Matching in Planar Graphs

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Master of Technology

by

Rohit Gurjar

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

May, 2010

Certificate

It is certified that the work contained in the thesis entitled *Matching in Planar Graphs*, by *Rohit Gurjar*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Piyush P Kurur
Department of Computer Science and Engineering
I.I.T. Kanpur

May, 2010

ABSTRACT

The perfect matching problem is a well studied problem in the field of parallel algorithms. It has a close relation with complexity theory. There exist RNC algorithms to construct a perfect matching in a given graph [MVV87, KUW86], but no NC algorithm is known for it. In this work we are particularly interested in planar graphs. The problem of counting the number of perfect matchings, which intuitively should be harder than constructing a perfect matching, can be solved in NC for planar graphs [Kas67, Vaz89]. This leads us to believe that there should be an NC algorithm to construct a perfect matching in a planar graph. Our attempt here is to get such an algorithm. We look at three main ideas for constructing a perfect matching in a bipartite planar graph in NC due to Miller and Naor, Mahajan et al. and Kulkarni et al. [MN95, MV00, DKR08]. We try to generalize the latter two to the general planar case.

The first algorithm due to Mahajan et al. [MV00] involves traversing the matching polytope and reaching one of its vertices. The main problem in generalizing this approach seems to be a large number of bounding hyperplanes of the matching polytope. The second approach uses a known RNC algorithm, due to Mulmuley et al. [MVV87], which finds a perfect matching in a general graph. Kulkarni et al. [DKR08] and Korwar [Kor09] derandomized the algorithm for bipartite planar graphs. We give some ideas for the derandomization in the general planar case. We also discuss a similarity between the two algorithms given by Miller and Naor [MN95] and Korwar [Kor09]. Though we did not get any NC algorithm to find a matching in a planar graph, we hope the ideas presented here would help in getting one.

Dedicated to

IIT Kanpur

which showed me a glimpse

of the Truth

Acknowledgements

First of all I want to thank Dr. Piyush P Kurur for patiently guiding me through this work. The projects and courses I did under him, introduced me to a lot of interesting topics, which eventually encouraged me to pursue research in TCS. I found him an excellent teacher and advisor. It was an enjoyable experience to work with him. I am also grateful to other faculty members and fellow students at IITK for the wonderful learning environment they provided.

I thank Arpita for all the useful discussions and ideas. I would also like to thank Tejaswi for all the interesting discussions. Thanks are due to Himanshu and Anuj, for their invaluable assistance in preparing this document. Finally, I am indebted to all my friends at IITK, my family members and my school teachers for their constant support in all my endeavors.

Contents

Nomenclature

$\mathbf{e}$	incidence vector of the set $\{e\}$
$\mathbf{M}$	incidence vector of the set of edges M
Δ	symmetric difference
δu	set of edges incident on u
$\det A$	determinant of matrix A
$\mathcal{FP}(G)$	fractional matching polytope of G
$\mathbb{F}_2$	Galois field of size 2
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integers
$\mathcal{M}_w$	union of all the minimum weight matchings under w
M	number of perfect matchings
M_e	number of perfect matchings which contain the edge e
$\mathcal{M}(G)$	matching hyperplane of G
$\mathcal{M}_0(G)$	matching space of G
$\overline{T}$	complement of T in the edge set
$\mathcal{P}(G)$	matching polytope of G (referred as perfect matching polytope in standard texts)
$G(V_1, V_2)$	a bipartite graph G with V_1 and V_2 being the two independent sets of vertices
G^*	planar dual of G
$w_p(S)$	weight of the set of edges S treating coordinates of the point p as a weight assignment on the edges
x_e^p	coordinate of the point p corresponding to the edge e
$\mathbf{v}_C$	alternating vector corresponding to the cycle C

Chapter 1

Introduction

The perfect matching problem is of fundamental interest in combinatorics, algorithms and complexity theory for a variety of reasons. The problem has a close relation with complexity theory. Edmonds [Edm65] gave the first polynomial time algorithm to find a maximum matching in a graph. Interestingly, the algorithm motivated Edmonds to propose *polynomial time* as a measure of efficient computation. The problem is also interesting from the perspective of parallel algorithms. An important question in the theory of parallel computation is which problems are efficiently parallelizable, i.e. P vs. NC. In the study of the analogous question P vs. NP, people quickly realized that (using self reducibility) search problems and their decision problems are equivalent. A similar line of thought for the P vs. NC question seems to fail (see [KUW85]). Many decision problems have simple parallel algorithms, but no such algorithms are known for their corresponding search problems. Perfect matching in a planar graph is one among them. It is an open problem to find an NC algorithm to construct a perfect matching in a graph or even in a planar graph. However, there is an NC algorithm to find the number of perfect matchings in a planar graph. This is quite contrary to our intuition that search should be easier than counting, because intuitively counting involves going over all the matchings. For example, counting the number of perfect matchings in the general case is #P-hard [Val79], whereas the search version in P. In fact, Mahajan et al. [KMV08] showed that for any subclass of bipartite graphs closed under edge deletion, the search problem NC-reduces to the counting version. All these facts lead us to believe that there should be an NC algorithm for constructing a perfect matching in a planar graph. The focus here is to get such an algorithm.

Given a graph $G(V, E)$, a matching is a subset of the edge set E, such that no two edges have a common vertex. Any matching with the largest size in G is called a maximum matching. A perfect matching is a matching which covers all the vertices of G. There are various versions of matching related problems, let us consider the following three-

Decision Is there a perfect matching in a given graph?

Search Construct a perfect matching in a graph, if it exists.

Counting Find the total number of perfect matchings in a graph.

As it turns out some versions have easier algorithms in the case of bipartite graphs. A graph $G(V, E)$ is bipartite if the vertex set V can be partitioned into two disjoint sets such that every edge has one endpoint in each partition. Finding a maximum matching in a bipartite graph is closely related to the network flow problem. In the 1950's Ford and Fulkerson published the first papers [LFF56, LFF57, LFF62] on the theory of network flows. Flow theory can be used to prove most results in bipartite matchings [LP86, Section 2.4]. The matching problem for non-bipartite graphs turned out to be substantially more difficult. Almost a decade later Edmonds [Edm65] found the first efficient algorithm to find a maximum cardinality matching in a non-bipartite graph.

The decision and search versions have received considerable attention in the field of parallel algorithms. NC represents a class of problems which have efficient parallel algorithms. A problem is in the class NC if it can be solved in polylogarithmic time using a polynomial number of processors. In the earliest such result for the matching problem, based on the work of Tutte, Lovasz observed that there is an RNC (randomized NC) algorithm to detect if a given graph has a perfect matching [Lov79]. Later, Karp, Upfal and Wigderson [KUW86] and then Mulmuley, Vazirani, and Vazirani [MVV87] showed that constructing a perfect matching, if one exists, is in RNC^3 and RNC^2, respectively. However, no NC algorithm is known, even for the decision version.

Tutte defined the notion of the *Tutte matrix* (Theorem 1), which gave a way to get RNC algorithms for the decision and search versions of the matching problem. The Tutte matrix of a graph G is a skew-symmetric matrix which is obtained by replacing all the nonzero entries in its adjacency matrix with indeterminates. Tutte's theorem states that a graph has a perfect matching if and only if the determinant of its Tutte matrix is not

Class/Version	Decision	Search	Counting
General graphs	RNC & P	RNC & P	#P-hard
Planar graphs	NC	RNC & P	NC
Bipartite planar graphs	NC	NC	NC

Table 1.1: Complexity of different versions of the matching problem

uniformly zero. The determinant of the Tutte matrix is a multivariate polynomial, and we need to check if it is uniformly zero. Now, we know that the determinant of a matrix can be computed in NC [MV99]; and the Schwartz-Zippel Theorem (Theorem 2) tells us that we can compute a polynomial at a few points and can tell, with a good probability, if it is uniformly zero. This gives us an RNC algorithm for the decision version.

Mulmuley et al. [MVV87] observed that if we can assign logarithmic sized weights to the edges of a graph such that there is a unique minimum weight perfect matching, then a perfect matching can be constructed in NC. This, along with the Isolating Lemma - which states that assigning weights randomly would result in a unique minimum weight matching with a good probability - gives an RNC algorithm for constructing a perfect matching.

The case of planar graphs is quite interesting. A graph is planar if it can be embedded on a plane or a sphere, without its edges crossing. Although counting the number of perfect matchings in a graph is #P-hard [Val79], Kasteleyn [Kas67] gave a polynomial time algorithm for the planar case. Kasteleyn [Kas67] defined the notion of a *Pfaffian orientation* for a graph. An orientation is a substitution of either $+1$ or -1 for the indeterminates in the Tutte matrix. It is easy to show that the determinant of the Tutte Matrix is always a square, and its square root is called the *Pfaffian* (see [Lan65, Chapter XV, §9]). The Pfaffian of an oriented graph is just a sum over all the perfect matchings, except that each matching has a sign associated as well, dictated by the orientation [LP86, Section 8.3]. In a Pfaffian orientation, the Pfaffian of a graph turns out to be the number of perfect matchings in it. Kasteleyn showed that every planar graph has such an orientation. Finding such an orientation in a planar graph was shown to be in NC by Vazirani [Vaz89]. So, the problem of counting the number of perfect matchings in a planar graph falls into NC and so the decision version. Table 1 gives the complexity of various versions of the problem for different classes of graphs.

In the general case the counting version is much harder than the search version, which gives us the intuition that the same should hold in the planar case. But surprisingly, an NC algorithm for finding a perfect matching in a planar graph has proved quite elusive. For the case of bipartite planar graphs, Miller and Naor [MN95] first succeeded in giving an NC algorithm to find a perfect matching. They reduced the problem to the problem of computing a maximum flow in a planar graph with multiple sources and sinks. Let us say, in a bipartite graph G, V_1 an V_2 are the two independent sets of vertices. Then let the capacity of every edge be 1. Their idea is to treat every vertex in V_1 as a source with outgoing flow being 1, and every vertex in V_2 as a sink with incoming flow being 1. A maximum integral flow will be a perfect matching in G. They gave an NC algorithm to find a maximum flow in a planar graph with multiple sources and sinks and with known demands, which led to an NC algorithm to find a perfect matching in a bipartite planar graph.

Mahajan and Varadarajan [MV00] gave a different NC algorithm for the bipartite planar case. Their approach directly uses the fact that counting is in NC, and they also generalize it to bipartite graphs of small ($O(\log n)$) genus. Their idea is to first get a point inside the matching polytope and then traverse inside it to reach a vertex of the polytope. The *matching polytope* is the convex hull of all the matching points in the edge space. Mahajan et al. [MV00] use the counting algorithm to find a point inside the matching polytope, that is the centroid of all the matching points. Then they pick an even cycle, and alternately add and subtract some constant ϵ to its edges, so that they remain inside the polytope. The choice of ϵ is such that at least one edge in the cycle gets a weight 0. They give a way to pick a constant fraction of all the faces from a planar embedding, such that they are edge disjoint, which gives them even cycles. Hence, they can destroy a constant fraction of total edges at each step. So, they reach a perfect matching in a logarithmic number of steps. Mahajan and Kulkarni [KM04] generalized the algorithm for general planar graphs to find a vertex of the *fractional matching polytope*, however it is not always a matching.

Kulkarni et al. [DKR08] gave another NC algorithm for bipartite planar graphs by derandomizing the Isolating Lemma, i.e. they could find a weight assignment in NC, under which there exists a unique minimum weight matching. Later Korwar [Kor09] gave a simpler version of the same algorithm. They use the notion of a *nice cycle*, which is an even cycle such that after removing its vertices, the graph still has a perfect matching. Their idea is to assign weights to the edges such that for each nice cycle, the sum of weights for the two sets of alternate edges are not equal. They prove that this would result in a unique minimum weight matching. Let us call the difference between the two sums, the *alternating weight* of the nice cycle. Now, they use a planar embedding of the graph, and assign a linear form to each face, such that the alternating weight of any cycle can be written as a positive linear combination of these linear forms. Here, the bipartiteness of the graph plays a very important role. Finally, they get a weight assignment under which the linear form corresponding to each face takes a positive value. This leads to a stronger result than what is required that alternating weight for each cycle is nonzero. Thus, they can isolate a matching in NC.

Recently, based on the work of Goldberg et al. [GPST92], Mahajan et al. [KMV08] observed that for any class of bipartite graphs which is closed under edge deletion and where the number of perfect matchings can be computed in NC, there is a deterministic NC algorithm for finding a perfect matching. The problem is still open for general planar graphs.

Overview of the thesis

In this work we attempt to get an NC algorithm to construct a perfect matching for general planar graphs. We try to generalize two known NC algorithms for bipartite planar graphs, to the general case. First one is due to Mahajan et al. [MV00, KM04] where they traverse the matching polytope to reach a matching point. The second one involves derandomizing the Isolating Lemma due to the work of Kulkarni et al. [DKR08] and Korwar [Kor09]. In Chapter 2, we describe the concepts of the matching space and the matching polytope. In the next chapters, we look at both the approaches in this light. In Chapter 3, we try to generalize the first approach for non-bipartite planar graphs, but we are stuck as the number of bounding hyperplanes of the matching polytope is very large. In Chapter 4, we tried to generalize the approach given by Korwar, but it is not clear if we can always assign the linear forms as mentioned above. We also give an alternate proof for the idea that giving a nonzero alternating weight to each nice cycle, leads to a unique minimum weight matching. As a by product, we show that every 1-dimensional face of the matching polytope is also a face of the fractional matching polytope. We also explore other possible ways of derandomizing the Isolating Lemma. In the last section of Chapter 4 we describe the Miller and Naor algorithm and study its similarity with the isolation algorithm.

Chapter 2

Preliminaries

An *undirected graph* $G(V, E)$ is a set of *vertices* V and a set of unordered pairs of vertices E, called *edges*. Let n and m represent the cardinalities of the sets V and E respectively. Then, a graph can be represented as a $n \times n$ boolean symmetric matrix, with the entry in i^{th} row and j^{th} column being 1 if $(i, j) \in E$ and 0 otherwise.

Definition 1 (Matching)**.** *A* matching *is a subset of the edge set E, such that no two edges have a common vertex. A perfect matching is a matching which covers all the vertices of the graph.*

In this report, by the term *matching* we would mean a perfect matching unless specified otherwise. A graph can be drawn with the vertices represented as dots, and the edges represented as arcs between the two vertices of the corresponding pairs, called endpoints. An *embedding* of a graph on an orientable surface Σ is a drawing of the graph on Σ, such that the vertices are mapped to distinct points and the edges are mapped to non-crossing curves between the corresponding two endpoints. A *face* of an embedding is a maximal connected subset of Σ that does not intersect the image of any edge or vertex. A graph which can be embedded on a plane or a sphere is called a *planar graph* (see [GT87, Chapter 1]). The *dual graph* G^* of an embedded graph G is a graph whose vertices are the faces of G, and there is an edge between two vertices of G^* if and only if the corresponding two faces of G share an edge. Thus, every edge in G has a corresponding dual edge in G^*.

Another class of graphs, called *bipartite graphs*, seems very important with respect to the matching problems, as there are better algorithms (both sequential and parallel) for some matching problems in the case of bipartite graphs.

Definition 2 (Bipartite graph)**.** *A graph $G(V_1, V_2)$ is bipartite if the vertices can be divided into two disjoint sets V_1 and V_2, such that V_1 and V_2 are* independent sets*, i.e. each edge in G has its one end point in V_1 and other in V_2.*

In a bipartite graph, we can orient the edges such that every vertex has either all incoming edges or all outgoing edges. This property seems very crucial, because Miller and Naor [MN95] used it to give an NC algorithm to find a matching in a bipartite planar graph. The problem in the case of bipartite graphs can be reduced to the maximum flow problem. However the reduction does not seem easy for general graphs.

2.1 The class NC [Pap94]

The class NC has been defined to capture the notion of problems which have efficient parallel algorithms, just like as P has been claimed to capture the notion of efficient computability in the sequential context. Formally, a problem is in NC^i if there exists a constant k such that it can be solved in time $O(\log^i n)$ using $O(n^k)$ parallel processors. Equivalently, NC^i can be defined as a class of decision problems decidable by a uniform family of Boolean circuits $\mathcal{C} = (\mathsf{C}_0, \mathsf{C}_1, \dots)$ with C_n (specializing in the strings of length n) having depth at most $\log^i n$ and size polynomial in n. And,

$$\mathrm{NC} \stackrel{def}{=} \bigcup_{i \geq 0} \mathrm{NC}^i$$

A language L is in RNC if there is a uniform family of NC circuits, with the following additional properties: First, the circuit C_n has now $n + m(n)$ input gates, where $m(n)$ is a polynomial - intuitively, the additional gates are the random bits needed for the algorithm. If a string x of length n is in L, then for at least half of the $2^{m(n)}$ strings y of length $m(n)$, the circuit C_n outputs true on input $x; y$. If $x \notin L$, C_n outputs false on $x; y$ for all y.

2.2 Tutte matrix

The *Tutte matrix* gives a way to get the RNC algorithms for the decision and search versions of the matching problem. We define the Tutte matrix in the following theorem. (see [MR96, Section 12.4])

Theorem 1 (Tutte's Theorem)**.** *Let A be the $n \times n$ (skew-symmetric) Tutte matrix of indeterminates obtained from $G(V,E)$ as follows: a distinct indeterminate x_{ij} is associated with the edge (v_i, v_j), where $i < j$, and the corresponding matrix entries are given by $A_{ij} = x_{ij}$ and $A_{ji} = -x_{ji}$, that is,*

$$A_{ij} = \begin{cases} x_{ij} & (v_i, v_j) \in E \text{ and } i < j \\ -x_{ji} & (v_i, v_j) \in E \text{ and } i > j \\ 0 & (v_i, v_j) \notin E \end{cases}$$

Then G has a perfect matching if and only if $\det A$ *is not identically zero.*

Based on this thoerem, Lovasz [Lov79] gave an RNC (randomized NC) algorithm for the decision version. The idea is to evaluate $\det A$ with the indeterminates replaced by independently and uniformly chosen random values from a suitably large set of integers. If the graph has a perfect matching, then with a good probability, the determinant will be non-zero otherwise it will be zero. The following theorem [MR96, Section 7.2] proves the correctness of this strategy.

Theorem 2 (Schwartz-Zippel Theorem)**.** *Let $Q(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a multivariate polynomial of total degree d over a field $\mathbb{F}$. Fix any finite set $\mathbb{S} \subseteq \mathbb{F}$, and let $r_1, \ldots, r_n$ be chosen independently and uniformly at random from $\mathbb{S}$. Then*

$$Pr[Q(r_1, \ldots, r_n) = 0 | Q(x_1, \ldots, x_n) \not\equiv 0] \leq \frac{d}{|\mathbb{S}|}$$

As computing determinant is in NC [MV99], the above algorithm for deciding the existence of a perfect matching in a graph falls into RNC.

The search version was first shown to be in RNC by Karp, Upfal and Wigderson [KUW86]. Subsequently, Mulmuley, Vazirani and Vazirani [MVV87] gave a different algorithm based on the Tutte matrix. They used the fact that using the Tutte matrix we can do a weighted counting of matchings. Consider some weight assignment to the edges $w\colon E \to \mathbb{Z}$, and let the weight of a matching (or a set of edges) $M \subseteq E$ be the sum of weights of the edges in the matching (or the set of edges), i.e. $w(M) = \sum_{e \in M} w(e)$.

Definition 3. *In a graph $G(V, E)$, a weight assignment to its edges $w\colon E \rightarrow \mathbb{Z}$ is isolating if there exists a unique minimum (or maximum) weight perfect matching.*

We assume that the given graph has at least one matching then it is easy to see that there always exists an isolating weight assignment. Pick any matching M, and assign weights to the edges such that $w(e) = 0\ \forall e \in M$ and $w(e) = 1\ \forall e \notin M$. The weight assignment such constructed is trivially isolating and it isolates M.

Mulmuley et al. [MVV87] observed that if we can get an isolating weight assignment with small weights ($O(\log n)$ sized) then we can find the minimum (or maximum) weight matching in NC. Suppose for a graph G, we have an isolating weight assignment and the weight of the unique minimum weight matching is W. For an edge $(v_i, v_j) \in E$ with weight w_{ij}, substitute x_{ij} with $x^{w_{ij}}$ in the Tutte matrix A of the graph G, for some indeterminate x. Then the highest power of x that divides $\det A$ is x^{2W} [MR96, Section 12.4]. Now, we do the following for each edge in parallel. We remove an edge e from the graph then let A' is the new Tutte Matrix. If e is present in the minimum weight matching, the highest power of x that divides $\det A'$, will be greater than x^{2W}, otherwise equal to x^{2W}. Thus, we can isolate the minimum weight matching. Now, it remains to get an isolating weight assignment. The Isolating Lemma [MVV87] given below, gives a way to get an isolating weight assignment with a good probability. Observe that the set of all possible perfect matchings can be viewed as a family of subsets of E. The Isolating Lemma gives a more general result involving an arbitrary set family.

Lemma 1 (Isolating Lemma [MVV87])**.** *Suppose X is a finite set of elements $X = \{x_1, x_2, \ldots x_m\}$ and $\mathcal{F}$ is a family of subsets of X, i.e. $\mathcal{F} = \{S_1, S_2 \ldots S_k\}$, where $S_i \subseteq X$ for $1 \leq i \leq k$. Given a positive integer weight function $w\colon X \rightarrow \mathbb{Z}^+$, let us define the weight of a set $S \subseteq X$ as $w(S) = \sum_{x \in S} w(x)$. Let $w\colon X \rightarrow \{1, \ldots, 2m\}$ be a positive integer weight function defined by assigning to each element of X, a random weight chosen uniformly and independently from $\{1, \ldots, 2m\}$. Then,*

$$\mathbf{Pr}[\textit{there is a unique minimum weight set in } \mathcal{F}] \geq \frac{1}{2}$$

.

The lemma shows that assigning weights randomly would lead to an isolating weight

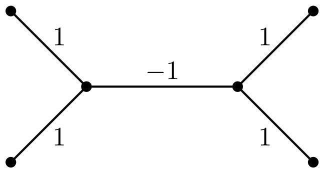

Figure 2.1: A bipartite graph with non-empty matching hyperplane but without any matching

assignment, with a good probability. Thus, we get an RNC algorithm to construct a matching in a given graph. Let us now look at the concepts of the matching space and the matching polytope.

2.3 The matching space

For the graph $G(V, E)$, consider the following m-dimensional space: $\mathbb{R}^E$, the set of all vectors whose entries are indexed by the edges of G. Every subset $F \subseteq E$ can be described by its incidence vector $\mathbf{F}$, the m-tuple $(x_e^F : e \in E)$, where

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

For a vertex u, let $\delta(u)$ denote the set of edges incident on u. Then, consider the following definition.

Definition 4. *The following equations define the* matching hyperplane $\mathcal{M}(G)$ *for a graph* G.

$$\sum_{e \in \delta(u)} x_e = 1 \ \forall u \in V \tag{2.1}$$

Observe that every matching point (incidence vector of a matching) lies in the matching hyperplane $\mathcal{M}(G)$. In fact, we will see later that every positive integral point (with all the coefficients being non-negative and integral) in the matching hyperplane is a matching. Note that it is not necessary that a graph with a non-empty matching hyperplane will always have a matching. Figure 2.1 shows a bipartite graph with a point in the matching hyperplane but it has no matching.

Note that the matching hyperplane $\mathcal{M}(G)$ is a coset of the subspace defined by the equations 2.2, let us call the subspace the *matching space* $\mathcal{M}_0(G)$ of G.

$$\sum_{e \in \delta(u)} x_e = 0 \; \forall u \in V \tag{2.2}$$

In other words, for any point $\mathbf{p} \in \mathcal{M}(G)$, $\mathbf{p} + \mathcal{M}_0(G) = \mathcal{M}(G)$. Standard matching theory [LP86, Chapter 7] tells us that if a graph is connected, then the matching space has the dimension $m - n + 1$ in the bipartite case and $m - n$ in the non-bipartite case. Let us denote the dimension of the matching space $\mathcal{M}_0(G)$ by $d(\mathcal{M}_0(G))$. This difference exists because, in the bipartite case the equations defining the matching space are not linearly independent. Let us see a detailed proof.

Lemma 2. [LP86, Lemma 7.6.1] *For a connected graph G, the dimension of the matching space $\mathcal{M}_0(G)$ is $m - n + 1$, if G is bipartite and is $m - n$, if G is non-bipartite.*

Proof. The claim is equivalent to saying that the matrix composed of the rows consisting of the left hand side of equations 2.2, i.e. the *vertex-edge incidence matrix* $A = (a_{ve})$ of G, has rank $n - 1$ when G is bipartite and n otherwise. First, let us say G is bipartite and (V_1, V_2) be its two disjoint, independent sets of vertices, i.e. each edge has its one endpoint in V_1 and other in V_2. That means, each column of A will have the value 1 in exactly two rows, corresponding to two vertices, one from each of V_1 and V_2. So, if we add up all the rows corresponding to the vertices in V_1 and subtract all the rows corresponding to the vertices in V_2, we get a zero vector. So, the rows are linearly dependent and rank of A is at most $n - 1$.

If the the rank of A is less than n then the rows must be linearly dependent, i.e. there is a non-trivial linear combination of them, which is zero. Suppose in the linear combination, the row corresponding to a vertex v has a coefficient α, then all its neighbouring vertices should have a coefficient $-\alpha$, because every column has exactly two $1's$, corresponding to the two endpoints of the edge. Continuing like this, we can see that every vertex has a coefficient α or $-\alpha$ and any two vertices with an edge between them will have coefficients with opposite signs. Hence, if G is non-bipartite, there is an odd cycle and $\alpha = 0$, hence A has rank n. If G is bipartite then this linear combination can be just a scalar multiple of the combination mentioned above, and hence A has rank $n - 1$. $\square$

2.3.1 A basis of the matching space

Observe that a vector corresponding to the difference of any two points in the matching hyperplane $\mathcal{M}(G)$, lies in the matching space $\mathcal{M}_0(G)$. So, if we want to move from a point to another in $\mathcal{M}(G)$, we need to move along a vector in $\mathcal{M}_0(G)$. Let us now try to construct a basis for the matching space $\mathcal{M}_0(G)$. Let $\mathbf{e}$ represent the incidence vector of the set $\{e\}$ for any $e \in E$. In this report by the term *cycle* we would mean a closed walk, i.e. a cycle with possibly repeated edges. For an even cycle $C = e_1, \dots e_{2k}$ in G, let us define the *alternating vector* corresponding to C as $\mathbf{v}_C = \sum_{i=1}^{2k} (-1)^i \mathbf{e}_i$ (or equivalently $-\mathbf{v}_C$). Consider the set $\mathcal{V}$ of alternating vectors corresponding to all the even cycles in G, i.e. $\mathcal{V} = \{\mathbf{v}_C : C \text{ is an even cycle in } G\}$. We claim that the set $\mathcal{V}$ spans the matching space. To construct a basis, we would need the concept of a pseudo-tree. We give different definitions for a pseudo-tree for bipartite and non-bipartite graphs. For a non-bipartite graph G, a *pseudo-tree* $T \subseteq E$ is a set of edges such that $T = T' \cup \{e\}$ for a spanning tree T' in G and an edge $e \in E \setminus T'$ and T has an odd cycle in it (a non-bipartite graph has at least one odd cycle). For a bipartite graph, a *pseudo-tree* is a spanning tree.

Lemma 3. *The set of alternating vectors $\mathcal{V}$ corresponding to all the even cycles, spans the matching space $\mathcal{M}_0(G)$.*

Proof. Clearly, each vector in $\mathcal{V}$ is in $\mathcal{M}_0(G)$. Now, to prove the lemma, we need to find a set of linearly independent alternating vectors, such that their number is equal to the dimension of the matching space. The set would be a basis for the matching space $\mathcal{M}_0(G)$. Let us first consider the case when G is bipartite and the dimension of the matching space $\mathcal{M}_0(G)$ is $m-n+1$. Let T be a pseudo-tree (spanning tree) in graph G, and let $\overline{T} = E \setminus T$, i.e. the non-tree edges. For each edge $e \in \overline{T}$, $T \cup e$ contains a cycle C_e, which must be even because G is bipartite. Let $\mathcal{C}_T = \{C_e : e \in \overline{T}\}$. Now, the set of alternating vectors $\mathcal{V}_T = \{\mathbf{v}_C : C \in \mathcal{C}_T\}$ corresponding to all the cycles in $\mathcal{C}_T$ is linearly independent, because each cycle in $\mathcal{C}_T$ contains an edge which no other cycle in $\mathcal{C}_T$ has. We know that the number of edges in any spanning tree is $n-1$. So, $|\overline{T}| = m - n + 1$ and hence the set $\mathcal{V}_T$ forms a basis for the matching space $\mathcal{M}_0(G)$.

The case of non-bipartite graph is a little more complex, where the dimension of the matching space is $m - n$. Let us pick an odd cycle C_0, shrink C_0 to a vertex then build a spanning tree T and then expand C_0 back, we get a pseudo-tree T. Clearly, T has total n edges. Let $\overline{T}$ have the rest of the edges, i.e. $\overline{T} = E \setminus T$. Now, let us construct a set of even cycles $\mathcal{C}_T$, like we did in the bipartite case. For each edge $e \in \overline{T}$, $T \cup e$ contains a cycle C_e other than C_0. If C_e is an even cycle, include it in $\mathcal{C}_T$. If C_e is odd, consider further two cases- (i) when C_e shares at least one edge with C_0 (ii) when C_e does not have any edge in common with C_0. In the first case, take the union of the two cycles C_0 and C_e, and remove the common edges, we will get an even cycle which contains e, include it in $\mathcal{C}_T$. In the second case, there exists a path P (possibly of length zero) between C_0 and C_e in $T \cup e$. Let P connect to C_0 at u and to C_e at v. Then the traversal of C_0 beginning at u, followed by the path P going from u to v, then the traversal of C_e beginning at v, followed by the path P going from v to u gives us an even cycle, include it in $\mathcal{C}_T$. Note that, in the alternating vector of this cycle, any edge in the path P gets the same sign in both the traversals because C_e and C_0 are odd cycles. Thus, the alternating vector has a coefficient 2 or -2 for each edge in P. Again each even cycle in $\mathcal{C}_T$ contains an edge which no other cycle in $\mathcal{C}_T$ has. Hence, the set of alternating vectors $\mathcal{V}_T = \{\mathbf{v}_C : C \in \mathcal{C}_T\}$ corresponding to all the cycles in $\mathcal{C}_T$ is linearly independent. Also, $|\overline{T}| = m - n$, hence the set $\mathcal{V}_T$ forms a basis for the matching space $\mathcal{M}_0(G)$. □

The above lemma tells us that, the difference of any two points in the matching hyperplane can be written as a linear combination of the alternating vectors corresponding to even cycles. For a bipartite planar graph G, its faces, which are all even cycles, give a natural way to construct a basis for the matching space. This basis plays a key role in both the NC algorithms for bipartite planar graphs [MV00, Kor09]. Consider the following lemma.

Lemma 4. *For a bipartite planar graph G, and its planar embedding Σ, let us say the alternating vector of the cycle corresponding to a face F is $\mathbf{v}_F$. Then the set $\mathcal{V}_\Sigma = \{\mathbf{v}_F :$ F is an inner face of $\Sigma\}$ forms a basis of the matching space $\mathcal{M}_0(G)$.*

Proof. It is clear that every vector in $\mathcal{V}_\Sigma$ lies in $\mathcal{M}_0(G)$. Suppose, the set $\mathcal{V}_\Sigma$ is linearly dependent, then there is a non-trivial linear combination of the vectors in $\mathcal{V}_\Sigma$, which is zero. We know that any edge is a part of exactly two faces. So, all the edges which are in the outer face, are part of only one of the inner faces. So, all the faces which are adjacent to the outer face will have a coefficient zero in the linear combination. Arguing so on, any face reachable from the outer face in the dual graph G^*, will have a coefficient zero. As, the dual graph G^* is connected, the linear combination becomes trivial. Thus, $\mathcal{V}_\Sigma$ is linearly independent. By *Euler's formula* we know that $|\mathcal{V}_\Sigma| = f - 1 = m - n + 1$ (f is the total number of faces), which is equal to the dimension of the matching space. Hence, the set $\mathcal{V}_\Sigma$ forms a basis of $\mathcal{M}_0(G)$. □

Similarly, for a non-bipartite planar graph, take an embedding in which the outer face is even. Then all the even faces (leaving the outer face) and pairs of odd faces with one fixed odd face (as discussed in Lemma 3) can form a basis.

2.3.2 The matching polytope

With the equations (2.1) defining the matching hyperplane, if we put a constraint $x_e \geq 0$ for each edge in E, we will get a bounded region called, the [1]*fractional matching polytope* $\mathcal{FP}(G)$. This would be a polytope, bounded by m hyperplanes, $x_e = 0 \; \forall e \in E$. Standard matching theory [LP86, Chapter 7] tells us that every matching is a vertex of $\mathcal{FP}(G)$. Consider the convex hull of all the matchings points in G, let us call it the *matching polytope* $\mathcal{P}(G)$, then clearly $\mathcal{P}(G) \subseteq \mathcal{FP}(G)$. In the case of bipartite graphs, the vertices of the fractional matching polytope are exactly the incidence vectors of matchings, i.e. $\mathcal{FP}(G) = \mathcal{P}(G)$. So, we will use the two terms (fractional matching polytope and matching polytope) interchangeably for bipartite graphs. The following constraints define the matching polytope $\mathcal{P}(G)$ for a bipartite graph $G(V, E)$.

$$\begin{aligned} \sum_{e \in \delta(u)} x_e &= 1 \; \forall u \in V \\ x_e &\geq 0 \; \forall e \in E \end{aligned}$$

On the other hand, in the general case the vertices of the fractional matching polytope are *half integral points* (a point in the set $\{0, 1/2, 1\}^m$). In this report by the term half integral point we would mean a vertex of the fractional matching polytope. A half integral point (Figure 2.2) has a corresponding disjoint set of edges E_p, forming a partial matching and another set of edges O_p forming some disjoint odd cycles (O_p is empty for a matching

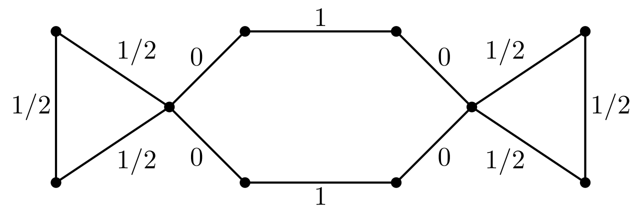

Figure 2.2: A half integral point in a graph

point). The incidence vector of p would be the tuple $(x_e^p : e \in E)$ such that,

$$x_e^p = \begin{cases} 1 & \text{if } e \in E_p \\ 1/2 & \text{if } e \in O_p \\ 0 & \text{otherwise} \end{cases}$$

Unlike the fractional matching polytope, it is not possible to describe the matching polytope by a small number of linear constraints (bounding hyperplanes). Its description is not unique, but we will discuss the one which is relevant to the algorithm given by Mahajan et al. [KM04]. Let us assume that G has an even number of vertices, which is necessary to contain a matching. An *odd cut* of the graph G is the set of edges connecting S to $V \setminus S$, where $S \subset V$ and cardinality of S is odd (for $|S| = 1$, the cut is a trivial cut). The following constraints define the matching polytope $\mathcal{P}(G)$ for a graph G [LP86, Section 7.3].

$$\begin{aligned} \sum_{e \in \delta(u)} x_e &= 1 \ \forall u \in V \\ x_e &\geq 0 \ \forall e \in E \\ \sum_{e \in C} x_e &\geq 1 \quad \text{for every non-trivial odd cut } C \end{aligned} \tag{2.3}$$

We know that the first two constraints define the fractional matching polytope, adding the third, we get the matching polytope. We can easily see that an half integral point, which is not a matching, lies outside the matching polytope. For any half integral point p take any odd cycle O in O_p, and consider the cut C separating O and $V \setminus O$. It is easy to see that the cut inequality (2.3) is not true for the cut C. For bipartite graphs, actually

[1]In this report, the terms fractional matching polytope and matching polytope refer to what are called fractional perfect matching polytope and perfect matching polytope respectively, in standard texts.

the first two constraints trivially imply the third one. So, we do not need the third one for bipartite graphs.

Both the algorithms for finding a matching in a bipartite planar graph, one given by Mahajan et al. [MV00] and other based on isolating weight assignment [DKR08, Kor09] are related to the concepts of matching space and matching polytope. In the next chapters we will describe both the algorithms relating to these concepts and will attempt to get an NC algorithm for general planar graphs. For NC algorithms of some standard graph theory problems like computing a spanning tree, shortest path and connected components, look at [GR88]. We also assume that the given graph is connected, because we can find the connected components in NC and work on each component parallelely.

Chapter 3

Seeking a vertex of the matching polytope

3.1 Bipartite planar graphs

Mahajan et al. [MV00] gave an NC algorithm to find a matching in a bipartite planar graph which explicitly uses the fact that counting can be done in NC for planar graphs. Let the number of matchings in G be M. Also, for an edge e, if we remove its endpoints from G and count the number of matchings in the remaining graph, we will get the number of matchings M_e in G in which e is present. Consider a point $p = (x_1^p, x_2^p \ldots x_m^p)$ in $\mathbb{R}^E$, such that $x_e^p = \mathsf{M}_e/\mathsf{M}\ \forall e \in E$. Then, it is easy to see that the point p is inside the matching polytope, in fact, it is the centroid of all the matching points. Now, their idea is to start from this point and move towards a vertex of the polytope. As G is bipartite, every cycle is an even cycle. Let us pick a cycle $C = e_1 e_2 \ldots e_{2k}$. Now, consider the alternating vector corresponding to C, $\mathbf{v}_C = \sum_{i=1}^{2k} (-1)^{i-1} \mathbf{e}_i$. It is clear that this vector lies in the matching space and by moving along it we remain inside the matching hyperplane. Now, we will move along this vector such that we do not go outside the matching polytope and reach one of the bounding hyperplanes ($x_e = 0$ for some edge e). Let us say, $x_{e_j}^p = \min_{e \in C} x_e^p$ for some edge $e_j \in C$. Consider the vector $\mathbf{v} = (-1)^j x_{e_j}^p \mathbf{v}_C$. Now, if we move to a point $\mathbf{q} = \mathbf{p} + \mathbf{v}$, we are still inside the matching polytope, because by the choice of $x_{e_j}^p$, we know that $x_{e_i}^q \geq 0\ \forall 1 \leq i \leq 2k$. Moreover we can see that $x_{e_j}^q = 0$. Now, remove the edge e_j from G. In other words, we have hit one of the faces of the polytope ($x_{e_j} = 0$) and now we will remain inside it. So the dimension of the search space has been reduced by one. Proceeding so on, we can reduce the dimension of the search space by destroying cycles. In the end we reach one of the vertices of the matching polytope, which is a matching.

To do this in a polylogarithmic number of steps, Mahajan et al. [MV00] again exploit the planarity. Consider a planar embedding of G. Each face gives us an even cycle. Now, pick some edge disjoint faces and process them simultaneously. They give a way to pick a constant fraction of the total faces at once, which are edge disjoint. Thus, they can reduce the dimension of the search space by a constant fraction in each step. Hence, we reach a matching in a logarithmic number of steps.

3.2 General planar graphs

Mahajan and Kulkarni [KM04] generalized the work of Mahajan and Varadarajan [MV00] to general planar graphs. First they choose an initial point in the matching polytope using the counting algorithm. In a non-bipartite graph not every face is an even cycle. Still, they gave a way to pick some edge disjoint even cycles with a bit more complex process, such that a constant fraction of the edges are removed in each step. So, we reach a vertex of the fractional matching polytope in a logarithmic number of steps. However, it is not necessarily a matching, it can be a half integral point.

Recall that for a non-bipartite graph G, the fractional matching polytope $\mathcal{FP}(G)$ and the matching polytope $\mathcal{P}(G)$ are not the same. To define the matching polytope we need some additional inequalities (2.3) corresponding to each non-trivial odd cut. In the algorithm given by Mahajan and Kulkarni [KM04], while traversing the matching hyperplane $\mathcal{M}(G)$, we remain inside the fractional matching polytope but we do not necessarily remain inside the matching polytope. So, to reach a matching point we can try the following ideas.

While moving in $\mathcal{M}(G)$, to ensure that we remain inside the matching polytope we should check that no odd cut inequality (2.3) gets violated. For a point p, let us define the weight of a set of edges $S \subseteq E$ as $w_p(S) = \sum_{e \in S} x_e^p$. Suppose, we are at a point p in $\mathcal{P}(G)$ and then we move to another point q. Now, we need to check if q lies inside the matching polytope, i.e. $w_q(C) \geq 1$ for every non-trivial odd cut C. Let us say $\mathcal{C}_0$ is the set of all non-trivial odd cuts. Let C_{min} be a non-trivial odd cut with the minimum weight

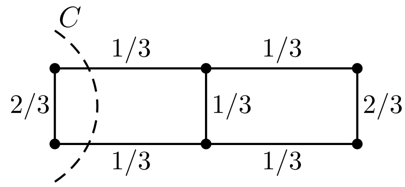

Figure 3.1: An even cut violating the cut constraint

i.e. $w_q(C_{min}) \leq w_q(C)\ \forall C \in \mathcal{C}_0$. If $w_q(C_{min}) \geq 1$ then clearly, q is inside the matching polytope. Although a minimum cut in a planar graph can be found in NC [Joh87], there is no NC algorithm to find a minimum odd cut. So, let us make the constraints more general, and say we apply the inequality to every non-trivial cut instead of odd cuts (we work on each connected component separately, after the removal of zero weight edges). If we succeed in maintaining the cut inequalities for every non-trivial cut, it is more than what is required. But as it turns out, this strategy does not work because our initial point itself might violate the cut inequality for some even cut. Figure 3.1 shows such a graph, with the initial point p (centroid of all the matching points) having a non-trivial even cut C such that $w_p(C) < 1$.

Suppose we can find a minimum odd cut in NC. If $w_q(C_{min}) \geq 1$ then q is inside the matching polytope, otherwise it violates some odd cut inequalities. Then, the idea is to move to a point between p and q such that the minimum odd cut for that point has weight greater than 1. There is a polynomial time algorithm for finding a minimum odd cut [PR82], where the main routine involves finding a minimum cut. As the problem of finding a minimum cut is in NC for planar graphs [Joh87], we believe there should be an NC algorithm for finding a minimum odd cut too. Hence, there should be an NC algorithm to decide if a given point is inside the matching polytope.

Conjecture 1. *For a planar graph G, given a point p in the matching hyperplane $\mathcal{M}(G)$, we can decide if p lies inside the matching polytope $\mathcal{P}(G)$, in NC.*

If this conjecture is true, we might get an NC algorithm to find a matching in a planar graph. Suppose, in a step, we choose the direction such that we move from a point p which is inside $\mathcal{P}(G)$ to a point q outside $\mathcal{P}(G)$. Now, the algorithm to decide if a point is inside $\mathcal{P}(G)$ might give us a way to find a point r on the line segment joining p and q ($\mathbf{r} = \lambda\mathbf{p} + (1-\lambda)\mathbf{q}$ for some $0 < \lambda < 1$), such that r lies on some face of $\mathcal{P}(G)$. We should also get the equation of the face on which r lies, that would be the weight of some odd cuts equal to 1. Now, we should put those cut equalities as constraints, and remain in that face afterwards. We must also make sure that dimension of the search space reduces by a constant fraction in one step. Certainly, even if conjecture 1 is true, we are not sure to get an NC algorithm for finding a matching. It would depend on the kind of the algorithm we get for deciding if a point is in $\mathcal{P}(G)$.

As we find it very difficult to remain inside $\mathcal{P}(G)$, we can try another possible way. Observe that we are not required to necessarily remain inside $\mathcal{P}(G)$. Instead, the idea can be to always go to a face of $\mathcal{FP}(G)$ which has at least one matching point in it, and again reduce the dimension of the search space by a constant fraction. This is equivalent to saying that we should find a constant fraction of edges such that after removing them, the graph still has a matching. Although we can remove such edges one by one using the counting algorithm, it is not clear how to find a constant fraction of total edges with this property.

Chapter 4

Isolation of a minimum weight matching

4.1 Bipartite planar graphs

Recall that, another way to get an NC algorithm for finding a matching is to derandomize the Isolating Lemma. In other words, we need an NC algorithm to compute a weight assignment to the edges of graph G such that there exists a unique minimum weight matching in graph G. Kulkarni et al. [DKR08] and Korwar [Kor09] derandomized the Isolating Lemma for bipartite planar graphs. Let us recall the their key ideas.

Definition 5 (Nice Cycle [LP86])**.** *In a graph G, a simple cycle C is called a nice cycle if the graph obtained after deleting the vertices in C has at least one perfect matching.*

Note that a nice cycle is always an even cycle. For any two matchings M_1 and M_2, their symmetric difference $M_1 \Delta M_2$ is a set of edge disjoint nice cycles. Each of these nice cycles has its edges lying alternately in M_1 and M_2. For an even cycle $C = e_1 \dots e_{2k}$, let $e_1 e_3 \dots e_{2k-1}$ and $e_2 e_4 \dots e_{2k}$ be the two sets of alternate edges. Let us define the *alternating weight* of C, under some weight assignment w, as $w_C = \sum_{i=1}^{2k} (-1)^i w(e_i)$ (or equivalently $-w_C$), i.e. the difference of the weights of the two sets of alternate edges. Note that the alternating weight of a nice cycle C gives us the difference of its contributions to the two matchings, whose symmetric difference contains the cycle C.

Lemma 5 (Nice Cycle Lemma [DKR08, Kor09])**.** *In a graph G, under a weight assignment* *w, if every nice cycle has a nonzero alternating weight, then w is an isolating weight assignment.*

Proof. Let us assume that w is not an isolating weight assignment, and M_1 and M_2 are any two distinct matchings with the minimum weight. We know that the set $M_1 \Delta M_2$ will be a collection of nice cycles, say $\{C_1, \ldots, C_l\}$. For each of these nice cycles, the two sets of alternate edges will have different weights as it has a nonzero alternating weight. Now, we will construct a matching with weight even lesser than that of M_1 and M_2. For each cycle in $\{C_1, \ldots, C_l\}$, taking the set of alternate edges which has lesser weight together with the set $M_1 \cap M_2$ gives us another matching M. As M_1 and M_2 have equal weights, M is different from either of the two and $w(M) < w(M_1) = w(M_2)$. By contradiction, we prove that w is an isolating weight assignment. □

Now, the task is to get a weight assignment under which every nice cycle has a nonzero alternating weight. It does not seem very easy, as there can be exponentially many nice cycles. To tackle this, Korwar [Kor09] defines the notion of the *even cycle space*, which is a subspace of the edge-space (over $\mathbb{F}_2$) spanned by the even cycles of the graph. In case of bipartite graphs the inner faces of the graph, in any planar embedding, forms a basis of the even cycle space. With each basis cycle $C = e_1 \ldots e_{2k}$, they associate a linear form $\sum_{i=1}^{2k} (-1)^i x_{e_i}$. The linear forms are such that any edge gets opposite signs in the two basis cycles (faces) it is present. Then one can show that the alternating sum corresponding to any cycle can be written as a *conical* (positive linear) combination of the linear forms associated with the basis elements. Now, if we get a weight assignment under which the linear form corresponding to each face has a positive value then every cycle will have a nonzero alternating weight, which is a stronger condition than what is required.

A planar embedding can be found in NC (see [KR88]). Assigning linear forms to the faces with the above mentioned properties is also easy in bipartite graphs. Korwar computes a weight assignment under which the linear form corresponding to each face has a value 1. We can get such a weight assignment by solving a few linear equations ($O(n)$). So, an isolating weight assignment can be found in NC for bipartite planar graphs. Unfortunately, in the general case it does not seem easy to get the linear forms such that the alternating sum corresponding to each nice cycle can be written as a conical combination of them.

Note that the even cycle space defined by Korwar [Kor09] and the matching space are similar. The even cycle space is a vector space over $\mathbb{F}_2$ spanned by the even cycles of the graph and the matching space is a vector space over $\mathbb{R}$ spanned by the alternating vectors corresponding to the even cycles of the graph. In fact, the linear forms computed by Korwar [Kor09] form a basis of the matching space. So in other words, for general planar graphs, we want to find a basis of the matching space such that the alternating vector corresponding to every nice cycle can be written in a positive (or negative) linear combination of the basis vectors. In the next section, we will discuss our attempt towards this problem.

4.2 General planar graphs

In an attempt to get an isolating weight assignment we first discuss a few ideas and then we also try to generalize Korwar's idea to general planar graphs. Let us extend the definition of a weight assignment $w\colon E \to \mathbb{R}$ to the matching hyperplane $\mathcal{M}(G)$ as follows. Define $w\colon \mathcal{M}(G) \to \mathbb{R}$ such that $w(p) = \mathbf{w}.\mathbf{p}$ for any point $p \in \mathcal{M}(G)$, where $\mathbf{w} \in \mathbb{R}^E$ is the vector representation of the weight assignment w. We can write the weight of a matching as $w(M) = \mathbf{w}.\mathbf{M}$ where, $\mathbf{M}$ is the incidence vector of the matching M. So, if we see $\mathbf{w}$ as an objective function in $\mathbb{R}^E$ then clearly, w is isolating if and only if $\mathbf{w}.\mathbf{x}$ has a unique minima (or maxima) in $\mathcal{P}(G)$. We will use these notions to look at the Nice Cycle Lemma with the perspective of the matching polytope.

4.2.1 Assigning zero to the pseudo-tree edges

The isolating weight assignments for bipartite planar graphs given by Kulkarni et al. [DKR08] and Korwar [Kor09] have the property that for some tree, all the tree edges get a weight zero. Now, one can ask the question that for any general graph, does there always exists an isolating weight assignment with the tree edges having weight 0. We give an affirmative answer to this question with a slight modification. Recall that we defined the concept of a pseudo-tree in Chapter 2. For a non-bipartite graph a *pseudo-tree* $T = T' \cup e$ is the union of a spanning tree T' and some edge $e \in E \setminus T'$ such that T has an odd cycle. For a bipartite graph a *pseudo-tree* is just a spanning tree. We claim that for any pseudo-tree T, there exists an isolating weight assignment which assigns zero to all the pseudo-tree edges. This is possible because the matching space is not full dimensional and the edges in $\overline{T}$ can represent the basis elements for the matching space. Recall that in Chapter 2, for a graph G we constructed a basis $\mathcal{V}_T$ of the matching space $\mathcal{M}_0(G)$ using a pseudo-tree T. For each edge $e \in \overline{T}$, $\mathcal{V}_T$ has a corresponding vector $\mathbf{v}_e$ such that for all the edges in $\overline{T}$, $\mathbf{v}_e$ has a nonzero coordinate only for the edge e, i.e. $\mathbf{v}_e.\mathbf{e} = 1$ and $\mathbf{v}_e.\mathbf{e}' = 0\ \forall e' \in \overline{T} \setminus e$. Now, consider the following claim.

Claim 1. *Let T be a pseudo-tree in a graph G. Then for any weight assignment w there exists another weight assignment w_0 which assigns 0 to all the pseudo-tree edges, i.e. $w_0(e) = 0\ \forall e \in T$ such that $w(p) = w_0(p)\ \forall p \in \mathcal{M}_0(G)$.*

Proof. Let us say the complement of the set T is $\overline{T} = E \setminus T$ and the basis constructed using T is $\mathcal{V}_T = \{\mathbf{v}_e : e \in \overline{T}\}$, where $\mathbf{v}_e$ represents the vector having a nonzero coordinate for the edge $e \in \overline{T}$. We can write a point $p \in \mathcal{M}_0(G)$ as a linear combination of the basis vectors, i.e. $p = \sum_{e \in \overline{T}} \alpha_e \mathbf{v}_e$. Now,

$$\begin{aligned} w(p) &= \mathbf{w}.\mathbf{p} \\ &= \sum_{e \in \overline{T}} \alpha_e \mathbf{w}.\mathbf{v}_e \end{aligned}$$

Consider another weight assignment w_0 such that $w_0(e) = \mathbf{w}.\mathbf{v}_e\ \forall e \in \overline{T}$ and $w_0(e) = 0\ \forall e \in T$. Now, we know that for any edge $e \in \overline{T}$, $\mathbf{v}_e.\mathbf{e} = 1$ and $\mathbf{v}_e.\mathbf{e}' = 0\ \forall e' \in \overline{T} \setminus e$. So, it is easy to see that $\mathbf{w}_0.\mathbf{v}_e = \mathbf{w}_0.\mathbf{e} = w_0(e) = \mathbf{w}.\mathbf{v}_e\ \forall e \in \overline{T}$. Hence,

$$\begin{aligned} w_0(p) &= \mathbf{w}_0.\mathbf{p} \\ &= \sum_{e \in \overline{T}} \alpha_e \mathbf{w}_0.\mathbf{v}_e \\ &= \sum_{e \in \overline{T}} \alpha_e \mathbf{w}.\mathbf{v}_e \\ &= w(p) \end{aligned}$$

Hence, w_0 is a weight assignment with the required properties. □

Now, consider the following lemma.

Lemma 6. *Let T be a pseudo-tree in a graph G. Then there exists an isolating weight assignment w_0 (with small weights) which assigns zero to all the pseudo-tree edges.*

Proof. Let w be an isolating weight assignment (with small weights) for a graph G, as we know there always exists one. Now, construct a weight assignment w_0 corresponding to w as we did in the above proof. w_0 assigns zero to all the pseudo-tree edges. We also showed that $w_0(p) = w(p)\ \forall p \in \mathcal{M}_0(G)$. Fix a point $p \in \mathcal{M}(G)$, we know that $\mathcal{M}(G) = \mathbf{p} + \mathcal{M}_0(G)$, i.e. for any point $q \in \mathcal{M}(G)$ we can write $\mathbf{q} = \mathbf{p} + \mathbf{q}_0$ for some point $q_0 \in \mathcal{M}_0(G)$. So,

$$\begin{aligned}
w_0(q) &= w_0(p) + w_0(q_0) \\
&= w_0(p) + w(q_0) \\
&= w_0(p) + w(q) - w(p) \\
w_0(q) - w(q) &= w_0(p) - w(p)
\end{aligned}$$

So, $w_0(q) - w(q)$ is a constant for every point $q \in \mathcal{M}(G)$. Hence, as w is isolating, w_0 is also isolating. So, w_0 is an isolating weight assignment which assigns zero to all the pseudo-tree edges. □

Exploiting some other properties of matchings, in fact, we can have an isolating weight assignment with more edges having weight zero (in case of non-bipartite graphs), but these properties are not related to the dimensionality of the matching polytope. Consider a generalization of a pseudo-tree. Suppose $S \subseteq E$ is a set of edges with no even cycles (which can only be a collection of trees in a bipartite graph). Then we can assign weight 0 to all the edges in S and still have an isolating weight assignment. Consider the following lemmas.

Lemma 7. *Let $S \subseteq E$ be a set of edges with no even cycles, and $\overline{S} = E \setminus S$. Then the intersections of any two distinct matchings M_1 and M_2 with $\overline{S}$ will be different, i.e. $M_1 \cap \overline{S} \neq M_2 \cap \overline{S}$.*

Proof. Consider two distinct matchings M_1 and M_2. Suppose they have the same intersection with $\overline{S}$, i.e. $M_1 \cap \overline{S} = M_2 \cap \overline{S}$. Then any edge $e \in M_1 \setminus M_2$ will be in S, because if we assume e to be in $\overline{S}$ then $e \in M_2$, which is not possible. Similarly, any edge

$e \in M_2 \setminus M_1$ will be in S. Thus, $M_1 \setminus M_2 \cup M_2 \setminus M_1 = M_1 \Delta M_2 \subseteq S$. But we know that $M_1 \Delta M_2$ is a set of even cycles, and hence cannot be a subset of S. Hence, by contradiction $M_1 \cap \overline{S} \neq M_2 \cap \overline{S}$. □

Lemma 8. *Let $S \subseteq E$ be a set of edges with no even cycles. Then there exists an isolating weight assignment $w\colon E \to \mathbb{Z}$ with small weights (logarithmic size), such that $w(e) = 0$ $\forall e \in S$.*

Proof. Consider any matching M^*, such that $|M^* \cap \overline{S}|$ is minimum for any matching, i.e. $|M^* \cap \overline{S}| \leq |M \cap \overline{S}|$ for any matching M. Note that there can be more than one choices for M^*. Now, assign a weight 0 to all the edges in $S \cup (M^* \cap \overline{S})$ and assign 1 to the rest of the edges. Clearly M^* has a weight 0 under this weight assignment. We further claim that M^* is the only matching with weight 0. Consider any matching M distinct from M^*. From Lemma 7 we know that $M \cap \overline{S}$ is different from $M^* \cap \overline{S}$ and by the choice of M^* we know that $M \cap \overline{S} \not\subseteq M^* \cap \overline{S}$. So, M has at least one edge with weight 1. Hence, the weight assignment so assigned is isolating. □

These lemmas give us an approach for getting an isolating weight assignment. As every matching has a different intersection with $\overline{S}$, we just need to isolate a subset of $\overline{S}$ which is a part of a matching, let us see in detail. For a set of edges $S \subseteq E$, let us define a parameter $\lambda_S = \min_{M \in \text{matchings}} |\overline{S} \cap M|$. Suppose, we can find a set of edges $S \subseteq E$ with no even cycles such that λ_S is small, i.e. $\lambda_S = O(1)$, then we can easily find an isolating weight assignment. Consider the following lemma.

Lemma 9. *For a graph $G(V, E)$ if we are given a set of edges $S \subseteq E$ with no even cycles such that λ_S is small, i.e. $\lambda_S \leq c$ for some constant c then an isolating weight assignment can be found in NC.*

Proof. We know that there is a matching M_0 such that $|M_0 \cap \overline{S}| = \lambda_S$ is minimum among all the matchings. We will isolate M_0 as follows: as $\lambda_S \leq c$, for each subset $S_0 \subseteq \overline{S}$ such that $|S_0| \leq c$, we do the following in parallel. We isolate S_0, i.e. assign 0 to all the edges in $S_0 \cup S$ and 1 to others. We know that for at least one choice of S_0, there exists a matching M_0 such that $M_0 \cap \overline{S} = S_0$ and $|S_0| = \lambda_S$. For that choice of S_0 the weight assignment would isolate the matching M_0, because any other matching has a different and larger

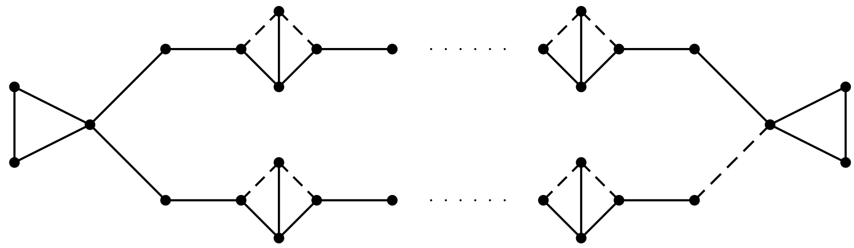

Figure 4.1: A tree S_p made from a half Integral point p with $\lambda_{S_p} = O(n)$

intersection with $\overline{S}$. As $\lambda_S \leq c$, the number of subsets with cardinality $\leq c$ will be $O(n^c)$, i.e. polynomial in n. Hence, we can get an isolating weight assignment in NC. □

We know that a half integral point in a planar graph can be found in NC [KM04]. So, we can try to construct such a set from a half integral point. Given a half integral point p, treat its disjoint edges and odd cycles as vertices and build a spanning tree, then expand the cycles and edges back, we get a set S_p. The set S_p such constructed would be without any even cycles. Observe that if the half integral point p is a matching point then $\lambda_{S_p} = 0$. But, it is not clear if λ_{S_p} for this set would be small in general. All we can say is that the approach will not work for every tree. Figure 4.1 shows a tree (solid edges) S_p formed from a half integral point, we can clearly see that $\lambda_{S_p} = O(n)$. So, it is not clear how to construct a set S_p with no even cycles and $\lambda_{S_p} = O(1)$, in NC.

We can try another idea to get a matching using a half integral point. Let us say for a planar graph, we have a weight assignment w which is not isolating, but it has a property that union of all the minimum weight matchings $\mathcal{M}_w$ is sufficiently small, even then we can find a matching in NC. Assign a Pfaffian orientation to the edges so that the matchings do not cancel each other in the determinant of the Tutte matrix. Then using an idea similar to that of Mulmuley et al. we can find the set $\mathcal{M}_w$, and remove other edges. Now, if $\overline{\mathcal{M}_w}$ is sufficiently large, i.e. it has a constant fraction of the edges, then we can remove a constant fraction of the edges in each step. Hence, we will be done in a logarithmic number of steps.

Observe that the weight assignment corresponding to a matching isolates itself. So, we can expect that a weight assignment corresponding to a half integral point will have a similar behavior. It may not be an isolating weight assignment, but the isolation process may at least remove a constant fraction of edges. As it turns out, this is not the case, Figure

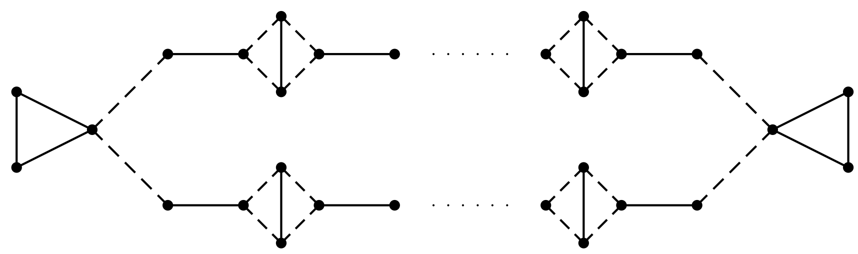

Figure 4.2: The half Integral point as a weight assignment would eliminate only a constant number of edges

4.2 shows a half integral point p in a graph. With the weight assignment w corresponding to p there are exponentially many minimum weight matchings and size of $\overline{\mathcal{M}_w}$ is just a constant.

4.2.2 Nice cycle lemma in the matching polytope

Let us now try to generalize the idea of Korwar [Kor09] to general planar graphs. First, let us see their approach with the perspective of the matching polytope. We will give an alternate proof for the Nice Cycle Lemma (Lemma 5). In a graph G, consider a nice cycle $C = e_1 \dots e_{2k}$ in G, let $C_1 = \{e_1, e_3, \dots e_{2k-1}\}$ and $C_2 = \{e_2, e_4, \dots e_{2k}\}$ be the two sets of alternate edges. We know that after deleting the vertices present in the cycle C from G, we have at least one matching M' in the remaining graph. The union of M' with the two sets of alternate edges gives us two matchings $M_1 = M' \cup C_1$ and $M_2 = M' \cup C_2$. We claim that the line L joining $\mathbf{M}_1$ and $\mathbf{M}_2$ is a 1-dimensional face of the matching polytope $\mathcal{P}(G)$. Note that the alternating vector corresponding to the cycle C, $\mathbf{v}_C = \mathbf{M}_1 - \mathbf{M}_2$ is parallel to L. Let us call all such 1-dimensional faces, faces corresponding to the nice cycle C.

Lemma 10. *In a graph G, let M_1 and M_2 are two matchings such that $M_1 \Delta M_2$ contains exactly one nice cycle C, then the line segment L joining the points $\mathbf{M}_1$ and $\mathbf{M}_2$ will be a 1-dimensional face of $\mathcal{FP}(G)$ and hence a 1-dimensional face of $\mathcal{P}(G)$.*

Proof. Let us assume that L is not a 1-dimensional face of $\mathcal{FP}(G)$, then for any point $\mathbf{M}_{12}$ on L (say $\frac{\mathbf{M}_1+\mathbf{M}_2}{2}$), there exists a vector ϵ, not parallel to $\mathbf{v}_C = \mathbf{M}_1 - \mathbf{M}_2$, such that $\mathbf{M}_{12} + \epsilon$ and $\mathbf{M}_{12} - \epsilon$ both lie in the polytope $\mathcal{FP}(G)$. We know that the coordinates of M_{12} corresponding to the edges in $\overline{C}$ are 0 or 1, i.e. $x_e^{M_{12}} = 0$ or $1 \ \forall e \in \overline{C}$. As for a

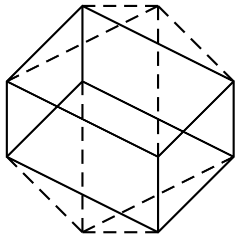

Figure 4.3: Matching polytope (solid edges) inside the fractional matching polytope

point to be in $\mathcal{FP}(G)$, coordinates must lie in $[0, 1]$, ϵ can have nonzero coefficients only corresponding to the edges in C, i.e. $x_e^\epsilon = 0 \; \forall e \in \overline{C}$. As the vector ϵ lies in the matching space it should follow the condition that $\sum_{e \in \delta(v)} x_e^\epsilon = 0 \; \forall v \in V$, so ϵ must be parallel to $\mathbf{v}_C$. Hence, the line L is a 1-dimensional face of $\mathcal{FP}(G)$ and since $\mathbf{M}_1, \mathbf{M}_2 \in \mathcal{P}(G)$, L is a 1-dimensional face of $\mathcal{P}(G)$. □

Furthermore we claim that for every 1-dimensional face L of $\mathcal{P}(G)$ there exists a nice cycle whose alternating vector is parallel to L. Let us say the end points of L are M_1 and M_2. We claim that $M_1 \Delta M_2$ contains only one nice cycle C, then we can see that the alternating vector $\mathbf{v}_C$ is parallel to L.

Lemma 11. *In a graph G, let us say L is a 1-dimensional face of the polytope $\mathcal{P}(G)$, and $\mathbf{M}_1$ and $\mathbf{M}_2$ are the two endpoints. Then $M_1 \Delta M_2$ contains only one nice cycle.*

Proof. Suppose $M_1 \Delta M_2$ has more than one cycle, say $C_1, C_2 \dots C_k$. Let $\mathbf{v}_{C_1}, \mathbf{v}_{C_2} \dots \mathbf{v}_{C_k}$ be their alternating vectors and $\mathbf{M}_1 - \mathbf{M}_2 = \sum_{i=1}^{k} \mathbf{v}_{C_i}$. Consider the mid-point of $\mathbf{M}_1$ and $\mathbf{M}_2$, $\mathbf{M}_{12} = \frac{\mathbf{M}_1 + \mathbf{M}_2}{2}$. Now, let us say $\epsilon = (\mathbf{v}_{C_1} - \mathbf{v}_{C_2} \cdots - \mathbf{v}_{C_k})/2$, then observe that $\mathbf{M}_{12} + \epsilon = \mathbf{M}_2 + \mathbf{v}_{C_1}$ and $\mathbf{M}_{12} - \epsilon = \mathbf{M}_1 - \mathbf{v}_{C_1}$. Now, $\mathbf{M}_2 + \mathbf{v}_{C_1}$ and $\mathbf{M}_1 - \mathbf{v}_{C_1}$ are the matching points obtained from $\mathbf{M}_2$ and $\mathbf{M}_1$ by exchanging the two sets of alternate edges in C_1. So, both the points are inside the matching polytope. Hence we get a direction ϵ, not parallel to $\mathbf{M}_1 - \mathbf{M}_2$, such that by moving in both the directions (+ve and -ve) along it from M_{12} we remain inside the matching polytope. Thus, L is not a 1-dimensional face of $\mathcal{P}(G)$, which is a contradiction. Hence $M_1 \Delta M_2$ has only one nice cycle. □

From the above two results we can make an interesting inference that any 1-dimensional face of the matching polytope $\mathcal{P}(G)$ is also a 1-dimensional face of the fractional matching polytope $\mathcal{FP}(G)$. See Figure 4.3 for clarification, it shows a matching polytope inside a fractional matching polytope with the above property.

Corollary 1. *For a graph G, any 1-dimensional face of the matching polytope $\mathcal{P}(G)$ is also a 1-dimensional face of the fractional matching polytope $\mathcal{FP}(G)$.*

Proof. Consider a 1-dimensional face L of $\mathcal{P}(G)$. Let its end points be the matchings M_1 and M_2. From Lemma 11 we know that the set $M_1 \Delta M_2$ has only one even cycle. But Lemma 10 tells us that if $M_1 \Delta M_2$ has only one even cycle then L is a 1-dimensional face of $\mathcal{FP}(G)$. Hence L is a 1-dimensional face of $\mathcal{FP}(G)$. □

Now, from Lemmas 10 and 11 we can deduce that, for a graph G, every nice cycle C has a corresponding set of 1-dimensional faces of $\mathcal{P}(G)$ which are parallel to the alternating vector $\mathbf{v}_C$ and every 1-dimensional face of $\mathcal{P}(G)$, has such a corresponding nice cycle. Now, this gives us another way of proving the Nice Cycle Lemma (Lemma 5). With a weight assignment w we can associate an objective function $\mathbf{w}.\mathbf{x}$, and let us say the hyperplane H corresponding to the objective function is defined by the equation $\mathbf{w}.\mathbf{x} = k$ for some constant k. Now, a nice cycle C having a non-zero alternating weight means $\mathbf{w}.\mathbf{v}_C \neq 0$, where $\mathbf{v}_C$ is the alternating vector corresponding to C. In other words the hyperplane H is not parallel to any of the $1-$dimensional faces corresponding to C. The condition in Lemma 5 implies that H is not parallel to any of the 1-dimensional faces of $\mathcal{P}(G)$. Hence we can easily see that the objective function $\mathbf{w}.\mathbf{x}$ will have a unique extrema in $\mathcal{P}(G)$. Hence, w is isolating. From this proof it is easy to see that the condition - every nice cycle has a nonzero alternating weight - is sufficient but not necessary. The hyperplane H should not be parallel to any 1-dimensional face which has the minimum weight matching as an endpoint, but it can be parallel to other 1-dimensional faces.

4.2.3 Finding a conical basis

We can look at Korwar's idea [Kor09] in a following way. Let us denote the vector parallel to a 1-dimensional face L of the matching polytope as $\mathbf{v}_L$. Now, the problem is to get a vector $\mathbf{w}$ such that $\mathbf{w}.\mathbf{v}_L \neq 0$ for every 1-dimensional face L of the matching polytope $\mathcal{P}(G)$. For a bipartite planar graph, Korwar [Kor09] used the alternating vectors corresponding to the inner faces of the graph, to form a basis $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of the matching space $\mathcal{M}_0(G)$ ($k = \text{dimension}(\mathcal{M}(G)) = f - 1$). Then they proved that the vector $\mathbf{v}_L$ corresponding to each 1-dimensional face L of $\mathcal{P}(G)$, can be written as a positive (or equivalently negative)

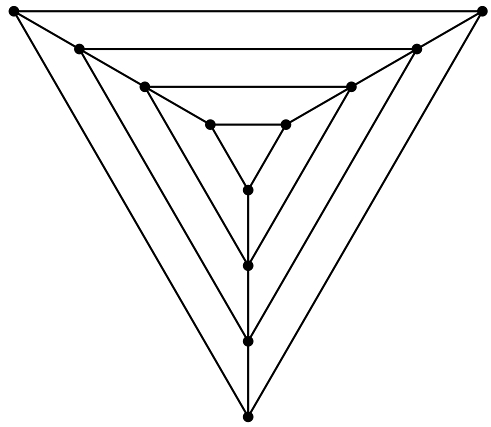

Figure 4.4: A planar graph where the set of alternating vectors corresponding to all but one even faces does not extend to a conical basis

linear combination of $\mathbf{v}_1, \ldots, \mathbf{v}_k$. Let us call a basis with this property, a *conical basis*. Once we get a conical basis, then we can easily get a vector $\mathbf{w}$ which gives a positive dot product with each vector in the conical basis, which will ensure that $\mathbf{w}.\mathbf{v}_L \neq 0$ for every 1-dimensional face L of $\mathcal{P}(G)$. The question is, can we get a conical basis for general planar graphs?

Now, let us try to generalize the idea to the non-bipartite case. For a non-bipartite graph G the dimension of the matching space $\mathcal{M}_0(G)$ is $f-2$. Given a planar embedding, a natural way is to take the alternating vectors corresponding to all but one even faces, and pair up the odd faces, to form the basis. But as it turns out the alternating vectors corresponding to the even faces do not always extend to a conical basis. Figure 4.4 shows such a graph, where this way of forming a conical basis does not work. So, there are no clear ideas about how to get a conical basis in NC. Thus, proving the existence of a conical basis and finding it in NC both remain open problems.

4.3 Similarities with the Miller and Naor algorithm

Recall that the first NC algorithm for bipartite planar graphs was given by Miller and Naor [MN95]. Although they used network flows to get a maximum matching, we can analyse their algorithm in the matching space. In fact, the notion of the circulation space used by them, and the matching space are same for the bipartite case. As we will see, they used the same basis which was later used by Korwar [Kor09], and it plays an important role in their algorithm. Below, we will discuss their algorithm and will look at its similarities with the isolation algorithm [Kor09, DKR08].

Miller and Naor [MN95] gave an NC algorithm to find a maximum flow in a planar network with multiple sources and sinks and with known demands at each source and sink. In a graph G, a flow is a weight assignment to the edges with a direction such that the total incoming flow is equal to the total outgoing flow on all the vertices other than sources and sinks. Every edge has a capacity for the flow; a feasible flow satisfies the capacity constraint for each edge. A maximum flow is a feasible flow with the total outgoing flow from the sources being maximum.

For a bipartite graph $G(V_1, V_2)$ they used the standard reduction from matching to flow, i.e. every edge is directed from V_1 to V_2 with a capacity 1, every vertex in V_1 acts as a source and every vertex in V_2 acts as a sink. Each source and sink has a demand equal to 1. Let G' denote the reduced graph with all the edge capacities assigned. It is easy to see that the edges in a maximum integral flow in G' would form a perfect matching in G. Moreover, every feasible flow in the reduced graph G', satisfying the demands at each source and sink, corresponds to a point in the matching polytope $\mathcal{P}(G)$. Miller and Naor [MN95] defined the notion of a circulation which is a flow with net incoming flow at every vertex (including sources and sinks) being 0. The space spanned by all the circulations is the circulation space. We can see that the circulation space for G' and the matching space $\mathcal{M}_0(G)$ are same.

Miller and Naor [MN95] defined a potential function $\alpha\colon F \to \mathbb{Z}$ on the faces of a planar graph. The potential function on the faces can be seen as assigning a circulation to each face. The set of unit circulations for each face forms a basis of the circulation space. The circulations in all the faces are oriented in the same direction (say, clockwise). Then the flow in an edge e will be $\alpha(f_1) - \alpha(f_2)$ where f_1 and f_2 are the right and left faces adjacent to the edge e, respectively. The unit circulation in a face of G' corresponds to the alternating vector of that face in G. The unit circulation basis also has the property that an edge e gets flows in two opposite directions from its adjacent faces. So, in a way they used the same basis which was later used by Korwar [Kor09, 3.2.1] in the isolation algorithm. The key idea of Miller and Naor is to compute a potential function on the faces to get any feasible circulation given some capacity constraints.

Miller and Naor first changed the problem of computing a flow to that of computing a circulation. They take a spanning tree and add some new edges parallel to the tree edges with some capacities, such that a circulation in the new graph will be equivalent to a maximum flow. Intuitively, they use the spanning tree to redirect a flow, satisfying the demands at each source and sink, from the sinks back to the sources and update the edge capacities accordingly. In other words, they first take an integral point $p \in \mathcal{M}(G)$ in the matching hyperplane. Now, we know that $\mathcal{M}(G) = p + \mathcal{M}_0(G)$ for any point $p \in \mathcal{M}(G)$. So, they bring down the matching polytope to the matching space by subtracting p, let us call the new polytope $\mathcal{P}_0(G)$, i.e. $\mathcal{P}_0(G) = \mathcal{P}(G) - p$. Clearly, the constraints defining $\mathcal{P}_0(G)$ will be the following.

$$\begin{aligned} \sum_{e \in \delta(u)} x_e &= 0 \; \forall u \in V \\ x_e + x_e^p &\geq 0 \; \forall e \in E \end{aligned}$$

For any point $q \in \mathcal{P}_0(G)$ we can easily see that $x_e^q \in [-x_e^p, 1 - x_e^p] \; \forall e \in E$. Now, this gives us a new capacity constraint for each edge. Now, idea is to get a circulation in G', i.e. a vertex M_0 of $\mathcal{P}_0(G)$ and then $M = M_0 + p$ will give us a maximum flow or a matching in G.

To compute the circulation, Miller and Naor used the basis consisted of unit circulations corresponding to the faces of the graph. For a potential function α on the faces , let q be the corresponding circulation or $q \in \mathcal{M}_0(G)$. Then, the flow in an edge e will be $x_e^q = \alpha(f_i) - \alpha(f_j)$, where f_i and f_j are the faces adjacent to the edge e. Now, for q to be a feasible circulation or for $q \in \mathcal{P}_0(G)$ we need to ensure that $-x_e^p \leq x_e^q \leq 1 - x_e^p \; \forall e \in E$. In other words for any two adjacent faces i and j, we must ensure that $\alpha(f_j) - \alpha(f_i) \leq x_e^p$ and $\alpha(f_i) - \alpha(f_j) \leq 1 - x_e^p$. Now, they use the beautiful observation that these conditions are similar to that of the *distance function* (shorted distance from a fixed vertex) in the dual graph. So, they construct a directed weighted dual graph where each primal edge has two corresponding dual edges oriented in opposite directions. If f_i and f_j are adjacent and they share an edge e, then they assign weight x_e^p to (f_i, f_j) and $1 - x_e^p$ to (f_j, f_i) in the dual graph. If we compute the shortest distance to each face from one fixed face f_0

(say the outer face), and assign $\alpha(f_i) = d(f_0, f_i)$ (we also need to prove that there are no negative weight cycles, for the distance function to be defined properly). Then q will be a feasible circulation or the point q will lie inside $\mathcal{P}_0(G)$. As, our initial flow p was integral, and the distance function is also integral, the flow q is integral. Now, $M = p + q$ will give us maximum integral flow, i.e. an integral point in $\mathcal{P}(G)$, which will be a matching.

We can see that Miller and Naor [MN95] used the same basis constructed from the inner faces, which was later used by Korwar [Kor09]. The basis used in both the algorithms has the property that an edge gets opposite signs in the two basis elements it is present. Hence we can observe a similarity between the two algorithms.

Chapter 5

Conclusion

We have tried to generalize two approaches for finding a matching in a bipartite planar graph, one traversing the matching polytope [MV00, KM04] and the other based on derandomizing the Isolating Lemma [DKR08, Kor09], to the general planar case. Based on our attempts, we believe that for general graphs, the main difficulty is that the fractional matching polytope and the matching polytope are different as opposed to their being the same in the bipartite case. So, while in the bipartite case we just need to find a vertex of the fractional matching polytope, in the general case we need to find an integral vertex of the fractional matching polytope, which is generally more difficult. Instead, if we try to work with the matching polytope, in the general case its description is not as easy as the fractional matching polytope. We have an inequality for every odd cut in the graph, which results in an exponential number of constraints. So, it seems difficult to work with the matching polytope in the general case.

The inner faces of a bipartite planar graph give a natural way to construct a basis of the matching space. Mahajan et al. [MV00] use this basis to move inside the matching polytope for a bipartite planar graph. Although Mahajan et al. [KM04] generalized the above approach to navigate inside the fractional matching polytope of a general planar graph, it does not seem easy to stay within the matching polytope, as there are exponentially many bounding hyperplanes. Further we studied a related problem of deciding if a given point is inside the matching polytope, because it seems natural to use this to stay within the polytope. The problem involves finding a minimum odd cut, for which we do not have any NC algorithm. However the following two algorithms - a polynomial time algorithm to find a minimum odd cut [PR82] and an NC algorithm to find a minimum cut in planar graphs [Joh87] - gives us the hope that a minimum odd cut can be found in NC for planar graphs. We conjecture that deciding if a point is inside the matching polytope, can be done in NC for a planar graph. We also believe that this would lead to an NC algorithm to find a matching in a planar graph.

The basis constructed from the inner faces also plays a key role in the isolation algorithm [Kor09]. It acts as a conical basis of the matching space for a bipartite planar graph. We first analyzed the approach with the perspective of the matching polytope and gave an alternate proof for the Nice Cycle Lemma. Then we tried to get a conical basis in the general planar case, but there does not seem to be an easy way to construct it. We also looked at other ways of derandomizing the Isolating Lemma. One of our aims was to find a similarity between all the three algorithms for bipartite planar graphs. As we saw in Section 4.3 the same natural basis, constructed from the faces, is used by Miller and Naor [MN95] in their algorithm. As this basis is important in all the three algorithms, it seems there is a degree of similarity between them. Among the three, the isolation algorithm [Kor09] and the Miller and Naor algorithm [MN95] are more similar as the orientation of each basis element also plays a key role in them.

Bibliography

[DKR08] Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. In *STACS*, pages 229–240, 2008.

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

[GPST92] Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Using interior-point methods for fast parallel algorithms for bipartite matching and related problems. *SIAM J. Comput.*, 21(1):140–150, 1992.

[GR88] Alan Gibbons and Wojciech Rytter. *Efficient parallel algorithms.* Cambridge University Press, New York, NY, USA, 1988.

[GT87] Jonathan L. Gross and Thomas W. Tucker. *Topological graph theory.* Wiley-Interscience, New York, NY, USA, 1987.

[Joh87] Donald B. Johnson. Parallel algorithms for minimum cuts and maximum flows in planar networks. *J. ACM*, 34(4):950–967, 1987.

[Kas67] P W Kastelyn. Graph theory and crystal physics. *Graph Theory and Theoretical Physics*, pages 43–110, 1967.

[KM04] Raghav Kulkarni and Meena Mahajan. Seeking a vertex of the planar matching polytope in NC. In *In Proceedings of the 12th European Symposium on Algorithms ESA, LNCS*, pages 472–483. Springer, 2004.

[KMV08] Raghav Kulkarni, Meena Mahajan, and Kasturi R. Varadarajan. Some perfect matchings and perfect half-integral matchings in NC. *Chicago Journal of Theoretical Computer Science*, 2008(4), September 2008.

[Kor09] Arpita Korwar. Matching in planar graphs. Master's thesis, Indian Institute of Technology Kanpur, 2009.

[KR88] Philip N. Klein and John H. Reif. An efficient parallel algorithm for planarity. *J. Comput. Syst. Sci.*, 37(2):190–246, 1988.

[KUW85] R M Karp, E Upfal, and A Wigderson. Are search and decision programs computationally equivalent? In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, STOC '85, pages 464–475, New York, NY, USA, 1985. ACM.

[KUW86] Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986.

[Lan65] Serge Lang. *Algebra.* Addison-Wesley, 1965.

[LFF56] Jr. L.R. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian Journal of. Mathematics*, 8:399–404, 1956.

[LFF57] Jr. L.R. Ford and D.R. Fulkerson. A simple algorithm for finding maximal network flows and an application to the hitchcock problem. *Canadian Journal of. Mathematics*, 9:210–218, 1957.

[LFF62] Jr. L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979.

[LP86] L. Lovasz and M.D. Plummer. *Matching Theory.* (North-Holland mathematics studies). Elsevier Science Ltd, 1986.

[MN95] Gary L. Miller and Joseph Naor. Flow in planar graphs with multiple sources and sinks. *SIAM Journal on Computing*, 24:1002–1017, 1995.

[MR96] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*, volume 28. ACM, New York, NY, USA, 1996.

[MV99] Meena Mahajan and V. Vinay. Determinant: Old algorithms, new insights. *SIAM J. Discret. Math.*, 12(4):474–490, 1999.

[MV00] Meena Mahajan and Kasturi R. Varadarajan. A new NC algorithm for finding a perfect matching in bipartite planar and small genus graphs (extended abstract). In *STOC'00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 351–357, New York, NY, USA, 2000. ACM.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. In *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 345–354, New York, NY, USA, 1987. ACM.

[Pap94] Christos M. Papadimitriou. *Computational complexity.* Addison-Wesley, Reading, Massachusetts, 1994.

[PR82] Manfred W. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.

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

[Vaz89] Vijay V. Vazirani. 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.