On the Matching Problem for Special Graph Classes

Thanh Minh Hoang
Institute for Theoretical Computer Science
University of Ulm, Germany
Email: thanh.hoang@uni-ulm.de

***Abstract*—An even cycle in a graph is called *nice* by Lovász and Plummer in [LP86] if the graph obtained by deleting all vertices of the cycle has some perfect matching. In the present paper we prove some new complexity bounds for various versions of problems related to perfect matchings in graphs with a polynomially bounded number of nice cycles.**

We show that for graphs with a polynomially bounded number of nice cycles the perfect matching decision problem is in SPL, it is hard for FewL, and the perfect matching construction problem is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}} \cap \oplus\mathbf{L}$. Furthermore, we significantly improve the best known upper bounds, proved by Agrawal, Hoang, and Thierauf in the STACS'07-paper [AHT07], for the polynomially bounded perfect matching problem by showing that the construction and the counting versions are in $\mathbf{C}_=\mathbf{L} \cap \oplus\mathbf{L}$ and in $\mathbf{C}_=\mathbf{L}$, respectively. Note that $\mathrm{SPL}, \oplus\mathrm{L}, \mathbf{C}_=\mathbf{L}$, and $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$ are contained in NC^2.

Moreover, we show that the problem of computing a maximum matching for bipartite planar graphs is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$. This solves Open Question 4.7 stated in the STACS'08-paper by Datta, Kulkarni, and Roy [DKR08] where it is asked whether computing a maximum matching even for bipartite planar graphs can be done in NC. We also show that the problem of computing a maximum matching for graphs with a polynomially bounded number of even cycles is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$.

***Keywords*-Perfect matchings, maximum matchings, NC-computations.**

I. INTRODUCTION

A set M of edges in an undirected graph G such that no two edges of M share a vertex is called a matching in G. A matching with maximal size is called a maximum matching. A maximum matching is perfect if it covers all vertices in the graph. Graph matchings because of their fundamental properties are one of the most fundamental and well-studied objects in mathematics and in theoretical computer science (see e.g. [LP86], [KR98]). In the wide research-topic on graph matchings, perfect matchings and maximum matchings w.r.t. parallel computations receive a great attention.

From the viewpoint of complexity theory it is well-known that a maximum matching can be constructed efficiently in polynomial time [Edm65]. Hence the problem of deciding whether a graph has a perfect matching (short: DECISION-PM) and the problem of computing a perfect matching in a graph (short: SEARCH-PM) are in $\mathbf{P}$. Regarding parallel computations, computing a maximum matching is known to be in randomized $\mathbf{NC}$ [KUW86], [MVV87], and particularly in nonuniform $\mathbf{SPL}$ [ARZ99] (see Section II for more detail on the complexity classes). Therefore, both problems DECISION-PM and SEARCH-PM are in nonuniform $\mathbf{SPL}$. But it is an important open question whether even DECISION-PM is in uniform $\mathbf{NC}$. Note that if SEARCH-PM would be in $\mathbf{NC}$ then also DECISION-PM. Note further that there is a huge gap among the complexities of the search and the counting version of the perfect matching problem (short: COUNTING-PM) because computing the number of all perfect matching in a bipartite graph is known to be #$\mathbf{P}$-complete [Val79].

By Tutte's Theorem [Tut47] (see next section for more detail), DECISION-PM can be reduced to the problem of testing if a symbolic determinant is zero. This algebraic setting puts DECISION-PM into a special case of the well-studied problem *Polynomial Identity Testing* (short: PIT), the problem of testing if a polynomial given in an implicit form, like an arithmetic circuit or a symbolic determinant, is zero. PIT can be solved by a randomized algorithm using the Schwartz-Zippel Lemma [Sch80], [Zip79], but whether the method can be derandomized is a prominent open question. Due to a result by Impagliazzo and Ka-

Supported by DFG grant HO 4337/1-1.

banets [KI04] stating that the problem of derandomizing PIT is computationally equivalent to the problem of proving lower bounds for arithmetic circuits, the matching problem attracts great attention.

In this paper we continue with the line of research that tries to characterize exactly the complexity of the matching problem. The motivation for the work comes directly from the crucial importance of the matching problem mentioned above. Since it is open whether the perfect matching problem is in $\mathbf{NC}$, diverse special cases of the problem have been studied and solved before. For example, $\mathbf{NC}$ algorithms are known for DECISION-PM for planar graphs [Kas67], [Vaz89] and for $K_{3,3}$-free graphs [Vaz89]. SEARCH-PM is also known in $\mathbf{NC}$ for regular bipartite graphs [LPV81], strongly chordal graphs [DK86], dense graphs [DHK93], for bipartite planar graphs [MN95], [MV00], [DKR08], for $K_{3,3}$-free bipartite graphs [KMV08], and for graphs with a polynomially bounded number of perfect matchings [GK87], [AHT07].

In the first part of the paper, in Section III, we investigate the complexity of the perfect matching problem for graphs with a polynomially bounded number of so-called *nice cycles*. An even cycle C in a graph G is called nice [LP86] if the graph obtained from G by deleting all vertices of C has some perfect matching. The nice cycles play a crucial role for deterministic isolations of perfect matchings (see Lemma 3.1 in Section III), thereby a deterministic isolation in $\mathbf{NC}$ would bring both the decision and search versions of the perfect matching problem in $\mathbf{NC}$. Thus, towards a derandomization of the perfect matching problem, our considered promise problems in Section III is not purposeless. Moreover, this promise problem is a generalization of the polynomially bounded perfect matching problem which has been studied in [GK87], [AHT07], since on the one hand the number of all nice cycles in any graph is at most the square of the number of all perfect matchings in it and on the other hand the number of all perfect matchings in a graph with a polynomially bounded number of nice cycles might be exponentially big. The results in Section III can be summarized as follows:

- Following a general paradigm for derandomizing polynomial identity testing by Agrawal [Agr03] and introducing a method different from the one in [AHT07] for solving the polynomially bounded perfect matching problem, we show that for graphs with a polynomially bounded number of nice cycles both the decision and search versions of the perfect matching problem are respectively in $\mathbf{SPL}$ and $\mathbf{L}^{\mathbf{C_{=}L}} \cap \oplus\mathbf{L}$, which are contained in $\mathbf{NC}^2$.
- We improve significantly the best known upper bounds $\mathbf{L}^{\mathbf{C_{=}L}}$ and $\mathbf{NC}^1(\mathbf{GapL})$, proved in [AHT07] for the construction and the counting versions of the polynomially bounded perfect matching problem, to $\mathbf{C_{=}L} \cap \oplus\mathbf{L}$ and $\mathbf{C_{=}L}$, respectively.

Moreover, the results and techniques presented in Section III give evidence that in general the perfect matching problem might be solvable within $\mathbf{NC}$ by a method we describe in Section V.

In the second part of the paper, in Section IV, we show an algebraic method for constructing a maximum matching once some weight function for isolating a maximum matching is given. Thereby we solve Open Question 4.7 in [DKR08] which asked whether the problem of constructing a maximum matching even for bipartite planar graphs is in $\mathbf{NC}$. In particular, we show that the maximum matching problem for bipartite planar graphs is in $\mathbf{L}^{\mathbf{C_{=}L}}$. Furthermore, using the results from Section III we show that the maximum matching problem for graphs with a polynomially bounded number of even cycles is also in $\mathbf{L}^{\mathbf{C_{=}L}}$. These results are significant because the considered problems were not known to be in $\mathbf{NC}$ previously.

II. PRELIMINARIES

A. Algebraic Graph Theory

We describe some basic notions on graph matchings. For more detail we refer the readers to [LP86], [MSV99], or to standard textbooks in linear algebra and graph theory.

Let $G = (V, E)$ be an undirected graph with n vertices, $V = \{1, 2, \ldots, n\}$, and m edges $E = \{e_1, \ldots, e_m\} \subseteq V \times V$. A *matching* in G is a set $M \subseteq E$, such that no two edges in M have a vertex in common. A matching M is called *perfect* if M covers all vertices of G, i.e. $|M| = \frac{1}{2}\, n$, M of maximal size is called *maximum*. The weight of a matching in a weighted graph is defined as the sum of all weights of the edges in the matching.

Graph G can be presented by its *adjacency matrix*. This is an $n \times n$ symmetric matrix $A \in \{0, 1\}^{n \times n}$

where $A_{i,j} = 1$ iff $(i,j) \in E$, for all $1 \leq i,j \leq n$. Assign weights $w(i,j)$ to edges (i,j) to get the *weighted* graph G. Assign orientations to the edges of weighted graph G, i.e. edge (i,j) gets one of two orientations, from i to j or from j to i, to obtain an orientation $\vec{G}$ for which we have a so-called *Tutte skew-symmetric matrix* T as follows:

$$T_{i,j} = \begin{cases} A_{i,j}\ w(i,j)\ , & \text{if an edge of } \vec{G} \\ & \text{is directed from } i \text{ to } j, \\ -A_{i,j}\ w(i,j)\ , & \text{otherwise.} \end{cases}$$

In the case when all directed edges of $\vec{G}$ are oriented from smaller to larger vertices, the orientation $\vec{G}$ and the matrix T are called *canonical*. The *Pfaffian* of a skew-symmetric matrix T from an orientation $\vec{G}$, denoted by $\text{pf}(T)$ or $\text{pf}(\vec{G}, w)$, is defined as follows:

$$\text{pf}(\vec{G}, w) \quad = \sum_{\text{perfect matching } M \text{ in } G} \text{sign}(M) \cdot \text{value}(M)$$

where $\text{sign}(M) \in \{-1, +1\}$ is the *sign* of M that depends on the orientation $\vec{G}$, and

$$\text{value}(M) = \prod_{(i,j) \in M} w(i,j)$$

is the *value* of M that depends on the weighting scheme for G. It is known from linear algebra that $\det(S) = \text{pf}^2(S)$ if S is a skew-symmetric matrix of even order, and $\text{pf}(S) = 0$ for all skew-symmetric matrices of odd order. We refer the reader to [LP86], [MSV99] for more detail.

Assign indeterminates $x_{i,j}$ to the edges (i,j) of a graph G to get the graph $G(X)$. Let $T(X)$ be the canonical Tutte skew-symmetric matrix of $G(X)$. The perfect matching problem can be decided by a randomized algorithm using the following theorem and the Schwartz-Zippel Lemma [Sch80], [Zip79].

Theorem 2.1 (Tutte [Tut47]): Graph G has no perfect matching iff $\text{pf}(T(X)) = 0$.

An orientation such that all perfect matchings in G have the same sign $+1$ (or -1) is called a *Pfaffian orientation* [Kas67]. Hence the number of perfect matchings in a graph G can be computed by finding a Pfaffian orientation in it and then by computing the Pfaffian. But there are graphs which do not admit any Pfaffian orientation, the complete bipartite graph $K_{3,3}$ is an example of them. However, planar graphs [Kas67] and $K_{3,3}$-free graphs [Vaz89] admit always Pfaffian orientations which are computable in $\mathbf{NC}$, and thus the number of all perfect matchings in such a graph can be computed efficiently.

B. Complexity Classes

The complexity classes $\mathbf{P}$, $\mathbf{L}$, $\mathbf{NP}$, and $\mathbf{NL}$ are well known. We mention briefly some other classes we work with. We refer the reader to [AO96], [ABO99], [ARZ99] for more detail.

The class $\mathbf{NC}^k$, for a fixed positive integer k, consists of families of Boolean circuit with $\wedge$-, $\vee$-gates of fan-in 2, and $\neg$ -gates, of depth $O(\log^k n)$ and of polynomial size. Moreover, we have

$$\mathbf{NC} = \bigcup_{k \geq 0} \mathbf{NC}^k.$$

The class $\mathbf{AC}^0$ is defined as the set of families of Boolean circuit with (unbounded fan-in) $\wedge$-, $\vee$-gates, and $\neg$ -gates, of constant-depth and of polynomial-size. It is know that

$$\mathbf{AC}^0 \subseteq \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC}^2 \subseteq \mathbf{NC} \subseteq \mathbf{P}.$$

For an $\mathbf{NL}$ machine M, we denote the number of accepting and rejecting computation paths on input x by $\#acc_M(x)$ and $\#rej_M(x)$, respectively. $\mathbf{FewL}$ is the class of languages accepted by $\mathbf{NL}$ machines with at most a polynomial number of accepting computations [BDHM91].

The class $\mathbf{GapL}$ consists of all functions gap_M

$$\forall x: \ gap_M(x) = \#acc_M(x) - \#rej_M(x),$$

where M is an $\mathbf{NL}$-machine. This class is characterized by the determinant of integer matrices [Dam91], [Tod91], [Vin91], [Val92]. Note that the problem of computing the determinant of an integer matrix is in $\mathbf{NC}^2$ [Ber84]. $\mathbf{GapL}$ is closed under addition, subtraction, multiplication, and restricted composition [AO96], [AAM03]. The following classes are related to $\mathbf{GapL}$.

- $\oplus\mathbf{L}$ is the class of sets A for which there exists a function $f \in \mathbf{GapL}$ such that
$$\forall x: \ x \in A \Longleftrightarrow f(x) \not\equiv 0 \pmod 2.$$
Obviously, we have $\mathbf{L}^{\oplus\mathbf{L}} = \oplus\mathbf{L}$.
- $\mathbf{C_=L}$ (*Exact Counting in Logspace*) consists of all problems of verifying a $\mathbf{GapL}$-function, i.e. it is the class of sets A for which there exists a function $f \in \mathbf{GapL}$ such that
$$\forall x: \ x \in A \Longleftrightarrow f(x) = 0.$$

$\mathbf{C_=L}$ is known to be closed under intersection and union [AO96], but it is open if it is closed under complement.

- The Hierarchy over $\mathbf{C_=L}$ collapses to $\mathbf{L}^{\mathbf{C_=L}}$ [ABO99] which is equal to $\mathbf{AC}^0(\mathbf{C_=L})$, the class of all problems $\mathbf{AC}^0$-reducible to $\mathbf{C_=L}$. The problem of computing the rank of an integer matrix is complete for $\mathbf{L}^{\mathbf{C_=L}} = \mathbf{AC}^0(\mathbf{C_=L})$ [ABO99].
- $\mathbf{SPL}$ [ARZ99] is the class of all languages for which their characteristic functions are in $\mathbf{GapL}$:

$$\mathbf{SPL} = \{L \in \Sigma^* | \chi_L \in \mathbf{GapL}\}.$$

It is known that $\mathbf{SPL}$ is closed under complement, moreover $\mathbf{L}^{\mathbf{SPL}} = \mathbf{SPL}$. Note that the inclusion $\mathbf{NL} \subseteq \mathbf{SPL}$ remains open.

We list some known inclusions among the mentioned classes:

$$\mathbf{L} \subseteq \mathbf{FewL} \subseteq \mathbf{SPL} \subseteq \mathbf{C_=L} \subseteq \mathbf{L}^{\mathbf{C_=L}} \subseteq \mathbf{NC}^2,$$
$$\mathbf{SPL} \subseteq \oplus\mathbf{L} \subseteq \mathbf{NC}^2,$$
$$\mathbf{L} \subseteq \mathbf{FewL} \subseteq \mathbf{NL} \subseteq \mathbf{C_=L},$$
$$\mathbf{L} \subseteq \mathbf{GapL} \subseteq \mathbf{NC}^2.$$

The Pfaffian of an integer skew-symmetric matrix is known to be in $\mathbf{GapL}$ [MSV99]. Given a univariate polynomial matrix $A(x)$, i.e. the elements of $A(x)$ are polynomials in x of logarithmic bit length in the degree, the problem of computing det$(A(x))$ is known to be in $\mathbf{GapL}$ [AAM03], i.e. all the coefficients of det$(A(x))$ are computable in $\mathbf{GapL}$. By following the latter and the combinatorial setting for Pfaffians in [MSV99], it is not hard to show that in the case when univariate polynomial matrix $A(x)$ is skew-symmetric, all the coefficients of pf$(A(x))$ are $\mathbf{GapL}$-computable.

We denote the decision, the search, and the counting version of the perfect matching problem by DECISION-PM, SEARCH-PM, and COUNTING-PM respectively. By SEARCH-MM we denote the problem of computing a maximum matching in a graph.

III. ISOLATING AND COMPUTING PERFECT MATCHINGS

In this section we show that the perfect matching problem for graphs with a polynomially bounded number of nice cycles is in $\mathbf{NC}^2$. It is well known that for any computational problem the decision version is reducible to the search version. Thus, in order to obtain an upper bound for the perfect matching problem we can concentrate on SEARCH-PM. Our method for searching a perfect matching consists of two standard steps:

(a) isolating a perfect matching by a weight function,
(b) computing the isolated perfect matching.

A. Isolating a Perfect Matching

Given a graph $G = (V, E)$ with n vertices $V = \{1, 2, \ldots, n\}$, m edges $E = \{e_1, e_2, \ldots, e_m\} \subseteq V \times V$, and with at most n^k nice cycles, where k is a fixed positive integer, (recall that an even cycle C in G is called nice if the graph obtained by deleting from G all vertices of C has some perfect matching or it is empty). We show how to deterministically isolate a perfect matching in G.

Let w be a weight function for the edges of G, i.e. edge e gets the weight $w(e)$, for every e. Observe that a simple cycle C (in G) with $2l$ edges, $l > 0$, has exactly two perfect matchings, for example N_1 and N_2, each of them is of size l. By $W(N_1)$ and $W(N_2)$ we denote the weights of N_1 and N_2, respectively. Recall that the weight of a matching is the sum of the weights labelled on its edges. The difference of the weights of the two perfect matchings in an even cycle is called in [DKR08] the *circulation* of the cycle, i.e.:

$$\text{circulation}(C) = |W(N_1) - W(N_2)|.$$

This function has been used in Lemma 3.2 of [DKR08] for isolating a perfect matching as follows: *if all the cycles of a bipartite graph have nonzero circulations, then the minimum weight perfect matching in it is unique*. It is straightforward to make a generalization of Lemma 3.2 in [DKR08]; for the sake of completeness we prove the following lemma.

Lemma 3.1 (Extension of Lemma 3.2 in [DKR08]): If all nice cycles in a weighted graph have nonzero circulations, then there is a unique minimum weight perfect matching in it.

Proof: Assume for a moment that there are two minimum weight perfect matchings M and N in a weighted graph G. Observe that the symmetric difference of these two perfect matchings in G is a set of vertex-disjoint cycles, i.e. we can write $M \triangle N = \{C_1, \ldots, C_l\}$ for some positive integer l, where $C_1, \ldots, C_l$ are nice. Observe further that these nice cycles are alternating cycles from M and N. For $1 \leq i \leq l$, C_i has exactly two perfect matchings, say

M_i and N_i, i.e. $C_i = M_i \cup N_i$, where M_i and N_i are respectively subsets of M and N.

If $l = 1$ then we have

$$\begin{aligned} \text{circulation}(C_1) &= |W(M_1) - W(N_1)| \\ &= |W(M) - W(N)| = 0 \end{aligned}$$

which is a contradiction because nice cycles have nonzero circulations.

In the case when $l > 1$, for $1 \leq i \leq l$, observe that $(M \setminus M_i) \cup N_i$ and $(N \setminus N_i) \cup M_i$ are also perfect matchings in G. Since circulation$(C_i) \neq 0$ one of these perfect matchings has a smaller weight than the weight of M and N. This is a contradiction to the assumption that M and N are minimum weighted. The proof of the lemma is complete.

■

By Lemma 3.1 the circulations of nice cycles play a central role for isolating a perfect matching in graphs. But note that the converse of Lemma 3.1 is not true. For example: we can easily assign integer weights to 6 edges of K_4, the complete graph with 4 vertices, so that the minimum weight perfect matching is unique but there is a nice cycle of zero circulation.

We call a weight function *admissible* for G if it assigns positive integers with a logarithmically bounded number of bits to the edges in G so that a minimum weight perfect matching becomes unique. By Lemma 3.1, in order to isolate deterministically a perfect matching we can determine an admissible weight function such that all nice cycles in the graph get nonzero circulations. We show the following lemma for isolating a perfect matching in graphs having a polynomially bounded number of nice cycles.

Lemma 3.2: Let $G = (V, E)$ be an undirected graph with $|V| = n$ vertices and m edges $E = \{e_1, e_2, \ldots, e_m\}$, and let the number of nice cycles in G be at most n^k, for some positive constant k. Then there exists a prime number $p < 2n^k(m+1)$ such that the weight function $w_p : E \mapsto \mathbb{Z}_p$ where $w_p(e_i) = 2^i \bmod p$ is admissible for G.

Proof: Assign 2^i to every edge e_i in G. Then each nice cycle C in G has a nonzero circulation because two perfect matchings defined in C have different weights. Consider the product of all the circulations:

$$Q = \prod_{C \text{ is a nice cycle}} \text{circulation}(C).$$

Since the number of nice cycles in G is at most n^k and since $0 < \text{circulation}(C) < 2^{m+1}$ holds for every nice cycle C, we get $0 < Q < 2^{n^k(m+1)}$. It is well-known from Number Theory that

$$\prod_{\text{all primes } p_i \leq 2N} p_i > 2^N, \text{ for all } N > 2.$$

Therefore, there exists a prime $p < 2n^k(m+1)$ such that p is not a factor of Q, i.e. we have $Q \bmod p \neq 0$, or equivalently:

$$\text{circulation}(C) \bmod p \neq 0$$

for all nice cycles C in G. Hence by Lemma 3.1 a minimum weight perfect matching becomes unique under the weight function $w_p : E \rightarrow \mathbb{Z}_p$ where

$$w_p(e_i) = 2^i \bmod p, \text{ for } i = 1, 2, \ldots, m.$$

Note that all the prime numbers $q < 2n^k(m+1)$ and the weight functions w_q are computable in logspace. This completes the proof of the lemma.

■

Observe that the set of all nice cycles in any graph is the union of all cycles formed up from all symmetric differences of two different perfect matchings. Hence it is easy to see that the number of all nice cycles in a graph is at most the square of the number of all perfect matchings in the graph. Therefore, Lemma 3.2 can be used also for isolating a perfect matching in graphs with polynomially bounded number of perfect matchings.

Note that it is still open if there is an **NC**-computable admissible weight function for an arbitrary graph (without any restriction of the number of nice cycles). This open question is similar to the open question of whether the Isolating Lemma [MVV87] for randomly isolating a perfect matching can be derandomized. We believe that there is an affirmative answer to this open question. In Section V we give a discussion on this topic. The Isolating Lemma can be stated in general as follows:

Lemma 3.3 (Isolating Lemma [MVV87]): Let U be a universe of size m and S be a considered family of subsets of U. Let $w : U \rightarrow \{1, \ldots, 2m\}$ be a random weight function. Then with probability at least $\frac{1}{2}$ there exists a unique minimum weight subset in S.

B. Computing a Perfect Matching

Theorem 3.4: For each fixed $k > 0$, there is a $\mathbf{GapL}$-function h_k such that on input a graph G with at most n^k nice cycles (where n is the number of vertices in G), $h_k(G) = 1$ if G has a perfect matching, and $h_k(G) = 0$ if G has no perfect matching. Furthermore, a perfect matching in G, if one exists, can be constructed in $\mathbf{L}^{\mathbf{C_=L}} \cap \oplus\mathbf{L}$.

Proof: Let $G = (V, E)$ be a graph with n vertices, m edges $E = \{e_1, \ldots, e_m\}$, and with at most n^k nice cycles, for some positive constant k. Let U be the set of all prime numbers smaller or equal to $2n^k(m+1)$. Define the weight functions $w_p : E \to \mathbb{Z}_p$, for each $p \in U$, where $w_p(e_i) = 2^i \bmod p$ for every edge e_i.

Let x be an indeterminate. Assign $x^{w_p(e)}$ to each edge e in G to get the graphs $G_p(x)$, for every $p \in U$. By $G_p^{(-e)}(x)$ we denote the result of deleting edge e from $G_p(x)$. The canonical Tutte skew-symmetric matrices of $G_p(x)$ and $G_p^{(-e)}$ we denote respectively by $T_p(x)$ and by $T_p^{(-e)}(x)$.

Considering the Pfaffian polynomials of these matrices we observe that the value of a perfect matching M becomes $x^{W(M)}$ where $W(M)$ is the weight of M, the coefficient of $x^{W(M)}$ in the polynomial is the sum of all signs of all perfect matchings having the same weight $W(M)$. Define $K = n^{k+1}(m+1)$. Obviously, all primes from U are at most K, for every $n \geq 2$. Now we can write:

$$\begin{aligned} \mathrm{pf}(T_p(x)) &= c_{p,0} + c_{p,1}x^1 + \cdots + c_{p,K}x^K, \\ \mathrm{pf}(T_p^{(-e)}(x)) &= c_{p,0}^{(-e)} + c_{p,1}^{(-e)}x^1 + \cdots + c_{p,K}^{(-e)}x^K. \end{aligned}$$

It is clear that all $\mathrm{pf}(T_p(x))$ and $\mathrm{pf}(T_p^{(-e)}(x))$ vanish if G has no perfect matching.

Consider the case when G has some perfect matching. By Lemma 3.2 there exists some $p \in U$ such that the graph G under w_p has a unique minimum weight perfect matching. Let M_0 be this unique matching and let I be its weight under w_p. Observe that the coefficient of x^I in $\mathrm{pf}(T_p(x))$, occurred as the lowest nonzero coefficient in the polynomial, should be

$$c_{p,I} = \mathrm{sign}(M_0) \in \{+1, -1\},$$

or equivalently $c_{p,I}^2 = 1$. Recall from Section II that all the coefficients of the polynomials we consider are computable in $\mathbf{GapL}$. Therefore

$$h_k(G) = 1 - \prod_{\substack{0 \leq i \leq K \\ p \in U}} (1 - c_{p,i}^2)$$

is a *zero-one-valued* $\mathbf{GapL}$-function that can be seen as the characteristic function for the problem of testing if G has a perfect matching, i.e. $h_k(G) = 1$ iff G has some perfect matching.

It remains to show that we can construct a perfect matching of G in $\mathbf{L}^{\mathbf{C_=L}} \cap \oplus\mathbf{L}$.

Observe that if w_p is admissible for G, then G has the unique minimum weight perfect matching M_0 with weight $0 \leq I \leq K$. Thus we have

$$c_{p,I}^2 = 1 \text{ and } c_{p,I}^{(-e)} = \begin{cases} 0\,, & \text{if } e \in M_0 \\ c_{p,I}\,, & \text{otherwise.} \end{cases}$$

Therefore, in $\mathbf{C_=L}$ we can construct all edge-sets $M_{p,i}$ as follows:

$$e \in M_{p,i} \text{ iff } c_{p,i}^2 = 1 \textbf{ and } c_{p,i}^{(-e)} = 0,$$

for each edge e, for all $p \in U$ and $0 \leq i \leq K$.

It is easy to see that the same edge-sets will be constructed by the same procedure in $\mathbb{Z}_2$, i.e. in $\oplus\mathbf{L}$ we can construct all the sets $M_{p,i}$. After that we can easily determine and output in logspace all perfect matchings from the constructed edge-sets $M_{p,i}$. Note that at least one edge-set, namely $M_{p,I}$ from our construction, is indeed a perfect matching in G. Our construction is in $\mathbf{L}^{\mathbf{C_=L}} \cap \oplus\mathbf{L}$ because $\mathbf{L}^{\oplus\mathbf{L}} = \oplus\mathbf{L}$. This completes the proof of the theorem.

■

Note that the formulation "DECISION-PM *for graphs with a polynomially bounded number of nice cycles is in* $\mathbf{SPL}$", used sometimes in the paper, is not completely formal. It means that, for every fixed positive integer k, there is a $\mathbf{GapL}$-function h_k such that on input a graph G with at most n^k nice cycles, where n is the number of vertices of G, $h_k(G)$ is zero-one valued and it tests the existence of perfect matching in G, but this $\mathbf{GapL}$-function h_k might be not zero-one-valued for graphs outside the considered promise. We further note that it is easy to see in the proof of Theorem 3.4 that DECISION-PM for graphs with a polynomially bounded number of nice cycles is in $\mathbf{C_=L} \cap \mathbf{coC_=L}$. (This avoids any possibility of confusion.)

Allender et al. [ARZ99] show that in general a perfect matching can be constructed in nonuniform $\mathbf{SPL}$. Unfortunately, in the proof of Theorem 3.4 we do not know how to perform in (uniform) $\mathbf{SPL}$ the decision of which prime p from U is "right" for isolating a minimum weight perfect matching.

C. The Polynomially Bounded Perfect Matching Problem

The best known upper bounds for the polynomially bounded perfect matching problem, taken from [GK87], [AHT07], are given in the following theorem.

Theorem 3.5 (Agrawal et al. [AHT07]): For graphs with a polynomially bounded number of perfect matchings, the perfect matching decision problem is in $\mathbf{coC_{=}L}$, the counting problem is in $\mathbf{AC}^0(\mathbf{C_{=}L})$, and all the perfect matchings can be constructed in $\mathbf{NC}^1(\mathbf{GapL})$.

Note that $\mathbf{coC_{=}L} \subseteq \mathbf{AC}^0(\mathbf{C_{=}L}) = \mathbf{L}^{\mathbf{C_{=}L}} \subseteq \mathbf{NC}^1(\mathbf{GapL}) \subseteq \mathbf{NC}^2$ where $\mathbf{NC}^1(\mathbf{GapL})$ is the class of all problems $\mathbf{NC}^1$-reducible to the determinant (see e.g. [AO96], [ABO99]).

Obviously, upper bounds for the decision version of the polynomially bounded perfect matching problem come directly from the preceded subsection, i.e. the decision version of the problem is in $\mathbf{SPL}$. We show the following:

Theorem 3.6: For graphs with a polynomially bounded number of perfect matchings

(1) DECISION-PM is hard for $\mathbf{FewL}$,
(2) SEARCH-PM is in $\mathbf{C_{=}L} \cap \oplus\mathbf{L}$, and COUNTING-PM is in $\mathbf{C_{=}L}$.

Proof: (1) We omit the proof that DECISION-PM is hard for $\mathbf{FewL}$ since it is straightforward by modifying the reduction from the directed connectivity problem, which is $\mathbf{NL}$-complete, to the bipartite unique perfect matching problem [HMT06], or to the bipartite perfect matching problem [CSV84].

(2) Let $G = (V, E)$ be an undirected graph with n vertices, m edges $|E| = \{e_1, \dots, e_m\}$, and with at most n^k perfect matchings, for a fixed $k > 0$. We show how to construct *all perfect matchings* in G. Our construction consists of two steps as follows:

(a) compute a prime p such that the function

$$w_p : \; w_p(e_i) = 2^i \bmod p$$

isolates all perfect matchings in G,

(b) construct all perfect matchings from the Pfaffian polynomials.

Consider Step (a). Let's call a prime p from Step (a) *"right"* if w_p isolates all perfect matchings in G. Observe that under the function $w : \; w(e_i) = 2^i$, the weights of all perfect matchings are pairwise different, i.e. we have

$$0 \; < \; |W(M) - W(N)| \; < \; 2^{m+1}$$

where $W(M)$ and $W(N)$ are the weights of perfect matching M and N, respectively. Therefore, we have the following estimation

$$\begin{aligned} 0 \;<\; Q \;&=\; \prod_{\text{all } M \neq N} |W(M) - W(N)| \\ &<\; 2^{(m+1)\binom{n^k}{2}} \\ &<\; 2^{\frac{1}{2}(m+1)n^{2k}}. \end{aligned}$$

In analogy to Lemma 3.2, define U as the set of all prime numbers smaller or equal to $(m+1)n^{2k}$. Then there exists a prime $p \in U$ such that $Q \bmod p \neq 0$, i.e. the weight function w_p defined on such a prime p isolates all perfect matchings in G. Define $K = (m+1)n^{2k}$. Then K is bigger than every prime from U. Therefore, K is also an upper bound on the degrees of the polynomial $\mathrm{pf}(T_p(x))$, i.e. we can write

$$\mathrm{pf}(T_p(x)) = c_{p,0} + c_{p,1}x^1 + \cdots + c_{p,K}x^K.$$

We observe further that a prime $p \in U$ is "right" iff in $\mathrm{pf}(T_p(x))$ *all coefficients are from* $\{-1, 0, +1\}$ **and** *the number of nonzero coefficients is maximum*. Note that the latter is the number of all perfect matchings in G when p is "right", i.e. for the "right" prime p, the $\mathbf{GapL}$-function

$$g_p = \sum_{i=0}^{K} c_{p,i}^2$$

computes the number of all perfect matchings in G. Note that if prime $p \in U$ is not "right" then anyhow we have $0 \leq c_{p,i}^2 \leq n^{2k}$ and $0 \leq g_p \leq Kn^{2k}$, for every $0 \leq i \leq K$ and $p \in U$.

Define

$$h_q \;=\; \sum_{i=0}^{K} (c_{q,i}^2 - 1)\; c_{q,i}^2,$$

for every $q \in U$. We see that $h_p = 0$ iff all coefficients in $\mathrm{pf}(T_p(x))$ are from $\{-1, 0, +1\}$. Moreover, for each

$0 \leq i \leq K$ and for each $q \in U$, we have

$$0 \leq (c_{q,i}^2 - 1)\ c_{q,i}^2 < n^{4k}$$

since the number of perfect matchings in G is at most n^k. Hence $0 \leq h_q < Kn^{4k}$ holds for every $q \in U$.

Finally, in order to determine the "right" prime p we define the following $\mathbf{GapL}$-functions

$$H_{q,q'} \;=\; \prod_{a=1}^{Kn^{4k}} (h_{q'} - a) \prod_{a=0}^{Kn^{2k}} (g_q - g_{q'} - a),$$

for each pair of primes q and q' from U. Then it is easy to see that

$$H_{q,q'} = 0 \text{ iff } h_{q'} \neq 0 \textbf{ or } g_q = g_{q'} + a,$$

where a is some nonnegative integer smaller or equal to Kn^{2k}. Moreover, in the case when $H_{q,q'} = 0$ we get $g_q > g_{q'}$ as long as $h_{q'} = 0$. Therefore, in $\mathbf{C_=L}$ we can select a "right" prime p from U as follows:

$$p \text{ is "right" iff} \quad h_p = 0 \textbf{ and } H_{p,q} = 0,$$

for all $p \neq q \in U$. Note that $\mathbf{C_=L}$ is closed under intersection and union.

Consider Step (b). In $\mathbf{C_=L}$ we can construct the edge-sets $M_{p,i}$ corresponding to $c_{p,i} \in \{-1, +1\}$ in $\text{pf}(T_p(x))$ as stated in the proof of Theorem 3.4. Note that after Step (b) we do not check again whether the constructed edge-sets are perfect matchings. This shows that all perfect matchings in G can be constructed in $\mathbf{C_=L}$.

The problem SEARCH-PM is in $\oplus\mathbf{L}$ by following the proof of Theorem 3.4.

The number of all perfect matchings in G can be computed in $\mathbf{C_=L}$ by verifying $g_p = a$, for some $a \leq n^k$, and by testing if p from U is "right".

This completes the proof of the theorem. ∎

IV. Isolating and Computing a Maximum Matching

In this section we investigate the maximum matching problem. W.l.o.g., assume that the considered graphs in this section have no perfect matching. We show the following lemma.

Lemma 4.1: Given a weight function w that assigns logarithmic bit long positive integers to the edges of a graph G such that the weight of a maximum matching in G becomes unique, the problem of computing a maximum matching in G is $\mathbf{L}^{\mathbf{C_=L}}$-reducible to the problem of computing a perfect matching in a subgraph of G.

Proof: Let $G = (V, E)$ be a graph with n vertices and m edges. Let M be a maximum matching of G, and let $|M| = l$ for some positive integer l. Suppose the weight of M is unique under the weight function w. By G_M we denote the subgraph of G, obtained by deleting $n - 2l$ vertices which are not covered by M.

Observe that the maximum matching M in G becomes perfect and unique in G_M under the weight function w. Therefore, the computation of M can be done by computing G_M and then by extracting a perfect matching in G_M.

Let x be an indeterminate. By $G(x)$ we denote the graph G by assigning $x^{w(e)}$ to every edge e of G. By this weighting scheme we obtain $G_M(x)$ from G_M. Let $T_G(x)$ and $T_{G_M}(x)$ be the canonical Tutte skew-symmetric matrix of $G(x)$ and of $G_M(x)$, respectively.

Since in G_M the weight of the perfect matching M is unique under w, the Pfaffian polynomial $\text{pf}(T_{G_M}(x))$ should be nonzero and the order of $T_{G_M}(x)$ should be $2l$. Hence we have

$$\det(T_{G_M}(x)) = \text{pf}^2(T_{G_M}(x)) \neq 0, \text{ and}$$
$$\text{rank}(T_{G_M}(x)) = 2l.$$

Moreover, since l is maximum, $T_{G_M}(x)$ is a maximum size nonsingular sub-matrix of the polynomial skew-symmetric $T_G(x)$. As a consequence we get

$$\text{rank}(T_G(x)) = \text{rank}(T_{G_M}(x)) = 2l.$$

Conversely, consider an n-bit vector $\vec{b}$ associated to a maximal set of linearly independent columns of $T_G(x)$. We call vector $\vec{b}$ a *column-basis* of $T_G(x)$. Observe that the subgraph $G_{\vec{b}}$ of G that contains all vertices i of G such that $\vec{b}_i = 1$ has always perfect matchings of the size l, and these matchings are maximum in G. Thus, in order to compute a subgraph having a perfect matching which is a maximum matching in G we can compute a column-basis of $T_G(x)$.

The problem of computing a column-basis of an integer matrix [zG93] is reducible to the rank of an integer matrix. The latter is known to be in $\mathbf{L}^{\mathbf{C_=L}}$ [ABO99]. It is not hard to show that, for polynomial matrices, i the problem of computing a column-basis is also in $\mathbf{L}^{\mathbf{C_=L}}$. Our proof of this statement consists of two following parts: a) *the problem of* *computing a column-basis is reducible to the problem of computing the rank* and b) *the rank can be computed in* $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$.

a) Given is an $n \times n$ univariate polynomial matrix $A(x)$ where the degrees of its elements are at most n^c, for some positive constant c. Let $\vec{a}_1(x), \ldots, \vec{a}_n(x)$ be its columns. One has to compute a column-basis of $A(x)$.
Let $A_i(x)$ be the matrix formed by the first i columns $\vec{a}_1(x), \ldots, \vec{a}_i(x)$ of $A(x)$, for all $1 \leq i \leq n$. It is well known from linear algebra that a column-basis can be selected as the collection of all $\vec{a}_i(x)$ where
$$\text{rank}(A_{i-1}(x)) + 1 = \text{rank}(A_i(x)),$$
for every $1 \leq i \leq n$. Therefore, the computation of a column-basis is reduced to the problem of computing the rank of a polynomial matrix.

b) Given is an $n \times m$ univariate polynomial matrix $B(x)$, where the degrees of the matrix-elements are at most n^c, for some positive constant c. One has to compute rank$(B(x))$.
It is known that 2 rank$(B(x))$ = rank$(C(x))$ where
$$C(x) = \begin{pmatrix} \mathbf{0} & B(x) \\ B^t(x) & \mathbf{0} \end{pmatrix}$$
and $B^t(x)$ is the transpose of $B(x)$. Since $C(x)$ is an $N \times N$ symmetric matrix, where $N = m + n$, we can compute rank$(C(x))$ by the characteristic polynomial
$$\chi_C(x) = \det(yI - C(x)),$$
where y is an indeterminate, as follows:
Let
$$\chi_C(x) = y^N + p_{N-1}(x)y^{N-1} + \cdots + p_0(x),$$
where $p_i(x)$ is a polynomial in x. Then for some $0 \leq j \leq N$ we have rank$(C(x)) = j$ iff
$$p_0(x) = \cdots = p_{N-j-1}(x) = 0 \textbf{ and } p_{N-j}(x) \neq 0.$$
Consider one of the polynomials $p_i(x)$. If $p_i(x) = 0$ then it is clear that $p_i(a) = 0$ for all a's. Otherwise, if $p_i(x) \neq 0$ then there exists an integer $a \in S = \{0, 1, \ldots, \deg(p_i(x))\}$ such that $p_i(a) \neq 0$ because of the fact that $p_i(x)$ has at most $\deg(p_i(x))$ real roots. Since
$$\deg(p_i(x)) \leq N\ n^c = (m + n)\ n^c,$$
for all $0 \leq i \leq N - 1$, where c is a constant, we define
$$S = \{0, 1, \ldots, (m + n)\ n^c\}.$$
Then the rank of $B(x)$ is equal to the maximum number in the set $\{$ rank$(B(a)) \mid a \in S \}$.
The rank of an integer matrix is known to be in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$ [ABO99]. Therefore, rank$(B(x))$ is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$.

The proof of the lemma is complete.

■

Obviously, Lemma 4.1 is very useful for the problem of computing a maximum matching, and for the problem of computing the *matching number* (this is the size of maximum matchings in a graph [LP86], it is not known whether the matching number can be computed in $\mathbf{NC}$). Thereby, it is clear that a deterministic isolation of some maximum matchings plays an important role. Such an isolation is known for bipartite planar graphs:

Lemma 4.2 (Datta et al. [DKR08]): In logspace one can assign polynomially bounded weights to the edges of a bipartite planar graph so that the circulation of any cycle is nonzero.

We show the following theorem for a positive answer to Open Question 4.7 stated in [DKR08] whether a maximum matching in bipartite planar graphs can be computed in $\mathbf{NC}$.

Theorem 4.3: The maximum matching problem for bipartite planar graphs is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$.

Proof: By Lemma 4.1, a subgraph G_M of a given bipartite planar graph G can be computed in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$ so that perfect matchings in G_M are maximum in G. Computing a perfect matching for bipartite planar graphs is known to be in $\mathbf{SPL}$ [DKR08]. Since $\mathbf{SPL} \subseteq \mathbf{C}_=\mathbf{L} \subseteq \mathbf{L}^{\mathbf{C}_=\mathbf{L}}$, the maximum matching problem for bipartite planar graphs is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$.

■

A method within $\mathbf{NC}$ for computing a maximum matching under the promise that the input graphs have a polynomial number of even cycles is given by the following theorem.

Theorem 4.4: The maximum matching problem for graphs with a polynomially bounded number of even cycles is in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$.

Proof: Let G be a graph with a polynomially bounded number of even cycles. In analogy to Lemma 3.2 we can show that there exists a small prime p such that all even cycles in G have nonzero circulations under $w_p : E \mapsto \mathbb{Z}_p$ where

$$w_p(e_i) = 2^i \bmod p,$$

for every edge e_i. Thus, all nice cycles in any subgraph H of G such that perfect matchings in H are maximum matchings in G have nonzero circulations under w_p. Hence by Lemma 3.1 H has a unique minimum weight perfect matching. By Lemma 4.1 such a graph H can be computed in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$. By Theorem 3.4 a perfect matching in H can be computed in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$. Therefore a maximum matching in G can be computed in $\mathbf{L}^{\mathbf{C}_=\mathbf{L}}$. The proof of the theorem is complete.

■

V. Discussion

As seen in the paper, isolations of graph matchings play a crucial role for a potential $\mathbf{NC}$ algorithm for both the decision and the search versions of the matching problem. Deterministic isolations of perfect matchings have been shown only for bipartite planar graphs [DKR08] and for graphs with a polynomially bounded number of nice cycles (the present paper). We conjecture that the method stated below can be used for isolating a perfect matching in general graphs.

Assign to each edge e_i of the graph G a polynomial $g_i(x)$ in x such that the circulation polynomial $p_C(x)$ of each even cycle C is nonzero in the ring $\mathbb{Z}[x]$. For example: $g_i(x) = a_i x^i$ for arbitrary small integers a_i. Consider $p_C(x)$ in the field $\mathbb{F} = \mathbb{Z}_P[x]/(h(x))$ where P is a small prime number and $h(x)$ is an irreducible polynomial in the polynomial ring $\mathbb{Z}_P[x]$. Since $\mathbb{F}$ has $P^{\deg(h)}$ elements, we have to choose $h(x)$ of constant degree, say $\deg(h(x)) \leq l$ for a constant l. If all the polynomials $p_C(x)$ are nonzero in $\mathbb{F}$, then there exists $a \in \mathbb{Z}_Q$, where Q is a small prime number of size at least $P^l \leq n^{kl}$, such that all the circulation-polynomials do not vanish at point a. Formally, we have

$$(p_C(x) \bmod P, h(x)) \bmod Q, x - a \neq 0$$

for all cycles C under the weight function w:

$$w(e_i) = (a_i x^i \bmod P, h(x)) \bmod Q, x - a,$$

for every edge e_i, for $i = 1, 2, \ldots, m$.

The main problem we have to solve is how to define $g_i(x), h(x)$, and P such that $p_C(x)$ is in $\mathbb{F} \setminus 0$ for every nice cycle C. A positive answer to this question would give a deterministic isolation as described. Note that the described isolation works for bipartite planar graphs.

Acknowledgment

The author wishes to thank Meena Mahajan, Prajakta Nimbhorkar, and Jacobo Torán for fruitful discussions on the topic of the paper. The author thanks Thomas Thierauf for comments on possibility of confusion of the formulation of the promise problems being in $\mathbf{SPL}$.

The author is grateful to the CCC anonymous referees for their comments and suggestions.

This research was supported by DFG grant HO 4337/1-1.

References

[AAM03] E. Allender, V Arvind, and M. Mahajan. Arithmetic complexity, Kleene closure, and formal power series. *Theory of Computing Systems*, 36(4):303–328, 2003.

[ABO99] E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8:99–126, 1999.

[Agr03] M. Agrawal. On derandomizing tests for certain polynomial identities. In *18th Annual IEEE Conference on Computational Complexity (CCC)*, page 355. IEEE Computer Society, 2003.

[AHT07] M. Agrawal, T. M. Hoang, and T. Thierauf. The polynomially bounded perfect matching problem is in $\mathbf{NC}^2$. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS, pages 489–499. Springer Verlag, 2007.

[AO96] E. Allender and M. Ogihara. Relationship among $\mathbf{PL}$, $\#\mathbf{L}$, and the determinant. *RAIRO-Theoretical Informatics and Applications*, 30:1–21, 1996.

[ARZ99] E. Allender, K. Reinhardt, and S. Zhou. Isolating, matching, and counting: uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59:164–181, 1999.

[BDHM91] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and importance of logspace-mod-classes. In *8th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 360–371. Springer, 1991.

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

[CSV84] A. K. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SIAM Journal on Computing*, 13(2):423–439, 1984.

[Dam91] C. Damm. DET = $L^{(\#L)}$. Technical Report Informatik-Preprint 8, Fachbereich Informatik der Humboldt Universitaet zu Berlin, 1991.

[DHK93] E. Dahlhaus, P. Hajnal, and M. Karpinski. On the parallel complexity of hamiltonian cycles and matching problem in dense graphs. *Journal of Algorithms*, 15:367–384, 1993.

[DK86] E. Dahlhaus and M. Karpinski. The matching problem for strongly chordal graphs is in $\mathbf{NC}$. Technical Report 855-CS, University of Bonn, 1986.

[DKR08] S. Datta, R. Kulkarni, and S. Roy. Deterministically isolating a perfect matching in bipartite planar graphs. In *25th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS, pages 229–240. Springer Verlag, 2008.

[Edm65] J. Edmonds. Maximum matching and a polyhedron with 0-1 vertices. *Journal of Research National Bureau of Standards*, 69:125–130, 1965.

[GK87] D. Grigoriev and M. Karpinski. The matching problem for bipartite graphs with polynomially bounded permanent is in $\mathbf{NC}$. In *28th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 166–172. IEEE Computer Society Press, 1987.

[HMT06] T. M. Hoang, M. Mahajan, and T. Thierauf. On the bipartite unique perfect matching problem. In *33rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 453–464. Springer-Verlag, 2006.

[Kas67] P. W. Kasteleyn. Graph theory and crystal physics. In F. Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.

[KI04] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[KMV08] R. Kulkarni, M. Mahajan, and K. R. Varadarajan. Some perfect matchings and perfect half-integral matchings in nc. *Chicago Journal of Theoretical Computer Science (CJCS)*, 2008(4), 2008.

[KR98] M. Karpinski and W. Rytter. *Fast parallel algorithms for graph matching problems*. Oxford University Press, Inc., New York, NY, USA, 1998.

[KUW86] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is random $\mathbf{NC}$. *Combinatorica*, 6(1):35–48, 1986.

[LP86] L. Lovász and M. Plummer. *Matching Theory*. Noth-Holland, 1986.

[LPV81] G. Lev, M. Pippenger, and L. Valiant. A fast parallel algorithm for routing in permutation networks. *IEEE Transactions on Computers*, C-30:93–100, 1981.

[MN95] G. Miller and J. Naor. Flow in planar graphs with multiple sources and sinks. *SIAM Journal of Computing*, 24(5):1002–1017, 1995.

[MSV99] M. Mahajan, P. Subramanya, and V Vinay. A combinatorial algorithm for pfaffians. In *5th Annual International Conference on Computing and Combinatorics (COCOON)*, Lecture Notes in Computer Science 1627, pages 134–143. Springer-Verlag, 1999.

[MV00] M. Mahajan and K. R. Varadarajan. A new $\mathbf{NC}$-algorithm for finding a perfect matching in bipartite planar and small genus graphs (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing (STOC)*, pages 351–357. ACM Press, 2000.

[MVV87] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Tod91] S. Toda. Counting problems computationally equivalent to the determinant. Technical Report CSIM 91-07, Dept. of Computer Science and Information Mathematics, University of Electro-Communications, Chofu-shi, Tokyo 182, Japan, 1991.

[Tut47] W. T. Tutte. The factorization of linear graphs. *London Math Society*, 22:107–111, 1947.

[Val79] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[Val92] L. Valiant. Why is Boolean complexity theory difficult. In M. S. Paterson, editor, *Boolean Function Complexity*, London Mathematical Society Lecture Notes Series 169. Cambridge University Press, 1992.

[Vaz89] V. Vazirani. $\mathbf{NC}$ algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Information and computation*, 80(2):152–164, 1989.

[Vin91] V Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *6th IEEE Conference on Structure in Complexity Theory (CCC)*, pages 270–284. IEEE Computer Society Press, 1991.

[zG93] J. Von zur Gathen. Parallel linear algebra. In J. H. Reif, editor, *Synthesis of parallel algorithms*, pages 573–617. Morgan Kaufmann, 1993.

[Zip79] R. Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM*, Lecture Notes in Computer Science 72, pages 216–226. Springer, 1979.