Electronic Colloquium on Computational Complexity, Report No. 162 (2026)

Weighted Bipartite Matching is in $\mathrm{Mod}_p\mathsf{L}$

Mingzi Xiao
Peking University
2400012950@stu.pku.edu.cn

Aug 27, 2026

Abstract

The recent paper [CGG+26] showed that deciding whether a bipartite graph has a perfect matching can be reduced to deciding whether a determinant, whose value may be assigned to any sufficiently large field $\mathbb{F}$, equals to zero. In the second part of their work, [CGG+26] also generalized the algebraic method and proposed an NC algorithm that computes the maximum *weighted* perfect matching of bipartite graphs. However, the method they used relies on the positivity of non-zero sum of squares and does not generalize to finite fields. In this paper we further leverage their ideas and show that the algebraic method works in finite fields as well, therefore maximum weighted perfect bipartite matching is in $\mathrm{Mod}_p\mathsf{L}$.

1 Introduction

Here we briefly present the ideas in [CGG+26] and introduce several notions.

A *bipartite graph* $G = (L \cup R, E)$ consists of vertices $L \cup R$ and edges $E \subseteq L \times R$. We consider *balanced bipartite graphs*, that is, $|L| = |R| = n$. A *matching* is a subset of E that covers each vertex of G at most once. A *perfect matching* is a matching of size n. For an unbalanced bipartite graph with $|L| < |R|$, define a *left-saturated matching* as a matching that covers L. A *weighted bipartite graph* consists of G and a weight function $w : E \to [0, W] \cap \mathbb{Z}$. Here we let W be a polynomial of n.

Determining whether a bipartite graph has a perfect matching, and searching for such a matching, is a classic problem. If a perfect matching exists, we can further ask what is the maximum weight of a perfect matching. In [CGG+26], it is shown that both problems can be solved in NC. Specifically, deciding whether G has a perfect matching and finding the maximum weight of a perfect matching can be done in NC^2 while searching for a matching can be done in NC^3. We focus on the two decision problems.

For the perfect matching problem, they constructed a matrix $M(G)$ of size $nD \times n^2 r$ so that $M(G)$ has full row rank iff G has a perfect matching. Here the parameters are set to $D = n(r - \delta)$ where $\delta \ge n$ and $r > n^2\delta$. Roughly speaking, $M(G)$ is a matrix that has slightly more columns than rows. The construction takes 3 steps:

1. Let A be the $n \times n$ adjacent matrix of G, where $A_{ij} = 1$ iff $(i, j) \in E$.
2. Expand A to a $n \times n^2$ matrix $\hat{A}$. Here $\hat{A} = (A \mid A_1)$, where $(A_1)_{i((n-1)i+j)} = 1$ $(1 \le j \le n - 1)$, and the other entries are 0.
3. Replace each entries of $\hat{A}$ with $D \times r$ blocks, thus producing $M(G)$. Specifically, replace each 0 with 0 matrices, and replace each $\hat{A}_{i,j} = 1$ with *γ-folded Vandermonde matrix* $V(\alpha_j)$. Here γ is an element in $\mathbb{F}$ with an order more than D and $\{\alpha_i\gamma^j \mid 1 \le i \le n^2, 0 \le j < r\}$ is a set with $n^2 r$ distinct elements. The γ-folded Vandermonde matrix $V(\alpha) = ((\alpha\gamma^{i-1})^{j-1})_{ij}$. The construction is closely related to *subspace designs*. (Refer to [CGG+26], Section 2 and 3 for details.)

ISSN 1433-8092

It is proven that

Proposition 1.1 ([CGG+26], Theorem 3.1)**.** *$M(G)$ has full rank $\iff$ G has a perfect matching.*

Since whether $M(G)$ has full rank can checked by computing $\det(M(G)M(G)^\top)$, the decision problem can be done in NC^2.

To compute the maximum weight of a perfect matching, they modified $M(G)$ by multiplying the (i,j) block of G by $t^{w(i,j)}$, here t is a variable and $w(i,j)$ is the weight of (i,j). Call this matrix $M(G,w)$. Then it is shown that the maximum weight of G can be computed from the highest degree in $\det(M(G,w)M(G,w)^\top)$, assume we adjust the setting of the parameters to $\delta \geq n$ and $r > 4n^3\delta W$.

Proposition 1.2 ([CGG+26], Lemma 4.4)**.** *Assume the ground field $F = \mathbb{Q}$. Let $\mathcal{D}_w(t) = \det(M(G,w)M(G,w)^\top)$. Then $|\deg(\mathcal{D}_w(t)) - 2rw^*| \leq 4n^3\delta W$. Here w^* is the maximum weight.*

Since $r > 4n^3W$, w^* is the nearest integer to $\deg(\mathcal{D}_w(t))/(2r)$.

Let $M(G,w)[S]$ be the $nD \times nD$ minor of $M(G,w)$ whose columns are in $S \subseteq [n^2r]$. Proposition 1.2 is based on two claims:

Lemma 1.3 ([CGG+26], Claim 4.5)**.** *$\forall S, \deg(\det(M(G,w)[S])) \leq rw^* + 2n^3\delta W$.*

Lemma 1.4 ([CGG+26], Claim 4.6 and following discussions)**.** *$\exists S, \deg(\det(M(G,w)[S])) \geq rw^* - 2n^3\delta W$.*

Finally, the Cauchy-Binet formula states that

$$\mathcal{D}_w(t) = \det(M(G,w)M(G,w)^\top) = \sum_S \det(M(G,w)[S])^2.$$

Consider the set of S that maximizes $\deg(\det(M(G,w)[S]))$. Positivity of squares guarantees that the coefficients won't cancel out, which yields Proposition 1.2. However, this treatment doesn't apply to fields with positive character, such as finite fields.

2 The Isolation

To avoid collision, we compute the following determinant

$$\begin{aligned}\mathcal{D}_w(t,y) &= \det(M(G,w)\operatorname{diag}(1,y,\ldots,y^{n^2r-1})M(G,w)^\top) \\ &= \sum_S \det(M(G,w)[S])^2 y^{v(S)} \in F[t,y].\end{aligned}$$

Here $v(S) = \sum_{i\in S}(i-1)$.

We will prove the following fact:

Proposition 2.1. *Let $\mathcal{B}$ be the set of S that maximizes $\deg(\det(M(G,w)[S]))$. Then $\{v(S) \mid S \in \mathcal{B}\}$ are distinct.*

Therefore, to compute w^*, we only need to check the term in $\mathcal{D}_w(t,y)$ which maximizes the degree in t.

Proposition 2.1 is a special case of the Isolation Lemma: we can randomly assign polynomially bounded weights to each element so that every $S \in \mathcal{B}$ has distinct sum of weights. We don't have a deterministic way to assign weights in general, but when $\mathcal{B}$ are the bases of a matroid any approach suffices as long as the weights of the elements are nonidentical. This is can be shown by the greedy method to compute the maximally weighted basis in a matroid. So Proposition 2.1 can be proven directly from

Proposition 2.2. *$\mathcal{B}$ forms the bases of a matroid.*

Proof. This is related to the theory of valuated matroids. One can refer to [DW92] for a detailed study. We also provide a self-contained proof here.

To prove that $\mathcal{B}$ forms the bases of a matroid, we only need two show that for $A, B \in \mathcal{B}$, the *exchange property* is satisfied: let $e \in A \setminus B$, then $A - e + f \in \mathcal{B}$ for some f in $B \setminus A$. Once this holds, we can take the independent family as the sets $S \subseteq$ some member in $\mathcal{B}$.

Let Δ be the maximum degree of the determinants of $nD \times nD$ minors in $M(G, w)$, $p_S(t) = \det(M(G, w)[S])$. Then $\deg p_A(t) = \deg p_B(t) = \Delta$.

The Grassmann-Plücker relations state that

$$p_A(t)p_B(t) = \sum_{f \in B} \pm p_{A-e+f}(t)p_{B-f+e}(t)$$

where the $\pm$ is caused by the positions of e and f. Since $\deg p_S \leq \Delta$ and the LHS of (3) has degree 2Δ, some term on the RHS must also have degree 2Δ. This term guarantees that $A - e + f$ and $B - f + e \in \mathcal{B}$. ☐

3 Explicit Construction for computing in $\mathrm{Mod}_p\mathsf{L}$

We follow a route that is similar to [App26]. One can refer to it for a definition of the complexity class $\mathrm{Mod}_p\mathsf{L}$. In general, the goal is to explicitly construct a directed acyclic graph G_{acyc} with sources s, t for each instance x of the problem, so that x is in the language iff the number of paths from s to t is not a multiple of p. Let d_u be the number of paths from u to t, we have $d_t = 1$,

$$d_u = \sum_{(u,v) \in G_{acyc}} d_v.$$

We can also allow edge weights in $\mathbb{F}_p$. That is, if $w(u, v)$ is the weight of edge (u, v), we have

$$d_u = \sum_{(u,v) \in G_{acyc}} w(u, v)d_v.$$

This can be simulated with the unweighted construction using multiple parallel edges.

Multiplication between a given matrix and a vector can be easily simulated. Suppose we have $v = (d_{v_1}, d_{v_2}, \cdots, d_{v_n})^\top$ being the column vector, $M \in \mathbb{F}^{m \times n}$. To compute Mv, we can construct $u = (d_{u_1}, d_{u_2}, \cdots, d_{u_m})^\top = Mv$ by letting $w(u_i, v_j) = M_{ij}$.

We can further allow edge weights and (d_u) in $\mathbb{F}_q$ instead of $\mathbb{F}_p$, here $q = p^m$. This is because we can use a group of m vertices as a large "vertex" to represent a member in $\mathbb{F}_q$ (as it is an m-dimensional $\mathbb{F}_p$-linear space), and multiplying $t \in \mathbb{F}_q$ by w can be viewed as a linear transformation.

To compute $\mathcal{D}_w(t, y)$, we first substitute $t := y^{n^2 r}$ so that the term $t^i y^j$ uniquely corresponds to $y^{n^2 ri+j}$. Let $\alpha_1, \alpha_2, \cdots, \alpha_K$ be K distinct elements in $\mathbb{F}_q$. Here $K = 2(nDW \cdot n^2 r + n^2 r) > \deg(\mathcal{D}_w(y^{n^2 r}, y))$. Let $\mathcal{D}_w^{(i)} = \mathcal{D}_w(\alpha_i^{n^2 r}, \alpha_i)$. $\mathcal{D}_w^{(i)}$, which is the determinant of a given matrix, can be computed by the Samuelson-Berkowitz algorithm (see [Ber84]) in parallel. The Samuelson-Berkowitz algorithm states that the coefficients of the characteristic polynomial is the product of some matrices (one of which is a vector) which can be computed from the original matrix, so it can be simulated in $\mathrm{Mod}_p\mathsf{L}$. This involves computing some edge weight $w(u, v)$ on the fly. However, if $w(u, v)$ can be simulated in $\mathrm{Mod}_p L$, we can choose u, v as the source and the sink respectively, which is equivalent to an edge with weight $w(u, v)$.

Now suppose $d_{u_i} = \mathcal{D}_w^{(i)}$, we need to check if the polynomial interpolated from $\mathcal{D}_w^{(i)}$ has a large degree. We can simply multiply (d_{u_i}) by the inverted Vandermonde matrix. Let (d_{v_i}) be the coefficient of y^i in $\mathcal{D}_w(y^{n^2 r}, y)$. We would like to check if $\exists i \geq 2w^* r \cdot n^2 r$, $d_{v_i} \neq 0$. Let $d_{v_i}^{(j)}(j \in [m])$ be the components of d_{v_i}. We can compute

$$1 - \prod_{i \geq 2w^* r \cdot n^2 r, j \in [m]} (1 - (d_{v_i}^{(j)})^{p-1}) \bmod p.$$

From Fermat's Little theorem, this value is 1 iff such i exists. Again, the product of variables can be computed in sequence.

References

[App26] Benny Applebaum. From nondeterministic to modular branching programs, uniformly and in logspace. (TR26-131), August 2026.

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3):147–150, 1984.

[CGG+26] Abhranil Chatterjee, Sumanta Ghosh, Rohit Gurjar, Roshan Raj, and Thomas Thierauf. Bipartite matching is in nc. (TR26-100), June 2026.

[DW92] Andreas W.M Dress and Walter Wenzel. Valuated matroids. *Advances in Mathematics*, 93(2):214–250, 1992.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il