10. Bipartite Graph Perfect Matching (PIT & MVV)

There are already many polynomial sequential algorithms that can be used for matching, but we can make use of Polynomial Identity Testing to come up with a randomized, parallelizable algorithm.

Edmond's Matrix

Given a graph $G = (L, R, E)$, we define a matrix X_G with entries defined as follows.

$$X_{ij} = \begin{cases} x_{ij} & \text{if exists an edge } (i, j), i \in L, j \in R \\ 0 & \text{otherwise} \end{cases}$$

This matrix is known as Edmond's matrix. x_{ij} is not some constant, they are distinct variables (hence there are n^2 variables possible)

We define the determinant of the matrix as

$$\text{Det}(X) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \prod_{i \in [n]} x_{i\sigma(i)}$$

Here, σ is a permutation that belongs to the set S_n, the set of all permutations of length n.

When is $\text{Det}(X) \neq 0$?

If the determinant of the matrix is 0, then we know that there exists some permutation $\sigma \in S_n$ such that each $x_{i\sigma(i)} \neq 0$. Remember that no two terms can cancel each other out since all the variables are distinct, which is why this condition is sufficient.

Theorem - Graph G has a perfect matching $\iff \text{Det}(X_G) \neq 0$

G has PM $\implies \text{Det}(X_G) \neq 0$

Suppose G has a perfect matching, then we essentially have a bijection mapping every $l_i \in L$ to a unique $r_i \in R$, and bijections can be seen as a permutation, hence we know that

$$\prod_{i=1}^{n} X_{i\sigma(i)} = \prod_{i=1}^{n} x_{i\sigma(i)}$$

is a non-zero monomial in the polynomial, and since we know that the monomials can't cancel each other out, we have a non-zero determinant.

G has PM $\Longleftarrow \mathrm{Det}(X_G \neq 0)$

Suppose $\mathrm{Det}(X_G) \neq 0$, this means that there exists at least one permutation σ such that all terms $X_{i\sigma(i)}$ are not 0. This indicates that each vertex $l_i \in L$ got matched to vvertex $r_{\sigma(i)} \in R$. Since σ is a bijection, we know that matching is perfect.

Using this theorem, we can come up with a sequential algorithm

Sequential Algorithm for Bipartite Matching

1. $M \leftarrow \{\}$
2. For each $e = (i, j) \in E$,
    1. $G' = (L \backslash \{i\}, R \backslash \{j\}, E \backslash \{e\})$
    2. If $\mathrm{Det}(X_G) \neq 0$
        1. $G \leftarrow G'$
        2. $M \leftarrow M \cup \{e\}$
3. Return M

The running time of this algorithm is m into the time complexity for computing the determinant, $O(mn^3)$

Note that depending upon the order of edges we take, we could get different perfect matchings

Theorem - If A is an $n \times n$ matrix, then $\mathrm{Det}(A), \mathrm{adj}(A), A^{-1}$ can all be computed in $O(\log^2 n)$ time over $O(n^{3.5})$ processors.

Using this theorem, we devise a parallel randomized algorithm for perfect matching.

Parallel Algorithm for Perfect Matching

Theorem - There is a randomized algorithm that finds a perfect matching in $O(\log^2 n)$ time using $O(mn^{3.5})$ processors with probability $\frac{1}{2}$.

Key Idea - We randomly assign edge weights in the graph, and show that if a perfect matching exists, there is a unique minimum perfect matching.

Isolation Lemma

Let S be a finite subset of $\mathbb{R}$. Let $T_1, T_2, \ldots, T_k$ be some subsets of $\{1, 2, \ldots, m\}$. For each $i \in \{1, 2, \ldots, m\}$ assign a weight uniformly and randomly from S. We define the weight of T_j as

$$\mathrm{wt}(T_j) = \sum_{i \in T_j} \mathrm{wt}(i)$$

The *Isolation Lemma* states that

$$\mathbb{P}[\exists j \text{ s.t } T_j \text{ has unique min wt}] \geq 1 - \frac{m}{|S|}$$

Proof

Suppose $\exists j, j'$ such that T_j and $T_{j'}$ attain the minimum weight

This implies that

$$\sum_{i \in T_j} \mathrm{wt}(i) = \sum_{i' \in T_{j'}} \mathrm{wt}(i')$$

Let $e \in T_j \backslash T_{j'}$, we can take $\mathrm{wt}(e)$ out to write the above equation as

$$\mathrm{wt}(e) + \sum_{i \neq e, i \in T_j} \mathrm{wt}(i) = \sum_{i' \in T_{j'}} \mathrm{wt}(i)$$

To calculate the probability of having two non minimum matchings when using randomized values, we can use the **Principle of Deferred Decisions**.
Since we know that no matter what the sum of weights are in $T_j \backslash \{e\}, T_{j'}$, there is at most one value of $\mathrm{wt}(e)$ that can satisfy the equation, hence we can say that the probability is upper bounded by $\frac{1}{|S|}$.

We define a *bad case* as 2 sets $T_j, T_{j'}$ that have the same weight.

We define an event E_i such that

$$\min_j\{\text{wt}(T_j)|i \in T_j\} = \min_{j'}\{\text{wt}(T_{j'})|i \notin T_{j'}\}$$

If E_i occurs, it implies there exists two unique matchings that achieve the minimum.

If $\bigcap_{i=1}^m \bar{E}_i$ occurs, then $\exists$ a unique minimum wt of T_j. (Sufficient, but not necessary condition).

$$\mathbb{P}\left[\bigcap_{i=1}^m \bar{E}_i\right] = 1 - \mathbb{P}\left[\bigcup_{i=1}^m E_i\right] \geq 1 - \sum_{i=1}^m \mathbb{P}[E_i] \geq 1 - \frac{m}{|S|}$$

Hence proved.

Different Matrix Definition

Initially, we assign each edge a weight i.u.a.r from S.
Then, we define a new matrix W as

$$W_{ij} = \begin{cases} 2^{\text{wt}(i,j)} & \text{if } i \in L, j \in R, (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Now, we'll get the determinant of this matrix as follows

$$\begin{aligned} \text{Det}(W) &= \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n W_{i\sigma(i)} \\ &= \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{\substack{i=1 \\ (i,\sigma(i)) \in E \\ \forall i}}^n 2^{\text{wt}(i,\sigma(i))} \\ &= \sum_{\substack{\sigma \in S_n \\ \sigma \in M}} \text{sgn}(\sigma) 2^{\text{wt}(M_\sigma)} \end{aligned}$$

Where M is the set of all perfect matchings, and M_σ is the matching resulting from σ.

If r is the minimum weight matching, then we know that 2^r divides $\text{Det}(W)$.

$$\begin{aligned} \text{Det}(W) &= \text{sgn}(M_1)2^{\text{wt}(M_1)} + \text{sgn}(M_2)2^{\text{wt}(M_2)} + \cdots + \text{sgn}(M_t)2^{\text{wt}(M_t)} \\ &= 2^{\text{wt}(M_0)}[\text{sgn}(M_0) + \text{rest}] \end{aligned}$$

Where M_0 is the minimum weight matching

Lemma to process edges independently

We denote the matrix where row i and column j is removed as $W^{(i,j)}$.
Let M_0 be the unique minimum weight perfect matching in G, let $r = \mathrm{wt}(M_0)$, then

$$(i,j) \in M_0 \iff \frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt(i, j)}}}{2^r} \text{ is odd}$$

The significance of this lemma is that if we know r, we can figure out if some edge $e \in E$ belongs to the matching independently of the other edges, allowing us to parallelize the processing of edges.

$(i,j) \in M_0 \implies \frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt(i, j)}}}{2^r}$ is odd

First, we show that if M_0 is the unique minimum weight perfect matching, then $\frac{\mathrm{Det}(W)}{2^{\mathrm{wt}(M_0)}}$ is odd.
We've seen earlier that we can write the determinant as

$$\mathrm{Det}(W) = 2^{\mathrm{wt}(M_0)}[\mathrm{sgn}(M_0) + \mathrm{rest}]$$

we know that $\mathrm{sgn}(M_0)$ is either -1 or 1, both of which are odd. We also know that the rest of the terms are all even as they all have the coefficient $2^{\mathrm{wt}(M_i) - \mathrm{wt}(M_0)}$ (or is just 0 if $\sigma \notin M$), and unique minimum weight guarantees $\mathrm{wt}(M_i) - \mathrm{wt}(M_0) > 0$.
Therefore $[\mathrm{sgn}(M_0) + \mathrm{rest}]$ is odd, hence

$$\frac{\mathrm{Det}(W)}{2^{\mathrm{wt}(M_0)}}$$

is odd.

Now, if we have $(i,j) \in M_0$, we claim that $M_0 \backslash \{(i,j)\}$ forms a unique minimum weight perfect matching for the graph $G' = (L \backslash \{i\}, R \backslash \{j\}, E')$.
This can be proven using proof by contradiction -

1. If this isn't the minimum weight matching, say there is some M' perfect matching for subgraph G' such that it's weight is less than that of $M_0 \backslash \{(i,j)\}$, we can get a better minimum perfect matching than M_0 by using $M' \cup \{(i,j)\}$, contradicting that M_0 is the minimum weight PM.
2. If there doesn't exist a unique minimum perfect matching, say there's M'' such that it has the same weight as $M_0 \backslash \{(i,j)\}$, we can then construct another

minimum matching for graph G, using $M'' \cup \{(i,j)\}$, contradicting the assumption that M_0 is a unique minimum PM.

If the minimum weight matching of G is 2^r, then for G' it'll be $2^{r-\mathrm{wt}(i,j)}$.
Since we know there exists a unique minimum PM for G', we can use the earlier claim to say that

$$\frac{\mathrm{Det}(W^{(i,j)})}{2^{r-\mathrm{wt}(i,j)}} = \frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt}(i,j)}}{2^r}$$

is odd as well.

$$(i,j) \in M_0 \Longleftarrow \frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt(i,\,j)}}}{2^r} \text{ is odd}$$

We can instead show that if $(i,j) \notin M_0$, then $\frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt(i,\,j)}}}{2^r}$ is even.

$$\mathrm{Det}(W) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} W_{i\sigma(i)}$$

We now break this summation into two terms, the permutations that contain $\sigma(i) = j$ and those where $\sigma(i) \neq j$ (Signify if that edge is picked or not).

$$\mathrm{Det}(W) = \left(\sum_{\sigma'} \mathrm{sgn}(\sigma') \prod_{i' \neq i} 2^{\mathrm{wt}(i',\sigma(i'))}\right) 2^{\mathrm{wt}(i,j)} + \left(\sum_{\substack{\sigma \\ \sigma(i) \neq j}} \mathrm{sgn}(\sigma) \prod 2^{\mathrm{wt}(i,\sigma(i))}\right)$$

Here, σ' signifies the permutations $[n] \backslash \{i\} \to [n] \backslash \{j\}$.

The first term essentially corresponds to $\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt}(i,j)}$.

Since $(i,j) \notin M_0$, we know that M_0 will be found in the second term above.
Hence, we can rewrite the second term as

$$2^r[\mathrm{sgn}(M_0) + \sum_{\substack{\sigma' \\ \sigma'(i) \neq j \\ \sigma' \neq M_0}} \mathrm{sgn}(\sigma') 2^{\mathrm{wt}(\sigma') - r}]$$

Hence, we can rewrite the formulation of $\mathrm{Det}(W)$ along with dividing all the terms to get

$$\frac{\mathrm{Det}(W)}{2^r} = \frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt}(i,j)}}{2^r} + [\pm 1 + \sum_{\substack{\sigma' \\ \sigma'(i) \neq j \\ \sigma' \neq M_0}} \mathrm{sgn}(\sigma') 2^{\mathrm{wt}(\sigma') - r}]$$

We know that the LHS and the second term in the RHS are both odd. Therefore, the first term must be even for the equality to be possible.

Hence, if $(i, j) \notin M_0$, then

$$\frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt}(i,j)}}{2^r}$$

is even.

We then use this lemma to make the parallel algorithm

Algorithm

1. Compute $\mathrm{Det}(W)$. The determinant is bounded by $n! \times 2^{\max\{S\} \cdot n} \approx 2^{n \log n \pm O(n)} \times 2^{\max\{S\} \cdot n}$. Hence the bit complexity is polynomial as $|S| = O(m)$.
2. We then binary search on r, to find the largest value such that 2^r divides $\mathrm{Det}(W)$.
3. For all edges $(i, j) \in E$, we parallely and independently
    1. Compute $\frac{\mathrm{Det}(W^{(i,j)})2^{\mathrm{wt}(i,j)}}{2^r}$
    2. If the result is odd, then $(i, j) \in M_0$, else not.

We know that each edge requires $O(n^{3.5})$ processors to compute the determinant in $O(\log^2 n)$ time, hence we require $O(mn^{3.5})$ processors to compute in $O(\log^2 n)$ time.