Matchings and algebra

Peter Hajnal

Bolyai Institute, University of Szeged, Hungary

2023 fall

The problem

The problem

Given G simple bipartite graph, $|L| = |U| = n$.

The problem

The problem

Given G simple bipartite graph, $|L| = |U| = n$.

Is there a perfect matching in G?

The problem

The problem

Given G simple bipartite graph, $|L| = |U| = n$.

Is there a perfect matching in G?

The assumption of the bipartite property is not necessary, but it makes our life easier.

Coding with matrices

Definition

Let G be a simple graph.

Coding with matrices

Definition

Let G be a simple graph.

The adjacency matrix of G, denoted as A_G, is a matrix with rows and columns identified with vertices, furthermore

$$(A_G)_u v = \begin{cases} 1, & u \text{ and } v \text{ are adjacent,} \\ 0, & u \text{ and } v \text{ are not adjacent,} \end{cases}$$

Coding with matrices

Definition

Let G be a simple graph.

The adjacency matrix of G, denoted as A_G, is a matrix with rows and columns identified with vertices, furthermore

$$(A_G)_u v = \begin{cases} 1, & u \text{ and } v \text{ are adjacent,} \\ 0, & u \text{ and } v \text{ are not adjacent,} \end{cases}$$

Definition

Let G be a bipartite graph with color classes $L\dot{\cup}U$.

Coding with matrices

Definition

Let G be a simple graph.

The adjacency matrix of G, denoted as A_G, is a matrix with rows and columns identified with vertices, furthermore

$$(A_G)_u v = \begin{cases} 1, & u \text{ and } v \text{ are adjacent,} \\ 0, & u \text{ and } v \text{ are not adjacent,} \end{cases}$$

Definition

Let G be a bipartite graph with color classes $L \dot\cup U$.

The bipartite adjacency matrix of G is a matrix $B_G \in \mathbb{R}^{L \times U}$:

$$(B_G)_{\ell,u} = \begin{cases} 1, & \ell \text{ and } u \text{ are adjacent,} \\ 0, & \ell \text{ and } u \text{ are not adjacent,} \end{cases}$$

The relation of matrices

The relation of matrices

In a bipartite graph there is no adjacency between lower vertices, and no adjacency between upper vertices.

The relation of matrices

In a bipartite graph there is no adjacency between lower vertices, and no adjacency between upper vertices.

In A_G this property means a big block of 0's.

The relation of matrices

In a bipartite graph there is no adjacency between lower vertices, and no adjacency between upper vertices.

In A_G this property means a big block of 0's.

$$A_G = \begin{pmatrix} 0 & B_G \\ B_G^T & 0 \end{pmatrix}$$

The relation of matrices

In a bipartite graph there is no adjacency between lower vertices, and no adjacency between upper vertices.

In A_G this property means a big block of 0's.

$$A_G = \begin{pmatrix} 0 & B_G \\ B_G^T & 0 \end{pmatrix}$$

B_G is a condensed version of the adjacency matrix.

The simple bipartite graph G and B_G

The simple bipartite graph G and B_G

The B_G matrix is a code of the simple bipartite graph G.

The simple bipartite graph G and B_G

The B_G matrix is a code of the simple bipartite graph G. There is a dictionary between the linear algebraic language and graph theoretical language.

The simple bipartite graph G and B_G

The B_G matrix is a code of the simple bipartite graph G. There is a dictionary between the linear algebraic language and graph theoretical language.

$$\begin{aligned}
\text{possible edges of } G &\equiv \text{positions of } B_G = A \times F \\
E(G) &\equiv \text{the 1's of } B_G \\
|A| = |F| &\equiv B_G \text{ is a square matrix} \\
\text{M is a matching} &\equiv \text{1's s.t. there is maximum one 1} \\
&\quad\ \text{in each row and column} \\
\text{M is a perfect matching} &\equiv n \text{ 1's s.t. there is exactly one 1} \\
&\quad\ \text{in each row and column} \\
&\equiv n \text{ 1's s.t. their product is a} \\
&\quad\ \text{term of the determinant}
\end{aligned}$$

The determinant

Reminder

$$\det M = \sum_{\pi \in S_n} \operatorname{sign}(\pi) \prod_{i=1}^{n} M_{i\pi(i)}.$$

The determinant

Reminder

$$\det M = \sum_{\pi \in S_n} \operatorname{sign}(\pi) \prod_{i=1}^{n} M_{i\pi(i)}.$$

Corollary

If $\det B_G \neq 0$ then the above expansion of $\det B_G$ has a non-zero term.

The determinant

Reminder

$$\det M = \sum_{\pi \in S_n} \mathrm{sign}(\pi) \prod_{i=1}^{n} M_{i\pi(i)}.$$

Corollary

If $\det B_G \neq 0$ then the above expansion of $\det B_G$ has a non-zero term.

This is equivalent to the fact that there is a perfect matching in G.

The determinant

Reminder

$$\det M = \sum_{\pi \in S_n} \mathrm{sign}(\pi) \prod_{i=1}^{n} M_{i\pi(i)}.$$

Corollary

If $\det B_G \neq 0$ then the above expansion of $\det B_G$ has a non-zero term.

This is equivalent to the fact that there is a perfect matching in G.

The reverse direction of the Corollary is not true.

The permanent

The permanent

Definition

The permanent of $M \in \mathbb{R}^{n\times n}$

$$\operatorname{per} M_{n\times n} = \sum_{\pi\in S_n} \prod_{i=1}^{n} M_{i\pi(i)}$$

The permanent

Definition

The permanent of $M \in \mathbb{R}^{n\times n}$

$$\operatorname{per} M_{n\times n} = \sum_{\pi\in S_n} \prod_{i=1}^{n} M_{i\pi(i)}$$

Observation

(i) per $B_G \neq 0$ iff there is a perfect matching in G.

(ii) per B_G is the number of perfect matchings in G.

The permanent

Definition

The permanent of $M \in \mathbb{R}^{n\times n}$

$$\text{per } M_{n\times n} = \sum_{\pi\in S_n} \prod_{i=1}^{n} M_{i\pi(i)}$$

Observation

(i) per $B_G \neq 0$ iff there is a perfect matching in G.

(ii) per B_G is the number of perfect matchings in G.

Unfortunately computing per B_G is $\#P$-hard.

The permanent

Definition

The permanent of $M \in \mathbb{R}^{n\times n}$

$$\text{per } M_{n\times n} = \sum_{\pi \in S_n} \prod_{i=1}^{n} M_{i\pi(i)}$$

Observation

(i) per $B_G \neq 0$ iff there is a perfect matching in G.

(ii) per B_G is the number of perfect matchings in G.

Unfortunately computing per B_G is $\#P$-hard.

Reminder

There are efficient algorithms to compute the determinant of a square matrix.

Polynomials

Polynomials

Definition

$X_G \in \mathbb{R}[x_e : e \in E(G)]^{n\times n}$: $\forall e \in E(G)$ we substitute the 1 ofB_G, corresponding to e with x_e.

Polynomials

Definition

$X_G \in \mathbb{R}[x_e : e \in E(G)]^{n\times n}$: $\forall e \in E(G)$ we substitute the 1 ofB_G, corresponding to e with x_e.

Theorem

$det(X_G)$ is a non-zero polynomial if and only if there is a perfect matching in G.

Polynomials

Definition

$X_G \in \mathbb{R}\left[x_e : e \in E(G)\right]^{n\times n}$: $\forall e \in E(G)$ we substitute the 1 ofB_G, corresponding to e with x_e.

Theorem

$det(X_G)$ is a non-zero polynomial if and only if there is a perfect matching in G.

Observation

(i) The number of perfect matchings in G is the number of monomials of $\det(X_G)$.

(ii) It is possible that to write down $\det(X_G)$ takes too long. But evaluate $\det(X_G)|_{x_e=\alpha_e}$ is easy.

A randomized algorithm

A randomized algorithm

Randomized algorithm

A randomized algorithm

Randomized algorithm

Random substitution: For each edge e generate $r_e \in \{1, \ldots, N\}$, a random value (uniform distribution, independence).

A randomized algorithm

Randomized algorithm

Random substitution: For each edge e generate $r_e \in \{1, \ldots, N\}$, a random value (uniform distribution, independence).

Calculating DET: Compute $det(X_G)|_{x_e=r_e}$.

A randomized algorithm

Randomized algorithm

Random substitution: For each edge e generate $r_e \in \{1, \dots, N\}$, a random value (uniform distribution, independence).

Calculating DET: Compute $det(X_G)|_{x_e=r_e}$.

Evaluation:
If the value is non-0, then the output is „There is perfect matching”.

A randomized algorithm

Randomized algorithm

Random substitution: For each edge e generate $r_e \in \{1, \ldots, N\}$, a random value (uniform distribution, independence).

Calculating DET: Compute $det(X_G)|_{x_e=r_e}$.

Evaluation:
If the value is non-0, then the output is „There is perfect matching”.
If the value is 0, then the output is „Probably there is no perfect matching”.

Error

Error

Our algorithm can make errors. But how?

Error

Our algorithm can make errors. But how?

- „There is a perfect matching": the answer is guaranteed to be correct.

Error

Our algorithm can make errors. But how?

- „There is a perfect matching": the answer is guaranteed to be correct.
- „Probably there is no perfect matching":

Error

Our algorithm can make errors. But how?

- „There is a perfect matching": the answer is guaranteed to be correct.
- „Probably there is no perfect matching":
  - if $det(X_G)$ is the 0 polynomial, then the answer is correct;

Error

Our algorithm can make errors. But how?

- „There is a perfect matching": the answer is guaranteed to be correct.
- „Probably there is no perfect matching":
  - if $det(X_G)$ is the 0 polynomial, then the answer is correct;
  - if $det(X_G)$ is a non-0 polynomial, and we generate an unfortunate r_e's, one of the roots of $det(X_G)$: the algorithm fails.

Error

Our algorithm can make errors. But how?

- „There is a perfect matching": the answer is guaranteed to be correct.
- „Probably there is no perfect matching":
  - if $det(X_G)$ is the 0 polynomial, then the answer is correct;
  - if $det(X_G)$ is a non-0 polynomial, and we generate an unfortunate r_e's, one of the roots of $det(X_G)$: the algorithm fails.

Our goal is to reduce the probability of error.

Schwartz' lemma

Schwartz' lemma

Theorem (Schwartz-lemma)

Let $p(x_1,\ldots,x_k)\in\mathbb{R}[x_1,\ldots,x_k]$ be a non-zero polynomial and let $r_i\in\{1,\ldots,N\}$ be uniform independent random variables, $(1\le i\le k)$. Then

$$\mathbb{P}(p(r_1,\ldots,r_k)=0)\le\frac{\deg p}{N}$$

Proof by induction on k: $k = 1$

Proof by induction on k: $k = 1$

For $k = 1$ we have $p \in \mathbb{R}[x]$.

Proof by induction on k: $k = 1$

For $k = 1$ we have $p \in \mathbb{R}[x]$.

We know that $|\{r \in \mathbb{R} : p(r) = 0\}| \leq \deg p$,

Proof by induction on k: $k = 1$

For $k = 1$ we have $p \in \mathbb{R}[x]$.

We know that $|\{r \in \mathbb{R} : p(r) = 0\}| \leq \deg p$, hence the probability that the value of r is a root of p is at most $\frac{\deg p}{N}$.

Proof by induction on k: The induction step

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\ldots,x_k) = p_\alpha(x_1,\ldots,x_{k-1})\cdot x_k^\alpha + p_{\alpha-1}(x_1,\ldots,x_{k-1})\cdot x_k^{\alpha-1} + \cdots + p_0(x_1,\ldots,x_{k-1}),$$

where $p_\alpha(x_1,\ldots,x_{k-1})$ is a non-0 polynomial.

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\dots,x_k) = p_\alpha(x_1,\dots,x_{k-1})\cdot x_k^\alpha + p_{\alpha-1}(x_1,\dots,x_{k-1})\cdot x_k^{\alpha-1} + \dots + p_0(x_1,\dots,x_{k-1}),$$

where $p_\alpha(x_1,\dots,x_{k-1})$ is a non-0 polynomial.

It is obvious that $\deg p \geq \deg p_\alpha + \alpha$.

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\dots,x_k)=p_\alpha(x_1,\dots,x_{k-1})\cdot x_k^\alpha+p_{\alpha-1}(x_1,\dots,x_{k-1})\cdot x_k^{\alpha-1}+\cdots+p_0(x_1,\dots,x_{k-1}),$$

where $p_\alpha(x_1,\dots,x_{k-1})$ is a non-0 polynomial.

It is obvious that $\deg p\geq \deg p_\alpha+\alpha$.

Let $R_k=\{(r_1,\dots,r_k):p(r_1,\dots,r_k)=0\}$, the set of choices for $(r_1,\dots,r_k)$, that is a root of p.

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\dots,x_k) = p_\alpha(x_1,\dots,x_{k-1})\cdot x_k^\alpha + p_{\alpha-1}(x_1,\dots,x_{k-1})\cdot x_k^{\alpha-1} + \cdots + p_0(x_1,\dots,x_{k-1}),$$

where $p_\alpha(x_1,\dots,x_{k-1})$ is a non-0 polynomial.

It is obvious that $\deg p \geq \deg p_\alpha + \alpha$.

Let $R_k = \{(r_1,\dots,r_k) : p(r_1,\dots,r_k) = 0\}$, the set of choices for $(r_1,\dots,r_k)$, that is a root of p. We need to bound $\mathbb{P}(R_k)$.

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\ldots,x_k) = p_\alpha(x_1,\ldots,x_{k-1})\cdot x_k^\alpha + p_{\alpha-1}(x_1,\ldots,x_{k-1})\cdot x_k^{\alpha-1} + \cdots + p_0(x_1,\ldots,x_{k-1}),$$

where $p_\alpha(x_1,\ldots,x_{k-1})$ is a non-0 polynomial.

It is obvious that $\deg p \geq \deg p_\alpha + \alpha$.

Let $R_k = \{(r_1,\ldots,r_k) : p(r_1,\ldots,r_k)=0\}$, the set of choices for $(r_1,\ldots,r_k)$, that is a root of p. We need to bound $\mathbb{P}(R_k)$.

Let $R_{k-1} = \{(r_1,\ldots,r_k) : p_\alpha(r_1,\ldots,r_{k-1})=0\}$.

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\dots,x_k) = p_\alpha(x_1,\dots,x_{k-1})\cdot x_k^\alpha + p_{\alpha-1}(x_1,\dots,x_{k-1})\cdot x_k^{\alpha-1} + \cdots + p_0(x_1,\dots,x_{k-1}),$$

where $p_\alpha(x_1,\dots,x_{k-1})$ is a non-0 polynomial.

It is obvious that $\deg p \ge \deg p_\alpha + \alpha$.

Let $R_k = \{(r_1,\dots,r_k) : p(r_1,\dots,r_k) = 0\}$, the set of choices for $(r_1,\dots,r_k)$, that is a root of p. We need to bound $\mathbb{P}(R_k)$.

Let $R_{k-1} = \{(r_1,\dots,r_k) : p_\alpha(r_1,\dots,r_{k-1}) = 0\}$.

Let
$Q = \{(r_1,\dots,r_k) : (r_1,\dots,r_{k-1}) \notin R_{k-1}, but\ (r_1,\dots,r_k) \in R_k\}$.

Proof by induction on k: The induction step

Assume that we know the claim for $k-1$ variables. Write p as:

$$p(x_1,\ldots,x_k) = p_\alpha(x_1,\ldots,x_{k-1})\cdot x_k^\alpha + p_{\alpha-1}(x_1,\ldots,x_{k-1})\cdot x_k^{\alpha-1} + \cdots + p_0(x_1,\ldots,x_{k-1}),$$

where $p_\alpha(x_1,\ldots,x_{k-1})$ is a non-0 polynomial.

It is obvious that $\deg p \geq \deg p_\alpha + \alpha$.

Let $R_k = \{(r_1,\ldots,r_k) : p(r_1,\ldots,r_k) = 0\}$, the set of choices for $(r_1,\ldots,r_k)$, that is a root of p. We need to bound $\mathbb{P}(R_k)$.

Let $R_{k-1} = \{(r_1,\ldots,r_k) : p_\alpha(r_1,\ldots,r_{k-1}) = 0\}$.

Let
$Q = \{(r_1,\ldots,r_k) : (r_1,\ldots,r_{k-1}) \notin R_{k-1}, but\ (r_1,\ldots,r_k) \in R_k\}$.

Easy to see that $R_k \subseteq R_{k-1} \cup Q$.

Proof by induction on k: The induction step (cont'd)

Proof by induction on k: The induction step (cont'd)

Based on the induction hypothesis we can bound the probability of R_{k-1}.

Proof by induction on k: The induction step (cont'd)

Based on the induction hypothesis we can bound the probability of R_{k-1}.

Based on the analysis of case $k = 1$ we can estimate the probability of Q.

Proof by induction on k: The induction step (cont'd)

Based on the induction hypothesis we can bound the probability of R_{k-1}.

Based on the analysis of case $k = 1$ we can estimate the probability of Q.

After summing our bounds we obtain

$$\mathbb{P}(R_k) \leq \mathbb{P}(R_{k-1}) + \mathbb{P}(Q) \leq \frac{\deg\ p_\alpha}{N} + \frac{\alpha}{N} \leq \frac{\deg\ p}{N}.$$

Proof by induction on k: The induction step (cont'd)

Based on the induction hypothesis we can bound the probability of R_{k-1}.

Based on the analysis of case $k = 1$ we can estimate the probability of Q.

After summing our bounds we obtain

$$\mathbb{P}(R_k) \leq \mathbb{P}(R_{k-1}) + \mathbb{P}(Q) \leq \frac{\deg\ p_\alpha}{N} + \frac{\alpha}{N} \leq \frac{\deg\ p}{N}.$$

This proves the claim of the Theorem.

Methods to reduce the error

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

We can reduce the error probability.

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

We can reduce the error probability.

(I) Increase the value of N.

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

We can reduce the error probability.

(I) Increase the value of N.

(II) Independently repeat the algorithm M times ($N = 2n$).

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

We can reduce the error probability.

(I) Increase the value of N.

(II) Independently repeat the algorithm M times ($N = 2n$). As soon as one determinant computation gives a non-0 value stop and announce "The graph has a perfect matching".

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

We can reduce the error probability.

(I) Increase the value of N.

(II) Independently repeat the algorithm M times ($N = 2n$). As soon as one determinant computation gives a non-0 value stop and announce "The graph has a perfect matching". If all M execution evaluate a 0 determinant, then announce "Probably the graph has NO perfect matching".

Methods to reduce the error

We can apply Schwartz' Lemma for $p = \det(X_G)$ ($\deg p = n(= |A| = |F|)$) we get that in the case of the choice $N = 2n$ the probability of error is at most $\frac{1}{2}$.

We can reduce the error probability.

(I) Increase the value of N.

(II) Independently repeat the algorithm M times ($N = 2n$). As soon as one determinant computation gives a non-0 value stop and announce "The graph has a perfect matching". If all M execution evaluate a 0 determinant, then announce "Probably the graph has NO perfect matching".

Theorem

After M independent repetition as above

$$\mathbb{P}(\text{error}) \leq \frac{1}{2^M}.$$

Break

Introduction

The problem

Let G be a bipartite graph, $c : E(G) \to \mathbb{R}^+$. Find a matching M, that $c(M) = \sum_{e \in M} c(e)$ has the maximum value.

Introduction

The problem

Let G be a bipartite graph, $c : E(G) \to \mathbb{R}^+$. Find a matching M, that $c(M) = \sum_{e \in M} c(e)$ has the maximum value.

Definition

For an edges set $M \subseteq E(G) = \{e_1, \ldots, e_m\}$ the characteristic vector of M is $\underline{\chi}_M = (v_i) \in \mathbb{R}^m$, where $v_i = 1$, if $e_i \in M$, otherwise 0.

Introduction

The problem

Let G be a bipartite graph, $c : E(G) \to \mathbb{R}^+$. Find a matching M, that $c(M) = \sum_{e \in M} c(e)$ has the maximum value.

Definition

For an edges set $M \subseteq E(G) = \{e_1, \ldots, e_m\}$ the characteristic vector of M is $\underline{\chi}_M = (v_i) \in \mathbb{R}^m$, where $v_i = 1$, if $e_i \in M$, otherwise 0.

Note that $m = |E(G)|$, hence

$$\mathbb{R}^{E(G)} \equiv \mathbb{R}^m.$$

Algebraization

Algebraization

$c(M) = \langle \underline{c}, \underline{\chi}_M \rangle$, where $\underline{c} \in \mathbb{R}^{E(G)}$.

Algebraization

$c(M) = \langle \underline{c}, \underline{\chi}_M \rangle$, where $\underline{c} \in \mathbb{R}^{E(G)}$.

The new form of our problem:

$$\begin{aligned} &\max\{\langle \underline{c}, \underline{\chi}_M \rangle : M \text{ is a matching}\} \\ &\qquad = \max\{\langle \underline{c}, \underline{x} \rangle : \underline{x} \in \{\underline{\chi}_M : M \text{ is a matching}\}\} \\ &\qquad = \max\{\langle \underline{c}, \underline{x} \rangle : \underline{x} \in \operatorname{conv} \{\underline{\chi}_M : M \text{ is a matching}\}\} \end{aligned}$$

Geometry

Geometry

Definition

Let $P \subseteq \mathbb{R}^m$ be a set of vectors/points. Then *the convex hull of P* is,

$$\text{conv } P = \{\sum_{i=1}^{k} \lambda_i \underline{p}_i : \lambda_i \geq 0, \sum \lambda_i = 1, \underline{p}_i \in P\}$$

the minimal convex set containing P.

Geometry

Definition

Let $P \subseteq \mathbb{R}^m$ be a set of vectors/points. Then *the convex hull of P* is,

$$\text{conv } P = \{\sum_{i=1}^{k} \lambda_i \underline{p}_i : \lambda_i \geq 0, \sum \lambda_i = 1, \underline{p}_i \in P\}$$

the minimal convex set containing P.

The vectors/points of the sets are called the convex combinations of the elements of P.

Geometry

Definition

Let $P \subseteq \mathbb{R}^m$ be a set of vectors/points. Then *the convex hull of P* is,

$$\text{conv } P = \{\sum_{i=1}^{k} \lambda_i \underline{p}_i : \lambda_i \geq 0, \sum \lambda_i = 1, \underline{p}_i \in P\}$$

the minimal convex set containing P.

The vectors/points of the sets are called the convex combinations of the elements of P.

Notation

The set conv $\{\underline{\chi}_M : M \text{ is a matching}\}$ is denoted as $MP(G)$.

The relation of the continuous and discrete set

The relation of the continuous and discrete set

$MP(G)$ is a geometrical/continuous set of points,
$\{\chi_M : M \text{ is a matching}\}$ is a combinatorial/discrete set of points.

The relation of the continuous and discrete set

$MP(G)$ is a geometrical/continuous set of points,
$\{\chi_M : M$ is a matching$\}$ is a combinatorial/discrete set of points.

$$\{\chi_M : M \text{ is a matching}\} \subset \text{conv}\,\{\underline{\chi}_M : M\text{is a matching}\}$$

The relation of the continuous and discrete set

$MP(G)$ is a geometrical/continuous set of points,
$\{\chi_M : M \text{ is a matching}\}$ is a combinatorial/discrete set of points.

$$\{\chi_M : M \text{ is a matching}\} \subset \text{conv}\,\{\underline{\chi}_M : M \text{is a matching}\}$$

Usually extending the set of feasible solutions effects the optimization problem.

The relation of the continuous and discrete set

$MP(G)$ is a geometrical/continuous set of points,
$\{\chi_M : M$ is a matching$\}$ is a combinatorial/discrete set of points.

$$\{\chi_M : M \text{ is a matching}\} \subset \text{conv}\,\{\underline{\chi}_M : M \text{is a matching}\}$$

Usually extending the set of feasible solutions effects the optimization problem. In our problem this not the case.

The relation of the continuous and discrete set

$MP(G)$ is a geometrical/continuous set of points,
$\{\chi_M : M \text{ is a matching}\}$ is a combinatorial/discrete set of points.

$$\{\chi_M : M \text{ is a matching}\} \subset \text{conv}\,\{\underline{\chi}_M : M\text{is a matching}\}$$

Usually extending the set of feasible solutions effects the optimization problem. In our problem this not the case.

$MP(G)$ convex, bounded, closed set.

The relation of the continuous and discrete set

$MP(G)$ is a geometrical/continuous set of points,
$\{\chi_M : M \text{ is a matching}\}$ is a combinatorial/discrete set of points.

$$\{\chi_M : M \text{ is a matching}\} \subset \text{conv}\,\{\underline{\chi}_M : M\text{is a matching}\}$$

Usually extending the set of feasible solutions effects the optimization problem. In our problem this not the case.

$MP(G)$ convex, bounded, closed set.

If we optimize a linear function on $MP(G)$ then the optimal value is obtained at a point $\underline{\chi}_M$:

$$\langle \underline{c}, \sum \lambda_i \underline{p}_i \rangle = \sum \lambda_i \langle \underline{c}, \underline{p}_i \rangle \leq \max \langle \underline{c}, \underline{p}_i \rangle.$$

Linear programming/LP

Linear programming/LP

Solving

$$\max\{\langle \underline{c}, \underline{x}\rangle : \underline{x} \in MP(G)\}$$

is an LP problem.

Linear programming/LP

Solving

$$\max\{\langle \underline{c}, \underline{x}\rangle : \underline{x} \in MP(G)\}$$

is an LP problem.

The standard usage of an LP algorithm requires the description of $MP(G)$ as an intersection of finitely many closed halfspaces (solution set of a system of linear inequalities).

Linear programming/LP

Solving

$$\max\{\langle \underline{c}, \underline{x}\rangle : \underline{x} \in MP(G)\}$$

is an LP problem.

The standard usage of an LP algorithm requires the description of $MP(G)$ as an intersection of finitely many closed halfspaces (solution set of a system of linear inequalities).

Bellow we describe few linear inequalities that is satisfied by the vectors of $\{\chi_M : M \text{ is a matching}\}$.

Linear programming/LP

Solving

$$\max\{\langle \underline{c}, \underline{x}\rangle : \underline{x} \in MP(G)\}$$

is an LP problem.

The standard usage of an LP algorithm requires the description of $MP(G)$ as an intersection of finitely many closed halfspaces (solution set of a system of linear inequalities).

Bellow we describe few linear inequalities that is satisfied by the vectors of $\{\chi_M : M \text{ is a matching}\}$.

Definition

Let

$$\widehat{MP}(G) = \{\underline{x} \in \mathbb{R}^{E(G)} : x_e \geq 0\,\forall e \in E(G), \text{ and } \sum_{e:v|e} x_e \leq 1\,\forall v \in V(G)\},$$

where $\underline{x} = (x_e : e \in E(G)) \in \mathbb{R}^{E(G)}$.

The relation of the two polytope

The relation of the two polytope

One inclusion is obvious

$$MP(G) \subseteq \widehat{MP}(G).$$

The relation of the two polytope

One inclusion is obvious

$$MP(G) \subseteq \widehat{MP}(G).$$

In general the inclusion is strict.

The relation of the two polytope

One inclusion is obvious

$$MP(G) \subseteq \widehat{MP}(G).$$

In general the inclusion is strict.

One example is $G = C_{2k+1}$.

The relation of the two polytope

One inclusion is obvious

$$MP(G) \subseteq \widehat{MP}(G).$$

In general the inclusion is strict.

One example is $G = C_{2k+1}$.

Let $\underline{x} \in \mathbb{R}^{E(G)}$ the vector with all $\frac{1}{2}$ components. $\underline{x} \in \widehat{MP}(G)$ and $\underline{x} \notin MP(G)$ ($\sum_{e \in E(G)} x_e = k + 1/4$ hyperplane separate $\underline{x}$ from $MP(G)$).

The goal

The goal

Our goal is to prove that if G is a bipartite graph then $MP(G) = \widehat{MP}(G)$.

The goal

Our goal is to prove that if G is a bipartite graph then $MP(G) = \widehat{MP}(G)$.

It is enough to show that the vertices of $\widehat{MP}(G)$ have integer coordinates.

The goal

Our goal is to prove that if G is a bipartite graph then $MP(G) = \widehat{MP}(G)$.

It is enough to show that the vertices of $\widehat{MP}(G)$ have integer coordinates.

Indeed: The elements of $\widehat{MP}(G)$ with integer coordinates are exactly the set $\{\chi_M : M \text{ is a matching}\}$!

The goal

Our goal is to prove that if G is a bipartite graph then $MP(G) = \widehat{MP}(G)$.

It is enough to show that the vertices of $\widehat{MP}(G)$ have integer coordinates.

Indeed: The elements of $\widehat{MP}(G)$ with integer coordinates are exactly the set $\{\chi_M : M \text{ is a matching}\}$!

The Lemma, that implies the Goal

The Lemma, that implies the Goal

Lemma

Let $Inc_G \in \mathbb{R}^{V \times E}$ be the vertex-edge incidence matrix of the bipartite graph G. Then each square submatrix R of Inc_G has determinant $-1, 0$, or 1.

The Lemma, that implies the Goal

Lemma

Let $Inc_G \in \mathbb{R}^{V \times E}$ be the vertex-edge incidence matrix of the bipartite graph G. Then each square submatrix R of Inc_G has determinant $-1, 0$, or 1.

If we want to see a generic vertex of $\widehat{MP}(G)$, then we take suitable $|E|$ defining inequalities of $\widehat{MP}(G)$. Substitute the inequality signs with equality signs. We obtain a system of linear equations, that has a unique solution.

The Lemma, that implies the Goal

Lemma

Let $Inc_G \in \mathbb{R}^{V \times E}$ be the vertex-edge incidence matrix of the bipartite graph G. Then each square submatrix R of Inc_G has determinant $-1, 0$, or 1.

If we want to see a generic vertex of $\widehat{MP}(G)$, then we take suitable $|E|$ defining inequalities of $\widehat{MP}(G)$. Substitute the inequality signs with equality signs. We obtain a system of linear equations, that has a unique solution.

The unique solution is the vertex, chosen.

The Lemma, that implies the Goal

Lemma

Let $Inc_G \in \mathbb{R}^{V \times E}$ be the vertex-edge incidence matrix of the bipartite graph G. Then each square submatrix R of Inc_G has determinant $-1, 0$, or 1.

If we want to see a generic vertex of $\widehat{MP}(G)$, then we take suitable $|E|$ defining inequalities of $\widehat{MP}(G)$. Substitute the inequality signs with equality signs. We obtain a system of linear equations, that has a unique solution.

The unique solution is the vertex, chosen.

To see the unique solution use Cramer's rule. Its coordinates are fractions of two determinants. The two determinants are integers, and the denominator is non-zero.

The Lemma, that implies the Goal

Lemma

Let $Inc_G \in \mathbb{R}^{V\times E}$ be the vertex-edge incidence matrix of the bipartite graph G. Then each square submatrix R of Inc_G has determinant $-1, 0$, or 1.

If we want to see a generic vertex of $\widehat{MP}(G)$, then we take suitable $|E|$ defining inequalities of $\widehat{MP}(G)$. Substitute the inequality signs with equality signs. We obtain a system of linear equations, that has a unique solution.

The unique solution is the vertex, chosen.

To see the unique solution use Cramer's rule. Its coordinates are fractions of two determinants. The two determinants are integers, and the denominator is non-zero. By the Lemma the denominator must be ± 1.

The Lemma, that implies the Goal

Lemma

Let $Inc_G \in \mathbb{R}^{V\times E}$ be the vertex-edge incidence matrix of the bipartite graph G. Then each square submatrix R of Inc_G has determinant $-1, 0$, or 1.

If we want to see a generic vertex of $\widehat{MP}(G)$, then we take suitable $|E|$ defining inequalities of $\widehat{MP}(G)$. Substitute the inequality signs with equality signs. We obtain a system of linear equations, that has a unique solution.

The unique solution is the vertex, chosen.

To see the unique solution use Cramer's rule. Its coordinates are fractions of two determinants. The two determinants are integers, and the denominator is non-zero. By the Lemma the denominator must be ± 1. This implies that the ratios are integers and the Goal is achieved.

The proof of the Lemma

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column. Using the induction hypothesis we are done.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column. Using the induction hypothesis we are done.

Case 2: Each column of R has two 1's.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column. Using the induction hypothesis we are done.

Case 2: Each column of R has two 1's. // we know that one is in a lower row, and one in an upper row.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column. Using the induction hypothesis we are done.

Case 2: Each column of R has two 1's. // we know that one is in a lower row, and one in an upper row. This implies that the sum of lower rows of R is the same as the sum of the upper rows.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column. Using the induction hypothesis we are done.

Case 2: Each column of R has two 1's. // we know that one is in a lower row, and one in an upper row. This implies that the sum of lower rows of R is the same as the sum of the upper rows. This implies a non-trivial linear combination of rows gives $\overrightarrow{0}$.

The proof of the Lemma

Let R be a square submatrix of size $k \times k$.

We prove the claim with induction on k.

The case $k = 1$ is obvious.

The rows of Inc_G (hence the rows of R two) can be classified as lower and upper rows.

Case 1: R has a column with at most one 1. We can expand the determinant according to this column. Using the induction hypothesis we are done.

Case 2: Each column of R has two 1's. // we know that one is in a lower row, and one in an upper row. This implies that the sum of lower rows of R is the same as the sum of the upper rows. This implies a non-trivial linear combination of rows gives $\overrightarrow{0}$. That means the determinant of R is 0.

Summary of our results

Summary of our results

Definition

A M is*totally unimodular*, if each square submatrix of it has determinant 0 or ± 1.

Summary of our results

Definition

A M is*totally unimodular*, if each square submatrix of it has determinant 0 or ± 1.

Corollary

If G is bipartite, then

a) Inc_G is totally unimodular,

b) $MP(G) = \widehat{MP}(G)$.

The LP algorithm

The LP algorithm

Algorithm based on linear programming

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

(Algebraization) Write down the linear inequalities describing $MP(G) = \widehat{MP}(G)$.

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

(Algebraization) Write down the linear inequalities describing $MP(G) = \widehat{MP}(G)$. Introduce the objective function is $\langle w, x\rangle$

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

(Algebraization) Write down the linear inequalities describing $MP(G) = \widehat{MP}(G)$. Introduce the objective function is $\langle w, x \rangle$

(Optimization) Solve the above LP problem (for example use the simplex method).

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

(Algebraization) Write down the linear inequalities describing $MP(G) = \widehat{MP}(G)$. Introduce the objective function is $\langle w, x \rangle$

(Optimization) Solve the above LP problem (for example use the simplex method).
// The solution is guaranteed to be an integer solution. Hence it is a 0/1 vector.

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

(Algebraization) Write down the linear inequalities describing $MP(G) = \widehat{MP}(G)$. Introduce the objective function is $\langle w, x \rangle$

(Optimization) Solve the above LP problem (for example use the simplex method).
// The solution is guaranteed to be an integer solution. Hence it is a $0/1$ vector.

(Combinatorialization) We interpret the optimal solution a characterization of an edge set M.

The LP algorithm

Algorithm based on linear programming

Given an edge weighted graph. Let $w \in \mathbb{R}^{E(G)}$ the vector of the weights. Find the maximum weight matching.

(Algebraization) Write down the linear inequalities describing $MP(G) = \widehat{MP}(G)$. Introduce the objective function is $\langle w, x \rangle$

(Optimization) Solve the above LP problem (for example use the simplex method).
// The solution is guaranteed to be an integer solution. Hence it is a $0/1$ vector.

(Combinatorialization) We interpret the optimal solution a characterization of an edge set M. M is the output.

This is the end!

Thank you for your attention!