Popular Matchings and Optimality

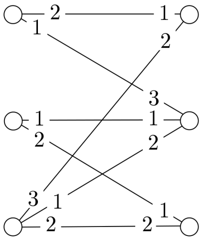
Kavitha Telikepalli

(Tata Institute of Fundamental Research, Mumbai)

Laboratoire d'Informatique de Paris-Nord (LIPN)
Université Sorbonne Paris Nord
May 14, 2024

The input

A bipartite graph where every vertex has a strict ranking of its neighbors.



We seek a matching that pairs up vertices as happily as possible.

How do we formalize such a pairing?

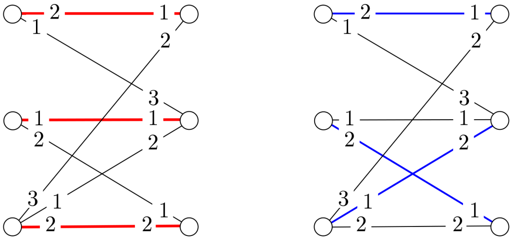
- Every agent a should be paired to the best possible job b such that

  b is willing to be matched to a.

Stability

The following property should hold for any agent/job v in our matching M:

- every neighbor ranked better than $M(v)$ is matched to a neighbor better than v.

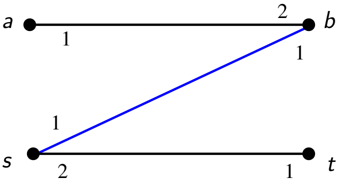


- The red matching satisfies this property; not the blue matching.
- The edge between the two middle vertices *blocks* the blue matching.
- A matching with no blocking edge is a stable matching.

Stable matchings

Stability is a very natural notion of "good matching".

The Gale-Shapley algorithm: agents propose and jobs dispose — this is a simple and clean algorithm.



As we saw, the above instance has a stable matching $\{sb\}$ of size 1.

- So the size of a stable matching might be only half the size of a maximum matching.
- This is the "price of stability".

Beyond stability

This motivates relaxing stability to a more *collective* or "democratic" notion.

Every vertex v has a ranking over all the possible matchings in G:

- $M \succ_v N$ if $M(v)$ is better than $N(v)$ in v's ranking;
- $M \prec_v N$ if $M(v)$ is worse than $N(v)$ in v's ranking;
- $M \sim_v N$ if $M(v) = N(v)$.

Beyond stability

This motivates relaxing stability to a more *collective* or "democratic" notion.

Every vertex v has a ranking over all the possible matchings in G:

- $M \succ_v N$ if $M(v)$ is better than $N(v)$ in v's ranking;
- $M \prec_v N$ if $M(v)$ is worse than $N(v)$ in v's ranking;
- $M \sim_v N$ if $M(v) = N(v)$.

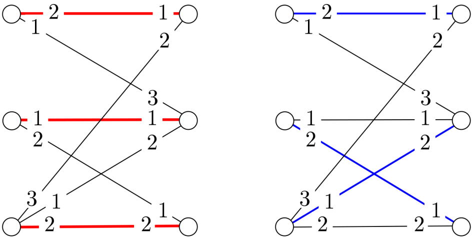
Any pair of matchings can be compared as follows:

- hold a head-to-head election between these two matchings.
- vertices are voters in this election.
- count the number of votes won by each matching.
- the matching with smaller number of votes loses this election.

The $\succeq$ and $\succ$ operators

Given two matchings M, N, we write $M \succeq N$ if

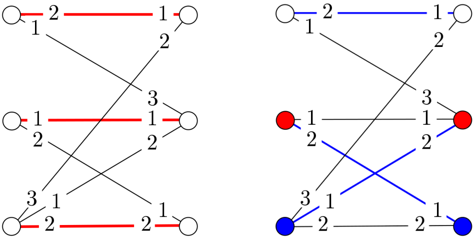
$$\underbrace{|\{v \in A \cup B : M(v) \succ_v N(v)\}|}_{\#\text{ of votes for } M} \geq \underbrace{|\{v \in A \cup B : N(v) \succ_v M(v)\}|}_{\#\text{ of votes for } N}$$



The $\succeq$ and $\succ$ operators

Given two matchings M, N, we write $M \succeq N$ if

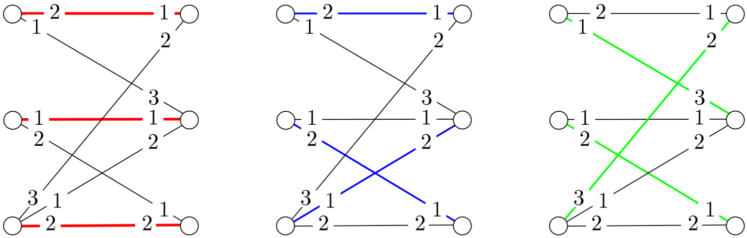
$$\underbrace{|\{v \in A \cup B : M(v) \succ_v N(v)\}|}_{\#\text{ of votes for } M} \geq \underbrace{|\{v \in A \cup B : N(v) \succ_v M(v)\}|}_{\#\text{ of votes for } N}$$



The $\succeq$ and $\succ$ operators

Given two matchings M, N, we write $M \succeq N$ if

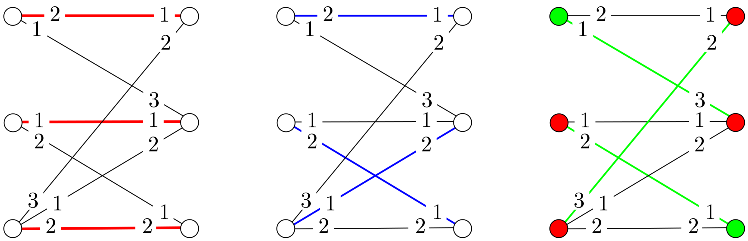
$$\underbrace{|\{v \in A \cup B : M(v) \succ_v N(v)\}|}_{\#\text{ of votes for } M} \geq \underbrace{|\{v \in A \cup B : N(v) \succ_v M(v)\}|}_{\#\text{ of votes for } N}$$



The $\succeq$ and $\succ$ operators

Given two matchings M, N, we write $M \succeq N$ if

$$\underbrace{|\{v \in A \cup B : M(v) \succ_v N(v)\}|}_{\#\text{ of votes for } M} \geq \underbrace{|\{v \in A \cup B : N(v) \succ_v M(v)\}|}_{\#\text{ of votes for } N}$$

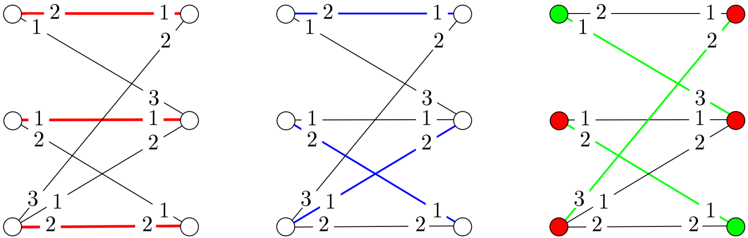


and $M \succ N$ if the inequality is strict (M defeats N).

The $\succeq$ and $\succ$ operators

Given two matchings M, N, we write $M \succeq N$ if

$$\underbrace{|\{v \in A \cup B : M(v) \succ_v N(v)\}|}_{\#\text{ of votes for } M} \geq \underbrace{|\{v \in A \cup B : N(v) \succ_v M(v)\}|}_{\#\text{ of votes for } N}$$



and $M \succ N$ if the inequality is strict (M defeats N).

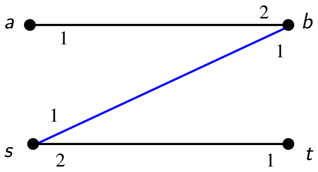
A matching M that does not lose any election is a popular matching.

Max-size popular matching

Every stable matching is popular [Gardenfors, 1975].

- Every stable matching is a min-size popular matching.

In the instance below $\{ab, st\}$ is a max-size popular matching.



Our goal now is to find the matching $\{ab, st\}$ here.

- We know that running the Gale-Shapley algorithm finds $\{sb\}$.

In order to find $\{ab, st\}$, a should get a second chance to propose to b.

- However b will again reject a.

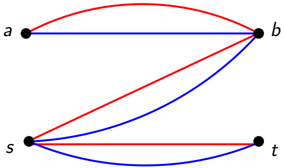
A new instance G'

IDEA. b should prefer a's second proposal to s's first proposal.

A new graph G' where every edge uv in G is replaced by uv and uv in G':

- that is, by two parallel edges: one red and the other blue.

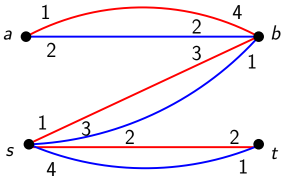
The corresponding graph G' is:



- Red edges correspond to *first-time* proposals.
- Blue edges correspond to *second-time* proposals.

A new instance G'

The graph G' with preferences is:



- Every agent prefers any red edge to any blue edge.
- Every job prefers any blue edge to any red edge.

The matching $\{ab, st\}$ is stable here.

- Ignoring colors, this is the desired matching $\{ab, st\}$.

Stable matchings in G'

Our algorithm in $G = (A \cup B, E)$

- Construct the red/blue graph $G' = (A \cup B, E')$.
- Run Gale-Shapley algorithm in G' to compute M'.
- Return the corresponding matching M in G.

Stable matchings in G'

Our algorithm in $G = (A \cup B, E)$

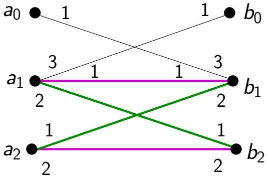
- Construct the red/blue graph $G' = (A \cup B, E')$.
- Run Gale-Shapley algorithm in G' to compute M'.
- Return the corresponding matching M in G.

CLAIM. M is popular matching in G.

- Furthermore, M is more popular than any larger matching.

Thus M is a max-size popular matching in G.

Max-size popular matchings



There are two max-size popular matchings here: purple and green.

- Only the green matching occurs as a stable matching in the red/blue graph G'.
- The purple matching cannot be realized as a stable matching in G'.

Hence not every max-size popular matching occurs as a stable matching in G'.

Dominant matchings

Any stable matching in the red/blue graph G' has an interesting property:

- M is a popular matching that is more popular than all larger matchings.

DEF. Call a popular matching that defeats all larger matchings dominant.

Dominant matchings

Any stable matching in the red/blue graph G' has an interesting property:

- M is a popular matching that is more popular than all larger matchings.

DEF. Call a popular matching that defeats all larger matchings dominant.

Dominant matchings in $G \equiv$ Stable matchings in the red/blue graph G'.
[Cseh and K, 2018]

- Given $e \in E$: is there a popular matching in $G = (A \cup B, E)$ with e?

Popularity with a forced edge

Forced edge e

1. Check if there is a stable matching in G with edge e.
2. Check if there is a dominant matching in G with edge e.
3. If the answer in steps 1 and 2 is no then return "no".

Popularity with a forced edge

FORCED EDGE e

1. Check if there is a stable matching in G with edge e.
2. Check if there is a dominant matching in G with edge e.
3. If the answer in steps 1 and 2 is no then return "no".

- Any popular matching decomposes into a stable part and a dominant part.
- This leads to the correctness of the above algorithm.

Popularity with a forced pair of edges

Given a pair of edges e_1, e_2 in G:

- Is there a popular matching in G with both e_1 and e_2?

There are polynomial time algorithms to determine if there is a stable matching with a given set $\{e_1, \ldots, e_k\}$ of edges [Gusfield and Irving, 1989].

- there need not be a stable / dominant matching with both e_1 and e_2.

Popularity with a forced pair of edges

Given a pair of edges e_1, e_2 in G:

- Is there a popular matching in G with both e_1 and e_2?

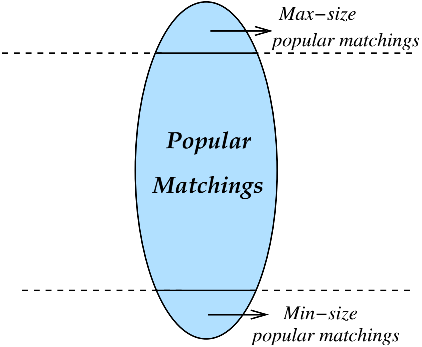
There are polynomial time algorithms to determine if there is a stable matching with a given set $\{e_1, \ldots, e_k\}$ of edges [Gusfield and Irving, 1989].

- there need not be a stable / dominant matching with both e_1 and e_2.

THEOREM. The above problem is NP-hard [Faenza, K, Powers, Zhang, 2019].

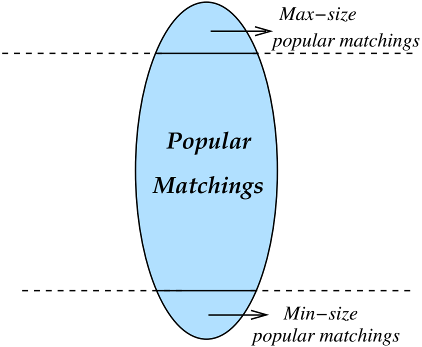
The easy subclasses

Finding a max-size (similarly, min-size) popular matching in G is easy.



The easy subclasses

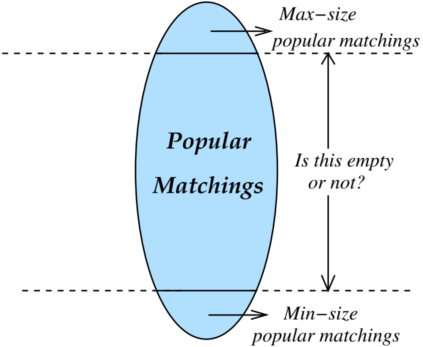
Finding a max-size (similarly, min-size) popular matching in G is easy.



Algorithmic question: Is there *any* popular matching in G that is neither a max-size nor a min-size popular matching?

The easy subclasses

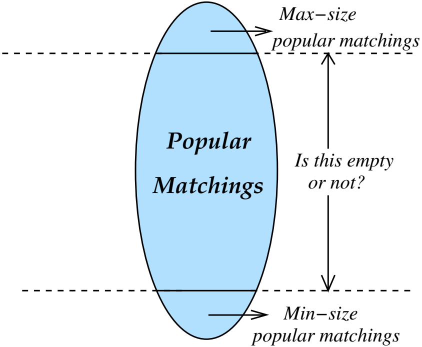
Finding a max-size (similarly, min-size) popular matching in G is easy.



Algorithmic question: Is there *any* popular matching in G that is neither a max-size nor a min-size popular matching?

The easy subclasses

Finding a max-size (similarly, min-size) popular matching in G is easy.



It is NP-hard to decide if G admits a popular matching that is neither a max-size nor a min-size popular matching.

Min-cost popular matchings

Suppose every edge has a cost.

- It is NP-hard to find a min-cost popular matching.

Finding a min-cost stable matching in G is easy.

- Efficient combinatorial [Irving, Leather, and Gusfield, 1987] and LP-based [Vande Vate, 1987; Rothblum, 1992] algorithms.
- The stable matching polytope has a linear-size description.

A relaxation of the min-cost popular matching problem:

- the min-cost popular mixed matching problem.

Mixed matchings

A mixed matching is a probability distribution over matchings, i.e.,

$$\Pi = \{(M_0, p_0), \ldots, (M_k, p_k)\},$$

where $M_0, \ldots, M_k$ are matchings in G and $\sum_i p_i = 1$ and $p_i \geq 0 \ \forall i$.

- A mixed matching is a lottery over matchings.

Mixed matchings

A mixed matching is a probability distribution over matchings, i.e.,

$$\Pi = \{(M_0, p_0), \ldots, (M_k, p_k)\},$$

where $M_0, \ldots, M_k$ are matchings in G and $\sum_i p_i = 1$ and $p_i \geq 0\ \forall i$.

- A mixed matching is a lottery over matchings.

For any two matchings M and N:

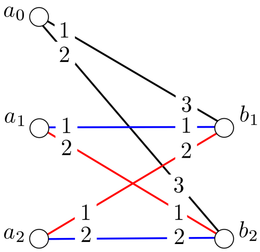
$$\text{let } \Delta(N, M) \;=\; \#\text{ of votes for } N - \#\text{ of votes for } M.$$

- Define $\Delta(N, \Pi) = \sum_i p_i \cdot \Delta(N, M_i)$.

Definition. Call a mixed matching Π popular if $\Delta(N, \Pi) \leq 0\ \forall$ matchings N.

An interesting example

This instance has only one popular matching $S = \{a_1b_1, a_2b_2\}$.



$M = \{a_1b_2, a_2b_1\}$ is unpopular; it is defeated by $N = \{a_0b_2, a_1b_1\}$.

- However the mixed matching $\Pi = \{(S, \frac{1}{2}), (M, \frac{1}{2})\}$ is popular:
  - observe that $\Delta(N, \Pi) = \frac{1}{2} \cdot \Delta(N, S) + \frac{1}{2} \cdot \Delta(N, M) = \frac{1}{2} - \frac{1}{2} = 0$.

Thus a popular mixed matching need not be a probability distribution over popular matchings.

The popular fractional matching polytope

In bipartite graphs: a mixed matching $\equiv$ a fractional matching [Birkhoff-von Neumann theorem].

A fractional matching is a point $\vec{x}$ in the matching polytope $\mathcal{M}_G$, i.e., $\vec{x} \in \mathbb{R}^m_{\geq 0}$ and $\sum_{e \in \delta(u)} x_e \leq 1$ for all $u \in A \cup B$.

$$\text{Let } \mathcal{F}_G = \{\vec{x} \in \mathcal{M}_G : \Delta(N, \vec{x}) \leq 0 \text{ for all matchings } N\}$$

$$\text{where } \Delta(N, \vec{x}) = \sum_{u \in A \cup B} \underbrace{\left(\sum_{v \prec_u N(u)} x_{uv} - \sum_{v \succ_u N(u)} x_{uv} \right)}_{u\text{'s vote for } N(u) \text{ vs } \vec{x}_u}$$

- $\mathcal{F}_G$ is the popular fractional matching polytope of G.
- That is, $\mathcal{F}_G$ is the convex hull of all popular fractional matchings in G.

Towards a compact extended formulation for $\mathcal{F}_G$

For any $\vec{x} \in \mathcal{M}_G$ and $e = ab \in E$:

- Let $\mathrm{wt}_x(e) = (a\text{'s vote for } b \text{ versus } \vec{x}_a) + (b\text{'s vote for } a \text{ versus } \vec{x}_b)$.

$$\text{So } \mathrm{wt}_x(e) = \underbrace{\left(\sum_{b' \prec_a b} x_{ab'} - \sum_{b' \succ_a b} x_{ab'}\right)}_{a\text{'s vote for } b \text{ vs } \vec{x}_a} + \underbrace{\left(\sum_{a' \prec_b a} x_{a'b} - \sum_{a' \succ_b a} x_{a'b}\right)}_{b\text{'s vote for } a \text{ vs } \vec{x}_b}.$$

Towards a compact extended formulation for $\mathcal{F}_G$

For any $\vec{x} \in \mathcal{M}_G$ and $e = ab \in E$:

- Let $\mathrm{wt}_x(e) = (a\text{'s vote for } b \text{ versus } \vec{x}_a) + (b\text{'s vote for } a \text{ versus } \vec{x}_b)$.

$$\text{So } \mathrm{wt}_x(e) = \underbrace{\left(\sum_{b' \prec_a b} x_{ab'} - \sum_{b' \succ_a b} x_{ab'}\right)}_{a\text{'s vote for } b \text{ vs } \vec{x}_a} + \underbrace{\left(\sum_{a' \prec_b a} x_{a'b} - \sum_{a' \succ_b a} x_{a'b}\right)}_{b\text{'s vote for } a \text{ vs } \vec{x}_b}.$$

- Similarly, $\mathrm{wt}_x(uu) = \underbrace{-\sum_{e \in \delta(u)} x_e}_{u\text{'s vote for itself vs } \vec{x}_u}$ for any $u \in A \cup B$.

Thus $\Delta(N, \vec{x}) = \mathrm{wt}_x(N)$ for any perfect matching N.

(note that N is augmented with self-loops to make it perfect)

- Hence $\vec{x}$ is popular $\iff$ $\mathrm{wt}_x(N) \le 0$ for all perfect matchings N.

LP for max-weight perfect matching

$$\max \sum_{e} \mathsf{wt}_x(e) \cdot y_e$$

$$\begin{aligned} \sum_{e \in \delta(u) \cup \{uu\}} y_e &= 1 \quad \forall u \in A \cup B \\ y_e &\geq 0 \quad \forall e \in E \cup \{\text{self-loops}\}. \end{aligned}$$

$\vec{x}$ is popular $\iff$ opt $= 0$.

LP for max-weight perfect matching

$$\max \sum_{e} \mathsf{wt}_x(e) \cdot y_e$$

$$\begin{aligned} \sum_{e \in \delta(u) \cup \{uu\}} y_e &= 1 \ \ \forall u \in A \cup B \\ y_e &\geq 0 \ \ \forall e \in E \cup \{\text{self-loops}\}. \end{aligned}$$

$\vec{x}$ is popular $\iff$ opt $= 0$.

The above LP allows us to test if the fractional matching $\vec{x}$ is popular or not.

- But we want to describe the polytope of all popular fractional matchings.

So we also have to add the constraints $\sum_{e \in \delta(u) \cup \{uu\}} x_e = 1 \ \forall u$ and $x_e \geq 0 \ \forall e$.

- Then the objective function becomes *quadratic* in variables x_e and y_e.

LP for max-weight perfect matching

So let us consider the dual LP for max-weight perfect matching.

Dual LP

$$\min \sum_{u} \alpha_u$$

$$\begin{aligned} \alpha_a + \alpha_b &\geq \mathsf{wt}_x(ab) \quad \forall\ ab \in E \\ \alpha_u &\geq \mathsf{wt}_x(uu) \quad \forall\ u \in A \cup B. \end{aligned}$$

$\vec{x}$ is popular $\iff$ $\exists$ dual feasible $\vec{\alpha}$ with $\sum_u \alpha_u = 0$ (dual certificate for $\vec{x}$).

LP for max-weight perfect matching

So let us consider the dual LP for max-weight perfect matching.

Dual LP

$$\min \sum_u \alpha_u$$

$$\begin{aligned} \alpha_a + \alpha_b &\geq \mathrm{wt}_x(ab) \ \ \forall\, ab \in E \\ \alpha_u &\geq \mathrm{wt}_x(uu) \ \ \forall\, u \in A \cup B. \end{aligned}$$

$\vec{x}$ is popular $\iff$ $\exists$ dual feasible $\vec{\alpha}$ with $\sum_u \alpha_u = 0$ (dual certificate for $\vec{x}$).

Let us add the constraints $\sum_{e \in \delta(u) \cup \{uu\}} x_e = 1 \ \forall u$ and $x_e \geq 0 \ \forall e$ to this LP.

- So this will be an LP in variables x_e's and α_u's.
- The set of optimal solutions $(\vec{x}, \vec{\alpha})$ is an extension of our polytope $\mathcal{F}_G$.

The popular fractional matching polytope $\mathcal{F}_G$

A compact extended formulation

$$\begin{aligned}
\alpha_a + \alpha_b &\geq \mathrm{wt}_x(ab) \quad \forall\ ab \in E \\
\alpha_u &\geq \mathrm{wt}_x(uu) \quad \forall\ u \in A \cup B \\
\sum_{e \in \delta(u) \cup \{uu\}} x_e &= 1\ \forall\ u \in A \cup B \\
x_e &\geq 0\ \forall\ e \in E \cup \{\textit{self-loops}\} \\
\sum_{u \in A \cup B} \alpha_u &= 0.
\end{aligned}$$

We can optimize over the above polytope (call it $\mathcal{F}'_G$) in polynomial time.

Thus we can find a min-cost popular mixed matching in polynomial time.

- A drawback of generalizing to mixed matchings is that the solution has become more complex to describe and difficult to implement.

The polytope $\mathcal{F}'_G$

The LP whose set of optimal solutions is $\mathcal{F}'_G$ has an unusual property:

- It is self-dual.
- So $x_{ab} > 0 \implies \alpha_a + \alpha_b = \mathrm{wt}_x(ab)$.
  (by complementary slackness)

Suppose G admits a perfect stable matching. Let $|A \cup B| = n$.

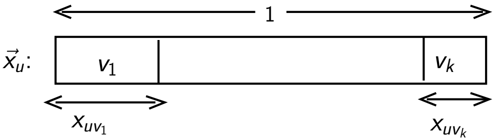
- Then every popular matching in G has a dual certificate in $\{\pm 1\}^n$.
- We show that $\mathcal{F}'_G$ (and thus $\mathcal{F}_G$) is integral in this special case.
  [Huang and K, 2021]

Our method is inspired by the proof of integrality of the formulation of the stable matching polytope [Teo and Sethuraman, 1998].

Integrality of $\mathcal{F}_G$ in the special case

Let $\vec{x} \in \mathcal{F}_G$. Then $(\vec{x}, \vec{\alpha}) \in \mathcal{F}'_G$ for some $\vec{\alpha} \in [-1, 1]^n$.

$\vec{x}_u = (x_{uv_1}, \ldots, x_{uv_k})$ is u's allocation in $\vec{x}$, for every $u \in A \cup B$.

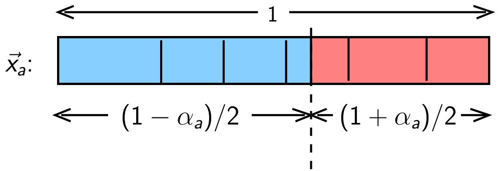


So $v_1, \ldots, v_k$ are the neighbors of u such that $x_{uv_i} > 0$ for $i = 1, \ldots, k$.

- For $a \in A$: arrange the entries of $\vec{x}_a$ in decreasing order of a's preference.
- For $b \in B$: arrange the entries of $\vec{x}_b$ in increasing order of b's preference.
- There is some $\alpha_u \in [-1, 1]$ for each $u \in A \cup B$. How do we interpret α_u?

Reordering the array $\vec{x}_a$

We will use a's α-value as follows.



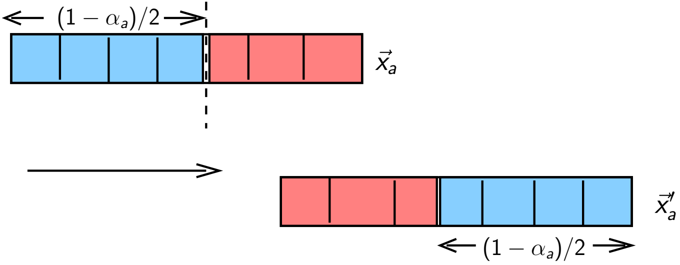
Call the initial $(1-\alpha_a)/2$ fraction of $\vec{x}_a$ the blue sub-array of $\vec{x}_a$.

Call the remaining $(1+\alpha_a)/2$ fraction of $\vec{x}_a$ the red sub-array of $\vec{x}_a$.

Swap the blue and red sub-arrays to form a reordered array $\vec{x}'_a$.

Reordering the array $\vec{x}_a$

Thus $\vec{x}_a \rightsquigarrow \vec{x}'_a$ by this swap.



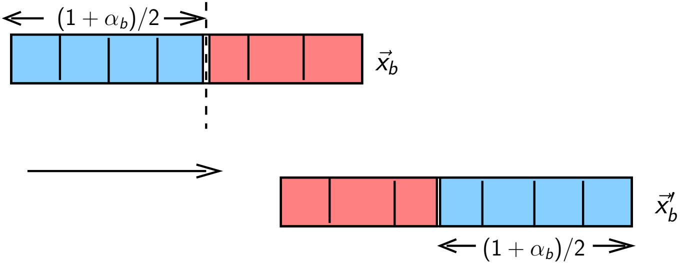
The order within the blue sub-array (similarly, the red sub-array) is preserved.

Reordering the array $\vec{x}_b$

We do an analogous transformation for $\vec{x}_b \rightsquigarrow \vec{x}'_b$ for any $b \in B$.

The initial $(1+\alpha_b)/2$ fraction of $\vec{x}_b$ will be the blue sub-array of $\vec{x}_b$.

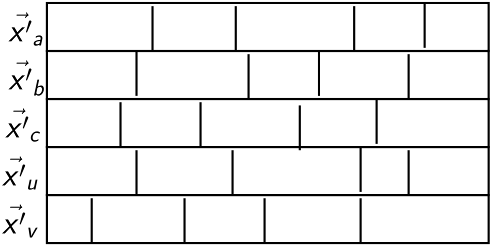
The latter $(1-\alpha_b)/2$ fraction of $\vec{x}_b$ will be the red sub-array of $\vec{x}_b$.



Swap the blue and red sub-arrays to form a reordered array $\vec{x}'_b$.

The table T

Form a table T whose rows are the reordered arrays.

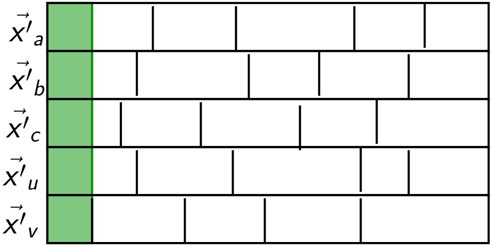


Sweep a vertical line along the table T decomposing it into columns.

$$M_i = \text{pairing defined by the } i\text{th column.}$$

The table T

Form a table T whose rows are the reordered arrays.

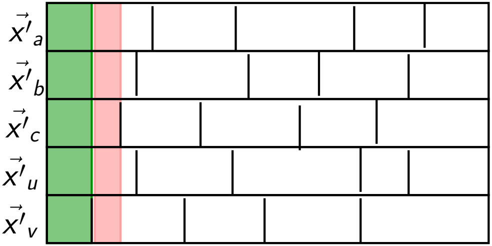


Sweep a vertical line along the table T decomposing it into columns.

M_1 = pairing defined by the first column.

The table T

Form a table T whose rows are the reordered arrays.

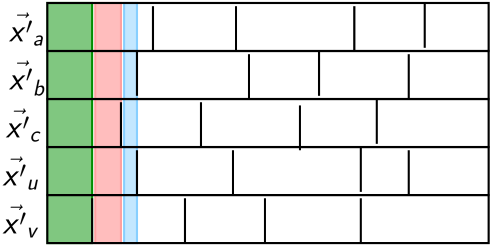


Sweep a vertical line along the table T decomposing it into columns.

M_2 = pairing defined by the second column.

The table T

Form a table T whose rows are the reordered arrays.

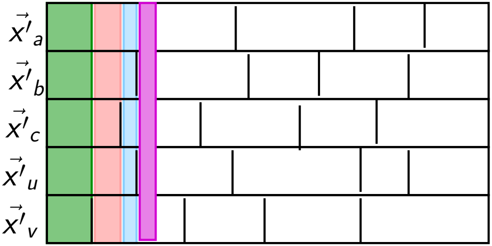


Sweep a vertical line along the table T decomposing it into columns.

M_3 = pairing defined by the third column.

The table T

Form a table T whose rows are the reordered arrays.



Sweep a vertical line along the table T decomposing it into columns.

M_4 = pairing defined by the fourth column.

Popularity of M_i

Thus $\vec{x} = \sum_i p_i \cdot M_i$, where p_i is the width of M_i's column.

- Self-duality of our LP $\Rightarrow$ each M_i is a matching in G.

We show a dual certificate $\vec{\alpha_i} \in \{\pm 1\}^n$ for each M_i.

- M_i corresponds to a red cell in $\vec{x}'_a$ (where $a \in A$) $\Rightarrow \alpha_i(a) = 1$;
- M_i corresponds to a red cell in $\vec{x}'_b$ (where $b \in B$) $\Rightarrow \alpha_i(b) = -1$;
- M_i corresponds to a blue cell in $\vec{x}'_a$ (where $a \in A$) $\Rightarrow \alpha_i(a) = -1$.
- M_i corresponds to a blue cell in $\vec{x}'_b$ (where $b \in B$) $\Rightarrow \alpha_i(b) = 1$.

This vector $\vec{\alpha_i}$ will be a feasible solution to the dual LP.

- Moreover, $\sum_{u \in A \cup B} \alpha_i(u) = 0$.
- Thus $\vec{\alpha_i}$ is a dual certificate for M_i.

Integrality of our formulation

Thus we have $(\vec{x}, \vec{\alpha}) = \sum_i p_i \cdot (M_i, \vec{\alpha_i})$ where:

- $\sum_i p_i = 1$ and $p_i \geq 0 \, \forall \; i$;
- each M_i is a matching;
- $\vec{\alpha}_i \in \{\pm 1\}^n$ is M_i's dual certificate (so M_i is popular).

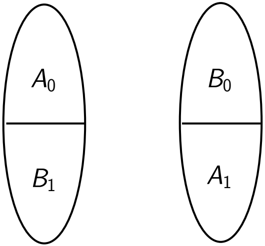
Hence $\mathcal{F}'_G$ (and thus $\mathcal{F}_G$) is integral.

- This is for the special case when G admits a perfect stable matching.

So we know how to formulate the popular matching polytope of G in this special case.

Half-integrality of the polytope $\mathcal{F}_G$ in the general case

Corresponding to $G = (A \cup B, E)$, let us build the following graph H:



Every vertex $u \in A \cup B$ has two copies u_0 and u_1 in H.

H is made up of 2 copies of G along with the "self-loop" edges $u_0 u_1$ $\forall u$.

- H admits a perfect stable matching.
- So $\mathcal{F}_H$ is integral.

Half-integrality of the polytope $\mathcal{F}_G$ in the general case

We can define natural maps $f : \mathcal{F}_G \to \mathcal{F}_H$ and $h : \mathcal{F}_H \to \mathcal{F}_G$:

- f will copy e's x-value on e_0 and e_1: $x_{e_0} = x_{e_1} = x_e$;
- h will be a "halving map": $x_e = (x_{e_0} + x_{e_1})/2$.
- so $h \circ f(\vec{x}) = \vec{x}$ for any $\vec{x} \in \mathcal{P}_G$.

Integrality of $\mathcal{F}_H \implies f(\vec{x}) = \{(M_i, p_i) : i = 1, \ldots, k\}$ for popular matchings $M_1, \ldots, M_k$ in H.

- So $\vec{x} = h \circ f(\vec{x}) = \{(h(M_i), p_i) : i = 1, \ldots, k\}$.

Half-integrality of the polytope $\mathcal{F}_G$ in the general case

We can define natural maps $f : \mathcal{F}_G \to \mathcal{F}_H$ and $h : \mathcal{F}_H \to \mathcal{F}_G$:

- f will copy e's x-value on e_0 and e_1: $x_{e_0} = x_{e_1} = x_e$;
- h will be a "halving map": $x_e = (x_{e_0} + x_{e_1})/2$.
- so $h \circ f(\vec{x}) = \vec{x}$ for any $\vec{x} \in \mathcal{P}_G$.

Integrality of $\mathcal{F}_H \implies f(\vec{x}) = \{(M_i, p_i) : i = 1, \ldots, k\}$ for popular matchings $M_1, \ldots, M_k$ in H.

- So $\vec{x} = h \circ f(\vec{x}) = \{(h(M_i), p_i) : i = 1, \ldots, k\}$.

Each $h(M_i)$ will be a popular half-integral matching in G.

- Thus $\vec{x}$ is a convex combination of popular half-integral matchings.
- So the popular fractional matching polytope $\mathcal{F}_G$ is half-integral.
- Hence there is always a min-cost popular mixed matching $\Pi = \{(M_0, \frac{1}{2}), (M_1, \frac{1}{2})\}$.

The popular matching polytope

Let $\mathcal{P}_G$ be the popular matching polytope of G.

- The extension complexity of $\mathcal{P}_G$ is $2^{\Omega\left(\frac{m}{\log m}\right)}$, where $|E| = m$. [Faenza and K, 2022]

Can we relax *popular* to "approximately popular" for the sake of tractability?

- M is popular $\implies$ no matching wins more votes than M.

DEFINITION. Call a matching M quasi-popular if no matching wins more than twice as many votes as M.

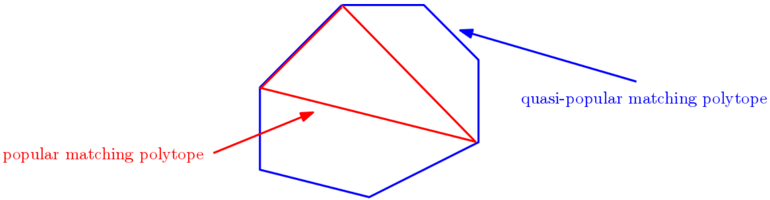
- Can a min-cost quasi-popular matching be efficiently computed?
- No, we show it is NP-hard to compute a min-cost quasi-popular matching.

Quasi-popular matchings

The min-cost popular / quasi-popular matching problems are NP-hard to approximate up to any factor.

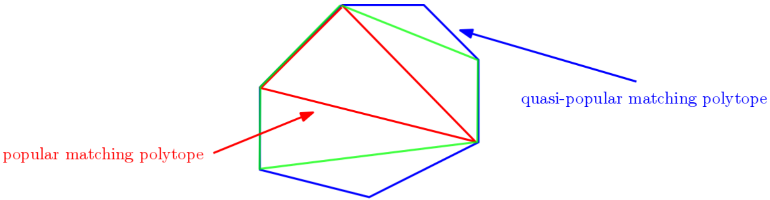
- A Bicriteria Approximation. Can we efficiently find a quasi-popular matching of cost at most that of a min-cost popular matching?
- Interestingly, the answer is *yes*.

Our technique



- The polytopes $\mathcal{P}_G$ and $\mathcal{Q}_G$ have near-exponential extension complexity.

Our technique



- The polytopes $\mathcal{P}_G$ and $\mathcal{Q}_G$ have near-exponential extension complexity.
- We show an integral polytope $\mathcal{C}$ sandwiched between $\mathcal{P}_G$ and $\mathcal{Q}_G$ such that $\mathcal{C}$ has a compact extended formulation.
- Optimizing over $\mathcal{C}$ leads to the efficient bicriteria algorithm.

The popular matching polytope in a special case

Special case:

- G has a perfect stable matching $\implies \mathcal{F}_G$ is integral.

From general case to special case:

- Assume for simplicity that $|A| = |B|$.
- **Our idea:** augment G with some new edges so that the resulting graph G^* has a perfect stable matching.

The popular matching polytope in a special case

Special case:

- G has a perfect stable matching $\implies \mathcal{F}_G$ is integral.

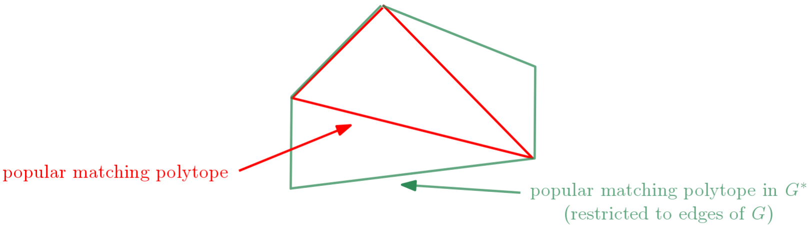
From general case to special case:

- Assume for simplicity that $|A| = |B|$.
- **Our idea:** augment G with some new edges so that the resulting graph G^* has a perfect stable matching.
- Pair up unstable vertices appropriately; add a new edge between each pair.

So the popular matching polytope of G^* has a compact extended formulation.
(by [Huang and K, 2021])

The sandwiched integral polytope

Every popular matching in G can be extended to a perfect popular matching in G^* using these new edges.

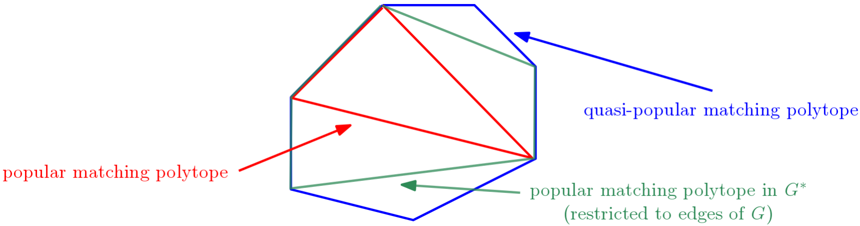


.

The sandwiched integral polytope

Every popular matching in G can be extended to a perfect popular matching in G^* using these new edges.

Every popular matching in G^*, when restricted to edges of G, is quasi-popular in G.

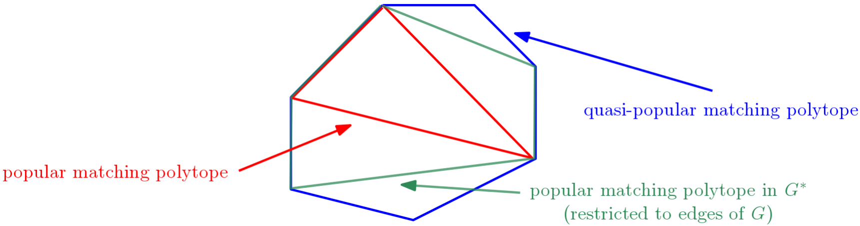


.

The sandwiched integral polytope

Every popular matching in G can be extended to a perfect popular matching in G^* using these new edges.

Every popular matching in G^*, when restricted to edges of G, is quasi-popular in G.



G^* has a perfect stable matching, so $\mathcal{F}_{G^*}$ = popular matching polytope of G^*.

The compact extended formulation of $\mathcal{F}_{G^*}$ is an extension of an integral polytope $\mathcal{C}$ where: $\mathcal{P}_G \subseteq \mathcal{C} \subseteq \mathcal{Q}_G$.

Another relaxation of popularity

Suppose N is a "very unpopular" matching.

- Let us not give N the power to block other matchings.
- That is, we waive the constraint $\Delta(N, \vec{x}) \leq 0$ in the popular matching polytope formulation to get a more relaxed formulation.

We seek a relaxation that admits a compact formulation.

- Then we can efficiently optimize over this polytope.

How do we define "very unpopular" matchings?

Another relaxation of popularity

Call a matching S supporting if $\exists$ a popular mixed matching Π whose support contains S.

- So $\exists$ popular $\Pi = \{(M_0, p_0), \ldots, (M_k, p_k)\}$ such that $S = M_i$ for some i.

A non-supporting matching cannot form a popular mixture even with the help of other matchings.

- A matching that is not supporting will be considered "very unpopular".

Call a matching M fairly popular if $\Delta(S, M) \leq 0$ $\forall$ supporting matchings S.

- Though M need not be popular, any matching that defeats M is *uninteresting* wrt popularity.

Fairly popular matchings

Can a min-cost fairly popular matching be computed in polynomial time?

- Interestingly, the answer is *yes*.

Fairly popular matchings

Can a min-cost fairly popular matching be computed in polynomial time?

- Interestingly, the answer is *yes*.

The following statements are equivalent.

- S is a supporting matching.
- No popular mixed matching defeats S.
- S matches all stable vertices and $S \subseteq E_p$ where E_p is the set of popular fractional edges in G.

The set E_p is the set of popular edges in the graph H (this is two copies of G).

Fairly popular matchings

This simple characterization of supporting matchings allows us to show:

- M is fairly popular $\iff$ M can be realized as a stable matching in a certain colorful multigraph.
- The min-cost stable matching algorithm in this multigraph finds a min-cost fairly popular matching in G in polynomial time.
- The fairly popular matching polytope admits a compact extended formulation as the stable matching polytope of this colorful multigraph.

A further relaxation

Call a matching M pseudo-popular if $\Delta(P, M) = 0$ $\forall$ popular matchings P.

- So M is undefeated by all popular matchings.
- What is the complexity of deciding if a given matching M is pseudo-popular?
- This is a coNP-hard problem.

A further relaxation

Call a matching M pseudo-popular if $\Delta(P, M) = 0$ $\forall$ popular matchings P.

- So M is undefeated by all popular matchings.
- What is the complexity of deciding if a given matching M is pseudo-popular?
- This is a coNP-hard problem.

By contrast, it is easy to decide if a given matching M is undefeated by all popular mixed matchings.

- Such an M is a supporting matching.

  (by our characterization of supporting matchings)
- So M has to match all stable vertices and $M \subseteq E_p$.

References

1. Á. Cseh and T. Kavitha.
   Popular edges and dominant matchings. Math. Programming, 2018.
2. Y. Faenza, T. Kavitha, V. Powers, and X. Zhang.
   Popular matchings and limits to tractability. In SODA 2019.
3. C.-C. Huang and T. Kavitha.
   Popularity, mixed matchings, and self-duality. Math. of Operations Research, 2021.
4. Y. Faenza and T. Kavitha.
   Quasi-popularity, optimality, and extended formulations. Math. of Operations Research, 2022.
5. T. Kavitha.
   Fairly popular matchings and optimality. In STACS 2022.

Thank you! Any questions?