Weighted Popular Matchings

Julián Mestre
University of Maryland

The setup

x_1

x_2

x_3

x_4

A

B

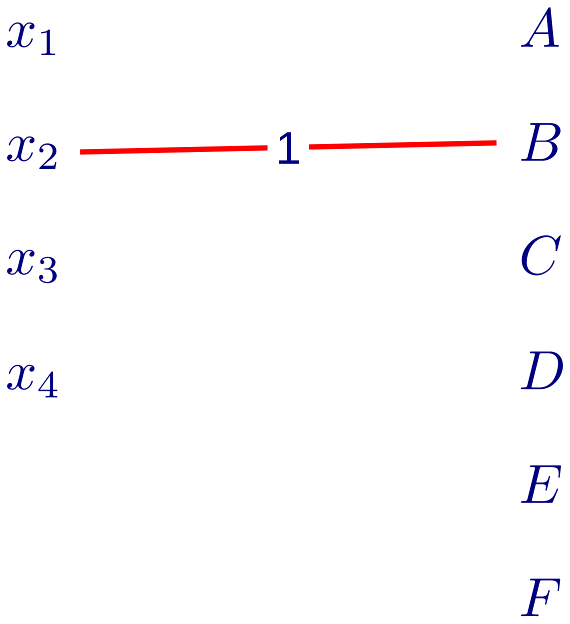
C

D

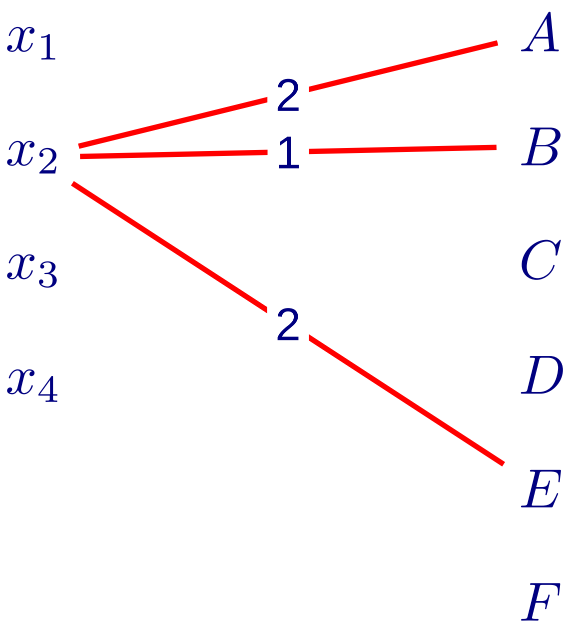
E

F

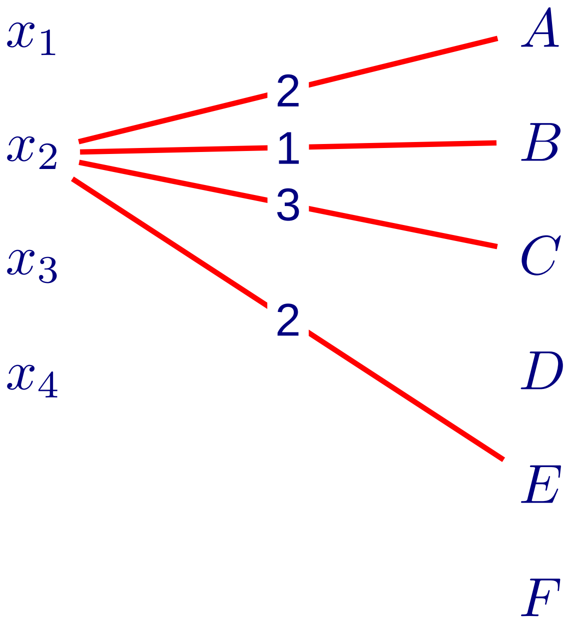
The setup



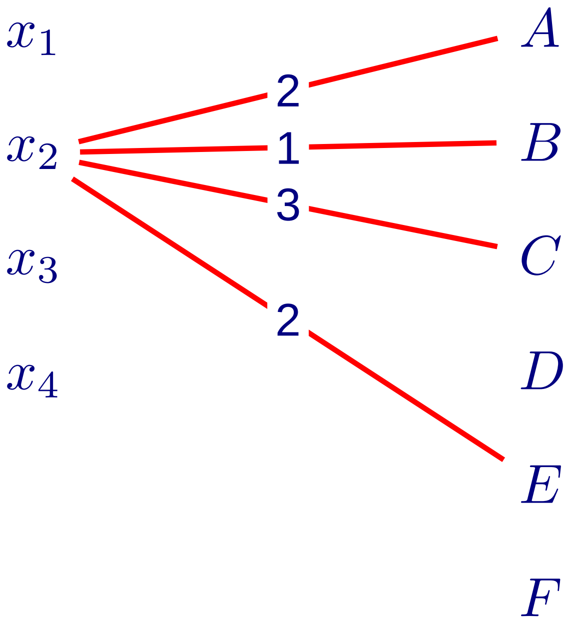
The setup



The setup



The setup



Objective: to find a “good” matching

Popular matchings

Applicants are presented two matchings: M and R.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

More applicants prefer M over R, we say M is *more popular than* R.

Popular matchings

Applicants are presented two matchings: M and R.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

More applicants prefer M over R, we say M is *more popular than* R.

Popular matchings

Applicants are presented two matchings: M and R.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

More applicants prefer M over R, we say M is *more popular than* R.

Popular matchings

Applicants are presented two matchings: M and R.

x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

More applicants prefer M over R, we say M is *more popular than* R.

Def: M is *popular* if $\nexists$ R more popular than M

What’s known?

Popular matchings may not always exist [G 75]

$$\begin{array}{c|ccc} x_1 & A & B & C \\ x_2 & A & B & C \\ x_3 & A & B & C \end{array}$$

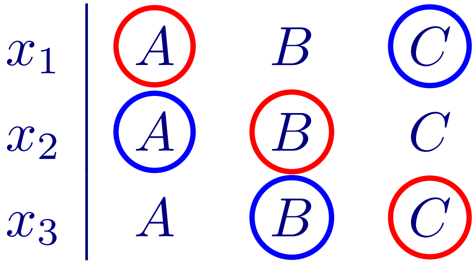
What's known?

Popular matchings may not always exist [G 75]

$$\begin{array}{c|ccc} x_1 & \enclose{circle}{A} & B & C \\ x_2 & A & \enclose{circle}{B} & C \\ x_3 & A & B & \enclose{circle}{C} \end{array}$$

What’s known?

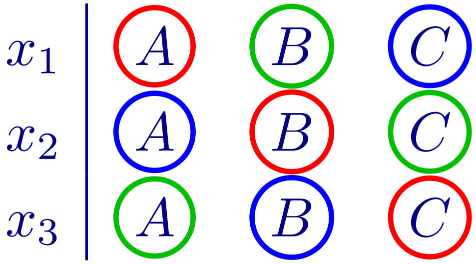
Popular matchings may not always exist [G 75]



x_1	A	B	C
x_2	A	B	C
x_3	A	B	C

What’s known?

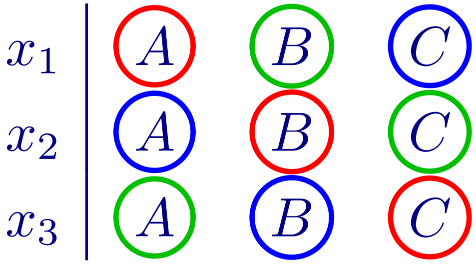
Popular matchings may not always exist [G 75]



x_1	A	B	C
x_2	A	B	C
x_3	A	B	C

What’s known?

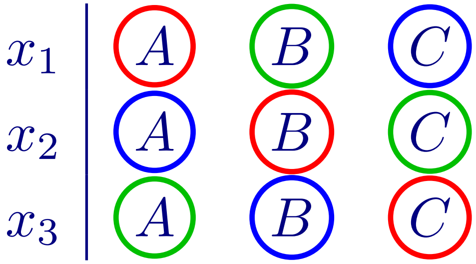
Popular matchings may not always exist [G 75]



But we can determine if one exists and if so produce one [AIKM 05]

What’s known?

Popular matchings may not always exist [G 75]



x_1	A	B	C
x_2	A	B	C
x_3	A	B	C

But we can determine if one exists and if so produce one [AIKM 05]

- for strict preferences in $O(n+m)$ time
- general case in $O(\sqrt{n}m)$ time

Some thoughts

- What do we do if there is no popular matching?
- What if there is a lot of contention for a few “good” jobs?
- We may want to give priorities to the applicants

Weighted popular matchings

Applicants are presented two matchings: M and R.

4	3	6	3	2	2	4
x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

M is more popular than R because:

weight of applicants who prefer M over R = 9 > 5 = weight of applicants who prefer R over M.

Weighted popular matchings

Applicants are presented two matchings: M and R.

4	3	6	3	2	2	4
x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

M is more popular than R because:

weight of applicants who prefer M over R = 9 > 5 = weight of applicants who prefer R over M.

Weighted popular matchings

Applicants are presented two matchings: M and R.

4	3	6	3	2	2	4
x_1	x_2	x_3	x_4	x_5	x_6	x_7
M	R	$-$	M	R	M	$-$

M is more popular than R because:

weight of applicants who prefer M over R = 9 > 5 = weight of applicants who prefer R over M.

High level idea

- Find alternative algorithmic-friendly characterization.
- M is popular $\Longrightarrow$ M is well-formed

  M is popular $\not\Longleftarrow$ M is well-formed
- M is popular $\Longleftrightarrow$ M is well-formed + some pruning

High level idea

- Find alternative algorithmic-friendly characterization.
- M is popular $\Longrightarrow$ M is well-formed

  M is popular $\not\Longleftarrow$ M is well-formed
- M is popular $\Longleftrightarrow$ M is well-formed + some pruning

High level idea

- Find alternative algorithmic-friendly characterization.
- M is popular $\Longrightarrow$ M is well-formed

  M is popular $\not\Longleftarrow$ M is well-formed
- M is popular $\Longleftrightarrow$ M is well-formed + some pruning

High level idea

- Find alternative algorithmic-friendly characterization.
- M is popular $\Longrightarrow$ M is well-formed

  M is popular $\not\Longleftarrow$ M is well-formed
- M is popular $\Longleftrightarrow$ M is well-formed + some pruning

Caveat: From now on we only consider strict preferences

Some definitions

- Every applicant x has weight $w(x)$.
- Partition applicants into $C_1, C_2, \ldots, C_k$ with $w_1 > w_2 \ldots > w_k$ such that $w(x) = w_i$ for all $x \in C_i$.

First and Second jobs

- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job

$w(x_1) = 7$	x_1	A	B	C	
$w(x_2) = 4$	x_2	A	C	D	
$w(x_3) = 2$	x_3	C	A	D	E
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job

$w(x_1) = 7$	x_1	$\textcircled{A}$	B	C	
$w(x_2) = 4$	x_2	A	C	D	
$w(x_3) = 2$	x_3	C	A	D	E
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job

$$\begin{array}{lc|cccc}
w(x_1) = 7 & x_1 & \textcircled{A} & \textcircled{B} & C & \\
w(x_2) = 4 & x_2 & A & C & D & \\
w(x_3) = 2 & x_3 & C & A & D & E \\
w(x_4) = 2 & x_4 & A & D & C & E
\end{array}$$

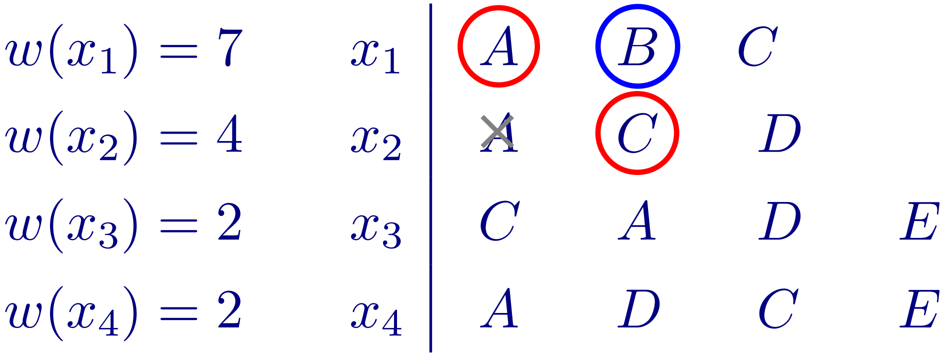
First and Second jobs

- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job

$w(x_1) = 7$	x_1	$\enclose{circle}{A}$	$\enclose{circle}{B}$	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	C	D	
$w(x_3) = 2$	x_3	C	A	D	E
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

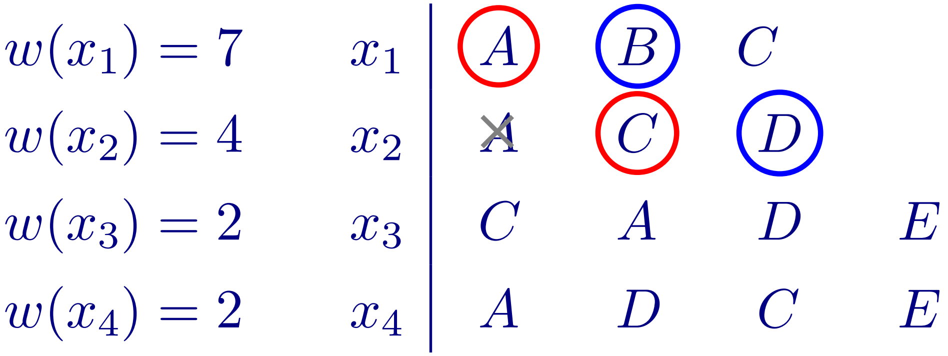
- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job



$w(x_1) = 7$	x_1	$\textcircled{A}$	$\textcircled{B}$	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	$\textcircled{C}$	D	
$w(x_3) = 2$	x_3	C	A	D	E
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

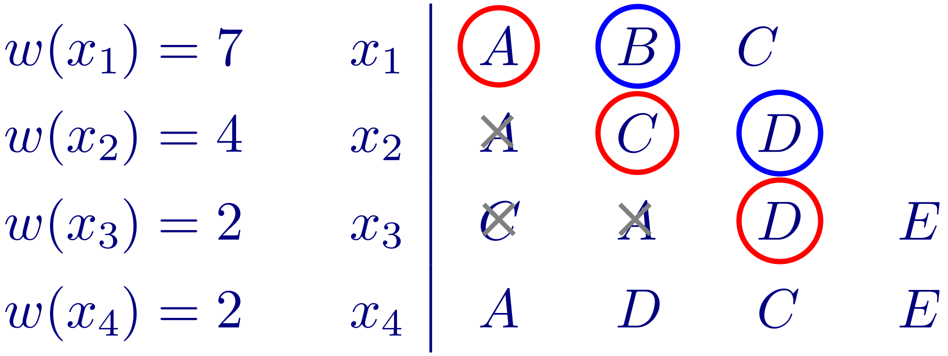
- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job



$w(x_1) = 7$	x_1	$\textcircled{A}$	$\textcircled{B}$	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	$\textcircled{C}$	$\textcircled{D}$	
$w(x_3) = 2$	x_3	C	A	D	E
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

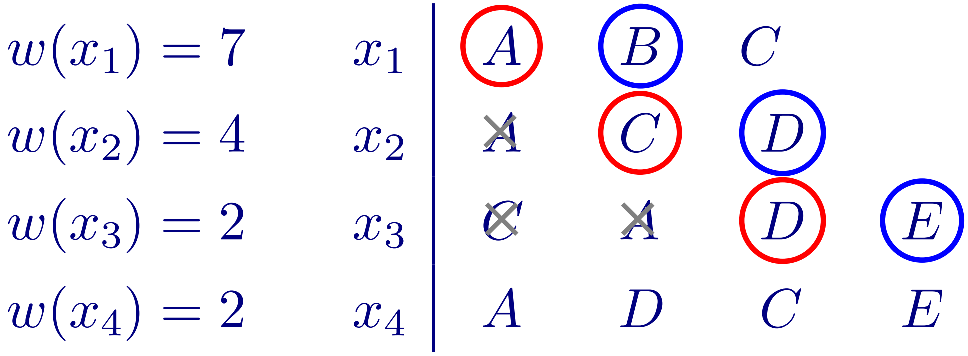
- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job



$w(x_1) = 7$	x_1	A	B	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	C	D	
$w(x_3) = 2$	x_3	$\cancel{C}$	$\cancel{A}$	D	E
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

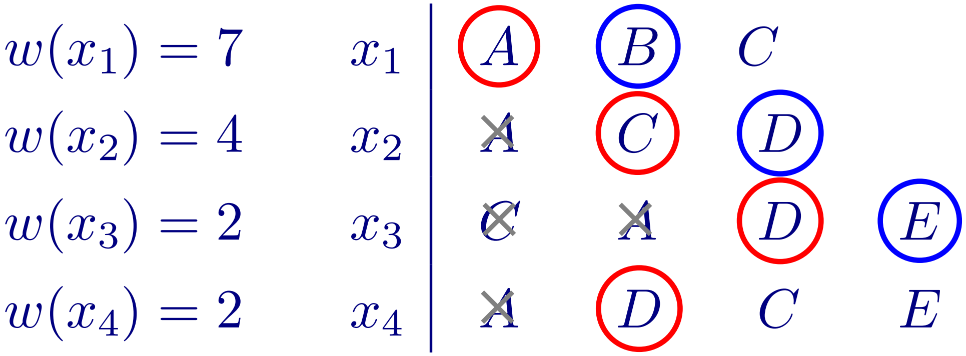
- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job



$w(x_1) = 7$	x_1	Ⓐ	Ⓑ	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	Ⓒ	Ⓓ	
$w(x_3) = 2$	x_3	$\cancel{C}$	$\cancel{A}$	Ⓓ	Ⓔ
$w(x_4) = 2$	x_4	A	D	C	E

First and Second jobs

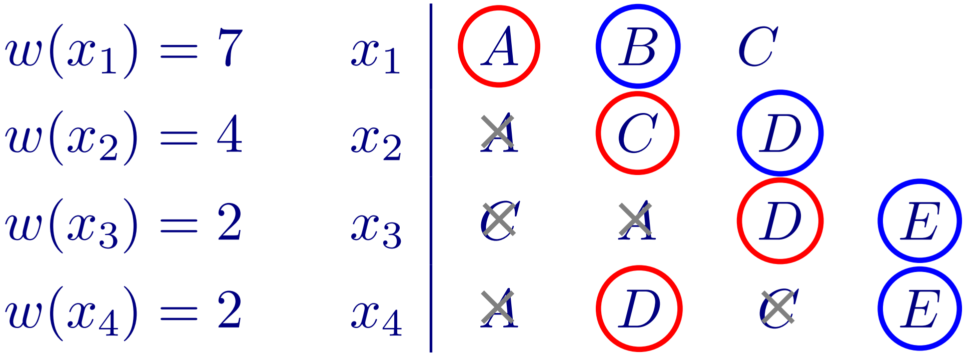
- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job



$w(x_1) = 7$	x_1	A	B	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	C	D	
$w(x_3) = 2$	x_3	$\cancel{C}$	$\cancel{A}$	D	E
$w(x_4) = 2$	x_4	$\cancel{A}$	D	C	E

First and Second jobs

- For $x \in C_1$ let $f(x)$ the first job on x's list, this is an f_1-job
- For $x \in C_i$ let $s(x)$ the first non $f_{\leq i}$-job.
- Fox $x \in C_i$ let $f(x)$ the first non $f_{<i}$-job, this is an f_i-job



$w(x_1) = 7$	x_1	A	B	C	
$w(x_2) = 4$	x_2	$\cancel{A}$	C	D	
$w(x_3) = 2$	x_3	$\cancel{C}$	$\cancel{A}$	D	E
$w(x_4) = 2$	x_4	$\cancel{A}$	D	$\cancel{C}$	E

Well-formed matchings

Def.: M is well-formed if:

(i) Every f_i-job j is matched to $x \in C_i$ and $f(x) = j$.

(ii) Every applicant x is matched to $f(x)$ or $s(x)$

Well-formed matchings

Def.: M is well-formed if:

(i) Every f_i-job j is matched to $x \in C_i$ and $f(x) = j$.

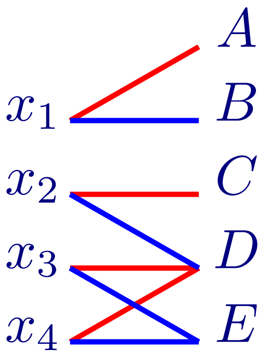
(ii) Every applicant x is matched to $f(x)$ or $s(x)$

Theorem 1. *If M is popular then M is well-formed*

Example: well-formed matching

$$w(x_1) = 7$$
$$w(x_2) = 4$$
$$w(x_3) = 2$$
$$w(x_4) = 2$$

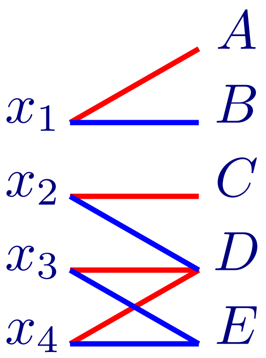
x_1	A	B	C	
x_2	A	C	D	
x_3	C	A	D	E
x_4	A	D	C	E

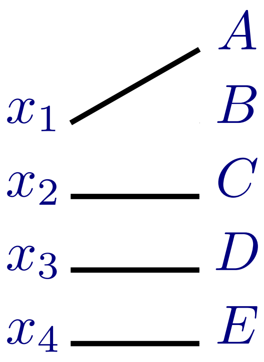


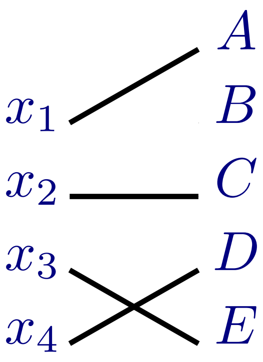
Example: well-formed matching

$$w(x_1) = 7$$
$$w(x_2) = 4$$
$$w(x_3) = 2$$
$$w(x_4) = 2$$

x_1	A	B	C	
x_2	A	C	D	
x_3	C	A	D	E
x_4	A	D	C	E







Promotion path

Def. A promotion path in a well-formed matching M is a sequence $p_0, x_0, p_1, x_1, \ldots, p_s, x_s$. Such that for all i

(i) $(x_i, p_i) \in M$ and $f(x_i) = p_i$.

(ii) x_i likes p_{i+1} better than p_i.

The *cost* of the path is defined as $w(x_s) - w(x_0) - \ldots - w(x_{s-1})$.

Promotion path

Def. A promotion path in a well-formed matching M is a sequence $p_0, x_0, p_1, x_1, \ldots, p_s, x_s$. Such that for all i

(i) $(x_i, p_i) \in M$ and $f(x_i) = p_i$.

(ii) x_i likes p_{i+1} better than p_i.

The *cost* of the path is defined as $w(x_s) - w(x_0) - \ldots - w(x_{s-1})$.

Intuition: can use the path to free p_0, but must “pay” the cost

Pruning and Labeling

Start with $G' = (V, E')$ where $(x, f(x)), (x, s(x)) \in E'$.

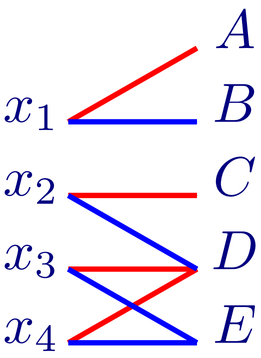
Prune edges and assign a label $\lambda(p)$ to every f_i-job p such that:

- In *every* well-formed matching M *included in the pruned graph* the minimum cost promotion path out of p has cost exactly $\lambda(p)$
- No pruned edge can be part of a popular matching.

Example: pruning and labeling

$$\begin{aligned} w(x_1) &= 7 \\ w(x_2) &= 4 \\ w(x_3) &= 2 \\ w(x_4) &= 2 \end{aligned}$$

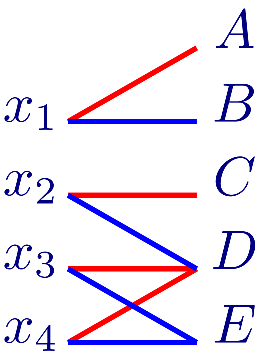
x_1	A	B	C	
x_2	A	C	D	
x_3	C	A	D	E
x_4	A	D	C	E



Example: pruning and labeling

$w(x_1) = 7$
$w(x_2) = 4$
$w(x_3) = 2$
$w(x_4) = 2$

x_1	A	B	C	
x_2	A	C	D	
x_3	C	A	D	E
x_4	A	D	C	E

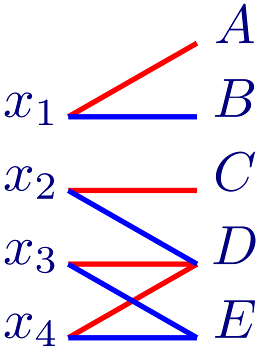


$\lambda(A) = 7$

Example: pruning and labeling

$$\begin{aligned} w(x_1) &= 7 \\ w(x_2) &= 4 \\ w(x_3) &= 2 \\ w(x_4) &= 2 \end{aligned}$$

$$\begin{array}{c|cccc} x_1 & A & B & C & \\ x_2 & A & C & D & \\ x_3 & C & A & D & E \\ x_4 & A & D & C & E \end{array}$$



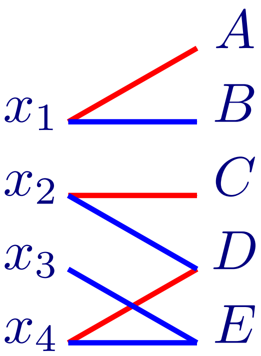
$$\lambda(A) = 7$$

$$\lambda(C) = 3$$

Example: pruning and labeling

$w(x_1) = 7$
$w(x_2) = 4$
$w(x_3) = 2$
$w(x_4) = 2$

x_1	A	B	C	
x_2	A	C	D	
x_3	C	A	D	E
x_4	A	D	C	E



$\lambda(A) = 7$

$\lambda(C) = 3$
$\lambda(D) = 2$

Back to popular

Theorem 2. *Well-formed matchings in the pruned graph are popular.*

- Let M be a well-formed matching and M' some matching.
- Let y be an applicant who likes M' better than M, construct promotion path out of $M'(y)$.
- Build set of maximal (inclusion-wise) paths

Back to popular

Theorem 2. *Well-formed matchings in the pruned graph are popular.*

- Let M be a well-formed matching and M' some matching.
- Let y be an applicant who likes M' better than M, construct promotion path out of $M'(y)$.
- Build set of maximal (inclusion-wise) paths

Back to popular

Theorem 2. *Well-formed matchings in the pruned graph are popular.*

- Let M be a well-formed matching and M' some matching.
- Let y be an applicant who likes M' better than M, construct promotion path out of $M'(y)$.
- Build set of maximal (inclusion-wise) paths

Back to popular

Theorem 2. *Well-formed matchings in the pruned graph are popular.*

- Let M be a well-formed matching and M' some matching.
- Let y be an applicant who likes M' better than M, construct promotion path out of $M'(y)$.
- Build set of maximal (inclusion-wise) paths

Preference lists with ties

We need to revise the definitions:

- For every $x \in C_i$, define $f(x)$ as highest ranked non-$f_{<i}$-job.
- What's an f_i-jobs? Critical to get a maximum matching.
- Well-formed.
- Promotion path.

To Recap

Develop an alternative algorithmic-friendly characterization.

- M is popular $\Longrightarrow$ M is well-formed.
- M is popular $\not\Longleftarrow$ M is well-formed.
- Prune the graph guided by labels and promotion paths.
- M is popular $\Longleftrightarrow$ M is well-formed in the pruned graph.

To Recap

Develop an alternative algorithmic-friendly characterization.

- M is popular $\Longrightarrow$ M is well-formed.
- M is popular $\not\Longleftarrow$ M is well-formed.
- Prune the graph guided by labels and promotion paths.
- M is popular $\Longleftrightarrow$ M is well-formed in the pruned graph.

We can find a popular matching, or determine that none exists:

- in $O(n+m)$ time for strict preference lists
- in $O(k\sqrt{n}m)$ time in the general case

Open problems

- Different definitions of popular matching
- Almost popular matchings

Thanks for your attention!

... and for staying until the end!