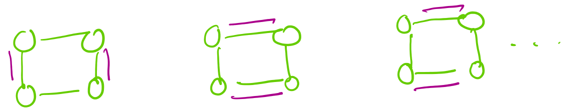
COPT Notes

20 April 2024 22:33

Given a graph, a graph may have exponentially many maximum matching

Example:



n components

there are 4n vertices and edges

but 2^n perfect (maximum) matching

Problem: Given a graph, return a maximum matching if exists.

Simpler problem: Given a Bipartite graph $G(U, V, E)$ and a guarantee that a perfect matching exists, find the perfect matching

↳ So how to do it.

We will start by assigning random weights to the edges of the graph chosen uniformly from $[1, 2m]$

With probability more than $\frac{1}{2}$ (by isolating lemma), there exists unique minimum weight perfect matching.

We will force our parallel algorithm to take this matching.

Now, $U = \{u_1, \ldots u_n\}$

$V = \{v_1, \ldots v_n\}$

and D be $n \times n$ matrix st

$$D_{ij} = 1 \quad \text{if } (u_i, v_j) \in E$$
$$= 0 \quad \text{otherwise}$$

B be the matrix st

$$B_{ij} = 2^{w_{ij}} \quad \text{if } D_{ij} = 1$$

$$B_{ij} = 0 \quad \text{otherwise.}$$

<u>Claim 1</u>: Suppose the minimum weight perfect matching (M) is unique with weight w. then $|B| \neq 0$,

$2^{w} \mid B$ and $2^{w+1} \nmid B$.

<u>Proof</u>:- <u>we know</u>,

$$\det(A) = \sum_{\sigma \in S_n} (-1)^{\operatorname{sgn}(\sigma)} \prod_{i=1}^{n} A_{i,\sigma(i)},$$

$$|B| = \sum_{\sigma \in S_n} \underbrace{\operatorname{Sgn}(\sigma)}_{\text{sign}} * \underbrace{\text{value}(\sigma)}_{\text{value}}$$

now, as every permutation is a potential perfect matching, the determinant

will be [illegible] value * Sgn. As

perfect matching, the determinant

will be sum of value * Sgn. As

we assigned 0 or $2^{w_{ij}}$ to the B_{ij}.

and as M has weight w (which is

unique), $|B| = \pm 2^{w} \pm 2^{w+1} \cdot k + 0$

minimum weight perfect matching

Perfect matching

not a Perfect matching

Clearly. $|B| \neq 0$.

$2^{w} | B$

$2^{w+1} \nmid B$.

Claim 2: The edge (u_i, v_j) belongs to M iff

$$\frac{|B_{ij}| \, 2^{w_{ij}}}{2^{w}} \text{ is odd.}$$

First of all, $|B_{ij}|$ mean minor of B_{ij} and

that basically indicates we are taking all the

permutations where i^{th} vertex is matched to

j^{th} vertex.

$$|B_{ij}| 2^{w_{ij}} = \sum_{\sigma : \sigma(i) = j} \mathrm{Sgn}(\sigma) \cdot \mathrm{Value}\ \sigma$$

w is the weight of the unique minimum

weight perfect matching.

Now. $\frac{|B_{ij}| 2^{w_{ij}}}{2^{w}}$ is odd

... ith vertex to be matched with jth

$\Rightarrow$ If we fix the ith vertex to be matched with jth vertex, and this edge actually is a part of minimum weight perfect matching then, all the perfect matching having $\sigma(i)=j$ will have weight either 0 or $\pm K \cdot 2^{w+1}$ (as $\exists$ an unique minimum weight matching)

$$\therefore \frac{|B_{ij}|\, 2^{w_{ij}}}{2^w} = \frac{0 \pm 2^w \pm K 2^{w+1}}{2^w}$$

$$= \pm 1 \pm 2K \quad \text{odd}$$

Now if w_{ij} is not a part of minimum weight perfect matching, then all the perfect matchings having $\sigma(i)=j$ have weight $\geq 2^{w+1}$ or 0

So, $$\frac{|B_{ij}|\, 2^{w_{ij}}}{2^w} = \frac{\pm K \cdot 2^{w+1}}{2^w}$$

$$= \pm 2K \quad \text{even}$$

Now, the algorithm to find M will be straightforward :-

1> a graph is given, add weight to each edge uniformly and independently from [1, 2m]

↳ By isolating lemma, with probability $\geq \frac{1}{2}$ we have a unique minimum weight perfect matching.

2> Compute $|B|$ and obtain w by observing 2^w is the highest power of 2 that divides $|B|$

↳ this follows from claim 1

↳ w is the weight of that minimum weight perfect matching

3> Compute $adj(B)$, $(i,j)^{th}$ entry of it will be $|B_{ij}|$

↳ This step calculates adjoint which is computationally as hard as calculating determinant.

↳ This step can be parallelized as matrix inversion has a parallel algorithm.

4> For each edge, compute $\frac{|B_{ij}| 2^{w_{ij}}}{2^w}$ in parallel.

if it is odd, add the edge to the collection, else don't add it