

Matching Problem in Different Complexity Classes

Rishav Gupta

2023

Chennai Mathematical Institute

Table of Contents

1. Abstract
2. Preliminaries
 - 2.1 Definitions
 - 2.2 Complexity Classes
3. Polynomially Bounded Perfect Matching Problem in NC^2
4. On the Matching Problem for Special Graphs
5. Planar Maximum Matching : Towards a Parallel Algorithm

Abstract

Abstract

I'll present the following three papers .

- The polynomially bounded Perfect Matching is in NC^2 [Agrawal,Hoang,Thierauf] [AHT07]
- On the Matching Problem for Special Graph Classes[Hoang] [Hoa10]
- Planar Maximum Matching : Towards a Parallel Algorithm[Datta,Kulkarni,Kumar,Mukherjee] [Dat+18]

The first paper will show all perfect matchings of a graph with a polynomially bounded number of perfect matchings can be constructed in NC^2 .

The second paper will show for polynomially bounded number of perfect matchings, SEARCH-PM is in $C=L \cap \bigoplus L$ and COUNTING-PM is in $C=L$.

The third paper will show that Maximum-Matching Search NC reduces to Maximum-Matching Decision in planar graphs.

More generally , it was shown that for a class of graphs closed under vertex deletions and edge contractions , there is an NC algorithm for Maximum Matching Search in the class with oracle queries to Maximum-Matching Size , Maximum-Bipartite-Matching Search and Perfect-Matching Search all for same class of graph.

Preliminaries

- **Matching** - Let $G = (V, E)$ be a graph then a matching M is just a subset of edges $M \subseteq E$ such that no two edges in M have a vertex in common.
- **Perfect Matching** - Let $G = (V, E)$ be a graph then a perfect matching M is just a subset of edges $M \subseteq E$ such that no two edges in M have a vertex in common and M covers all the vertices of G .
- **Maximum Matching** - The matching of maximum cardinality is known as maximum matching.

Definitions

- **Bipartite Tutte Matrix** - Let G be bipartite, that is we can partition the nodes into $V = L \cup R$ such that there are no edges in L and in R . We assume w.l.o.g. that $|L| = |R| = n$, otherwise G has no perfect matching. The bipartite adjacency matrix of G is the $n \times n$ matrix $A = (a_{i,j})$, where

$$a_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E, \text{ for } i \in L \text{ and } j \in R, \\ 0 & \text{otherwise.} \end{cases}$$

The bipartite Tutte matrix of G is the $n \times n$ matrix $T = (t_{i,j})$, where

$$t_{i,j} = a_{i,j}x_{i,j}$$

for indeterminates $x_{i,j}$. The determinant of T is

$$\det(T) = \sum_{\pi \in S_n} \text{sign}(\pi) \prod_{i=1}^n a_{i,\pi(i)} x_{i,\pi(i)}$$

Definitions

General Tutte Matrix - Let G be a graph with n nodes. W.l.o.g. assume that n is even, otherwise G has no perfect matchings. Let $A = (a_{i,j})$ be the $n \times n$ adjacency matrix of G . Note that A is symmetric. The skew-symmetric Tutte matrix of G is the $n \times n$ matrix $T = (t_{i,j})$, where

$$t_{i,j} = \begin{cases} a_{i,j}x_{i,j}, & \text{if } i \leq j, \\ -a_{j,i}x_{j,i}, & \text{otherwise,} \end{cases}$$

for indeterminates $x_{i,j}$. The Pfaffian of T is

$$\text{pf}(T) = \sum_{M \in PM(G)} \text{sign}(M) \cdot \prod_{\substack{(i,j) \in M \\ i < j}} a_{i,j}x_{i,j}.$$

The sign is defined as follows. Consider perfect matching

$$M = \{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\} \in PM(G)$$

for $k = n/2$. By convention, we have $i_l < j_l$ for all l . The sign of M is defined as the sign of the permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \cdots & n-1 & n \\ i_1 & j_1 & i_2 & j_2 & \cdots & i_k & j_k \end{pmatrix} \in S_n \quad (1)$$

It is known that the sign of M does not depend on the order in which the edges are given, i.e. the sign is well defined. $\text{pf}(T)$ is a multi-linear polynomial. Each non-vanishing term $\text{sign}(M)$.

$\prod_{(i,j) \in M, i < j} x_{i,j}$ corresponds to one perfect matching $M \in PM(G)$. The Pfaffian and the determinant of a matrix are known to be closely related as we know that .

$$\det(T) = (\text{pfaffian}(T))^2$$

Theorem (Tutte's Theorem)

A graph G has a perfect matching $\iff \det(T(G)) \neq 0$.

Definitions

- **Nice Cycle** - Let G be a graph which has a perfect matching. An even cycle C of G is said to be nice if $G \setminus C$ also has a perfect matching.
- **Circulation** - The absolute value difference of weights of the two perfect matchings in an even cycle C is called the circulation of the cycle C and is denoted by $\text{circ}(C)$. (Since an even cycle contains exactly two perfect matchings.)
- **Non Vanishing Circulation** - A non vanishing circulation is just a weight assignment to the edges so that for all nice cycles N , $\text{circ}(N) > 0$.

- **Pseudo Deterministic algorithm-** An algorithm A for a relation R is said to be pseudo deterministic if there exists some function s such that when A is executed on input x , outputs $s(x)$ with high probability, and s satisfies $(x, s(x)) \in R$.

Complexity Classes

- NC^i - The class of boolean functions that can be computed with polynomial size , fanin 2, depth $\mathcal{O}(\log^i(n))$.
- AC^i - The class of boolean functions that can be computed with polynomial size , unbounded fanin for both and gates and or gates , depth $\mathcal{O}(\log^i(n))$.
- SAC^i - The class of boolean functions that can be computed with polynomial size ,unbounded fanin for or gates and fanin 2 for and gates , depth $\mathcal{O}(\log^i(n))$.
- NC -

$$NC = \bigcup_{i \in \mathbb{N}} NC^i$$

Complexity Classes

- AC -

$$AC = \bigcup_{i \in \mathbb{N}} AC^i$$

- SAC -

$$SAC = \bigcup_{i \in \mathbb{N}} SAC^i$$

- $\#L$ - A function $f \in \#L$ if and only if there exists a NL machine such that $f(x)$ counts the number of certificates which lead to acceptance in the NL machine.
- $GAP - L$ - A function h is in $GAP - L$ if it can be written as $f - g$ where f and g are in $\#L$.
- SPL - The class of decision problems such that \exists a $GAP - L$ function f such that -

Complexity Classes

- If $x \in L$ then $f(x) = 1$.
- If $x \notin L$ then $f(x) = 0$.
- $C=L$ (*Exact counting in Logspace*) - It is the class of sets A for which there exists a function $f \in GAP - L$ such that

$$\forall x : x \in A \iff f(x) = 0$$

.

Complexity Classes

- $\oplus L$ - $\oplus L$ is the class of sets A for which there exists a function $f \in GAP - L$ such that

$$\forall x : x \in A \iff f(x) \not\equiv 0 \pmod{2}$$

.

Now we have the following relation between the classes -

$$L \subseteq SPL \subseteq C=L \subseteq L^{C=L} \subseteq NC^2$$

$$SPL \subseteq \oplus L \subseteq NC^2$$

$$NL \subseteq C=L$$

$$L^{\oplus L} = \oplus L$$

Polynomially Bounded Perfect Matching Problem in NC^2

Case of Bipartite Graphs

Theorem - *All perfect matchings of a bipartite graph with polynomially bounded number of perfect matchings can be constructed in NC^2 .*

Proof - Let $B_m(x) = (b_{i,j}^{(m)}(x))$ where $b_{i,j}^{(m)}(x) = a_{i,j} \cdot p_{i,j} \cdot x^{m^{ni+j} \bmod r}$ and $a_{i,j} = 1$ if and only if the i th vertex of bipartition X is connected to the j th vertex of the bipartition Y .

Also we have $p_{i,j}$ are pairwise distinct primes, r is a prime such that $r > n^2 \cdot p^2(n)$. By prime number theorem we can choose $\max\{p_{(i,j)} \mid 1 \leq i, j \leq n\} = \mathcal{O}(n^3)$.

Case of Bipartite Graphs

Lets denote $d_m(x) = \det(B_m(x)) =$

$$\sum_{\pi \in S_n} \text{sign}(\pi) \left(\prod_{i=1}^n a_{i, \pi(i)} p_{i, \pi(i)} \right) x^{e_m(\pi)}$$

where $e_m(\pi) = \sum_{i=1}^n (m^{ni + \pi(i)} \bmod r)$.

Lemma - Let $\pi_1, \pi_2, \dots, \pi_t \in S_n$ for some $t \leq p(n)$. Then there exists an $m < r$ such that $e_m(\pi_i) \neq e_m(\pi_j)$, for all $i \neq j$ and $i, j \in [1 \dots n]$.

Proof - The values $e_m(\pi_i)$ can be seen as evaluations of polynomials over the field \mathbb{Z}_r in the following way. Define

$$q_\pi(z) = \sum_{i=1}^n z^{ni + \pi(i)}$$

Case of Bipartite Graphs

Then we have

$$e_m(\pi_i) \equiv q_{\pi_i}(m) \pmod{r}$$

To prove the lemma we will use the fact that the degree of $q_{\pi_i}(z)$ is bounded by $n^2 + n \leq 2n^2$. Now any two of them can agree on at most $2n^2$ points. So if the field size is larger than $\binom{t}{2}2n^2$ we have a point where all the q_{π_i} polynomials, pairwise differ. And by our choice of r we have that property as

$$\binom{t}{2}2n^2 \leq t^2 n^2 \leq p^2(n)n^2 < r$$

.

Case of Bipartite Graphs

It follows that if G has t perfect matchings for some $t \leq p(n)$, then there exists an $m < r$ such that

$$d_m(x) = \sum_{k=0}^D c_k^{(m)} x^k$$

where precisely t of the coefficients are non zero and are of form

$$c_k^m = \text{sign}(\pi) \prod_{i=1}^n p_{i,\pi(i)}$$

for some $\pi \in S_n$ such that $k = e_m(\pi)$. To compute these coefficients we define the Vandermonde matrix $V = (v_{i,j})$ where $v_{i,j} = i^j$ for $0 \leq i, j \leq D = r \cdot n$. (Basically we are just doing interpolation).

Case of Bipartite Graphs

Define the vectors

$$d_m = (d_m(0), d_m(1), \dots, d_m(D))^T$$

.

$$c_m = (c_0^{(m)}, c_1^{(m)}, \dots, c_D^{(m)})$$

We have that

$$c_m = V^{-1} d_m$$

Lemma - Computation of c_m is in NC^2 .

Proof - The matrices V and $B_m(x)$ can be computed in NC^1 for any $x \leq D$. Vector d_m can be computed by computing the determinant of matrix $B_m(x)$ for different values of x can be computed in NC^2

Case of Bipartite Graphs

(Using the trace method). Also V^{-1} can be computed in NC^2 . (Using Csanky's Algorithm).

Proof of main theorem - So what we do is calculate the vectors c_m for all $m < r$ then focus on the one with maximum number of nonzero entries because that value of m will isolate all the perfect matchings (since we proved the existence of such a m) then for all non zero coefficients we can test $c_k^{(m)} \equiv 0 \pmod{p_{ij}}$. (Since $p_{i,j}$ are $\mathcal{O}(n^3)$). Hence we can construct all perfect matchings of G in NC^2 .

So basically the idea is the p_{ij} is used store the information of matching edges and the value r is used to isolate the matchings.

Case of General Graphs

Theorem *All perfect matchings of a graph with a polynomially bounded number of perfect matchings can be constructed in \mathbf{NC}^2 .*

Let $G = (V, E)$ be an undirected graph with $|V| = n$ nodes. We assume that n is even, otherwise G has no perfect matchings. Let $A = (a_{i,j})$ be the adjacency matrix of G . Let p be a polynomial and assume that G has at most $p(n)$ perfect matchings. We define matrices $B_m(x) = \left(b_{i,j}^{(m)}(x)\right)$ in a similar fashion as before. The definition is now according to the Tutte matrix of G :

$$b_{i,j}^{(m)}(x) = \begin{cases} a_{i,j} p_{i,j} x^{m^{n+j} \bmod r}, & \text{if } i \leq j, \\ -a_{j,i} p_{j,i} x^{m^{n+j+i} \bmod r}, & \text{otherwise,} \end{cases}$$

Case of General Graphs

for pairwise different primes $p_{i,j}$ of size $O(n^3)$, an indeterminate x , a prime r such that $r > n^2 p^2(n)$, and $1 \leq m < r$.

The Pfaffian of $B_m(x)$ is a polynomial $p_m(x)$, where

$$\begin{aligned} p_m(x) = \text{pf}(B_m(x)) &= \sum_{M \in PM(G)} \text{sign}(M) \cdot \prod_{\substack{(i,j) \in M \\ i < j}} a_{i,j} p_{i,j} x^{m^{ni+j} \bmod r} \\ &= \sum_{M \in PM(G)} \text{sign}(M) \cdot \left(\prod_{\substack{(i,j) \in M \\ i < j}} a_{i,j} p_{i,j} \right) x^{e_m(M)} \end{aligned}$$

where

$$e_m(M) = \sum_{\substack{(i,j) \in M \\ i < j}} (m^{ni+j} \bmod r)$$

Case of General Graphs

are the exponents of x in $p_m(x)$. Similar as in previous lemma we have that there is some $m < r$ where the exponents $e_m(M)$ pairwise differ. Note that $e_m(M) \leq (r-1)n/2$. Let $D = (r-1)n/2$. Then we can write

$$p_m(x) = \sum_{k=0}^D c_k^{(m)} x^k$$

Define the Vandermonde matrix $V = (v_{i,j})$ by $v_{i,j} = i^j$, for $0 \leq i, j \leq D$. Define vectors

$$\mathbf{p}_m = \begin{pmatrix} p_m(0) & p_m(1) & \cdots & p_m(D) \end{pmatrix}^T$$
$$\mathbf{c}_m = \begin{pmatrix} c_0^{(m)} & c_1^{(m)} & \cdots & c_D^{(m)} \end{pmatrix}^T$$

Case of General Graphs

As in the bipartite case we have $\mathbf{p}_m = V \cdot \mathbf{c}_m$, from which we get $\mathbf{c}_m = V^{-1} \mathbf{p}_m$. Similar to the previous case \mathbf{c}_m can be computed in NC^2 .

This is because of the result that calculation of integral Pfaffian is in $\text{GAP} - L$ (Mahajan and Vinay 99) and the fact that integer Determinant is $\text{GAP} - L$ complete and can be calculated in NC^2 .

Case of Weighted Graphs

Theorem- Let G be a weighted graph with polynomially bounded weights such that G has a polynomially bounded number of perfect matchings of some weight w . Then all perfect matchings of G of weight w can be constructed in \mathbf{NC}^2 .

Proof- Define matrices $B_m(x, y) = \left(b_{i,j}^{(m)}(x, y) \right)$ in two variables x and y that incorporate the weights of G :

$$b_{i,j}^{(m)}(x, y) = \begin{cases} a_{i,j} p_{i,j} y^{w_{i,j}} x^{m^{ni+j} \bmod r}, & \text{if } i \leq j, \\ -a_{j,i} p_{j,i} y^{w_{j,i}} x^{m^{nj+i} \bmod r} & \text{otherwise,} \end{cases}$$

Case of Weighted Graphs

for pairwise different primes $p_{i,j}$ of size $O(n^3)$, indeterminates x and y , a prime r such that $r > n^2 p^2(n)$, and $1 \leq m < r$. The Pfaffian of $B_m(x, y)$ is a polynomial $p_m(x, y)$, where

$$\begin{aligned} p_m(x, y) &= \text{pf}(B_m(x, y)) \\ &= \sum_{M \in PM(G)} \text{sign}(M) \cdot \prod_{\substack{(i,j) \in M \\ i < j}} a_{i,j} p_{i,j} y^{w_{i,j}} x^{m^{ni+j} \bmod r} \\ &= \sum_{M \in PM(G)} \text{sign}(M) \cdot \left(\prod_{\substack{(i,j) \in M \\ i < j}} a_{i,j} p_{i,j} \right) y^{w(M)} x^{e_m(M)} \end{aligned}$$

Case of Weighted Graphs

where $e_m(M) = \sum_{(i,j) \in M, i < j} (m^{ni+j} \bmod r)$. By a similar argument as in previous lemma we have that there is some $m < r$ where the exponents $e_m(M)$ pairwise differ, and this suffices for our purpose.

The degree of x in $p_m(x, y)$ is bounded by $(r - 1)n/2$. Let $d = (r - 1)n/2 + 1$, so that the degree of x in $p_m(x, y)$ is strictly less than d . We transform $p_m(x, y)$ into polynomial $P_m(x)$ with just one variable by setting

$$P_m(x) = p_m\left(x, x^d\right).$$

Case of Weighted Graphs

Then we have

$$P_m(x) = \sum_{M \in PM(G)} \text{sign}(M) \cdot \left(\prod_{\substack{(i,j) \in M \\ i < j}} a_{i,j} p_{i,j} \right) x^{dw(M) + e_m(M)}$$

By our choice of d we have $d > e_m(M)$. Let w be any fixed weight and consider a perfect matching M of weight w . Then we have

$$dw < dw + e_m(M) < d(w + 1).$$

That is, the degrees of x in $P_m(x)$ for perfect matchings of different weights w are in disjoint intervals of the form $(dw, d(w + 1))$. Let D be the degree of $P_m(x)$.

Case of Weighted Graphs

We have $D \leq d(w_{\max} + 1)$, where w_{\max} is the maximum weight of any matching. Note that $w_{\max} \leq \max \{w_{i,j} \mid 1 \leq i, j \leq n\} n/2$. Let

$$P_m(x) = \sum_{k=0}^D c_k^{(m)} x^k.$$

We have seen earlier how to determine the coefficients $c_k^{(m)}$ and how to get the perfect matchings from these coefficients in NC^2 . (Using interpolation). Note that the perfect matchings of weight w are represented by the coefficients $c_k^{(m)}$ for $dw < k < d(w + 1)$.

Now, if there are at most $p(n)$ perfect matchings of weight w , then all of these will be listed by our NC^2 -circuit. Note however that we

Case of Weighted Graphs

might list perfect matchings of other weights as well. In case that the promise is for the minimum (or maximum) weight perfect matching, we may discard non-optimal perfect matchings.

On the Matching Problem for Special Graphs

Perfect matching construction in $L^{C=L} \cap \oplus L$

Lemma - Given a graph G and a weight function on the edges of the graph G . If all the nice cycles in the graph G have a non vanishing circulation then there exist a unique min-weight matching for the graph G and the given weight function.

Isolating Lemma - Let U be a universe of size m and F be a family of subsets of U . Let $w : U \rightarrow \{1, 2, \dots, M\}$ be weighting function. Then if we select weight functions uniformly at random then we have that the probability of getting a unique minimum subset in F is

$$\geq 1 - \frac{m}{M}$$

Perfect matching construction in $L^{C=L} \cap \oplus L$

Lemma- Let $G = (V, E)$ be an undirected graph with $|V| = n$ vertices and m edges $E = \{e_1, e_2, \dots, e_m\}$, 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 (A weight function is admissible if and only if all the nice cycles have non-vanishing circulations) 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:

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

Perfect matching construction in $L^{C=L} \cap \oplus L$

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

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

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:

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

Perfect matching construction in $L^{C=L} \cap \oplus L$

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

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

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.

Perfect matching construction in $L^{C=L} \cap \oplus L$

Also notice if the graph has polynomially bounded perfect matchings then it also has polynomially bounded nice cycles (more precisely number of nice cycles is upperbounded by the square of number of perfect matchings because each nice cycle can be associated a pair of perfect matchings such that their symmetric difference is that nice cycle.)

Perfect matching construction in $L^{C=L} \cap \oplus L$

Theorem - For each fixed $k > 0$, there is a 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 $\mathbf{L}^{C=L} \cap \oplus \mathbf{L}$.

Proof: Let $G = (V, E)$ be a graph with n vertices, m edges $E = \{e_1, \dots, e_m\}$, 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 $w_p : E \rightarrow \mathbb{Z}_p$, 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

Perfect matching construction in $L^{C=L} \cap \oplus L$

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 $K = n^{k+1}(m+1)$. Note that K upper bounds weight of any perfect matching for any prime $p < 2(n^k)(m+1)$. Now we can write:

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

Perfect matching construction in $L^{C=L} \cap \oplus L$

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

Consider the case when G has some perfect matching. By previous lemma 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 $\text{pf}(T_p(x))$, occurred as the lowest nonzero coefficient in the polynomial, should be

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

Perfect matching construction in $L^{C=L} \cap \oplus L$

or equivalently $c_{p,I}^2 = 1$. We know that all the coefficients of the polynomials we consider are computable in $GAP - L$ because of the following lemma.

Lemma - Given $A(x)$ to be a polynomial matrix with entries having logarithmic bit length in degree (basically polynomially bounded degree) then the coefficients of $\det(A(x))$ are computable in $GAP - L$.

Using the above lemma and the fact that $pf(A(x))$ for skew symmetric $A(x)$ is $GAP - L$ computable [Mahajan and Vinay 99] we get that coefficients can also be computed in $GAP - L$.

Hence we get that

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

Perfect matching construction in $L^{C=L} \cap \oplus L$

is a *zero-one-valued GAP* – L function such that $h_k(G) = 1$ if and only if G has some perfect matching.

This implies *DECISION-PM* for such graphs is in *SPL*.

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

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

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

Perfect matching construction in $L^{C=L} \cap \oplus L$

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

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

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 $\oplus L$ 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 $L^{C=L} \cap \oplus L$ because $L^{\oplus L} = \oplus L$. This completes the proof of the theorem.

Poly Bounded Perfect Matching Search in $C=L \cap \oplus L$

Theorem - For graphs with polynomially bounded number of perfect matchings we have that *SEARCH – PM* is in $C=L \cap \oplus L$ and *COUNTING – PM* is in $C=L$.

Proof - 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.

Poly Bounded Perfect Matching Search in $C_{=}L \cap \oplus L$

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

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

Poly Bounded Perfect Matching Search in $C_{=}L \cap \oplus L$

In analogy to previous lemma, 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 $K = \frac{(m+1)n^{2k+1}}{2}$. See that, K is also an upper bound on the degrees of the polynomial $\text{pf}(T_p(x))$, i.e. we can write

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

We observe further that a prime $p \in U$ is "right" iff in $\text{pf}(T_p(x))$ all coefficients are from $\{-1, 0, +1\}$ and the number of nonzero coefficients is maximum. Note that the latter is the number of all

Poly Bounded Perfect Matching Search in $C_{=L} \cap \oplus L$

perfect matchings in G when p is "right", i.e. for the "right" prime p , the 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

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

Poly Bounded Perfect Matching Search in $C_{=}L \cap \oplus L$

for every $q \in U$. We see that $h_p = 0$ if and only if all the coefficients in $pf(T_p(x))$ 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 GapL-functions

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

Poly Bounded Perfect Matching Search in $C=L \cap \oplus L$

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

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

where a is some nonnegative integer smaller or equal to Kn^{2k} .

Therefore, in $C=L$ we can select a "right" prime p from U as follows:

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

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

Consider Step (b). In $C=L$ 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 previous theorem. Note that after Step (b) we do not check again

Poly Bounded Perfect Matching Search in $C=L \cap \oplus L$

whether the constructed edge-sets are perfect matchings. (Earlier we needed to check this because if p wasn't the correct prime then the set might not have given a perfect matching). This shows that all perfect matchings in G can be constructed in $C=L$.

The problem SEARCH-PM is in $\oplus L$ by following the proof of previous theorem.

The number of all perfect matchings in G can be computed in $C=L$ 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.

Isolating and Computing Maximum Matching

Theorem- *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 $\mathbf{L}^{C=L}$ -reducible to the problem of computing a perfect matching in a subgraph of G .*

The above theorem will give us the following corollary-

Corollary- *Since $SPL \subseteq L^{C=L}$ and we know planar bipartite matching is in SPL and planar bipartite graphs are closed under taking subgraphs so we get that maximum matching problem for planar graphs is in $L^{C=L}$.*

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

Isolating and Computing Maximum Matching

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 skewsymmetric matrix of $G(x)$ and of $G_M(x)$, respectively.

Isolating and Computing Maximum Matching

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

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

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

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

Isolating and Computing Maximum Matching

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 is reducible to the rank of an integer matrix. The latter is known to be in $\mathbf{L}^{\mathbf{C}=\mathbf{L}}$ [Allender, Beals, Ogihara-99]. Now we will show that, for polynomial matrices, the problem of computing a columnbasis is also in $\mathbf{L}^{\mathbf{C}=\mathbf{L}}$. Our proof of this statement consists of two following parts:

Isolating and Computing Maximum Matching

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 $L^{C=L}$. 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 $\vec{a}_1(x), \dots, \vec{a}_n(x)$ 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 $\vec{a}_1(x), \dots, \vec{a}_i(x)$ 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

$$\text{rank}(A_{i-1}(x)) + 1 = \text{rank}(A_i(x)),$$

Isolating and Computing Maximum Matching

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 $\text{rank}(B(x))$. It is known that $2 \text{rank}(B(x)) = \text{rank}(C(x))$ where

$$C(x) = \begin{pmatrix} \mathbf{0} & B(x) \\ B^t(x) & \mathbf{0} \end{pmatrix}$$

Isolating and Computing Maximum Matching

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 $\text{rank}(C(x))$ 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 $\text{rank}(C(x)) = j$ iff

$$p_0(x) = \cdots = p_{N-j-1}(x) = 0 \text{ and } p_{N-j}(x) \neq 0.$$

Isolating and Computing Maximum Matching

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 $a \in S = \{0, 1, \dots, \deg(p_i(x))\}$ 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

$$\deg(p_i(x)) \leq Nn^c = (m+n)n^c,$$

for all $0 \leq i \leq N-1$, where c is a constant, we define

$$S = \{0, 1, \dots, (m+n)n^c\}.$$

Then the rank of $B(x)$ is equal to the maximum number in the set $\{\text{rank}(B(a)) \mid a \in S\}$.

Isolating and Computing Maximum Matching

The rank of an integer matrix is known to be in $\mathbf{L}^{C=L}$. Therefore, $\text{rank}(B(x))$ is in $\mathbf{L}^{C=L}$.

The proof of the lemma is complete.

Planar Maximum Matching : Towards a Parallel Algorithm

Maximum Matching Search in Graphs with NVC

Notice- *Perfect-Matching and Maximum-Matching are equivalent in general graphs under Logspace Turing reductions.* Because we could to check if there is a matching of size M add $(n - 2M)$ auxillary vertices and connect them to all original vertices .And now run the perfect matching algorithm.Now do a binary search kind of thing to get the maximum matching.

- We know that planar perfect matching is in NC [Anari,Vazirani] but we cant use above reduction to put maximum planar matching in NC because it doesn't preserve planarity.
- For similar reason we can't extend directly for polynomially bounded case.(I think so.)

Maximum Matching Search in Graphs with NVC

We will show that given an undirected unweighted graph $G = (V, E)$ admitting non-zero circulations, finding a maximum matching is in *SPL*. The basic idea is to construct an auxiliary graph G' having the property that finding a maximum matching in G reduces to finding a min-weight generalized perfect matching (defined later) in G' . Assign non-zero circulation weights to the edges in G' which are also isolating weights for matchings. Then we extract a min-weight generalized perfect matching from G' which in turn extracts a maximum matching from G .

Note we assume that *NVC* of planar bipartite graphs can be computed .

We construct a graph $G' = (V', E')$ from G by adding vertex t_v with a self loop for each vertex $v \in V$ and join v and t_v using an

Maximum Matching Search in Graphs with NVC

undirected edge, as shown in Figure 1. Thus, $|V'| = 2n \equiv 0 \pmod{2}$. Define a weight function $w' : E' \mapsto \{0, 1\}$ for G' as follows. The original edges of G have weight 1, the self-loops are of weight zero and rest of the new edges have weight 1 (suffices to pick any weight $> 1/2$). We define a generalized matching as a set of disjoint edges (possibly) inclusive of self-loops. Various notions for matching naturally extends to generalized matchings. Call a generalized matching as perfect wherein every vertex is matched and as min-weight perfect if it is perfect and of minimum weight.

Lemma - *Any matching M in G can be extended to a generalized perfect matching in G' . Such that $w'(P) = n - |M|$*

Proof The proof is as follows -

Maximum Matching Search in Graphs with NVC

- Keep all the matching edges of M as it is in P .
- For all the matched vertices v match t_v using self loops in P .
- For all unmatched vertices v in M join v with t_v using the introduced edge .
- Now see that the weight of P is same as $|M|$ from previous matching edges and the non self loop edges have the total weight equal to $n - 2|M|$ so we get that the total weight is $n - |M|$.

Maximum Matching Search in Graphs with NVC

Observe !- *The min weight perfect matching in G' corresponds to the maximum matching in G .*

Lemma *With respect to the weight function $w : E' \mapsto [\mathbb{N}]$, the min-weight generalized perfect matching in G' is unique.*

Proof- To prove this we need some definitions first.

- Define a loop-path as a trail $(e_0, e_1, e_2, \dots, e_k)$ (for k odd, $k > 1$) where e_0 is a self loop, the subtrail $(e_1, e_2, \dots, e_{k-1})$ is a path of non-zero length and e_k is also a self loop.
- Define a 2-cycle as a length 2 directed cycle corresponding to an undirected edge as the underlying graph.

Maximum Matching Search in Graphs with NVC

- Define a 2-self loop as a closed walk (e, e) where e is a self loop.
- Define the alternating weight of a loop-path

$\mathcal{P}' = (e_0, e_1, e_2, \dots, e_k)$ (for $k \geq 2$) to be the alternating sum of the weight of the edges in \mathcal{P}' i.e.

$$AW(\mathcal{P}') = \sum_{i=0}^k (-1)^i w(e_i) = (w(e_0) - w(e_k)) + (-w(e_1) + w(e_2) - \dots + w(e_{k-1})).$$

- Let the graph G' has at most $c'n$ many edges for some constant c' . Define a weight function w'' on the edges of G' which assigns non-zero weights to the self-loops as follows,

$$w''(e) = \left\{ \begin{array}{ll} ic', & \text{if } e = (t_i, t_i) \ 1 \leq i \leq |V| \\ 0 & \text{otherwise} \end{array} \right\}$$

Maximum Matching Search in Graphs with NVC

Since for planar bipartite graphs we can compute NVC efficiently so we take this NVC and denote this non vanishing circulation weight for an edge e by $w'''(e)$ which are bounded by, say n^c for some constant c . We combine the weights w'' and w''' into a single weight w^* . Using bit shift, we define the new weight $w^*(e)$ on the edges of G' by $w^*(e) = w''(e) \cdot 2^{\lceil (c+1) \log_2(n) \rceil} + w'''(e)$ for $e \in E(G)$. The weights $w^*(e)$ are bounded by $w''(e) \cdot n^{c+1} \leq c'n \cdot n^{c+1} \leq c'n^{c+2}$. Notice that for the non self-loop edges $w^*(e)$ is bounded by $w'''(e) \leq n^c$.

Lemma- *With respect to the weighting scheme w^* , the alternating sum of each simple alternating cycle of G' and each loop-path is non-zero.* Proof. Using the weights $w'''(e)$ from Lemma 10, each

Maximum Matching Search in Graphs with NVC

simple alternating cycle of G has non-zero circulation, and since each simple cycle of G' is necessarily a simple cycle in G , thus every simple alternating cycle of G' has non-zero circulation. Now consider the looppath given by $\mathcal{P} = (e_0, e_1, e_2, \dots, e_k)$ (for $k \geq 2$). Then,

$$|AW(\mathcal{P})| = \left| \sum_{i=0}^k (-1)^i w^*(e_i) \right| \geq \\ |w^*(e_0) - w^*(e_k)| - |(w^*(e_1) - w^*(e_2) + \dots + (-1)^k w^*(e_{k-1}))|.$$

And,

$$\left| (w^*(e_1) - w^*(e_2) + \dots + (-1)^k w^*(e_{k-1})) \right| \\ < |w^*(e_1)| + |w^*(e_2)| + \dots + |w^*(e_{k-1})|$$

$$< (k-1) \cdot n^c \quad (\text{as } w^*(e_i) \leq n^c \text{ here})$$

$$< (c'n - 1) \cdot n^c \quad (k < |E(G')| \leq c'n) \\ \leq c'n^{c+1}$$

Maximum Matching Search in Graphs with NVC

Then $|A W(\mathcal{P})| > |w^*(e_0) - w^*(e_k)| - c'n^{c+1} \geq 0$ and thus every loop-path also has non-zero alternating weight.

Now we combine the weights w' and w^* into a single weight w . Using bit shift again, we define the new weight $w(e)$ on the edges of G' as $w(e) = w'(e) \cdot 2^{\lceil (c+2) \log_2(c'n) \rceil} + w^*(e)$ for $e \in E(G)$. The weights $w(e)$ are bounded by $w'(e) \cdot c'n^{c+2} \leq c'n^{c+2}$ as $w'(e) \in \{0, 1\}$.

Now we will summarize the weighting scheme -

- First we defined w' on G' as

$$w'(e) = \begin{cases} 1 & e \in G \\ 0 & e = (t_v, t_v) \\ 1 & e = (v, t_v) \end{cases} \quad (2)$$

Maximum Matching Search in Graphs with NVC

- Then we defined w'' on G' as

$$w''(e) = \begin{cases} ic' & e = (t_v, t_v) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

- Then we defined w''' on G which is just the non vanishing circulation weights on G . We trivially extend w''' to G' as the edges which are not in G we added 0 weight to them.
- Then we defined w^* as $w^*(e) = w''(e) \cdot 2^{\lceil (c+1) \log_2(n) \rceil} + w'''(e)$
- Then we defined the weight scheme $w(e)$ as $w(e) = w^*(e) + w'(e) \cdot 2^{\lceil (c+2) \log_2(c'n) \rceil}$

Maximum Matching Search in Graphs with NVC

Lemma *A min-weight generalized perfect matching of G' corresponding to the weight function w' is also a min-weight generalized perfect matching corresponding to the weight function w . Moreover, the alternating sum of the weights with respect to w of simple alternating cycles and loop-paths are non-zero.*

Proof of main lemma (unique min weight perfect matching) The components of the symmetric difference of any two generalized perfect matchings are either simple alternating cycles, loop-paths, 2-cycles or 2-self-loops. Suppose that there is more than one min-weight generalized perfect matching of G' , call them P_1 and P_2 , such that $P_1 \neq P_2$. Since $P_1 \neq P_2$, there exists at least one component of $P_1 \cup P_2$ which is a simple alternating cycle or an loop-path. And since $w''(e)$ assigns a non-zero alternating sum weight

Maximum Matching Search in Graphs with NVC

on all simple alternating cycle and loop-paths, this implies that the sum of weights of edges from one of P_1 and P_2 is lesser than the other. Swapping the edges between P_1 and P_2 in this component will give rise to a new generalized perfect matching having weight lower than both of P_1 and P_2 , which is a contradiction.

We use the determinant polynomial to compute the size of the maximum matching. Since we can get the least coefficient and take the square root of it to get the coefficient of it in the pfaffian. (Since the least degree in the determinant will be just the square of the term in pfaffian since the min weight thing is unique .) Now we compute the least degree term in determinant polynomial. Once the weight is known extract P the perfect matching by deleting edges in parallel

Maximum Matching Search in Graphs with NVC

and computing the perfect matching in $G' \setminus \{e\}$ If the min weight is unchanged implies that the edge is not in P . Once we get hold of P we can get M the maximum matching in the original graph as

$M = P \cap E(G)$. Now if $w(P) = W$ then $|M| = n - y$ where

$$y = \left\lfloor \frac{W}{2^{\lceil (c+2) \log_2(c'n) \rceil}} \right\rfloor.$$

Maximum Matching Search in Graphs with NVC

Main Theorem- *Maximum Matching Search in graphs with NVC is in SPL .*

Proof- Since the edge weights are polynomially bounded they can be computed in logspace. Moreover computation of the coefficients of a determinant is in $GAP - L$. Since we are only querying whether the coefficient is zero 1 upto the least degree coefficient hence the computation is in SPL . So we are able to extract a min-weight generalized perfect matching in the graph G' in $L^{SPL} = SPL$. Hence proved.

Reduction From Search to Decision

- In this part we will reduce the problem of finding a maximum a matching to oracle calls for determining the size of a maximum matching in the presence of a parallel algorithm to find a perfect matching and a parallel algorithm to solve the bipartite version of a maximum matching.
- We will use *Edmond Gallai Decomposition* for doing so.

Edmond Gallai Decomposition-

- We can decompose a graph G into the following 3 parts-
 - $D(G)$ - Set of vertices v such that there exist a maximum matching of G without matching v .
 - $A(G)$ - Set of vertices which are adjacent to $D(G)$ and not in $D(G)$.
 - $C(G) = V(G) \setminus (A(G) \cup D(G))$
- Now this decomposition has the following properties. Let M be a maximum matching of G then we have that-
 - M induces a perfect matching on $C(G)$.
 - M induces a near perfect matching on each component of $D(G)$.
 - M matches each vertex of $A(G)$ to a distinct component in $D(G)$.

Reduction From Search to Decision

- The set $D(G)$ can be constructed using subroutine to maximum matching size computation as
$$v \in D(G) \iff \nu(G) = \nu(G - v) \text{ where } \nu(G) \text{ denotes size of maximum matching in } G.$$
- Once $D(G)$ is constructed $A(G)$, $C(G)$ can be constructed easily .
- So to find a maximum matching we just need to do the following using *Edmond Gallai Decomposition* .
- First compute a perfect matching on $C(G)$.
- A perfect matching in each component of $D(G)$ minus an arbitrary vertex.

Reduction From Search to Decision

- A maximum matching in the bipartite graph formed by contracting each component of $D(G)$ into a single vertex and adding an edge to each vertex in $a \in A(G)$ such that the corresponding component had an edge to a .
- So we get the following lemma-
- **Lemma-** *For any class of graphs closed under vertex deletions and edge contractions, there is an NC algorithm for Maximum-Matching Search in the class with oracle queries to Maximum-Matching Size(For Edmond Gallai Decomposition), Maximum-Bipartite Search and Perfect-Matching Search all for the same class of graphs.*

Reduction From Search to Decision

- For planar graphs we know they are closed under vertex deletion and edge contraction , and perfect matching computation in planar graphs is in NC [Anari,Vazirani] and *Maximum-Bipartite Matching Search* can be done using previous algorithm as NVC is easy to find for planar bipartite graphs.
- So we get that for planar graphs *Maximum-Matching Search* NC reduces to *Maximum-Matching Decision or Size*.
- We know of a RNC^2 algorithm for maximum matching size.[Mulmuley,Vazirani,1987]
- Hence we get that *Maximum Matching Search* is in pseudo-deterministic NC for planar bipartite graphs.

The End !!

- [AHT07] Manindra Agrawal, Thanh Minh Hoang, and Thomas Thierauf. “The Polynomially Bounded Perfect Matching Problem Is in NC²”. In: *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*. Ed. by Wolfgang Thomas and Pascal Weil. Vol. 4393. Lecture Notes in Computer Science. Springer, 2007, pp. 489–499. DOI: [10.1007/978-3-540-70918-3_42](https://doi.org/10.1007/978-3-540-70918-3_42). URL: https://doi.org/10.1007/978-3-540-70918-3%5C_42.

- [Hoa10] Thanh Minh Hoang. “On the Matching Problem for Special Graph Classes”. In: *2010 IEEE 25th Annual Conference on Computational Complexity*. 2010, pp. 139–150. DOI: 10.1109/CCC.2010.21.

- [Dat+18] Samir Datta et al. “Planar Maximum Matching: Towards a Parallel Algorithm”. In: *29th International Symposium on Algorithms and Computation, ISAAC 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan*. Ed. by Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao. Vol. 123. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. ISBN: 978-3-95977-094-1. DOI: [10.4230/LIPIcs.ISAAC.2018.21](https://doi.org/10.4230/LIPIcs.ISAAC.2018.21). URL: <https://doi.org/10.4230/LIPIcs.ISAAC.2018.21>.