermediate steps, it needs to put weights in the exponent. Hence, inevitably we need to consider the bit complexity of the weights. Note that there are other parallel algorithms for the weighted perfect matching problem (e.g., [15]), but there too it is important that the weights are polynomially bounded integers.

It is not clear if the GWBPM problem (for $p \geq 2$) is in P (or even in NP) in the bit-complexity model. To the best of our knowledge, the existing algorithms for GWBPM require comparisons between two sums of pth roots. For $p \geq 2$, this naturally leads us to consider an approximate version of the problem. Let us define the δ-GWBPM problem, which asks for a perfect matching whose weight is at most δ more than the minimum-weight perfect matching. We aim to get an NC algorithm for the problem whenever $1/\delta$ is $\text{poly}(n)$.

1.1 Our Contribution

In this work, we study the parallel complexity of δ-GWBPM problem. First, it is natural to ask whether solving δ-GWBPM for some $\delta = 1/\operatorname{poly}(n)$ will already solve the GWBPM problem. In other words, by considering only $O(\log n)$ bit approximations of ℓ_p distances, can we hope to find the geometric minimum weight perfect matching? Our first result rules out this possibility. We show that for GWBPM (under ℓ_p metric for any $p \geq 2$), a super-linear number of bit approximations is required to distinguish the minimum-weight perfect matching from others.

▶ **Theorem 1.1.** *There is a set of $2n$ points in the $O(n^7) \times O(n^7)$ integer grid such that in the corresponding complete bipartite graph, the difference between the weights of the minimum weight perfect matching and another perfect matching is at most $1/(n-1)!$ (under any ℓ_p metric with $p \geq 2$).*

This theorem is proved in Section 2. The first part of the proof goes via a known counting technique [8], where we construct a geometric bipartite graph and argue that there must be two perfect matchings whose weights are distinct but very close. In the second part of the proof, we construct another geometric bipartite graph based on these two matchings, where one of the two is the minimum weight perfect matching.

Next, we come to our positive result. We affirmatively answer Question 1, by showing that the geometric minimum weight perfect matching problem that allows up to $\frac{1}{\text{poly}(n)}$ error, is in NC.

▶ **Theorem 1.2.** *The δ-GWBPM under ℓ_p metric ($p \geq 2$) is in NC, assuming the points are on a polynomially bounded integer grid, where δ is $1/\operatorname{poly}(n)$.*

This theorem is proved in Section 3. The main idea is to reduce the problem to bipartite planar matching and then use known techniques for the planar case [27].

1.2 Related Work

The classical Hopcroft-Karp algorithm computes a maximum-cardinality matching in a bipartite graph with n vertices and m edges in $O(m\sqrt{n})$ time [17]. After almost three decades, Madry [20] improved the running time to $O(m^{10/7} \operatorname{polylog} n)$ time, which was further improved to $O(m + n^{3/2} \operatorname{polylog} n)$ by Brand et al. [29]. The Hungarian algorithm computes the minimum-weight maximum cardinality matching in $O(mn + n^2 \log n)$ time [23]. In some recent breakthrough results [28, 9], they have shown that maximum-cardinality matching in bipartite graphs can be solved in near-linear time.

For two sets of points A and B in $\mathbb{R}^2$, the best known algorithm for computing GWBPM runs in $O(n^2 \operatorname{polylog} n)$ time [3, 4]. Moreover, if points have integer coordinates bounded by Δ, the running time can be improved to $O(n^{3/2} \operatorname{polylog} n \log \Delta)$ [24]. If coordinates of input points have real values, it is not known whether a subquadratic algorithm exists. However, for the non-bipartite case, Varadarajan [30] presented an $O(n^{3/2} \operatorname{polylog} n)$-time algorithm under any ℓ_p-norm. For bipartite matching, a large body of literature focused on obtaining approximate matching for points in $\mathbb{R}^d$. Varadarajan and Agarwal [31] presented an $O(n^{3/2}\varepsilon^{-d} \log^d n)$-time ε-approximation algorithm for geometric matching in $\mathbb{R}^d$. Later, Agarwal and Raghavendra [25] improved the running time. Recently, Agarwal et al. [2] presented a deterministic algorithm with running time $n \cdot (\varepsilon^{-1} \log n)^{O(d)}$ time, and computes a perfect matching whose cost is within a $(1 + \varepsilon)$ factor of the optimal matching under any ℓ_p-norm.

2 Lower Bound

In this section, we want to show that for a geometric bipartite graph with $n + n$ vertices, we need at least $\Omega(n \log n)$ bits of precision to distinguish the minimum weight perfect matching from others (under ℓ_p metric for any integer $p \geq 2$). We will show this by constructing a bipartite set of $2n$ points in the integer grid of size $O(n^7) \times O(n^7)$ such that the difference between the weights of the minimum weight perfect matching and the one with the next higher weight will be $1/(n-1)!$. Towards this, the first step is to construct a geometric graph where there are two perfect matchings whose weights differ by at most $1/(n-1)!$ (Claim 2.2). Here we use an argument based on the pigeonhole principle. A similar argument was used to show such a bound on the difference of two sums of square roots [8].

In the above construction, it is not necessary that one of the two perfect matchings is of minimum weight. In the second step, we show that the above geometric graph can be modified to construct another one where the same two perfect matchings appear, but now one of them is of minimum weight (Claim 2.4).

► **Construction 1.** *Consider the left hand side vertices $u_0, u_1, \ldots, u_{n-1}$ at points*

$$\{(0,0), (0,1), (0,2), \ldots, (0, n-1)\}.$$

Similarly, consider the right hand side vertices $v_0, v_1, \ldots, v_{n-1}$ at points

$$\{(q, n), (q, 2n), (q, 3n), \ldots, (q, n^2)\},$$

where $q = n^6$.

▷ Claim 2.1. The geometric bipartite graph in Construction 1 has all its perfect matchings with distinct weights (under ℓ_p metric for any integer $p > 1$).

Proof. Recall that edge weights are pth roots of integers. We will argue that the edge weights are linearly independent over rationals, which immediately implies that any two different subsets of edges cannot have equal weights. It is known that to show linear independence of a set of pth roots of integers, it suffices to show that they are *pairwise* linearly independent (see, for example, [7]). So, now we just argue that the edge weights are pairwise linearly independent.

First we observe that none of the edge weights is an integer. This is because from our construction, we have $n^6 < \sqrt[p]{n^{6p}+1} < w(e) \leq \sqrt[p]{n^{6p}+n^{2p}} < n^6+1$.

For the sake of contradiction, suppose we have two edges e and e', whose weights are linearly dependent. Then we have $aw(e) = bw(e')$ for some integers a and b. From here we get that $w(e)^p w(e')^p = (a/b)^p w(e)^{2p}$. That is, the product $w(e)^p w(e')^p$ is pth power of a rational number. Since it is an integer, it must be pth power of an integer. From our construction, for any edge e, we have $q < w(e) \leq \sqrt[p]{q^p + n^{2p}}$. Moreover, note that only one of the edges e or e' can match the upper bound. Hence,

$$(q^2)^p < w(e)^p w(e')^p < (q^p + n^{2p})^2. \tag{1}$$

Now, we consider two cases $p \geq 3$ and $p = 2$.

Case I ($p \geq 3$). As $w(e)^p w(e')^p$ is pth power of an integer, from Equation (1) we have

$$(q^2+1)^p \leq w(e)^p w(e')^p < (q^p + n^{2p})^2.$$

Comparing the first and the last terms, we get

$$pq^{2p-2} + \binom{p}{2} q^{2p-4} + \cdots + 1 < 2q^p n^{2p} + n^{4p}.$$

Putting $q = n^6$, we see that the above inequality is false. Hence, we get a contradiction.

Case II ($p = 2$). From Equation 1, we have

$$q^4 < w(e)^2 w(e')^2 < (q^2 + n^4)^2.$$

Since $w(e)^2 w(e')^2$ is square of an integer, we can write $w(e)^p w(e')^p = (q^2+\alpha)^p$ for some integer $0 < \alpha < n^4$.

For any edge e, let us denote by Δ_e, the difference in the y coordinates of the two endpoints of the edge. Then, the weight of an edge e can be written as $w(e) = \sqrt{q^2 + \Delta_e^2}$. Now, we have

$$w(e)^2 w(e')^2 = (q^2 + \Delta_e^2)(q^2 + \Delta_{e'}^2) = (q^2+\alpha)^2.$$

Equivalently,

$$q^2(\Delta_e^2 + \Delta_{e'}^2) + \Delta_e^2 \Delta_{e'}^2 = 2q^2\alpha + \alpha^2.$$

Observe that $\Delta_e^2 \Delta_{e'}^2 \leq n^8 < q^2$ (from construction) and also $\alpha^2 < n^8 < q^2$. Hence, we conclude from above that

$$\Delta_e^2 + \Delta_{e'}^2 = 2\alpha \text{ and } \Delta_e^2 \Delta_{e'}^2 = \alpha^2.$$

This implies that $\Delta_e = \Delta_{e'}$.

Now, we will argue that for any two distinct edges, we have $\Delta_e \neq \Delta_{e'}$, which will give us a contradiction. Indeed for the edge (u_i, v_j), we have $\Delta_e = jn - i$, which comes from a unique choice of $0 \leq i \leq n-1$ and $1 \leq j \leq n$. ◁

▷ Claim 2.2. In the geometric bipartite graph from Construction 1, there are two perfect matchings whose weights are different and differ by at most $1/(n-1)!$.

Proof. From Claim 2.1, all $n!$ perfect matchings have distinct weights. From the construction, any perfect matching has its weight between n^7 and $n\sqrt[p]{n^{6p}+n^{2p}} \leq n(n^6+1)$. The bound follows from the pigeonhole principle. ◁

Now, consider the two perfect matchings from Claim 2.2, say M_1 and M_2, whose weights differ by at most $1/(n-1)!$. Let M_1 be the one with a smaller weight. The union of two perfect matchings $M_1 \cup M_2$ is a set of vertex-disjoint cycles and edges. We are going to ignore the common edges between M_1 and M_2. Let $(e_1, e_2, \ldots, e_{2\ell})$ be the sequence of edges generated from the cycles in $M_1 \cup M_2$ as follows: arrange the cycles in an arbitrary order. For each cycle, start from that edge in M_1 which has its left endpoint with minimum y-coordinate, and traverse along the cycle till we hit the starting vertex. Note that the sequence $(e_1, e_2, \ldots, e_{2\ell})$ has edges alternating from M_1 and M_2. The new graph will be constructed by "unrolling" these cycles. The construction will be such that edges outside these cycles will be long, and hence, will not be a part of any minimum weight perfect matching. Recall that for any edge $e = (u_i, v_j)$ (Construction 1), we denote by Δ_e the difference in the y-coordinates of the endpoints, i.e., $jn - i$.

▶ **Construction 2.** *Consider the vertex t_0 at $(0,0)$. Let $y_0 = 0$. For $1 \leq k \leq 2\ell$, we place the vertex t_k at (kq, y_k), where*
- $y_k = y_{k-1} + \Delta_{e_k}$ *if k is odd*
- $y_k = y_{k-1} - \Delta_{e_k}$ *if k is even*

We add three more vertices: s_0 at $(0, -2\ell q)$, s_1 at $(\ell q, -2\ell q)$, and s_2 at $(2\ell q, -2\ell q)$. See Figure 1.

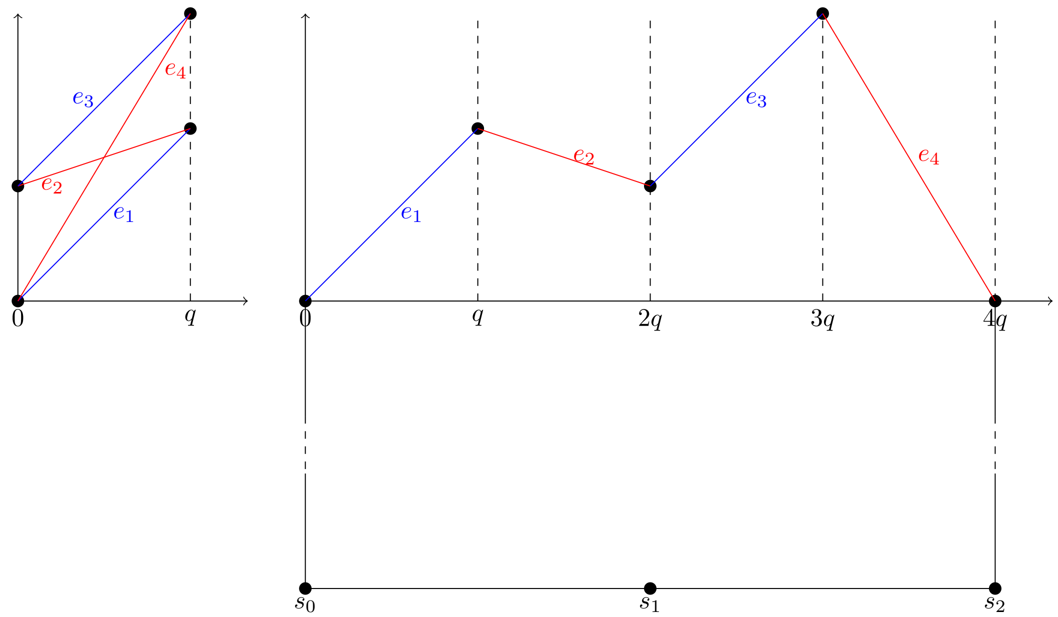

Figure 1 The left-hand side figure shows a cycle in the union of two perfect matchings. The right-hand side figure shows how we "unroll" this cycle.

Corresponding to perfect matchings M_1 and M_2, here we will have perfect matchings M_1' and M_2' as

$$M_1' = \{e_1, e_3, \ldots, e_{2\ell-1}, (t_{2\ell}, s_2), (s_0, s_1)\}$$

$$M_2' = \{e_2, e_4, \ldots, e_{2\ell}, (t_0, s_0), (s_1, s_2)\}.$$

The following are easy observations about Construction 2.

1. The edge lengths of $e_1, e_2, \dots, e_{2\ell}$ are exactly the same as their lengths in Construction 1.
2. $y_k \geq 0$ for each $1 \leq k \leq 2\ell$, because for each cycle, the cycle traversal starts from the lowest y coordinate on the left. Moreover, $y_{2\ell}$ must be zero, because any cycle traversal ends at the starting vertex.
3. Any pair of vertices are at least distance q apart.
4. $w(M_1') = w(M_1) + 3\ell q$ and $w(M_2') = w(M_2) + 3\ell q$.

▷ Claim 2.3. The minimum weight perfect matching in Construction 2 is M_1', with weight $w(M_1) + 3\ell q$.

Proof. Recall that weight of any edge e_k is at most $\sqrt[p]{q^p + n^{2p}} = \sqrt[p]{n^{6p} + n^{2p}} < n^6 + 1/(pn^{4p-6}) < q + 1/(pn^{4p-6})$. Hence, $w(M_1') \leq \ell(q + 1/(pn^{4p-6})) + 3\ell q = \ell(4q + 1/(pn^{4p-6}))$. We have already assumed that M_2' has weight higher than M_1'. Now, consider any perfect matching M other than M_1' and M_2'. We will consider different cases and argue that in each case M has a larger weight.

- If M matches s_1 with one of the t_k vertices, the weight of that edge will be at least $2\ell q$. The vertices s_0 and s_2 will either match with each other or to some t_k vertices. In either case, they will contribute at least $2\ell q$ to the weight. The remaining vertices must have at least $\ell - 3$ edges, each with weight at least q. Hence, the total weight will be at least $5\ell q - 3q$, which is larger than $w(M_1')$.
- Consider the case when M has (s_1, s_2) (the other case is similar) and s_0 is matched with one of the t_k vertices, other than t_0. Recall that $y_k \geq 0$ and $s_0 = (0, -2\ell q)$. Then the weight of (s_0, t_k) (for $k > 0$) is at least $\sqrt[p]{(2\ell q)^p + q^p} \geq 2\ell q + 2\ell q/(4\ell)^p$. The remaining 2ℓ vertices will have ℓ matching edges, each with weight at least q. Hence, the weight of the matching M will be at least $\ell q + 2\ell q + 2\ell q/(4\ell)^p + q\ell$. This is larger than $w(M_1') \leq 4\ell q + \ell/(pn^{4p-6})$ (as $q = n^6$ and $\ell \leq n$).
- Consider the case when M has (s_1, s_2) and (s_0, t_0). These two edges will add up to weight $3\ell q$. Since the matching M is different from M_1' and M_2', it must match a vertex t_k with another vertex t_j such that $j \neq \{k-1, k+1\}$. Then $|j - k|$ must be at least 3, because the graph is bipartite. The edge (t_k, t_j) will have weight at least $3q$. The other $\ell - 1$ edges will have weight at least q. Hence, the total weight is at least $4\ell q + 2q$, which is again larger than $w(M_1')$.
- The other cases when M has (s_0, s_1) matched are similar to the above two cases. ◁

Now, we finally come to our main claim.

▷ Claim 2.4. In the geometric bipartite graph from Construction 2, the difference between the minimum weight perfect matching and the perfect matching with the next higher weight is at most $1/(n-1)!$.

Proof. From Claim 2.3, we know that M_1' is the minimum weight perfect matching. We had observed that $w(M_1') - w(M_2') = w(M_1) - w(M_2)$. From Claim 2.2, this difference is at most $1/(n-1)!$. ◁

3 Geometric Bipartite Matching

In this section, we study the parallel complexity of δ-GWBPM (under ℓ_p metric for $p \geq 2$), and show that the problem lies in the class NC, for $\delta = 1/\operatorname{poly}(n)$.

First of all, we assume that no three vertex points are colinear. There is a simple fix to break colinearity by way of small perturbations in coordinates. Specifically, for the ith vertex at point (x_i, y_i), let us assign its new coordinates to be $(x_i + i/K, y_i + i^2/K)$, where

K is a large enough number. This specific perturbation guarantees that no three points are colinear. To see this, consider ith, jth, and kth vertices after the perturbation. They will be colinear if and only if the following matrix has zero determinant.

$$\begin{pmatrix} 1 & 1 & 1 \\ x_i + i/K & x_j + j/K & x_k + k/K \\ y_i + i^2/K & y_j + j^2/K & y_k + k^2/K \end{pmatrix}$$

Consider the coefficient of the term $1/K^2$ in the determinant, which is $(i-j)(j-k)(k-i) \neq 0$. Other terms in the determinant will be an integer multiple of $1/K$ and hence, cannot cancel this term, when K is large enough ($\text{poly}(n)$). This perturbation can cause additive error in the weights of perfect matchings, but the error will remain bounded by $O(n^3/K)$. Thus, the minimum weight perfect matching with respect to perturbed coordinates will be an GWBPM up to a $1/\text{poly}(n)$ additive error. To make the coordinate integral, we can multiply them by K. Now, give a brief overview of our ideas.

Our main idea is to design an isolating weight assignment for the given graph and then use the MVV algorithm. Let G be a complete bipartite graph of two sets of points A and B in $\mathbb{R}^2$. The MVV theorem asserts that if a graph has an isolating weight assignment, then the task of finding the minimum weight perfect matching in G can be accomplished in NC.

To construct an isolating weight assignment, we adopt the weight scheme introduced by Tewari and Vinodchandran [27], which was designed specifically for planar bipartite graphs. However, note that our graph is a complete bipartite graph and hence, far from planar. Our first key observation is that the union of minimum weight perfect matchings (with respect to ℓ_p distances or even approximate ℓ_p distances) forms a planar subgraph. Then one can hope to use the Tewari and Vinodchandran [27] weight scheme on this planar subgraph. However, we cannot compute this planar subgraph (i.e., the union of minimum weight perfect matchings). What proves to be useful is the fact that the Tewari-Vinodchandran weight scheme is black-box, i.e., it does not care what the underlying planar graph is, it only needs to know the points in the plane where vertices are situated. Finally, we combine the approximate distance function with the Tewari-Vinodchandran weight function on a smaller scale and apply it to the complete bipartite graph. We show that this combined weight function is indeed isolating.

Towards showing the planarity of the union of minimum weight perfect matchings, first, we establish the simple fact that for any convex quadrilateral, there is a significant difference between the sum of diagonals and the sum of any opposite sides.

▶ **Lemma 3.1.** *Consider a convex quadrilateral formed by a quadruple in an integer grid of size $N \times N$. The sum of lengths of its diagonals is larger than the sum of any two opposite sides. And the gap between the two sums in ℓ_p metric is at least $\frac{1}{4\sqrt[p]{2}N^{p^4-1}}$ ($p \geq 2$).*

Proof. Intuitively, the sum of the diagonals will be larger than the sum of any two opposite sides because of triangle inequality (the diagonals combined with any two opposite edges form two triangles). The significance of this gap arises from the fact that if the points are from a grid and are not collinear, then the angle between any side and the diagonal cannot be arbitrarily small. Formally, let the four corners of the quadrilateral be A, B, C, D (in cyclic order). See Figure 2. Let O be the intersection point of the diagonals AC and BD (diagonals always intersect in a convex quadrilateral). By triangle inequality, we have $|AO| + |OB| \geq AB$ and $|CO| + |OD| \geq CD$. Adding the two we get,

$$|AC| + |BD| \geq |AB| + |CD|.$$

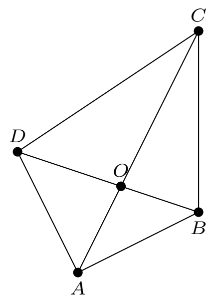

Figure 2 A convex quadrilateral with its two diagonals.

Now, we lower bound the gap. To calculate the lower bound of the gap we directly use the result of [6]. They gave an easy way to calculate the lower bound for an arithmetic expression over operators $+, -, *, /$ and $\sqrt[p]{}$, with integer operands. Our aim to lower bound the gap between, $|AC| + |BD|$ and $|AB| + |CD|$. Let us find the expression for the same where A, B, C, and D are the points on the $N \times N$ grid from $\mathbb{R}^2$. Let the co-ordinates of the points are $(i, j), (k, \ell), (m, n)$, and (o, p) respectively. Then the expression we want to lower bound is,

$$E = ||AC||_p + ||BD||_p - ||AB||_p - ||CD||_p.$$

$$E = \sqrt[p]{(m-i)^p + (n-j)^p} + \sqrt[p]{(o-k)^p + (p-l)^p} - \sqrt[p]{(k-i)^p + (l-j)^p} - \sqrt[p]{(o-m)^p + (p-n)^p}.$$

Our expression also uses only $+, -, *$ and $\sqrt[p]{}$ operators over the integer operands, we can use the Corollary 2 from [6]. It says that for any division-free expression E whose value ξ is nonzero, we have

$$(u(E)^{D(E)-1})^{-1} \leq |\xi| \leq u(E).$$

Here $u(E)$ represents the upper bound on the absolute value of E and $D(E)$ represents the product of indices of all the radicals involved in E. The detailed methodology for calculating $u(E)$ and $D(E)$ can be found in [6]. The values of $u(E)$ and $D(E)$ for our specific case turn out to be as follows:

$$u(E) = 4\sqrt[p]{2}N,$$

$$D(E) = p^4.$$

So the value of the expression E i.e ξ is bounded by,

$$\frac{1}{4\sqrt[p]{2}N^{p^4-1}} \leq |\xi| \leq 4\sqrt[p]{2}N$$

The statement of our Lemma 3.1 easily follows from this. ◀

3.1 Union of Near-Minimum Weight Perfect Matchings

In this subsection, we establish our main lemma that in a geometric bipartite graph G, the union of near-minimum weight perfect matchings forms a planar subgraph of G. This allows us to use the Tewari and Vinodchandran [27] isolating weight scheme for planar bipartite graphs. We first define a near-minimum weight perfect matching.

▶ **Definition 3.2.** *Let the vertices of the geometric bipartite graph lie in the $N \times N$ integer grid. A perfect matching is said to be of near-minimum weight under the ℓ_p metric if its weight is less than $w^* + 1/(8\sqrt[p]{2}N^{p^4-1})$, where w^* is the minimum weight of a perfect matching.*

▶ **Lemma 3.3.** *For a geometric bipartite graph G with vertices in the $N \times N$ integer grid, the union of near-minimum weight perfect matchings forms a planar graph (under ℓ_p metric for any $p \geq 2$).*

Proof. Let $A \cup B$ be the bipartition of the vertices. We will first show that no two edges in a near-minimum weight perfect matching M cross each other. For the sake of contradiction, let there be two edges $\{a_1, b_1\}$ and $\{a_2, b_2\}$ in M that cross each other, where $a_1, a_2 \in A$ and $b_1, b_2 \in B$. Since G is a complete bipartite graph, every vertex of set A must have an edge to every vertex of set B in G. We can construct another matching M' from M by replacing the crossing edges $\{a_1, b_1\}$ and $\{a_2, b_2\}$ with $\{a_1, b_2\}$ and $\{a_2, b_1\}$, respectively.

Note that (a_1, a_2, b_1, b_2) form a convex quadrilateral, since its diagonals a_1b_1 and a_2b_2 cross each other. From Lemma 3.1, we know that

$$|a_1b_2| + |a_2b_1| \leq |a_1b_1| + |a_2b_2| - 1/(4\sqrt[p]{2}N^{p^4-1}).$$

From here, we can conclude that $w(M') \leq w(M) - 1/(4\sqrt[p]{2}N^{p^4-1})$, where $w(M')$ and $w(M)$ are the weights of M' and M, respectively. This contradicts the fact that M is a near-minimum weight perfect matching.

Now, we will show that two edges belonging to two different near-minimum weight perfect matchings cannot cross. Consider two such near-minimum weight perfect matchings M_1 and M_2, where the edges $\{a_1, b_1\} \in M_1$ and $\{a_2, b_2\} \in M_2$ cross each other. Observe that the union of these two perfect matchings forms a set of vertex-disjoint cycles and a set of disjoint edges (which are common to both). There are two cases: (i) the edges $\{a_1, b_1\}$ and $\{a_2, b_2\}$ are part of one of these cycles and (ii) they are part of two different cycles. In each of the cases, we will create two new perfect matchings with significantly smaller weight, which will contradict the near-minimumness of M_1 and M_2.

Case (i): $\{a_1, b_1\}$ and $\{a_2, b_2\}$ are part of one cycle C. See Figure 3. Note that the edges of this cycle come alternatingly from M_1 and M_2 (shown in the figure in red and blue colors).

Figure 3 Construction of M_1' and M_2', when the crossing edges are part of one cycle.

Figure 4 Construction of M_1' and M_2', when the crossing edges are part of two different cycles.

We construct two distinct perfect matchings, M_1' and M_2', using M_1 and M_2. Removing the edges $\{a_1, b_1\}$ and $\{a_2, b_2\}$ from cycle C divides it into two parts. Note that both parts must have an even number of edges since the edges are alternating between M_1 and M_2. It follows that one of these parts is a path from a_1 to a_2, let us call it C_1. And the other one is a path from b_1 to b_2, let us call it C_2 (as shown in Figure 3).

Let us put $\{a_1, b_2\}$ into M_1' and $\{a_2, b_1\}$ into M_2'. For the edges in C_1, we put the M_1 edges into M_1' and the M_2 edges into M_2'. For the edges in C_2 we do the opposite, put the M_1 edges into M_2' and the M_2 edges into M_1'. For edges outside of the cycle C, we put edges from M_1 into M_1' and edges from M_2 into M_2'.

Case (ii): $\{a_1, b_1\}$ and $\{a_2, b_2\}$ are part of two different cycles. Let C_1 and C_2 be the paths obtained from removing $\{a_1, b_1\}$ and $\{a_2, b_2\}$ from the two cycles, respectively. See Figure 4. Here again we construct two distinct perfect matchings, M_1' and M_2', using a similar uncrossing of edges. Let us put both $\{a_1, b_2\}$ and $\{a_2, b_1\}$ into M_1'. For the edges in C_1, we put the M_1 edges into M_1' and the M_2 edges into M_2'. For the edges in C_2 we do the opposite, put the M_1 edges into M_2' and the M_2 edges into M_1'. For edges outside the two cycles, we put edges from M_1 into M_1' and edges from M_2 into M_2'.

Note that in both Case (i) and Case (ii), the newly constructed perfect matchings M_1' and M_2' together have the same edges as $M_1 \cup M_2$, except for $\{a_1, b_1\}$ and $\{a_2, b_2\}$ being replaced with $\{a_2, b_1\}$ and $\{a_1, b_2\}$.

Let w_1, w_2, w_1', w_2' be the weights of matchings M_1, M_2, M_1', M_2', respectively. Then,

$$w_1' + w_2' = w_1 + w_2 - |a_1b_1| - |a_2b_2| + |a_1b_2| + |a_2b_1|.$$

From Lemma 3.1, we have that

$$|a_1b_1| + |a_2b_2| - |a_1b_2| - |a_2b_1| \geq 1/(4\sqrt[p]{2}N^{p^4-1}).$$

Thus,

$$w_1' + w_2' \leq w_1 + w_2 - 1/(4\sqrt[p]{2}N^{p^4-1}).$$

Let w^* be the weight of the minimum weight perfect matching. Since M_1 and M_2 are of near-minimum weight, we have $w_1, w_2 < w^* + 1/(8\sqrt[p]{2}N^{p^4-1})$. Using this with the above inequality, we get $w_1' + w_2' < 2w^*$. This implies that at least one of the two matchings M_1' and M_2' have weight smaller than w^*, which is a contradiction. ◀

3.2 Weight scheme

Now, we come to the design of an isolating weight assignment for the graph and the proof of our main theorem. One of the components of our weight scheme is the isolating weight assignment W_{TV} constructed by Tewari and Vinodchandran [27] for planar bipartite graph. We will use the same weight scheme, but for any graph (not necessarily planar) embedded in the plane.

Consider a bipartite graph $G = (A, B, E)$ (not necessarily planar) with a straight-line embedding in $\mathbb{R}^2$. For any vertex u, let (x_u, y_u) be the associated point in $\mathbb{R}^2$. For an edge $e = (u, v)$, where $u \in A$ and $v \in B$, we define the weight function W_{TV} as follows:

$$W_{TV}(e) = (y_v - y_u) \times (x_v + x_u)$$

Then, the theorem below says that W_{TV} is isolating for bipartite planar graphs.

▶ **Theorem 3.4** ([27]). *Let G be a planar bipartite graph. Then with respect to weight function W_{TV} (defined using any planar embedding), the minimum weight perfect matching in G, if one exists, is unique.*

For a geometric bipartite graph, our main idea is to combine W_{TV} with the approximate distance function (up to a certain number of bits of precision) The purpose of combining W_{TV} is to break ties among minimum weight perfect matchings according to the approximate distance function.

Let $G(A, B, E)$ be a geometric bipartite graph on the $N \times N$ integer grid. Let $d(\cdot)$ be the weight function on the edges defined using the ℓ_p distance and let it naturally extend to subsets of edges. For any positive integer ℓ, let us define the approximate distance function $d_\ell \colon E \to \mathbb{Z}$ as

$$d_\ell(e) = \lfloor d(e) \times 2^\ell \rfloor.$$

First, let us show that the minimum weight perfect matchings with respect to approximate distance function remains near-minimum with respect to the exact distance function.

▷ Claim 3.5. For any positive integer ℓ, let M and M^* be minimum weight perfect matchings with respect to functions d_ℓ and d, respectively. Then,

$$d(M) < d(M^*) + n/2^\ell.$$

Proof. Observe that for any edge e, $2^\ell d(e) - 1 < d_\ell(e) \leq 2^\ell d(e)$. Hence, for perfect matching M,

$$2^\ell d(M) - n < d_\ell(M) \leq 2^\ell d(M).$$

Then, we can write

$$2^\ell d(M) < d_\ell(M) + n \leq d_\ell(M^*) + n \leq 2^\ell d(M^*) + n.$$

This implies that $d(M) < d(M^*) + n/2^\ell$. ◁

3.2.1 Weight scheme

For any integer ℓ, now let us define the combined weight function W_ℓ on the edges as follows:

$$W_\ell := (2nN^2 + 1) \times d_\ell + W_{TV}$$

Here, the scaling d_ℓ with a large number ensures that W_ℓ has the same ordering of perfect matchings as d_ℓ, and the W_{TV} function plays the role of tie breaking. Our next lemma says that when to take enough number of bits from the distance function and then combine it with W_{TV} as above, the resulting weight function is isolating.

▶ **Lemma 3.6.** *For any integer $\ell \geq (p^4 - 1) \log N + \log n + 3 + 1/p$, the minimum weight perfect matching in G with respect to the weight function W_ℓ is unique.*

Proof. First, observe that for any two perfect matchings M_1 and M_2,

$$d_\ell(M_1) > d_\ell(M_2) \implies W_\ell(M_1) > W_\ell(M_2).$$

This is because the maximum contribution of W_{TV} to the weight of a matching can be at most $n \times 2N^2$. Thus, we can write

$$\begin{aligned} W_\ell(M_1) - W_\ell(M_2) &= (2nN^2+1)(d_\ell(M_1) - d_\ell(M_2)) + W_{TV}(M_1) - W_{TV}(M_2) \\ &\geq (2nN^2+1)\cdot 1 + 0 - 2nN^2. \\ &\geq 1 \end{aligned}$$

It follows that the set of minimum weight perfect matchings with respect to W_ℓ is a subset of that with respect to d_ℓ. Now, we argue that these sets form a planar subgraph.

▷ Claim 3.7. The union of minimum weight perfect matchings with respect to d_ℓ forms a planar subgraph.

Proof. Let M and M^* be the minimum weight perfect matchings with respect to functions d_ℓ and d, respectively. From Claim 3.5 we have that $d(M) < d(M^*) + n/2^\ell$. By substituting $\ell \geq (p^4-1)\log N + \log n + 3 + 1/p$, we get that the gap is less than $1/(8\sqrt[p]{2}N^{p^4-1})$. Hence, M is a near-minimum weight perfect matching with respect to the $d(\cdot)$. Then, the claim follows from Lemma 3.3. ◁

To finish the proof of the lemma, let H be the subgraph formed by the union of minimum weight perfect matchings with respect to d_ℓ. Clearly, d_ℓ gives equal weights to all the perfect matchings in H. Thus, the function W_ℓ is the same as W_{TV} on H (up to an additive constant). From Theorem 3.4, we know that W_{TV} ensures a unique minimum weight perfect matching in the planar graph H. Hence, so does W_ℓ. ◀

3.2.2 Proof of the main theorem (Theorem 1.2)

Once we have shown how to construct an isolating weight assignment, we just need to use the algorithm of Mulmuley, Vazirani and Vazirani [21] to construct the minimum weight perfect matching.

▶ **Theorem 3.8** ([21]). *Given a graph $G = (V, E)$ with an isolating weight assignment on the edges that uses $O(\log n)$ bits, there is an* NC *algorithm to find the minimum-weight perfect matching.*

Now, we are ready to prove the main theorem. Suppose we are given a bipartite set of $2n$ points in $N \times N$ integer grid. Recall that the weight of an edge is defined to be the Euclidean distance between the endpoints. Our goal is to construct a perfect matching whose weight is at most $w^* + \delta$, where δ is the given error parameter and w^* is the minimum weight of a perfect matching. If we choose $\ell \geq \log(n/\delta)$, then from Claim 3.5, we know that a minimum weight perfect matching with respect to function $d_\ell(\cdot)$ will have the desired property.

We choose $\ell = \max\{\log(n/\delta), (p^4-1)\log N + \log n + 4\}$. Then we use the weight scheme W_ℓ with the MVV algorithm (Theorem 3.8). Recall that from Lemma 3.6, we have the isolation property required in Theorem 3.8. Finally, let us analyse the number of bits used by weight function W_ℓ. The maximum weight given to any edge by function $d(\cdot)$ is at most $\sqrt[p]{2}N$ and by function W_{TV}, it is at most $2N^2$. Thus, the maximum weight given to any edge by function W_ℓ will be at most $2^\ell \times \sqrt[p]{2}N \times (2nN^2+1) + 2N^2$. The number of bits in weight of any edge comes out to be $O(\log(Nn/\delta))$. Hence, we have an NC algorithm, whenever N and $1/\delta$ are polynomial in n.

4 Conclusion

In this work, we explored the parallel complexity of GWBPM problem. We established a lower bound which shows that for GWBPM, a linear number of bits is required to distinguish the minimum-weight perfect matching from others. Next, we showed that GWBPM problem (under ℓ_p metric for $p \geq 2$) that allows up to $\frac{1}{\text{poly}(n)}$ additive error, is in NC. The main question that arises from our work is whether the non-bipartite version of GWBPM is also in NC. Another possible extension is to consider the bipartite version in 3 or higher dimensions.

References

1. The open problems project: problem 33 sum of square roots. https://topp.openproblem.net/p33. Accessed: 2010-09-30.
2. Pankaj K Agarwal, Hsien-Chih Chang, Sharath Raghvendra, and Allen Xiao. Deterministic, near-linear ε-approximation algorithm for geometric bipartite matching. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1052–1065, 2022. doi:10.1145/3519935.3519977.
3. Pankaj K Agarwal, Hsien-Chih Chang, and Allen Xiao. Efficient algorithms for geometric partial matching. In *35th International Symposium on Computational Geometry (SoCG 2019)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.
4. Pankaj K Agarwal, Alon Efrat, and Micha Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 39–50, 1995. doi:10.1145/220279.220284.
5. Nima Anari and Vijay V. Vazirani. Planar graph perfect matching is in NC. *J. ACM*, 67(4):21:1–21:34, 2020. doi:10.1145/3397504.
6. Christoph Burnikel, Rudolf Fleischer, Kurt Mehlhorn, and Stefan Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica*, 27(1):87–99, 2000. doi:10.1007/S004530010005.
7. Richard Carr and Cormac O'Sullivan. On the linear independence of roots. *International Journal of Number Theory*, 05:161–171, 2007.
8. Qi Cheng and Yu-Hsin Li. Finding the smallest gap between sums of square roots. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics*, pages 446–455, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi:10.1007/978-3-642-12200-2_39.
9. Julia Chuzhoy and Sanjeev Khanna. Maximum bipartite matching in $n^{2+o(1)}$ time via a combinatorial algorithm. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 83–94, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649725.
10. Elias Dahlhaus and Marek Karpinski. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Applied Mathematics*, 84(1-3):79–91, 1998. doi:10.1016/S0166-218X(98)00006-7.
11. Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory of Computing Systems*, 47(3):737–757, 2010. doi:10.1007/S00224-009-9204-8.
12. Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.
13. Friedrich Eisenbrand, Matthieu Haeberle, and Neta Singer. An improved bound on sums of square roots via the subspace theorem. *CoRR*, abs/2312.02057, 2023. To appear in SoCG 2024. doi:10.48550/arXiv.2312.02057.
14. Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 754–763, 2016. doi:10.1145/2897518.2897564.

15 Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Using interior-point methods for fast parallel algorithms for bipartite matching and related problems. *SIAM J. Comput.*, 21(1):140–150, February 1992. `doi:10.1137/0221011`.

16 Dima Yu Grigoriev and Marek Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 166–172. IEEE, 1987.

17 John E Hopcroft and Richard M Karp. An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973. `doi:10.1137/0202019`.

18 Richard M Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 22–32, 1985. `doi:10.1145/22145.22148`.

19 László Lovász. On determinants, matchings, and random algorithms. In *FCT*, volume 79, pages 565–574, 1979.

20 Aleksander Madry. Navigating central path with electrical flows: From flows to matchings, and back. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 253–262. IEEE, 2013. `doi:10.1109/FOCS.2013.35`.

21 Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 345–354, 1987. `doi:10.1145/28395.383347`.

22 Joseph O'Rourke. Advanced problem 6369. *Amer. Math. Monthly*, 88(10):769, 1981.

23 Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

24 R Sharathkumar. A sub-quadratic algorithm for bipartite matching of planar points with bounded integer coordinates. In *Proceedings of the twenty-ninth annual symposium on Computational geometry*, pages 9–16, 2013. `doi:10.1145/2462356.2480283`.

25 R Sharathkumar and Pankaj K Agarwal. Algorithms for the transportation problem in geometric settings. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 306–317. SIAM, 2012. `doi:10.1137/1.9781611973099.29`.

26 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-NC. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 696–707. Ieee, 2017. `doi:10.1109/FOCS.2017.70`.

27 Raghunath Tewari and NV Vinodchandran. Green's theorem and isolation in planar graphs. *Information and Computation*, 215:1–7, 2012. `doi:10.1016/J.IC.2012.03.002`.

28 Jan Van Den Brand, Li Chen, Richard Peng, Rasmus Kyng, Yang P Liu, Maximilian Probst Gutenberg, Sushant Sachdeva, and Aaron Sidford. A deterministic almost-linear time algorithm for minimum-cost flow. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 503–514. IEEE, 2023. `doi:10.1109/FOCS57990.2023.00037`.

29 Jan van den Brand, Yin-Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Bipartite matching in nearly-linear time on moderately dense graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 919–930. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00090`.

30 Kasturi R Varadarajan. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 320–329. IEEE, 1998.

31 Kasturi R Varadarajan and Pankaj K Agarwal. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *SODA*, volume 99, pages 805–814, 1999. URL: `http://dl.acm.org/citation.cfm?id=314500.314918`.