

Bipartite Perfect Matching is in quasi-NC

Stephen Fenner¹, Rohit Gurjar^{*2}, and Thomas Thierauf^{*2}

¹University of South Carolina

²Aalen University, Germany

November 9, 2015

Abstract

We show that the bipartite perfect matching problem is in quasi-NC². That is, it has uniform circuits of quasi-polynomial size and $O(\log^2 n)$ depth. Previously, only an exponential upper bound was known on the size of such circuits with poly-logarithmic depth.

We obtain our result by an almost complete derandomization of the famous Isolation Lemma when applied to yield an efficient randomized parallel algorithm for the bipartite perfect matching problem.

1 Introduction

The perfect matching problem has been widely studied in complexity theory. It has been of particular interest in the study of derandomization and parallelization. The perfect matching problem, PM, asks whether a given graph contains a perfect matching.

The problem has a polynomial-time algorithm due to Edmonds [Edm65]. However, its parallel complexity is still not completely resolved as of today. The problem can be solved by randomized efficient parallel algorithms due to Lovász [Lov79], i.e., it is in RNC, but it is not known whether randomness is necessary, i.e., whether it is in NC. The class NC represents the problems which have efficient parallel algorithms, i.e., they have uniform circuits of polynomial size and poly-logarithmic depth. For the perfect matching problem, nothing better than an exponential-size circuit was known, in the case of poly-logarithmic bounded depth.

The construction version of the problem, SEARCH-PM, asks to construct a perfect matching in a graph if one exists. It is in RNC due to Karp et al. [KUW86] and MULMULEY et al. [MVV87]. The latter algorithm applies the celebrated Isolation Lemma. Both algorithms work with a weight assignment on the edges of the graph. A weight assignment is called *isolating* for a graph G if the minimum weight perfect matching in G is unique, if one exists. MULMULEY et al. [MVV87] showed that given an isolating weight assignment with polynomially bounded integer weights for a graph G , then a perfect matching in G can be constructed in NC. To get an isolating weight assignment they use randomization. This is where the Isolation Lemma comes into play.

*Supported by DFG grant TH 472/4

Lemma 1.1 (Isolation Lemma [MVV87]). *For a graph $G(V, E)$, let w be a random weight assignment, where edges are assigned weights chosen uniformly and independently at random from $\{1, 2, \dots, 2|E|\}$. Then w is isolating with probability $\geq 1/2$.*

Derandomizing this lemma means to construct such a weight assignment deterministically in NC. This remains a challenging open question. A general version of this lemma, which considers a family of sets and requires a unique minimum weight set, has also been studied. The general version is related to the polynomial identity testing problem and circuit lower bounds [AM08].

The Isolation Lemma has been derandomized for some special classes of graphs, e.g., planar bipartite graphs [DKR10, TV12], strongly chordal graphs [DK98], graphs with a small number of perfect matchings [GK87, AHT07]. In this work, we make a significant step towards the derandomization of the Isolation Lemma for bipartite graphs. In Section 3, we construct an isolating weight assignment for these graphs with quasi-polynomially large weights. Previously, the only known deterministic construction was the trivial one that used exponentially large weights. As a consequence we get that for bipartite graphs, PM and SEARCH-PM are in quasi-NC². That is, they can be solved by uniform Boolean circuits of quasi-polynomial size and quadratic logarithmic depth.

In Section 4 we show that our approach also gives an alternate NC-algorithm for SEARCH-PM in bipartite planar graphs. This case already has known NC-algorithms [MN95, MV00, DKR10]. Our algorithm is in NC³, while the previous best known upper bound is already NC² [MN95, DKR10].

Our result also gives an RNC-algorithm for PM in bipartite graphs which uses very few random bits. The original RNC-algorithm of Lovász [Lov79] uses $O(m \log n)$ random bits. This has been improved by Chari, Rohatgi, and Srinivasan [CRS95] to $O(n \log m/n)$ random bits. They actually construct an isolating weight assignment using these many random bits. To the best of our knowledge, the best upper bound today on the number of random bits is $(n + n \log m/n)$ by Chen and Kao [CK97], that is, the improvement to [CRS95] was to bring down the constant factor to two. In Section 5 we achieve an *exponential* step down to $O(\log^3 n)$ random bits. Note however, that we do *not* get an isolating weight assignment with polynomially bounded weights.

We give a short outline of the main ideas of our approach. For any two perfect matchings, the edges where they differ form disjoint cycles. For a cycle C , its circulation is defined to be the difference of weights of two perfect matchings which differ exactly on the edges of C . Datta et al. [DKR10] showed that a weight assignment which ensures nonzero circulation for every cycle is isolating. It is not clear if there exists such a weight assignment with small weights. Instead, we give nonzero circulations only to *small* cycles. Then, we consider the union of minimum weight perfect matchings. In the bipartite case, this union is significantly smaller than the original graph. In particular, at least half of its nodes have degree ≤ 2 . We keep repeating this procedure with weight assignments that give nonzero circulation to every cycle with a small number of degree > 2 nodes. In each round, the number of degree > 2 nodes decreases by half. In $\log n$ rounds, we get a unique minimum weight perfect matching.

2 Preliminaries

2.1 Matchings and Complexity

By $G(V, E)$ we denote a graph with vertex set V and edge set E . We consider only undirected graphs in this paper. A graph is *bipartite* if there exists a partition $V = L \cup R$ of the vertices such that all edges are between vertices of L and R . The *girth* of G is the length of the shortest cycle in G . If G has no cycle, then the girth is defined to be infinite.

In a graph $G(V, E)$, a *matching* $M \subseteq E$ is a subset of edges with no two edges sharing an endpoint. A matching which covers every vertex is called a *perfect matching*. For any weight assignment $w: E \rightarrow \mathbb{Z}$ on the edges of a graph, the *weight of a matching* M is defined to be the sum of weights of all the edges in M , i.e., $w(M) = \sum_{e \in M} w(e)$.

A weight function w is called *isolating for* G if there is a unique perfect matching of minimum weight in G .

A graph G is *matching-covered* if each edge in G participates in some perfect matching. In the literature, matching-covered is also called *1-extendable* and these notions require G to be *connected*. **Note:** in this paper, we use matching-covered also for non-connected graphs!

The *perfect matching problem* PM is to decide whether a given graph has a perfect matching. Its construction version SEARCH-PM is to compute a perfect matching of a given graph, or to determine that no perfect matching exists.

Analogous to NC, Barrington [Bar92] defined the class quasi-NC ^{k} as the class of problems which have uniform circuits of quasi-polynomial size $2^{\log^{O(1)} n}$ and poly-logarithmic depth $O(\log^k n)$. Here, *uniformity* means that local queries about the circuit can be answered in poly-logarithmic time (see [Bar92] for details). The class quasi-NC is the union of classes quasi-NC ^{k} , over all $k \geq 0$.

2.2 An RNC algorithm for Search-PM

Let us first recall the RNC algorithm of Mulmuley, Vazirani & Vazirani [MVV87] for the construction of a perfect matching (SEARCH-PM). Though the algorithm works for any graph, we will only consider bipartite graphs here.

Let G be a bipartite graph with vertex partitions $L = \{u_1, u_2, \dots, u_n\}$ and $R = \{v_1, v_2, \dots, v_n\}$, and weight function w . Consider the following $n \times n$ matrix A associated with G ,

$$A(i, j) = \begin{cases} 2^{w(e)}, & \text{if } e = (u_i, v_j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

The algorithm in [MVV87] computes the determinant of A . An easy argument shows that this determinant is the signed sum over all perfect matchings in G :

$$\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^n A(i, \pi(i)) \tag{1}$$

$$= \sum_{M \text{ perfect matching in } G} \operatorname{sgn}(M) 2^{w(M)} \tag{2}$$

Equation (2) holds because the product $\prod_{i=1}^n A(i, \pi(i))$ is nonzero if and only if the permutation π corresponds to a perfect matching. Here $\operatorname{sgn}(M)$ is the sign of the corresponding

permutation. If the graph G does not have a perfect matching, then clearly $\det(A) = 0$. However, even when the graph has perfect matchings, there can be cancellations due to $\text{sgn}(M)$, and $\det(A)$ may become zero. To avoid such cancellations, one needs to design the weight function w cleverly. In particular, if G has a perfect matching and w is isolating, then $\det(A) \neq 0$. This is because the term $2^{w(M)}$ corresponding to the minimum weight perfect matching cannot be canceled with other terms, which are strictly higher powers of 2.

Given an isolating weight assignment for G , one can easily construct the minimum weight perfect matching in NC . Let M^* be the unique minimum weight perfect matching in G . For every edge $e \in E$, compute the determinant of the matrix A_e associated with $G - e$. If the term $2^{w(M^*)}$ disappears from $\det(A_e)$, then $e \in M^*$. Doing this in parallel for each edge, we can find all the edges in M^* .

As already explained in the introduction, the Isolation Lemma delivers the isolating weight assignment with high probability. Moreover, the weights chosen by the Isolation Lemma are polynomially bounded. Therefore, the entries in matrix A have polynomially many bits. This suffices to compute the determinant in NC^2 [Ber84]. Hence, also the construction is in NC^2 . Put together, this yields an RNC-algorithm for SEARCH-PM.

2.3 Matching Polytope

Matchings are also one of the well-studied objects in polyhedral combinatorics. Matchings have an associated polytope, called the *perfect matching polytope*. We use some properties of this polytope to construct an isolating weight assignment. The perfect matching polytope also forms the basis of one of the NC -algorithms for bipartite planar matching [MV00].

The perfect matching polytope $\text{PM}(G)$ of a graph $G(V, E)$ with $|E| = m$ edges is a polytope in the edge space, i.e., $\text{PM}(G) \subseteq \mathbb{R}^m$. For any perfect matching M of G , consider its incidence vector $\mathbf{x}^M = (x_e^M)_e \in \mathbb{R}^m$ given by

$$x_e^M = \begin{cases} 1, & \text{if } e \in M, \\ 0, & \text{otherwise.} \end{cases}$$

This vector is referred as a *perfect matching point* for any perfect matching M . The *perfect matching polytope* of a graph G is defined to be the convex hull of all its perfect matching points,

$$\text{PM}(G) = \text{conv}\{ \mathbf{x}^M \mid M \text{ is a perfect matching in } G \}.$$

Any weight function $w: E \rightarrow \mathbb{R}$ on the edges of a graph G can be naturally extended to \mathbb{R}^m as follows: for any $\mathbf{x} = (x_e)_e \in \mathbb{R}^m$, define

$$w(\mathbf{x}) = \sum_{e \in E} w(e) x_e.$$

Clearly, for any matching M , we have $w(M) = w(\mathbf{x}^M)$. In particular, let M^* be a perfect matching in G of minimum weight. Then

$$w(M^*) = \min\{ w(\mathbf{x}) \mid \mathbf{x} \in \text{PM}(G) \}.$$

The following lemma gives a simple description of the perfect matching polytope of a bipartite graph G which is well known, see for example [LP86].

Lemma 2.1. *Let G be a bipartite graph and $\mathbf{x} = (x_e)_e \in \mathbb{R}^m$. Then $\mathbf{x} \in \text{PM}(G)$ if and only if*

$$\sum_{e \in \delta(v)} x_e = 1 \quad v \in V, \tag{3}$$

$$x_e \geq 0 \quad e \in E, \tag{4}$$

where $\delta(v)$ denotes the set of edges incident on the vertex v .

It is easy to see that any perfect matching point will satisfy these two conditions. In fact, all perfect matching points are vertices of this polytope. The non-trivial part is to show that any point satisfying these two conditions is in the perfect matching polytope [LP86, Chapter 7]. For general graphs, the polytope described by (3) and (4) can have vertices which are not perfect matchings. Thus, the description does not capture the perfect matching polytope for general graphs.

2.4 Nice Cycles and Circulation

Let $G(V, E)$ be a graph with a perfect matching. A cycle C in G is a *nice cycle*, if the subgraph $G - C$ still has a perfect matching. In other words, a nice cycle can be obtained from the symmetric difference of two perfect matchings. Note that a nice cycle is always an even cycle.

For a weight assignment w on the edges, the *circulation* $c_w(C)$ of an even length cycle $C = (v_1, v_2, \dots, v_k)$ is defined as the alternating sum of the edge weights of C ,

$$c_w(C) = |w(v_1, v_2) - w(v_2, v_3) + w(v_3, v_4) - \dots - w(v_k, v_1)|.$$

The definition is independent of the edge we start with because we take the absolute value of the alternating sum.

The circulation of nice cycles was one crucial ingredient of the isolation in bipartite planar graphs given by Datta et al. [DKR10]. An important property is as follows: Let M_1, M_2 be two perfect matchings such that their symmetric difference is precisely the nice cycle C . Then we have $c_w(C) = |w(M_1) - w(M_2)|$. That is, M_1 and M_2 have the same weight, if and only if $c_w(C) = 0$. Consequently, when all nice cycles have nonzero circulation, then the minimum perfect matching is unique.

Lemma 2.2 ([DKR10]). *Let G be a graph with a perfect matching, and let w be a weight function such that all nice cycles in G have nonzero circulation. Then the minimum perfect matching is unique. That is, w is isolating.*

We will construct an isolating weight function for bipartite graphs. However, our weight function will not necessarily have nonzero circulation on all nice cycles. We start out with a weight assignment which ensures nonzero circulations for a small set of cycles in a black-box way, i.e., without being able to compute the set efficiently. The following lemma describes a standard trick for this.

Lemma 2.3 ([CRS95]). *Let G be a graph with n nodes. Then, for any number s , one can construct a set of $O(n^2s)$ weight assignments with weights bounded by $O(n^2s)$, such that for any set of s cycles, one of the weight assignments gives nonzero circulation to each of the s cycles.*

Proof. Let us first assign exponentially large weights. Let e_1, e_2, \dots, e_m be some enumeration of the edges of G . Define a weight function w by $w(e_i) = 2^{i-1}$, for $i = 1, 2, \dots, m$. Then clearly every cycle has a nonzero circulation. However, we want to achieve this with small weights.

We consider the weight assignment modulo small numbers, i.e., the weight functions $\{w \bmod j \mid 2 \leq j \leq t\}$ for some appropriately chosen t . We want to show that for any fixed set of s cycles $\{C_1, C_2, \dots, C_s\}$, one of these assignments will work, when t is chosen large enough. That is, we want

$$\exists j \leq t \quad \forall i \leq s : c_{w \bmod j}(C_i) \neq 0.$$

This will be true provided

$$\exists j \leq t : \prod_{i=1}^s c_w(C_i) \not\equiv 0 \pmod{j}.$$

In other words,

$$\text{lcm}(2, 3, \dots, t) \nmid \prod_{i=1}^s c_w(C_i).$$

This can be achieved by setting $\text{lcm}(2, 3, \dots, t) > \prod_{i=1}^s c_w(C_i)$. The product $\prod_{i=1}^s c_w(C_i)$ is upper bounded by $2^{n^2 s}$. Furthermore, we have $\text{lcm}(2, 3, \dots, t) > 2^t$ for $t \geq 7$ (see [Nai82]). Thus, choosing $t = n^2 s$ suffices. Clearly, the weights are bounded by $t = n^2 s$. \square

3 Isolation in Bipartite Graphs

In this section we present our main result, an almost efficient parallel algorithm for the perfect matching problem.

Theorem 3.1. *For bipartite graphs, PM and SEARCH-PM are in quasi-NC².*

Our starting point is Lemma 2.2 which requires nonzero circulations for all nice cycles. Recall that the construction algorithm requires the weights to be polynomially bounded. As the number of nice cycles can be exponential in the number of nodes, even the existence of such a weight assignment is not immediately clear. Nonetheless, Datta et al. [DKR10] give a construction of such a weight assignment for bipartite planar graphs. For general bipartite graphs, this is still an open question.

Our approach is to work with a weight function which gives nonzero circulation to only small cycles. Lemma 2.3 describes a way to find such weights. The cost of this weight assignment is proportional to the number of small cycles. Further, it is a black-box construction in the sense that one does not need to know the set of cycles. It just gives a set of weight assignments such that one of them has the desired property.

3.1 Union of Minimum Weight Perfect Matchings

Let us assign a weight function for bipartite graph G which gives nonzero circulation to all small cycles. Consider a new graph G_1 obtained by the union of minimum weight perfect matchings in G . Our hope is that G_1 is significantly smaller than the original graph G . Note

that it is not clear if one can efficiently construct G_1 from G . This is because the determinant of the bi-adjacency matrix with weights in equation (1) from Section 2.2 can still be zero. As we will see, we do not need to construct G_1 , it is just used in the argument. Our final weight assignment will be completely black-box in this sense.

The following lemma is the main reason why our technique is restricted to bipartite graphs: it shows that the graph G_1 constructed from the minimum weight perfect matchings in G contains no other perfect matchings than these. In Figure 1, we give an example that this does not hold in general graphs.

Lemma 3.2. *Let $G(V, E)$ be a bipartite graph with weight function w . Let E_1 be the union of all minimum weight perfect matchings in G . Then every perfect matching in the graph $G_1(V, E_1)$ has the same weight – the minimum weight of any perfect matching in G .*

Proof. We use the description of the perfect matching polytope for bipartite graphs from Lemma 2.1. Let q be the weight of a minimum weight perfect matching in G . Recall that

$$\min\{w(\mathbf{x}) \mid \mathbf{x} \in \text{PM}(G)\} = q.$$

The intersection of $\text{PM}(G)$ with the hyperplane $H(\mathbf{x}) = \{\mathbf{x} \mid w(\mathbf{x}) = q\}$ is a face F of the polytope.

One can describe a face of a polytope by replacing some of the inequalities in the description of the polytope by equalities. For the perfect matching polytope, the inequalities are given by (4). Thus, for the face F there exists a set $S \subseteq E$ such that for any $\mathbf{x} = (x_e)_e$, we have $\mathbf{x} \in F$ if and only if

$$\sum_{e \in \delta(v)} x_e = 1 \quad v \in V, \tag{5}$$

$$x_e \geq 0 \quad e \in E \setminus S, \tag{6}$$

$$x_e = 0 \quad e \in S. \tag{7}$$

Clearly, for any minimum weight perfect matching M , its matching point \mathbf{x}^M satisfies the above three conditions as it lies on the face F . In particular, equation (7) implies that $S \cap M = \emptyset$. It follows that

$$E_1 \cap S = \emptyset. \tag{8}$$

Now, consider any perfect matching M' in the graph $G_1(V, E_1)$. By equation (8), we have $M' \cap S = \emptyset$. Hence its matching point $\mathbf{x}^{M'}$ satisfies equation (7). Since $\mathbf{x}^{M'}$ represents a perfect matching, it also satisfies (5) and (6). Thus, we have $\mathbf{x}^{M'} \in F$. Hence, $w(M') = w(\mathbf{x}^{M'}) = q$. \square

Next, we show that G_1 is significantly smaller than G . As all the perfect matchings in G_1 have the same weight, every nice cycle in G_1 has zero circulation. The following lemma proves an even stronger statement: *every* cycle in G_1 has zero circulation. It essentially comes from the fact that all cycles are in the linear span of nice cycles. Note that G_1 is matching-covered, as it is a union of perfect matchings.

Lemma 3.3. *Let H be a matching-covered bipartite graph. Let w be a weight function such that every perfect matching in H has the same weight. Then for each cycle C in H , we have $c_w(C) = 0$.*

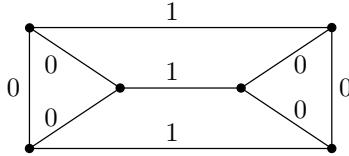


Figure 1: A non-bipartite weighted graph where every edge is contained in a minimum perfect matching of weight 1. However, the graph also has a perfect matching of weight 3. That is, Lemma 3.2 does not hold for non-bipartite graphs.

Proof. Let the weight of each perfect matching be q . Any point \mathbf{x} in the perfect matching polytope $\text{PM}(H)$ is a convex combination of perfect matchings. Therefore its weight $w(\mathbf{x})$ must also be q .

Let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t$ be all the perfect matchings points of H , i.e., the corners of $\text{PM}(H)$. Consider the average point $\mathbf{x} \in \text{PM}(H)$ of the matching points,

$$\mathbf{x} = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_t}{t}.$$

Because each edge participates in a perfect matching, every coordinate of $\mathbf{x} = (x_e)_e$ is nonzero. Now, consider a cycle C in H with edges (e_1, e_2, \dots, e_p) in cyclic order. We show that when we move from point \mathbf{x} along the cycle C , we remain inside the perfect matching polytope. This technique of moving along the cycle has been used by Mahajan and Varadarajan [MV00]. To elaborate, consider a new point $\mathbf{y} = (y_e)_e \in \mathbb{R}^m$ such that for all $e \in E$,

$$y_e = \begin{cases} x_e + (-1)^i \varepsilon, & \text{if } e = e_i, \text{ for some } 1 \leq i \leq p, \\ x_e, & \text{otherwise,} \end{cases}$$

for some $\varepsilon > 0$. We choose ε to be small enough so that $y_e \geq 0$ for all $e \in E$. Clearly, the vector $\mathbf{x} - \mathbf{y}$ has nonzero coordinates only on cycle C with alternating ε and $-\varepsilon$. Hence,

$$|w(\mathbf{x} - \mathbf{y})| = \varepsilon \cdot c_w(C). \quad (9)$$

We argue that \mathbf{y} fulfills the conditions of Lemma 2.1 and therefore also lies in the perfect matching polytope. Because $y_e \geq 0$ for all $e \in E$, it satisfies inequality (4) from Lemma 2.1. It remains to show that \mathbf{y} also satisfies

$$\sum_{e \in \delta(v)} y_e = 1 \quad v \in V. \quad (10)$$

To see this, let $v \in V$. We consider two cases:

1. $v \notin C$. Then $y_e = x_e$ for each edge $e \in \delta(v)$. Thus, we get (10) from equation (3) for \mathbf{x} .
2. $v \in C$. Let e_j and e_{j+1} be the two edges from C which are incident on v . By definition, $y_{e_j} = x_{e_j} + (-1)^j \varepsilon$ and $y_{e_{j+1}} = x_{e_{j+1}} + (-1)^{j+1} \varepsilon$. For any other edge $e \in \delta(v)$, we have $y_e = x_e$. Combining this with equation (3) for \mathbf{x} , we get that \mathbf{y} satisfies (10) for v .

We conclude that \mathbf{y} lies in the polytope $\text{PM}(H)$. Hence, $w(\mathbf{y}) = q$. It follows that $w(\mathbf{x} - \mathbf{y}) = w(\mathbf{x}) - w(\mathbf{y}) = q - q = 0$. From equation (9), we conclude that $c_w(C) = 0$. \square

We apply Lemma 3.3 to graph G_1 . Recall that by our weight function, each small cycle in G has a nonzero circulation. Therefore G_1 has no small cycles. The following lemma states this in a slightly more general form which will be required later.

Lemma 3.4. *Let G be a bipartite graph with weight assignment w such that some cycles C_1, C_2, \dots, C_t in G have nonzero circulations. Then the graph G_1 , obtained by the union of minimum weight perfect matchings in G , does not have any of the cycles C_1, C_2, \dots, C_t .*

We want to argue that the number of edges in G_1 has significantly decreased from that in G . We use the following result of Alon, Hoory, & Linial [AHL02] which states that the nodes of graphs with no small cycles have a small average degree.

Theorem 3.5 ([AHL02]). *Let H be a graph with n nodes, average degree $d \geq 2$ and girth g . Then,*

$$n \geq 2(d - 1)^{g/2-1}.$$

The proof is fairly easy for graphs where each node has degree $\geq d$: do a breadth-first search of the graph starting from an arbitrary node until depth $g/2 - 1$. When one reaches a node v via an edge e , there are $\geq d - 1$ edges incident on v other than e . So, the search looks like a $(d - 1)$ -ary tree of depth $g/2 - 1$. As there are no cycles of length $< g$, all the nodes in the tree should be distinct, which are $\geq 2(d - 1)^{g/2-1}$. This gives us the desired bound. Alon et al. generalized the argument to *average degree $\geq d$* .

If we put $g = 4 \log n - 2$ in Theorem 3.5 and take logarithms on both sides, we get

$$\begin{aligned} \log n &\geq 2(\log n - 1) \log(d - 1) + 1 \\ 1 &\geq 2 \log(d - 1) \\ 2 &\geq (d - 1)^2 \\ 1 + \sqrt{2} &\geq d. \end{aligned}$$

Hence, the length of small cycles in our weighting scheme is chosen to be $4 \log n - 2$. Then G_1 has average degree < 2.5 .

Corollary 3.6. *Let H be a graph with girth $g \geq 4 \log n - 2$. Then H has average degree < 2.5 .*

It follows that at least a constant fraction of the nodes in G_1 have degree ≤ 2 . Hence, one can say that the graph G_1 has significantly fewer edges than G . Now, the plan is to repeat this procedure for G_1 with a new weight function. However, G_1 has no small cycles. The standard cycle length counts the number of nodes in a cycle. The idea now is to count instead the number of degree > 2 nodes in a cycle. This can also be viewed as contracting degree 2 nodes with their neighbors and then considering again small cycles. When we give nonzero circulations to these cycles in G_1 and again take the union of minimum weight perfect matchings, these cycles disappear from G_1 . This, in turn, means that the number of degree > 2 nodes further reduces by a constant fraction. We continue this for $O(\log n)$ rounds until all cycles disappear and a single perfect matching remains. Now, we put all the ingredients together and formally define our weight assignment.

3.2 Constructing the Weight Assignment

Let $G(V, E) = G_0$ be the given bipartite graph with n nodes. Define $\ell = 4 \log n - 2$. Let w_0 be a weight function that gives nonzero circulation to all cycles containing $< \ell$ nodes of

degree > 2 in G_0 . Let k be the number of rounds for which we need to construct weight assignments. We will define subgraphs G_i and weight assignments w_i , for $i = 1, 2, \dots, k$, which will be obtained in successive rounds.

G_i : the union of minimum weight perfect matchings in G_{i-1} according to weight w_{i-1} .

w_i : a weight function such that any cycle in G_i with $< \ell$ nodes of degree > 2 has nonzero circulation.

Our final weight function w will be a combination of w_0, w_1, \dots, w_k . We combine them in a way that the weight assignment in a later round does not interfere with the order of perfect matchings given by earlier round weights. Let B be a number greater than the weight of any edge under any of these weight assignments. Then, define

$$w = w_0 B^k + w_1 B^{k-1} + \cdots + w_k B^0. \quad (11)$$

Clearly, the precedence decreases from w_0 to w_k . That is, for any two perfect matchings M_1 and M_2 in G_0 , we have $w(M_1) < w(M_2)$, if and only if there exists an $0 \leq i \leq k$ such that

$$\begin{aligned} w_j(M_1) &= w_j(M_2), & \text{for } j < i, \\ w_i(M_1) &< w_i(M_2). \end{aligned}$$

The number k of rounds is chosen to be the minimum number such that the number of degree > 2 nodes in the graph G_k is less than ℓ . We will see later that $k = O(\log n)$. First, we prove that the weight assignment w is isolating in G_0 .

Lemma 3.7. *The weight assignment w defined in (11) is isolating for G_0 .*

Proof. We split the proof in a sequence of claims. By the definition of the graphs G_i and the weight functions w_i , we have:

Claim 1. *For any $1 \leq i \leq k$, any two perfect matchings in G_i have the same weight according to w_j , for all $j < i$.*

We first prove the isolation in the last round.

Claim 2. *w is isolating in G_k .*

Proof. As the number of nodes in G_k with degree > 2 is $< \ell$, the weight assignment w_k gives a nonzero circulation to each cycle in G_k . From Lemma 3.4, the union of minimum weight perfect matchings in G_k has no cycles. In other words, it has a unique minimum weight perfect matching according to w_k .

From Claim 1, all perfect matchings in G_k have equal weight according to w_j , for all $j < k$. Thus, G_k has a unique minimum weight perfect matching according to w . This proves Claim 2. \square

Next we claim that the final weight w of the perfect matchings is decreasing in subsequent rounds.

Claim 3. *For any $1 \leq i \leq k$, let M_1 be a perfect matching in G_i and M_2 be a perfect matching in G_{i-1} which is not in G_i . Then $w(M_1) < w(M_2)$.*

Proof. From the construction of G_i and Lemma 3.2, it follows that

$$w_{i-1}(M_1) < w_{i-1}(M_2).$$

From Claim 1, we have $w_j(M_1) = w_j(M_2)$ for all $j < i - 1$. Due to the precedence of the weight assignments, we get that $w(M_1) < w(M_2)$. This proves Claim 3. \square

From Claim 3 we get that all the minimum weight perfect matchings in G_0 according to w come from G_k . Putting this together with Claim 2, we get that w is isolating in G_0 . This proves the lemma. \square

Next we want to give a bound on the number k of rounds we need. We show that in each round the number of nodes with degree > 2 decreases by half. For some $i \geq 1$, let $U \subseteq V$ be the set of degree > 2 nodes in G_{i-1} . Now, w_{i-1} assigns nonzero circulation to every cycle in G_{i-1} which has less than ℓ nodes from U . From Lemma 3.4, the union of minimum weight perfect matchings (i.e., G_i) has none of these cycles. Now, we want to show that at least half of the nodes in U have degree ≤ 2 in G_i .

Lemma 3.8. *Let $G(V, E)$ be a graph with n nodes. Let $U \subseteq V$ be its set of degree > 2 nodes. Let $G_1(V, E_1)$ be a matching-covered subgraph of G such that any cycle in G_1 contains at least $\ell = 4 \log n - 2$ nodes from U . Then at least half of the nodes in U have degree ≤ 2 in G_1 .*

Proof. Let $T = V - U$ be the set of nodes of degree ≤ 2 in G . In the following, by *degree of a node* we mean its degree in G_1 .

Observe first that any node of degree 1 in G_1 can only be connected to a node of degree 1. That is, they form a connected component in G_1 that consists of a single edge. This is because G_1 is matching-covered. We delete all degree 1 nodes from G_1 .

For the degree 2 nodes in T , let us identify them with one of their two neighbors. In more detail, let $(u_0, u_1, u_2, \dots, u_{p+1})$ be a path in G_1 , for some $p \geq 1$, such that the nodes u_0, u_{p+1} are in U and the nodes u_1, u_2, \dots, u_p are from T and have degree 2. Delete the nodes u_1, u_2, \dots, u_p and add an edge (u_0, u_{p+1}) as shown in Figure 2. Note that u_0 and u_{p+1} cannot have another path with nodes only coming from T , because by our assumption, any cycle in G_1 has many nodes from U .



Figure 2: A degree 2 node contracted with its neighbor.

In summary, we deleted all nodes in T and the nodes of degree 1 in U from G_1 . Let G'_1 be the resulting graph. Hence, the nodes in G'_1 are exactly those nodes in U whose degree is > 1 in G_1 . Note that the degree of any node in G'_1 is the same as in G_1 , as one can see in Figure 2.

By the assumption of the lemma, any cycle in G'_1 has length $\geq \ell$. From Corollary 3.6 we have that G'_1 has average degree < 2.5 . As G'_1 does not have any degree 1 nodes, it follows that at least half of its nodes have degree 2. This, in turn, means that at least half the nodes in U have degree ≤ 2 in G_1 . \square

We apply Lemma 3.8 to G_{i-1} and G_i .

Corollary 3.9. *For each $1 \leq i \leq k$, the number of degree > 2 nodes in G_i is at most half of that in G_{i-1} .*

We want a value of k such that G_k has fewer than ℓ nodes of degree > 2 . Clearly, $k = \log n - 1$ suffices. Now, the only thing that remains is to bound the values of the weights assigned. First we bound the number of cycles which have a small number of degree > 2 nodes.

Lemma 3.10. *In a graph with n nodes, the number of cycles with $< \ell$ nodes of degree > 2 is $\leq n^\ell$.*

Proof. We have n choices for the first node of the cycle. If the first node has degree 2 then the cycle has to contain both its neighbors. Hence, there is a unique choice for the second node. Otherwise there are at most n choices for the second node. In general, if the recently chosen node has degree 2, then there is just one choice for the next node. Otherwise there are at most n choices. As there are fewer than ℓ nodes of degree > 2 in the cycle, the number of choices made will be bounded by $n^{\ell-1}$. Combined with the choices for the first node we get the bound of n^ℓ . \square

It remains to bound the number of weight assignments we need to try and the values of the weights. From Lemma 3.10, we get that w_i needs to give nonzero circulation to at most $n^{4\log n - 2}$ cycles for each $0 \leq i \leq k$. Hence, the set of weight assignments we get from Lemma 2.3 has size $O(n^{4\log n})$ and the weights involved are bounded by $O(n^{4\log n})$. For each weight assignment w_i , one needs to try all choices given by Lemma 2.3, as we do not know which one would work. This means, in total, we need to try $O(n^{4(k+1)\log n}) = O(n^{4\log^2 n})$ weight assignments. Recall that the number B is the highest weight assigned by any w_i . Hence, we have $B = O(n^{4\log n})$. Therefore the weights in the assignment w in equation (11) are bounded by $n^{O(\log^2 n)}$. In other words, the weights have $O(\log^3 n)$ bits. Moreover, it is straightforward to see that these weight assignments can be constructed in quasi-NC¹ with circuit size $2^{O(\log^3 n)}$.

Lemma 3.11. *In quasi-NC¹, one can construct a set of $O(n^{4\log^2 n})$ weight functions $[n] \times [n] \rightarrow \mathbb{N}$, where the weights have $O(\log^3 n)$ bits, such that for any given bipartite graph with $2n$ nodes, one of the weight functions is isolating.*

Up to this point, we can conclude that the existence of a perfect matching in bipartite graphs can be decided in quasi-NC². But we can even construct a perfect matching: recall the construction algorithm of Mulmuley et al. [MVV87] from Section 2.2. They compute the determinant of the bi-adjacency matrix that has entry $2^{w(e)}$ for edge e . As our weights have $O(\log^3 n)$ bits, the determinant entries have quasi-polynomial bits. This determinant can still be computed in parallel, but with quasi-polynomial $(2^{O(\log^3 n)})$ size circuits. In particular, it can be computed in quasi-NC² by the algorithm of Berkowitz [Ber84]. Thus, our construction algorithm is in quasi-NC² with circuit size $2^{O(\log^3 n)}$. This finishes the proof of Theorem 3.1.

4 Bipartite Planar Graphs

The SEARCH-PM problem already has some known NC-algorithms in the case of bipartite planar graphs [MN95, MV00, DKR10]. The one by Mahajan and Varadarajan [MV00] is in NC^3 , while the other two are in NC^2 . Our approach from the previous section can be modified to give an alternate NC-algorithm for this case, in NC^3 however.

The weights in our scheme in Section 3.2 become quasi-polynomial for two reasons: (i) we want nonzero circulation for all cycles of length $< 4 \log n - 2$, and (ii) we need to combine the different weight functions from $\log n$ rounds using a different scale.

To solve the first problem we observe that for the planar case, it suffices to give nonzero circulation to constant length cycles. To see this, recall that Euler's polyhedron formula gives a linear bound on the number of edges of a planar graph (see for example [Har69, Corollary 11.1(a)]).

Theorem 4.1. *Let H be a planar graph with $n \geq 3$ nodes, m edges, and girth $g \leq n$. Then*

$$m \leq \frac{g}{g-2}(n-2)$$

The case not covered by Theorem 4.1 is when H has no cycles. Then we have $m \leq n-1$. Recall that the average degree of a graph is $d = 2m/n$. By Theorem 4.1, we get

$$d = \frac{2m}{n} \leq \frac{2g}{g-2} \frac{n-2}{n} < \frac{2g}{g-2}. \quad (12)$$

We want $d < 2.5$. By equation (12) this holds for $g \geq 10$. Note also that if a graph has no cycles, then the average degree is < 2 . Hence, we get the following consequence for planar graphs, which is analogous to Corollary 3.6.

Corollary 4.2. *Let H be a planar graph with girth $g \geq 10$. Then H has average degree < 2.5 .*

Since there are only polynomially many cycles of length < 10 , one can ensure nonzero circulations for them with polynomially bounded weights by Lemma 2.3.

To solve the second problem, we use the fact that in planar graphs, one can count the number of perfect matchings of a given weight in NC^2 by the Pfaffian orientation technique [Kas67, Vaz89]. As a consequence, we can actually construct the graphs G_i in each round in NC^2 . Thereby we avoid having to combine the weight functions from different rounds.

In more detail, in the i -th round, we need to compute the union of minimum weight perfect matchings in G_{i-1} according to w_{i-1} . For each edge e , we decide in parallel if deleting e reduces the count of minimum weight perfect matchings. If yes, then edge e should be present in G_i . As it takes $\log n$ rounds to reach a single perfect matching, the algorithm is in NC^3 .

5 An RNC-Algorithm with Few Random Bits

We can also present our result for bipartite perfect matching in an alternate way. For PM, instead of quasi-NC, we can get an RNC circuit but with only poly-logarithmically many random bits.

Theorem 5.1. *For bipartite graphs, there is an RNC-algorithm for PM which uses $O(\log^3 n)$ random bits.*

The rest of this section is devoted to prove Theorem 5.1. In our algorithm from Section 3, there are two reasons that we need quasi-polynomially large circuits: (i) we need to try quasi-polynomially many different weight assignments and (ii) each weight assignment has quasi-polynomially large weights.

To solve the first problem, we modify Lemma 2.3 to get a random weight assignment which works with high probability.

Lemma 5.2 ([CRS95, KS01]). *Let G be a graph with n nodes. There is a random weight assignment on the edges which uses $O(\log ns)$ random bits and assigns weights with $O(\log ns)$ bits such that for any set of s cycles, it gives nonzero circulation to each of the s cycles, with probability at least $1 - 1/n$.*

Proof. We follow the construction of Lemma 2.3 and give exponential weights modulo small numbers. Here, we use only prime numbers as moduli. Recall the weight function w defined by $w(e_i) = 2^{i-1}$. Let us choose a random number p among first t prime numbers. We take our random weight assignment to be $w \bmod p$. We want to show that with high probability this weight function gives nonzero circulation to every cycle in $\{C_1, C_2, \dots, C_s\}$. In other words, $\prod_{i=1}^s c_w(C_i) \not\equiv 0 \pmod{p}$. As the product is bounded by $2^{n^2 s}$, it has at most $n^2 s$ prime factors. Let us choose $t = n^3 s$. This would mean that a random prime works with probability at least $(1 - 1/n)$. As the t -th prime can only be as large as $2t \log t$, the weights are bounded by $2t \log t = O(n^3 s \log ns)$, and hence have $O(\log ns)$ bits. A random prime with $O(\log ns)$ bits can be constructed using $O(\log ns)$ random bits (see [KS01]). \square

Recall from Section 3.2 that for a bipartite graph G with n nodes, we had $k = \log n$ rounds and constructed one weight function in each round. We do the same here, however, we use the random scheme from Lemma 5.2 to choose the weight functions w_0, w_1, \dots, w_k . The probability that all of them provide nonzero circulation on their respective set of cycles $\geq 1 - (k+1)/n \geq 1 - \log n/n$ using the union bound.

Now, instead of combining them to form a single weight assignment, we use a different variable for each weight assignment. We modify the construction of matrix A from Section 2.2. For variables x_0, x_1, \dots, x_k , define

$$A(i, j) = \begin{cases} x_0^{w_0(e)} x_1^{w_1(e)} \cdots x_k^{w_k(e)}, & \text{if } e = (u_i, v_j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

From arguments similar to those in Section 2.2, one can write

$$\det(A) = \sum_{M \text{ perfect matching in } G} \operatorname{sgn}(M) x_0^{w_0(M)} x_1^{w_1(M)} \cdots x_k^{w_k(M)},$$

where $\operatorname{sgn}(M)$ is the sign of the corresponding permutation. From the construction of the weight assignments it follows that if the graph has a perfect matching then the lexicographically minimum term in $\det(A)$ comes from a unique perfect matching. Thus, we get the following lemma.

Lemma 5.3. $\det(A) \neq 0 \iff G \text{ has a perfect matching.}$

Recall that by Lemma 3.10, each w_i needs to give nonzero circulation to $O(n^{4 \log n - 2})$ cycles. Thus, the weights obtained by the scheme of Lemma 5.2 will use $O(\log^2 n)$ bits.

Hence, the polynomial computed by $\det(A)$ will have quasi-polynomially degree. To avoid this, we split the weights into $\log n$ bit chunks of length $r = O(\log n)$ each. For $0 \leq i \leq k$ and $0 \leq j \leq r$, define w_{ij} 's to be the unique weight functions which satisfy the following two conditions for each edge e :

$$\begin{aligned} w_{ij}(e) &< n, \\ w_i(e) &= \sum_{j=0}^r w_{ij}(e) n^j. \end{aligned}$$

For variables $\{y_{ij}\}_{i,j}$, define the matrix A' by

$$A'(i, j) = \begin{cases} \prod_{i=0}^k \prod_{j=0}^r y_{ij}^{w_{ij}(e)}, & \text{if } e = (u_i, v_j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, substituting $y_{ij} = x_i^{n^j}$ in A' gives us A . Hence, we get the following.

Lemma 5.4. $\det(A') \neq 0 \iff \det(A) \neq 0$.

Now, $\det(A')$ is a polynomial of individual degree $< n^2$ with $O(\log^2 n)$ variables, as both k and r are $O(\log n)$. To test if $\det(A')$ is nonzero one can apply the standard randomized polynomial identity test [Sch80, Zip79, DL78]. That is, to plug-in random values for variables y_{ij} , independently from $\{1, 2, \dots, n^3\}$. If $\det(A') \neq 0$, then the evaluation is nonzero with high probability.

Number of random bits: For a weight assignment w_i , we need $O(\log ns)$ random bits from Lemma 5.2, where s is $O(n^{4\log n - 2})$ by Lemma 3.10. Thus, the number of random bits required for all w_i 's together is $O(k \log^2 n) = O(\log^3 n)$. Finally, we need to plug-in $\log n$ random bits for each y_{ij} . This again requires $O(\log^3 n)$ random bits.

Complexity: The weight construction involves taking exponential weights modulo small primes by Lemma 5.2. Primality testing can be done in NC, as the numbers involved have $O(\log^2 n)$ bits [Rab80]. Thus, the weight assignments can be constructed in NC. Moreover, the determinant with polynomially bounded entries can be computed in NC.

In summary, we get an RNC-algorithm that uses $O(\log^3 n)$ random bits as claimed in Theorem 5.1.

Discussion

The major open question remains whether one can do isolation with *polynomially bounded* weights. Our construction requires quasi-polynomial weights for two reasons: (i) we work with $O(\log n)$ -length cycles and (ii) it takes $\log n$ rounds to reach a unique perfect matching and the graphs obtained in the successive rounds cannot be constructed. To get polynomially bounded weights one needs to circumvent both these problems.

For non-bipartite graphs, the isolation question is open even in the planar case. For this case, our approach fails in its first step: Lemma 3.2 no longer holds as demonstrated in Figure 1. Can one assign weights in a way which ensures that the union of minimum weight perfect matchings is significantly smaller than the original graph?

It needs to be investigated if our ideas can lead to isolation in other objects. For example, isolation of paths in a directed graph, which is related to the NL versus UL question.

Acknowledgements

We would like to thank Manindra Agrawal and Nitin Saxena for their constant encouragement and very helpful discussions. We thank Arpita Korwar for discussions on some techniques used in Section 5.

References

- [AHL02] Noga Alon, Shlomo Hoory, and Nathan Linial. The moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002.
- [AHT07] Manindra Agrawal, Thanh Minh Hoang, and Thomas Thierauf. The polynomially bounded perfect matching problem is in NC^2 . In *24th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *Lecture Notes in Computer Science*, pages 489–499. Springer Berlin Heidelberg, 2007.
- [AM08] Vikraman Arvind and Partha Mukhopadhyay. Derandomizing the isolation lemma and lower bounds for circuit size. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop, APPROX, and 12th International Workshop, RANDOM*, pages 276–289, 2008.
- [Bar92] David A. Mix Barrington. Quasipolynomial size circuit classes. In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, pages 86–93, 1992.
- [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.
- [CK97] Zhi-Zhong Chen and Ming-Yang Kao. Reducing randomness via irrational numbers. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing (STOC)*, pages 200–209, New York, NY, USA, 1997. ACM.
- [CRS95] Suresh Chari, Pankaj Rohatgi, and Aravind Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM Journal on Computing*, 24(5):1036–1050, 1995.
- [DK98] Elias Dahlhaus and Marek Karpinski. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Applied Mathematics*, 84(13):79 – 91, 1998.
- [DKR10] Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory of Computing Systems*, 47:737–757, 2010.
- [DL78] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978.
- [Edm65] Jack Edmonds. Path, trees, and flowers. *Canadian Journal of Mathematics*, 17:449467, 1965.

- [GK87] Dima Grigoriev and Marek Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC (extended abstract). In *28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 166–172, 1987.
- [Har69] Frank Harary. *Graph Theory*. Addison-Wesley Series in Mathematics. Addison Wesley, 1969.
- [Kas67] Pieter W. Kasteleyn. Graph theory and crystal physics. *Graph Theory and Theoretical Physics*, pages 43–110, 1967.
- [KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, pages 216–223, 2001.
- [KUW86] Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986.
- [Lov79] László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Computation Theory*, pages 565–574, 1979.
- [LP86] László Lovász and Michael D. Plummer. *Matching Theory*. (North-Holland mathematics studies). Elsevier Science Ltd, 1986.
- [MN95] Gary L. Miller and Joseph Naor. Flow in planar graphs with multiple sources and sinks. *SIAM Journal on Computing*, 24:1002–1017, 1995.
- [MV00] Meena Mahajan and Kasturi R. Varadarajan. A new NC algorithm for finding a perfect matching in bipartite planar and small genus graphs (extended abstract). In *STOC’00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 351–357, New York, NY, USA, 2000. ACM.
- [MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.
- [Nai82] Mohan Nair. On chebyshev-type inequalities for primes. *The American Mathematical Monthly*, 89(2):126–129, 1982.
- [Rab80] Michael O. Rabin. A probabilistic Algorithm for Testing Primality. *Journal of Number Theory*, 1980.
- [Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.
- [TV12] Raghunath Tewari and N. V. Vinodchandran. Green’s theorem and isolation in planar graphs. *Information and Computation*, 215:1–7, 2012.
- [Vaz89] Vijay V. Vazirani. NC algorithms for computing the number of perfect matchings in $K_{3,3}$ -free graphs and related problems. *Information and Computing*, 80(2):152–164, 1989.

- [Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226. Springer-Verlag, 1979.