| | |
|---|---|
| **COL758: Advanced Algorithms** | **Spring 2019** |

Lecture 16: February 28

*Lecturer: Naveen Garg* *Scribe: Akshay Kumar Pansari*



## 16.1 Red-Blue Matching Problem : When Unique Matching Exist

This was introduced in the previous class.
**Problem Statement:** Given a graph $G = (U, V, E)$ with each $e \in E$ is either red or blue, does there exist a perfect matching with exactly k red edges?

Consider the n × n matrix A, whose entries are given as

$$A(i,j) = \begin{cases} y, & (i,j) \in \text{E and is red} \\ 1, & (i,j) \in \text{E and is blue} \\ 0, & \text{otherwise} \end{cases}$$

Assume that there exist a unique red-blue perfect matching.

$$det(A) = p(y) = \sum_{n=0}^{n} a_i y^i$$

Let us define p(y) the determinant of A as a function of y. Then the p(y) will be a polynomial with degree atmost n. Observe that the Graph G will have a unique perfect matching with exactly k red edges if and only if p(y) has a term of the form $\pm y^k$.
We will use Lagrangian interpolation to find the polynomial p(y). We will need n+1 points say $(y_0, y_1, \cdots, y_n)$ to find all the coefficients of p(y). We will find the value of p(y) at these n+1 points $p(y_0), p(y_1), \ldots, p(y_n)$ by computing the determinant of A at $y = y_i$ for each i. Then we can easily find if there is a perfect matching with k red edges by observing the coefficient of $y^k$.

## 16.2 Red-Blue Matching Problem : More than one Perfect Matching Exist

In the previous section we saw the Red-Blue Matching Problem. One assumption that we took was that there is a unique Red-Blue perfect matching with k red edges. In this section, we will prove show that graph G will contain a unique minimum weight perfect matching with Probability more than 0.5.

### 16.2.1 Isolation Lemma

Assign every edge $e \in E$ in the Graph G has a weight $w_e$ from the set $S = \{1, 2, 3, \cdots, 2m\}$ picked at random and independent of other edge.

We can show that

$$\text{P(there exist a unique minimum weight perfect matching)} \geq \frac{1}{2} \tag{16.1}$$

This is known as "isolation lemma" which was proved for sets by Mulmuley, Vazirani, and Vazirani in 1987.
**Proof:**
Definition 1: We will call an edge $e$ ***confused*** if $\exists$ two minimum weight matchings $M_1, M_2$ such that $e \in M_1$ and $e \notin M_2$.

If no edge is confused then G has a unique minimum weight matching since $M_1$ will be equal to $M_2$.
For a given edge $e$,

$$\text{P } (e \text{ is confused}) \leq \frac{1}{2m} \tag{16.2}$$

To prove 16.1 we need to prove 16.2 first. The proof of 16.2 is as follows:

Let us arbitrarily choose an edge $e$. We choose weights for all the edges in E randomly except this particular edge $e$. We define $M_1$ to be the minimum weight Perfect Matching in graph G **without the edge** $e$ and $M_2$ be a minimum weight Perfect Matching **without the vertices of edge** $e = (u, v)$ where $u \in U$ and $v \in V$. By addition of edge e to $M_2$, we get a perfect matching of n edges. Let $W_1$ and $W_2$ be the respective weights of the matching.

**Note** that we have constructed $M_1$ and $M_2$ in such a way that $M_1$ contain n edges whereas $M_2$ contain n-1 edges so that we can add the edge $e$ to $M_2$ to make it a minimum weight perfect matching.

Case 1. If $W_2 \geq W_1$, then e is never confused so $M_1$ and $M_2$ can't be a minimum weight matchings.
Case 2. If $W_2 < W_1$ , then if the weight of e is $W_1 - W_2$ then we will get e to be confused.

Since we are picking weights randomly and independent of other edges from the set $S = \{1, 2, \cdots, 2m\}$ so

$$\text{P}(e \text{ is confused }) \leq \frac{1}{2m}$$

Here we have proved 16.2.

$$\text{P(some edge is confused)} \leq \frac{m}{2m} \leq \frac{1}{2} \tag{16.3}$$

which is equivalent of saying

$$\text{P(min weight perfect matching is not unique)} \leq \frac{1}{2} \Rightarrow \text{P(min weight perfect matching is unique} \geq \frac{1}{2}$$

Here we have proved 16.1.
In 16.3 we have used Union Bound which states that *the probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events.*

### 16.2.2 Red-Blue Matching when the matching is not unique

Given edges $e \in E$ with weights $w_e$ chosen uniformly from set S $= \{1, 2, 3, \cdots, 2m\}$. We define matrix $A$ as

$$A(i, j) = \begin{cases} 2^{w_e} y, & (i, j) \in \text{E and is red} \\ 2^{w_e}, & (i, j) \in \text{E and is blue} \\ 0, & \text{otherwise} \end{cases}$$

Every red-blue perfect matching M with k red edges contributes

$$\pm y^k 2^{\sum_e w_e} = \pm y^k 2^{w(M)} \qquad \text{to det(A)}$$

where $\sum_e w_e = w(M)$ where w(M) is the sum of the weight of the edges of the perfect matching and

$$det(A) = p(y) = \sum_{n=0}^{n} a_i y^i$$

$$\text{If G contains a red blue matching, then } a_k \text{, the coefficient of } y^k \text{ is non-zero with Probability} \geq \frac{1}{2} \quad (16.4)$$

This can be proved using isolation lemma. We can see that the

$$\text{coefficient of } y^k = \pm 2^{w(M_1)} \pm 2^{w(M_2)} \pm \cdots \pm 2^{w(M_t)}$$

where $\pm w(M_1)$ is the sum of the minimum weight perfect matching edges and $\pm w(M_t)$ is the sum of the perfect matching edges weights other than the minimum weight.
Taking $2^{w(M_1)}$ out as common we get:

$$\text{coefficient of } y_k = 2^{w(M_1)}(\pm 1 \pm 2^{w(M_2)-w(M_1)} \pm \cdots \pm 2^{w(M_t)-w(M_1)})$$

Note that the coloured term in the above expression will always be even and hence the coefficient of $y_k$ will always be odd if the minimum weight perfect matching is unique (as we will have either +1 or -1. This will ensure that the coefficient of $y_k$ will be odd). As shown above in 16.1 that this probability is $\geq \frac{1}{2}$.
If the minimum weight perfect matching is not unique then $\pm 1$ terms will cancel each other.

We can repeat this multiple times to be sure that the graph G has red-blue matching or not.
Also note that no deterministic polynomial time algorithm exist for this red-blue matching problem and only randomised algorithm exist. Hence, there is active research going on in this field.

### 16.2.3 Bipartite Graph Matching vs Red-Blue Matching

In Bipartite Matching we picked $x_{i,j}$ from the field $\mathbb{F}$ randomly and with Probability $\geq 1 - \frac{1}{n^2}$ we have det(A) $\neq 0$ , iff G has a perfect matching.
In Red Blue Matching we picked $w_{i,j}$ from the set S = $\{1, 2, 3, \cdots, 2m\}$ randomly and independently. With Probability $\geq \frac{1}{2}$ we can say that the minimum weight perfect matching is unique and coefficient of $y^k \neq 0$

**Homework:** Using the ideas for existence of red-blue perfect matching, give an algorithm to find the red-blue perfect matching if it exists like we did for Bipartite Matching.

### 16.2.4 Bipartite Graph Matching using Polynomial Number of Processors

Definition 2: **Parallel Random Access Machine (PRAM)** : Multiple processors are connected in parallel with a single shared memory unit. In this, n number of processors can perform independent operations on n number of data in a particular unit of time.

Definition 3: In complexity theory, the class NC (for "Nick's Class") is the set of decision problems decidable in polylogarithmic time on a parallel computer with a polynomial number of processors. In other words, a problem is in NC if there exist constants c and k such that it can be solved in time $O(log^c n)$ using $O(n^k)$ parallel processors.

Bipartite Graph Matching can be parallelised because this require only finding the determinant of a matrix and this can be done efficiently if we have large number (but polynomial) of parallel processors.

- Matrix multiplication can be done in $NC^1$ ($NC^k = O(log^k(n))$). Here n denotes size of the matrix $n \times n$.
- Also note that Addition of n numbers can be done in $NC^1$ time and we will require n parallel processor to do it.
- Determinant can be computed in $NC^2$ time.
- Hence Bipartite Matching can be solved in $RNC^2$ time where RNC stands for randomised NC. This is because we need to compute determinants only.

**Note** : This section 16.2.4 was discussed in the class in brief.