Graph Algorithms using Rank and Determinant

A thesis submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

Thesis Supervisor: Dr. Saket Saurabh

by
Vivek Krishna Pradhan
April, 2013

IISER PUNE

Indian Institute of Science Education and Research Pune
Sai Trinity Building, Pashan, Pune India 411021

This is to certify that this thesis entitled "Graph Algorithms using Rank and Determinant" submitted towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Pune, represents work carried out by Vivek Krishna Pradhan under the supervision of Dr. Saket Saurabh.

Vivek Krishna Pradhan

Thesis committee:
Dr. Saket Saurabh
Dr. Ayan Mahalanobis

Dr. A. Raghuram
Coordinator of Mathematics

Acknowledgments

I would like to thank my guide Dr. Saket Saurabh for helping me learn Graph Algorithms. I thank Fahad and Ashutosh who helped me during my initial days at IMSc. My batchmates Venkata Raghu (currently at IMSc) helped me understand how to work at IMSc and Behlool Sabir provided remote troubleshooting from IISER on various issues. Last but not the least I would like to thank Pranabendu Misra of IMSc for sitting patiently through all my presentations and setting a new speed record for proof-reading a MS thesis.

Abstract

Graph Algorithms using Rank and Determinant

by Vivek Krishna Pradhan

KEYWORDS: *Matchings, planar graph, spanning tree, algebraic algorithm*

Algebraic algorithms are important since they provide elegant and easy solutions to many problems. Even though most fast algorithms for graph problems exploit graphs structure or combinatorial properties, in some situations algebraic solutions out perform these algorithms. Also with future improvements to the basic matrix algorithms and with novel parallel algorithms emerging, algebraic algorithms are set to become faster. In this thesis I explore important algebraic graph theoretic algorithms. All the algorithms discussed here use the same tool at their core i.e. equivalence of computing the determinant, multiplying two matrices, inverting a matrix, and performing Gaussian Elimination. All of these operations take time $\mathcal{O}(n^\omega)$, where ω is the matrix multiplication constant (< 2.373)[1]. Problems discussed are Existence of Bipartite Perfect Matchings, Size of Maximum Bipartite Matching, Counting perfect matchings in planar graphs and counting the number of spanning trees.

Contents

Chapter 1

Introduction

There are many important algorithms that make use of matchings in graphs. There are many problems to study with respect to matchings, for example finding - maximal matchings, maximum matchings and perfect matchings in a graph. Also counting problems like, counting matchings or counting perfect matchings in a graph. Solving these graph problems has important applications in computational chemistry and thermodynamics. Some of these problems like counting perfect matchings are believed to be hard and have no efficient solutions and others like the maximum matching problem have polynomial time solutions.

Spanning trees are an important part of many graph algorithms. There are many algorithms for finding, enumerating and counting spanning trees. Optimization algorithms like minimum spanning trees is also well studied. There are many variants of these problems like the k-minimum spanning tree, degree constrained minimum spanning tree etc. Applications of finding spanning trees and minimum weight spanning trees come up in network design, circuit design, transportation and logistics. The problem of counting spanning trees has also been studied extensively. It has application in assessing reliability of a network represented by a graph.

Initially algorithms in matching theory were algebraic in nature. Early results in bipartite matchings were formulated in terms of matrices. Later results giving efficient algorithms to find matchings are purely graphical in nature. The algebraic algorithms may not be the most efficient but with improvements in algorithms for finding determinant and rank their efficiency will improve, but more importantly they are simple and easy to understand. This makes them important from an academic standpoint.

In this thesis I have studied some important algebraic algorithms in graph theory. First I introduce the necessary background in graphs and algorithms. Then, in Chapter 3, I discuss algorithms that decide the existence of various types of matchings, then move on to the problem of counting the number of perfect matchings in a planar graph. Then in Chapter 4, I discuss a theorem called the matrix tree theorem, which counts the number of spanning trees in a given graph. Proofs are adapted from a bachelor's thesis[2] and a phd thesis[3].

Chapter 2

Preliminaries

2.1 Definitions

Definition 1 (Graph)**.** *A graph $G = (V, E)$ consists of a finite set of vertices V and edges E. An edge is an unordered pair of vertices of the form (u, v) where $u, v \in V$.*

We say a vertex v is adjacent to another vertex u is $(u, v) \in E$.

Definition 2 (Planar Graph)**.** *A graph $G = (V, E)$ is a planar graph if it can be drawn on a plane in such a way that none of the its edges intersect only at their endpoints.*

Definition 3 (Bipartite Graph)**.** *A graph $G = (V, E)$ is a bipartite graph if the set of vertices can be divided into two disjoint sets U and V such that all edges $(u, v) \in E$ are of the form $u \in U$ and $v \in V$.*

Definition 4 (Matching)**.** *In graph $G = (V, E)$ a matching M is a set of edges, such that no two edges in M share a vertex. It is also called an independent edge set.*

Definition 5 (Spanning Tree)**.** *Given graph $G = (V, E)$ a spanning tree is an acyclic connected subgraph of G of the form $T = (V, \widetilde{E})$, with $|\widetilde{E}| = n - 1$*

Note that this tree spans all the vertices of G using the minimum number of edges.

Definition 6. *The sign of a permutation M of n elements $\{1, 2...n\}$ is defined as $(-1)^t$, where t is the number of times we need to exchange of two elements to get back the identity permutation i.e. $sgn(M) = (-1)^t$*

Chapter 3

Matchings - Existence and Counting

3.1 Introduction

Matching theory has contributed immensely to the development of many fields like graph theory, combinatorial optimization and computer science. For example, Edmond's seminal paper titled "Paths, Flowers and Tree"[4] inspired the definition of the class P, and started the field of polyhedral combinatorics. There are many classes of matchings. A few of them are bipartite, non-bipartite, perfect, maximum, weighted etc. Some solutions are combinatorial in nature i.e. they exploit graph and combinatorial properties. Another class of solutions are the algebraic solutions, these algorithms have less to do with the actual graph representation and work by using properties of some special matrices. In this section I explore some of the important algebraic solutions of some matching problems.

First I discuss the problem of deciding if there exists a perfect matching in a bipartite graph. Using a result by Edmonds we describe an algorithm for deciding if a perfect matching exists in a bipartite graph. The next section discusses how this algorithm can also be used to find the size of the maximum matching in a bipartite graph. Then finally we discuss an algorithm to count the number of perfect matchings in a planar graph.

Counting perfect matchings in general graphs was shown to belong to a class of problems called $\#P - Complete$ in 1976 by L. Valiant[5]. Hence it is believed that there is no efficient solution possible for counting perfect matchings in general graphs.

In fact this problem is hard even when restricted to bipartite graphs. Due to a result by Kasteleyn[6] there is a polynomial time algorithm to count the number of perfect matchings in planar graphs.

3.2 Preliminaries

Definition 7 (Adjacency Matrix). *Given a graph $G = (V, E)$ with $n = |V|$, a $n \times n$ matrix is called an adjacency matrix of the graph G if*

$$A_{ij} = \begin{cases} 1, & \textit{if } (i,j) \in E \\ 0, & \textit{otherwise} \end{cases}$$

Definition 8 (Maximum Matching). *In graph $G = (V, E)$ a matching M is a maximum matching if M contains the largest possible number of edges. There are many maximum matchings, but the size of the maximum matching is fixed.*

Definition 9 (Perfect Matching). *Given a graph $G = (V, E)$, a Matching M is called a perfect matching if it contains all the edges of the graph.*

Note that for a graph to have a perfect matching it must have an even number of edges.

Definition 10 (M-Alternating Cycle). *Given a graph $G = (V, E)$, a Perfect Matching M, an M-Alternating Cycle is a cycle that alternates between edges in M and edges not in M.*

Note that every M-Alternating Cycle is a even cycle i.e. it has an even number of edges.

Definition 11 (M,N-Alternating Cycle). *Given a graph $G = (V, E)$, Perfect Matchings M and N, an M,N-Alternating Cycle is a cycle that alternates between edges in M and edges in N.*

Definition 12 (dual Graph). *Given a planar graph $G = (V, E)$ the dual graph of G is a graph $G^* = (V^*, E^*)$, with $V^* = \{f_i | f_i$ is a face of the planar representation of $G\}$ and $E^* = \{(f_i, f_j)|$ There exists an edge between f_i and f_j in the planar representation of $G\}$.*

3.3 Bipartite Perfect Matching

In this section we give an algorithm that can decide if there exists a perfect matching in a given bipartite graph or not in polynomial time. We then define the $n \times n$ matrix

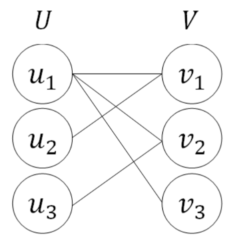


Figure 3.1: A bipartite graph

A (Perfect Matching Matrix) as,

$$A_{ij} = \begin{cases} x_e, & \text{if } v_i u_j \in E \\ 0, & \text{otherwise} \end{cases}$$

Here x_e is a unique variable for each edge in the graph.

Example The matrix for the bipartite graph in figure 3.1 is

$$A = \begin{matrix} & U \quad \rightarrow \\ \begin{matrix} V \\ \downarrow \\ \ \end{matrix} & \begin{pmatrix} x_1 & x_2 & 0 \\ x_3 & 0 & x_4 \\ x_5 & 0 & 0 \end{pmatrix} \end{matrix}$$

Theorem 1. *G has a perfect matching iff det(A)(Perfect Matching Matrix) is not the zero polynomial.*

Proof. Consider, the expansion of the determinant of the matrix A.

$$det(A) = \sum_{\sigma \in S^{2n}} Sgn(\sigma) \prod_{i=1}^{2n} A_{i\sigma(i)}$$

Notice that every term $\prod_{i=1}^{2n} A_{i\sigma(i)}$, is unique since different set of variables are chosen in each term. Additionally, if the edges $(1, \sigma(1)), (2, \sigma(2))...(n, \sigma(n))$ all exist then they form a perfect matching and also the product is non-zero. Since all the terms

are unique (except the zero terms) this proves that if there exists a perfect matching the determinant will be non-zero.
The converse is also true since any non-zero term in the determinant provides us with a perfect matching. □

3.3.1 A randomized algorithm

Since computing the determinant of a variable matrix is not easy, using the above result we can devise a randomized algorithm. We substitute each non-zero variable of A by a random element in a large enough finite field to obtain the matrix B. Now, finding the determinant of the matrix B is straight forward.

Lemma 1. *With high probability if* $det(A) \neq 0$, *then* $det(B) \neq 0$, *if we use elements from a large enough finite field.*

Proof. The proof of this lemma follows directly from the well known Schwartz-Zippel lemma.

Lemma 2. *If* $P \in \mathbb{F}[x_1, x_2 ... x_n]$ *is a non-zero polynomial of degree d,then* $P(r_1, r_2 ... r_n) = 0$ *with probability at most* $d/|\mathbb{F}|$, *where* $r_1, r_2 ... r_n$ *are random elements in* $\mathbb{F}$.

The determinant is a polynomial of degree n. So, if we choose the size of the field to be $|\mathbb{F}| > n^2$, then probability of error is less than $1/n$ □

Now, due to Bunch and Hopcroft, the determinant of A, the rank of A, the inverse of A can all be computed in $\mathcal{O}(n^\omega)$ time, with $\omega < 2.376$. Therefore we have a $\mathcal{O}(n^\omega)$ randomized algorithm to decide if a bipartite graph has a perfect matchings.

3.4 Maximum Bipartite Matching

Maximum bipartite problem can be solved by reducing the problem to the maximum flow problem. In this section we show that the results from the previous section(3.3) can be used to as an algebraic algorithm to find the size of the maximum bipartite matching. Note that the determinant of the matrix A defined in Section 3.3 will vanish if a perfect matching does not exist in the graph. We show here that the rank of the matrix gives us some information about the size of the maximum matching.

Lemma 3. *The size of the maximum bipartite matching is equal to the rank of the matrix A*

Proof. Consider a matching M of the graph $G = (V, E)$ of size k. Now M is a perfect matching of the subgraph G', where G' is formed by deleting vertices of G that are not in M. Note that the matrix $A'(k \times k)$ that corresponds to the Perfect Matching Matrix(section 3.3) of G' is a submatrix of A. From previous theorem 1, the matrix A' is a full rank matrix(of rank k). This shows that $rank(A) \geq rank(A')$ i.e. $rank(A) \geq k$

Now, consider a maximum rank submatrix of A, this represents a subgraph of G. From the theorem 1 (reversed), since the submatrix is a full rank matrix, there exists a perfect matching in the subgraph represented by this matrix. This shows that, $rank(A) \leq k$. □

3.5 Counting Perfect Matchings in a Planar graph

Definition 13. *We define $PM(n)$ to be the set of partitions of n elements to pairs.*

For example, $PM(4) = \{\{(1,2),(3,4)\},\{(1,3),(2,4)\},\{(1,4),(2,3)\}\}$. Also note that each partition of the set of n numbers can be thought of as a permutation into pairs. If the number of vertices is taken to be n then $PM(n)$ gives us all the potential perfect matchings of the graph G, i.e. if $M \in PM(n)$ represents a perfect matching then the quantity $A_{ij} = 1$ if $(i, j) \in M$. So to count the number of perfect matchings in a graph we need to compute the quantity

$$PerfMatch(G) = \sum_{M \in PM(n)} \prod_{(i,j) \in M} A_{ij}$$

Example Let us illustrate this by using a simple graph on four vertices, the square graph. The Adjacency Matrix of this graph is given by,

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

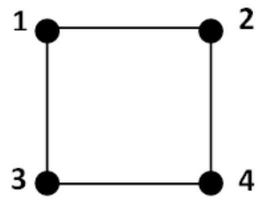


Figure 3.2: A square graph

and $PM(4) = \{\{(1,2),(3,4)\},\{(1,3),(2,4)\},\{(1,4),(2,3)\}\}$.

$$\begin{aligned} PerfMatch(G) &= \sum_{M \in PM(n)} \prod_{(i,j)\in M} A_{ij} \\ &= A_{12}A_{34} + A_{13}A_{24} + A_{14}A_{23} \\ &= 1 + 1 + 0 \\ &= 2 \end{aligned}$$

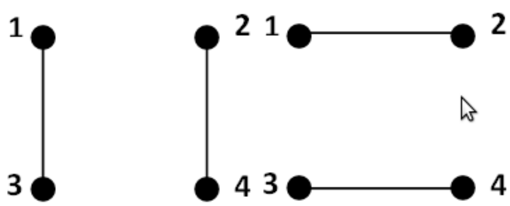


Figure 3.3: The two perfect matchings of the square graph

Definition 14 (Pfaffian). *We define the Pfaffian of a $n \times n$ matrix as $Pf(A) = \sum_{M \in PM(n)} sgn(M) \prod_{(i,j)\in M} A_{ij}$, where $sgn(M)$ is the sign of M as a permutation of n elements.*

Example Let us compute the pfaffian of the adjacency matrix of the square graph used before,

$$\begin{aligned} Pf(A) &= \sum_{M \in PM(n)} Sgn(M) \prod_{(i,j)\in M} A_{ij} \\ &= A_{12}A_{34} - A_{13}A_{24} + A_{14}A_{23} \\ &= 1 - 1 + 0 \\ &= 0 \end{aligned}$$

It is clear that the definitions of the pfaffian of a matrix A and the value that we want to compute $PerfMatch(G)$ are very similar. We are interested in computing the value of $PerfMatch(G)$ using the value of the Pfaffian, since due to the following result of Muir[7] it is easy to compute $Pf(A)$ for skew-symmetric matrices.

Lemma 4 (Muir, 1882)**.** *Let A be a skew-symmetric matrix, then $Pf(A)^2 = det(A)$, where $det(A)$ is the determinant of A.*

Since we can compute the pfaffian of a skew-symmetric matrix easily we now construct a skew-symmetric matrix from the graph G. We do the following:

1. From the given undirected graph G we construct the directed graph G' by orienting each edge of G in some direction.

2. Now the resultant adjacency matrix A' is skew-symmetric since if there is an edge $i \to j$ then we have $A'_{ij} = 1$ and $A'_{ji} = -1$.

3. We can now compute $Pf(A') = \sqrt{det(A')}$.

But, we want to be sure that

$$Pf(A') = \pm PerfMatch(G)$$
$$\sum_{M \in PM(n)} sgn(M) \prod_{(i,j)\in M} A'_{ij} = \pm \sum_{M \in PM(n)} \prod_{(i,j)\in M} A_{ij}$$

Since the terms are non-zero only when M is a perfect matching, we want that

$$sgn(M) \prod_{(i,j)\in M} A'_{ij} = \pm \prod_{(i,j)\in M} A_{ij}, \text{ where } M \text{ is a perfect matching.}$$

Hence, if we find a way to orient the edges in G such that $sgn(M)\prod_{(i,j)\in M} A'_{ij} = s$, where s is either +1 or -1 and it is the same for all M which are perfect matchings in G.

Definition 15. *The Orientation for which the above property is satisfied is said to be a Pfaffian Orientation of G i.e. if G' is a Pfaffian orientation of G then, for any two perfect matchings M and N, we must have $sgn(M)\prod_{(i,j)\in M} A'_{ij} = sgn(N)\prod_{(i,j)\in N} A'_{ij}$*

Example Recall that in a previous example the $Pf(A) = 0$, while $PerfMatch(G) = 2$, where G had A as the adjacency matrix. In this example we show that when using a Pfaffian orientation, $Pf(A) = PerfMatch(G)$.

The Adjacency Matrix of this graph is given by,

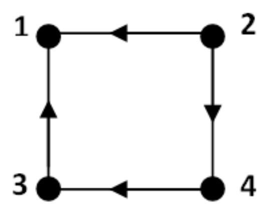


Figure 3.4: Pfaffian orientation of the square graph

$$A = \begin{pmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{pmatrix}$$

and $PM(4) = \{\{(1,2),(3,4)\},\{(1,3),(2,4)\},\{(1,4),(2,3)\}\}$.

$$\begin{aligned} Pf(G) &= \sum_{M\in PM(n)} Sgn(M) \prod_{(i,j)\in M} A_{ij} \\ &= A_{12}A_{34} - A_{13}A_{24} + A_{14}A_{23} \\ &= (-1)(-1) - (-1)(1) + 0 \\ &= 2 \end{aligned}$$

We have now reduced the problem of counting the number of perfect matchings to the problem of finding a pfaffian orientation. To show that we can find the pfaffian orientation of planar graphs we first need to prove the following lemma.

Definition 16 (Nice Cycle)**.** *An even cycle C such that, if C were removed, G would still have a perfect matching is called a nice cycle.*

Definition 17 (Oddly Oriented)**.** *If a cycle C, has has odd number of edges going in clockwise and anti-clockwise direction.*

Lemma 5. *Let G be a graph and G' be an orientation of G. Then G' is a pfaffian orientation if every nice cycle in G is oddly oriented in G'. Here*

Proof. Let us assume that G' is an orientation of G such that every nice cycle in G is oddly oriented in G'. Now let M and N be any two perfect matchings of G. Suppose C is a M-Alternating Cycle. We know that C is even. Notice that $M - C$ is a perfect matching for the graph formed by deleting C from G. Therefore an M-Alternating Cycle is a nice cycle and it is oddly oriented. Similarly an M,N-Alternating Cycle is oddly oriented.

Now, $M \cup N$ may contain many oddly oriented even cycles. Let us consider one such cycle C formed by edges $\widetilde{M} \subseteq M$ and $\widetilde{N} \subseteq N$. Let us write the cycle by listing the vertices in clockwise order, $\widetilde{M} = \{(i_1, i_2), (i_3, i_4)...(i_{2k-1}, i_{2k})\}$, then $\widetilde{N} = \{(i_2, i_3), (i_4, i_5)...(i_{2k}, i_1)\}$. Since C has odd number of edges oriented clockwise, one of $\widetilde{N}$ and $\widetilde{M}$ have even number of clockwise edges while the other has an odd number. Without loss of generality let us assume that $\widetilde{M}$ has even number of clockwise edges, say there are $2t$ number of edges. Therefore, $\prod_{(i,j)\in\widetilde{M}} A_{ij} = (-1)^{|\widetilde{M}|-2t}(+1)^{2t} = (-1)^{|\widetilde{M}|} = +1$.

Now let us compute $\prod_{(i,j)\in\widetilde{N}} A_{ij}$, we know $\widetilde{N}$ has odd number of clockwise edges say $2l+1$, therefore $\prod_{(i,j)\in\widetilde{N}} A_{ij} = (-1)^{|\widetilde{N}|-(2l+1)}(+1)^{2l+1} = -(-1)^{|\widetilde{N}|} = -1$

Now, suppose $M = \{\widetilde{M}_1 \cup \widetilde{M}_1 ... \widetilde{M}_\gamma\} \cup (M \cap N)$, here every $\widetilde{M}_i$ is a cycle of $M \cup N$. Therefore

$\prod_{(i,j)\in M} A_{ij} = \prod_{(i,j)\in\widetilde{M}_1} A_{ij} \cdot \prod_{(i,j)\in\widetilde{M}_2} A_{ij} \cdots \prod_{(i,j)\in\widetilde{M}_\gamma} A_{ij} \cdot \prod_{(i,j)\in M\cap N} A_{ij}$
$= 1^\gamma \cdot \prod_{(i,j)\in M\cap N} A_{ij}$

Similarly $\prod_{(i,j)\in N} A_{ij} = (-1)^\gamma \cdot \prod_{(i,j)\in M\cap N} A_{ij}$. Therefore,
$\prod_{(i,j)\in N} A_{ij} \cdot \prod_{(i,j)\in M} A_{ij} = (-1)^\gamma \cdot \left(\prod_{(i,j)\in M\cap N} A_{ij}\right)^2$.

The square of (± 1) is 1, so we get,

$$\prod_{(i,j)\in N} A_{ij} \cdot \prod_{(i,j)\in M} A_{ij} = (-1)^\gamma$$

Now, $Sgn(M) = Sgn(\widetilde{M}_1 \cup \widetilde{M}_1 ... \widetilde{M}_\gamma)$

$= Sgn(i_{11}, i_{12}...i_{1(2k_1)}, i_{21}, i_{22}...i_{2(2k_2)}......i_{\gamma 1}, i_{\gamma 2}...i_{\gamma(2k_\gamma)})$
$= (-1)Sgn(i_{12}, i_{13}...i_{11}, i_{21}, i_{22}...i_{2(2k_2)}......i_{\gamma 1}, i_{\gamma 2}...i_{\gamma(2k_\gamma)})$
Since we have done $2k_1 - 1$(odd) number of swaps of two numbers in this step
$\vdots$ Repeating the above steps γ number of times (for each $\widetilde{M_i}$)
$= (-1)^\gamma Sgn(i_{12}, i_{13}...i_{11}, i_{22}, i_{23}...i_{21}......i_{\gamma 2}, i_{\gamma 3}...i_{\gamma 1})$
$= (-1)^\gamma Sgn(\widetilde{N}_1 \cup \widetilde{N}_1...\widetilde{N}_\gamma) = (-1)^\gamma Sgn(N)$, i.e.

$$Sgn(M) \cdot Sgn(N) = (-1)^\gamma$$

Therefor multiplying the above two equations we get,

$$\left(Sgn(M) \prod_{(i,j)\in M} A_{ij}\right)\left(Sgn(N) \prod_{(i,j)\in N} A_{ij}\right) = (-1)^{2\gamma} = 1$$

This shows that if a graph G has an orientation G' such that every nice cycle in G is oddly oriented in G' then all terms of the form $Sgn(M)\prod_{(i,j)\in M} A_{ij}$, where M is a perfect matching have the same sign i.e. this is a pfaffian orientation. □

Now we prove the main result that shows the existence and a method to find the pfaffian orientation of planar graphs.

Theorem 2. *Let G be a planar graph. Then*

1. *G can be oriented efficiently so that each face except one has an odd number of edges oriented in clockwise (called clockwise odd orientation) and,*

2. *This is a Pfaffian Orientation of G.*

Proof. **Proof of Part-I**
We prove this by construction. Let us consider the dual graph G^* of the given planar graph. We find a spanning tree of this dual graph. Now we choose a face of our original graph starting from the leaf nodes of the spanning tree of the dual graph and orient all the edges arbitrarily except one edge. This remaining edge can be oriented in such a way that the face has an odd number of edges. The next face to be chosen has to be adjacent to the leaf nodes face on the spanning tree of the dual graph. We repeat the process until all the faces have been oriented in clockwise odd orientation, except the final face for which we may or may not be able to give a clockwise odd

orientation.

This method works since suppose we are orienting the face f_i, with $i < |faces|$. Then there exists a face f_k which is adjacent to f_i is yet to be oriented. This means that the edge that is shared by f_i and f_k is not yet given an orientation. Hence we can use this edge to correctly orient f_i. This property is true until the last face that we are orienting. So we can orient all the faces except one.

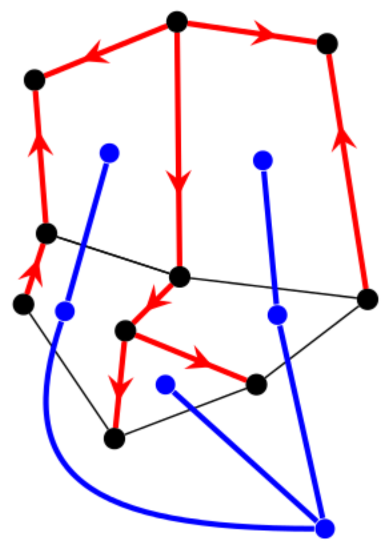


Figure 3.5: Spanning tree of the dual graph. All the directed edges can be oriented arbitrarily, the undirected edges can be used to make each face clockwise odd

Proof of Part-II

Due to the previous lemma, if suffices to show the following result.

Lemma 6. *Let G be a planar graph. If G' an orientation so that each internal face has an odd number of lines oriented clockwise, then every nice cycle in G is oddly oriented in G'*

We use the following version of Euler's formula:

Lemma 7. *For any cycle C, $e = v + f - 1$, where e is the number of edges inside C, v is the number of vertices inside C, and f is the number of faces inside C*

Let C be a nice cycle, let c_i be the number of clockwise lines on the boundary of face i in C. and c be the number of clockwise lines on C. Since our orientation is clockwise odd, we have, $c_i \equiv 1 \ mod \ 2$, so $f \equiv \sum_{i=1}^{f} c_i \ mod \ 2$.
But, $\sum_{i=1}^{f} c_i = c+e$, since each interior line is clockwise on one face and anti-clockwise on another and so is counted once.
This shows that, $f \equiv c + (v + f - 1) \ mod \ 2$, so $c \equiv (v - 1) \ mod \ 2$. But, since C is a nice cycle $v \equiv 0 \ mod \ 2$. This proves that C is oddly oriented.

□

Chapter 4

Counting Spanning Trees

4.1 Definitions

Definition 18 ("weakly" connected components)**.** *Given a directed graph $G = (V, E)$ a weakly connected component is a subgraph in which the undirected graph formed by replacing all directed edges with undirected edges, is connected.*

4.1.1 Incidence Matrix

Definition 19 (Incidence Matrix)**.** *Given a graph $G = (V, E)$ with $V = \{1, 2...n\}$ and $E = \{e_1, e_2...e_m\}$ be a directed graph. Then the incidence matrix S_G is a $n \times m$ matrix, defined as:*

$$(S_G)_{ij} = \begin{cases} 1, & \text{if } e_j \text{ ends in } i \\ -1, & \text{if } e_j \text{ starts in } i \\ 0, & \text{otherwise} \end{cases}$$

Note that for an undirected graph G any incidence matrix $S_{\overrightarrow{G}}$ of some arbitrarily oriented directed variant $\overrightarrow{G}$ of G can be taken as the incidence matrix. As an example consider the graph K_4 in figure 4.1, the complete graph on four vertices. The incidence matrix can be computed on any directed variant of this graph, let us use the directed

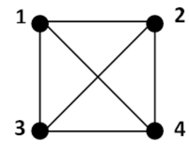


Figure 4.1: The complete graph on 4 vertices

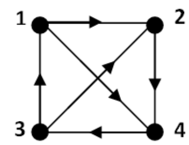


Figure 4.2: Directed variant of the complete graph on 4 vertices

version given in figure 4.2. The incidence matrix for this graph is:

$$S_{K_4} = \begin{pmatrix} -1 & 0 & 0 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & 0 & 1 & 0 \end{pmatrix}$$

We will exploit the following property of incidence matrices in subsequent proofs.

Theorem 3 (Incidence Matrix Rank). *The rank of the incidence matrix of a graph on n vertices is :*

$$rank(S_G) = n - |\text{“weakly” connected components of } G|$$

Proof. Let us assume that |“weakly” connected components of $G| = r$, then we can rearrange the columns of S_G to make S_G be comprised of the incidence matrices of the “weakly” connected components, $S_{G_1}, S_{G_2} ... S_{G_r}$ as shown

$$S_G = \begin{pmatrix} S_{G_1} & & \cdots & 0 \\ & S_{G_2} & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & S_{G_r} \end{pmatrix}$$

All entries outside of the submatrices S_{G_k} are zero since if there was a non-zero entry in any other place it would mean that there exists an edge from some S_{G_i} to some other S_{G_j}. Since each of these are “weakly” connected components such an edge cannot exist and hence all entries outside of the submatrices S_{G_k} are zero. Now further this shows that the number of independent columns of S_G = sum of the number of independent columns of all submatrices S_{G_i} i.e.

$$rank(S_G) = \sum_{i=1}^{r} rank(S_{G_i}), \text{where each } S_{G_i} \text{ is a ``weakly'' connected component of } G \tag{4.1}$$

Now if we prove that $rank(S_{G_i}) = n_i - 1$, where G_i is a connected graph and n_i is the number of vertices of G_i, then substituting in the above equation 4.1 get the statement of this theorem. Hence we only need to show:

Lemma 8. *$rank(S_{G_i}) = n_i - 1$, where G_i is a connected graph and n_i is the number of vertices of G_i*

Now, let us consider a connected graph G and its incidence matrix S_G. If we find cycles in the graph G we can delete an edge from the cycle. Analogously we can delete the column corresponding to such an edge without changing the $rank(S_G)$. Since if the edges $e_{k_1}, e_{k_2}...e_{k_t}$ form a cycle then the columns $k_1, k_2...k_t$ sum up to zero. Repeating this deletion of edges we can reduce this graph G to a spanning tree G' of G, such that $rank(S_{G'}) = rank(S_G)$. We find that G' is a graph having n vertices and $n-1$ edges since it a connected tree on n vertices. Hence the incidence matrix $S_{G'}$ is a $n \times n - 1$ matrix. We claim that all columns of $S_{G'}$ are linearly independent.

This can be proved by assuming the contrary i.e. there exist some set of columns $k_1, k_2...k_t$ and corresponding set of integers $\alpha_1, \alpha_2...\alpha_t$ such that $\sum_{i=1}^{t} \alpha_i S_{G'}(; , k_i) = 0$. But we know that if sum of columns of $S_{G'}$ adds to zero then they must form a cycle. This is a contradiction since G' is a tree it cannot have cycles.

Since all columns of $S_{G'}$ are independent it is a full rank matrix. This shows that $rank(S_G) = rank(S_{G'}) = n - 1$. Hence the lemma is proved. This in turn proves the theorem. □

Definition 20. *For any $n \times m$ matrix A define the $n-1 \times m$ matrix $\widetilde{A}$ as the matrix A without the n-th row.*

4.2 Kirchoff's Matrix Tree Theorem

Theorem 4 (Matrix Tree Theorem)**.** *The number of spanning trees of a graph G can be calculated as: $det(D_G)$, where $D_G = \widetilde{S}_G \cdot \widetilde{S}_G^T$*

Remark: Note that D_G is a $n-1 \times n-1$ matrix and it is of the form

$$(D_G)_{ij} = \begin{cases} deg(i), & \text{if } i = j \\ -1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Let us start the proof by proving some basic lemmas that will be useful to us in this proof.

Lemma 9. *Let $T = (V, E)$ be a undirected tree that is rooted at n. We can order and orient the edges so that e_i ends at the vertex i.*

Proof. We start with a undirected tree with vertices labeled from 1 to n. Since the edges are still unlabeled, we label the edges as $e_i := (p(i), i)$, where $p(i)$ is the parent of i. Now we have a undirected tree with edges labeled such one of the vertices in every edge e_i is i. Now we direct the edges such that e_i ends in i. Note that the orientation of each edge is such that e_i ends in i. □

Note that for every spanning tree of a graph G, there is a unique directed tree rooted at n and such that e_i ends in i. So for counting spanning trees it is enough to count all directed trees of this form.

Lemma 10. *Let $G = (V, E)$ with $|E| = n - 1$ be a directed graph which is not a tree. Then $det(\widetilde{S}_G) = 0$.*

Proof. Since G is a graph on n vertices with $n - 1$ edges and is not a tree, it cannot be a connected graph. To see this we know that it has at least one cycle and hence if we remove an edge from each cycle in G we end up with a acyclic graph G'. Number of edges in this graph $|E'| < n - 1$ since it had at least one cycle. We know that for an acyclic graph to be connected it requires at least $n - 1$ edges. Hence we have shown that the graph is not connected.
Now by Theorem 3 we know that
$rank(S_G) = rank(\widetilde{S}_G) = n - |$"weakly" connected components of $G|$, but we know

that |"weakly" connected components of $G| \geq 2$.
Hence, the $rank(\widetilde{S}_G) \leq n-2$ and $\widetilde{S}_G$ is a $n-1 \times n-1$ matrix. Since $\widetilde{S}_G$ is not a full rank matrix, $det(\widetilde{S}_G) = 0$. □

Lemma 11. *Let $T = (V, E)$ be a directed tree with $e_i \in E$ ending in $i \in V$. Then $det(\widetilde{S}_T) = 1$*

Proof. Since every edge $e_i \in E$ ending in $i \in V$, all the diagonal entries in the $n-1 \times n-1$ matrix will be 1 (recall definition 19). If we simultaneously reorder a row and a column the determinant will not change. We can reorder the vertices in $\widetilde{S}_T$ so that $p(i) > i$ while simultaneously reordering a corresponding edge to make sure that the edge $e_i \in E$ ending in $i \in V$. Then the matrix will look like

$$\widetilde{S}_T = \begin{pmatrix} 1 & * & \cdots & * \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Hence the $det(\widetilde{S}_T) = 1$. □

Proof of Matrix Tree Theorem: Notice that the i-th column of D_G can be thought to be composed of sum of incidence vectors that each correspond to an edge in an orientation $\overrightarrow{G}$ of G, such that the edge ends in the vertex i. Consider the i-th column of the determinant. It is a vector that contains the value $deg(i)$ as the i-th entry, rest of the terms are either -1 or zero. Note that the number of -1s in the vector can be $deg(i)$ or $deg(i) - 1$ (since row corresponding to n-th vertex is deleted). Hence we can split the i-th column into $deg(i)$ vectors each having the value 1 as the i-th entry, one entry as -1(all except one) and rest zero. These vectors now have the same property as the columns of $\widetilde{S}_G$ (representing edges). The i-th column may look like the following:

$$\begin{matrix} 1 & 2 & & i & & n-1 \\ \end{matrix}$$
$$\begin{pmatrix} 0 & -1 & \cdots & deg(i) & \cdots & -1 \end{pmatrix} =$$

Note that the above vector represents a vertex. This can be split into $deg(i)$ vectors that represent edges of some subgraph that end in the i-th vertex(columns of $\widetilde{S}_G$)

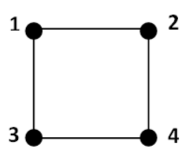
$$\begin{matrix} 1 & 2 & & i & & n-1 \\ \end{matrix} \qquad \begin{matrix} 1 & 2 & & i & & n-1 \\ \end{matrix}$$
$$\begin{pmatrix} 0 & -1 & \cdots & 1 & \cdots & 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 & 0 & \cdots & 1 & \cdots & -1 \end{pmatrix}$$

If we split the determinant using this property on every column (linearly of determinant in every column) then we obtain

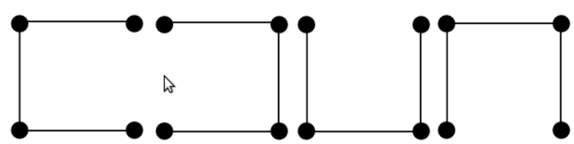
$$det(D_G) = \sum_{H \in \mathcal{H}} det(\widetilde{S}_H)$$

Here, $\mathcal{H}$ is the family of graphs that are subgraphs of G having $|E| = n - 1$ and the property that every edge ends in i. Using the previous two lemmas we know that the determinants in the summation output 1 if they correspond to a spanning tree and 0 otherwise. Hence we have $det(D_G) = \sum_{H \in \mathcal{H}} det(\widetilde{S}_H) =$ number of spanning trees of G

Example Let us take the example of a square graph and work through the above proof. For this graph,



(a) The square graph


(b) The four spanning trees of a square graph

$$\widetilde{D}_G = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

But, not that each column can be split as,

$$\begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} -1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Splitting D_G along its first column gives us,

$$det(\widetilde{D}_G) = \begin{vmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{vmatrix} = \begin{vmatrix} 1 & -1 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 2 \end{vmatrix} + \begin{vmatrix} 1 & -1 & -1 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{vmatrix}$$

Similarly splitting over other columns we can split $det(D_G)$ into determinants of incidence matrices of eight candidate spanning trees as shown,

$det(\widetilde{D}_G) =$

$$\begin{vmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{vmatrix} = \begin{matrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & -1 & -1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \end{matrix}$$

$$det(\widetilde{D}_G) = \begin{vmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{vmatrix} = \begin{matrix} 1+0+0+1 \\ + \\ 1+1+0+0 \end{matrix}$$

$det(\widetilde{D}_G) = 4$, this is the number of spanning trees of a square graph.

References

[1] V. V. Williams, "Breaking the coppersmith-winograd barrier. unpublished manuscript," 2011.

[2] J. Nguyen, "Perfect matchings and pfaffian orientation," Master's thesis, Universiteit van Amsterdam, 2008.

[3] D. Kamenetsky, *Ising Graphical Model.* PhD thesis, The Australian National University, 2010.

[4] J. Edmonds, "Paths, trees, and flowers," *Canad. J. Math.*, vol. 17, pp. 449–467, 1965.

[5] L. G. Valiant, "The complexity of computing the permanent," *Theor. Comput. Sci.*, vol. 8, pp. 189–201, 1979.

[6] P. W. Kasteleyn, "Graph theory and crystal physics," *Graph Theory and Theoretical Physics*, pp. 43–110, 1967.

[7] T. Muir, *A Treatise on the Theory of Determinants.* Macmillan and Co., 1882.