Disjoint Stable Matchings in Linear Time

Aaditya Ganesh[1], Vishwa Prakash[1], Prajakta Nimbhorkar[1,2], Geevarghese Philip[1,2]

WG 2021

[1]Chennai Mathematical Institute

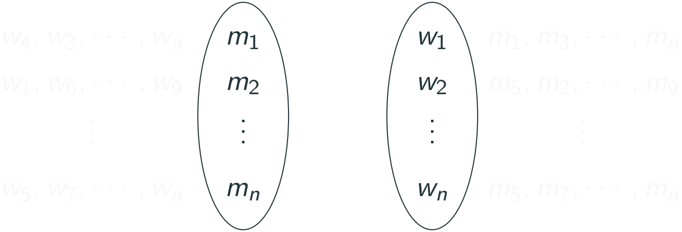
[2]UMI ReLaX

The Marriage Model

In the marriage model, we are given with a bipartite graph $G = (A \cup B, E)$, and for each $v \in A \cup B$ a strict ordering $\succ_v$ of its neighbours - given in it's preference list.

A: Set of menB: Set of women

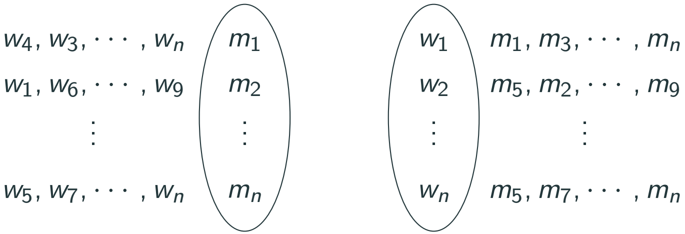
In the marriage model, we are given with a bipartite graph $G = (A \cup B, E)$, and for each $v \in A \cup B$ a strict ordering $\succ_v$ of its neighbours - given in it's preference list.



A: Set of menB: Set of women

The Marriage Model

In the marriage model, we are given with a bipartite graph $G = (A \cup B, E)$, and for each $v \in A \cup B$ a strict ordering $\succ_v$ of its neighbours - given in it's preference list.



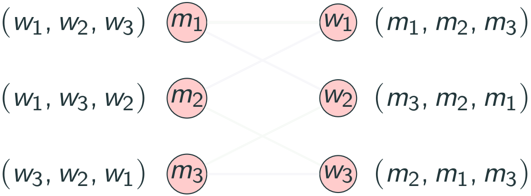
A: Set of men B: Set of women

Stability

A matching M is said to be stable if there is no edge $(m, w) \in E \backslash M$ such that:

$$w \succ_m M(m) \quad \text{and} \quad m \succ_w M(w)$$

That is, m and w prefer each other over their respective partners in M.



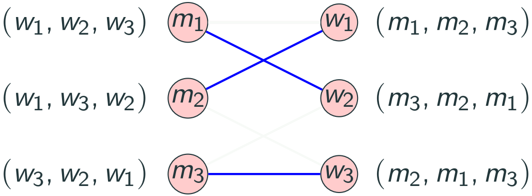
A stable matching always exists (Gale and Shapley, 1962) and can be found in linear time.

Stability

A matching M is said to be stable if there is no edge $(m, w) \in E \setminus M$ such that:

$$w \succ_m M(m) \quad \text{and} \quad m \succ_w M(w)$$

That is, m and w prefer each other over their respective partners in M.



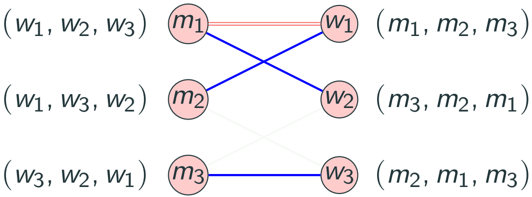
A stable matching always exists (Gale and Shapley, 1962) and can be found in linear time.

Stability

A matching M is said to be stable if there is no edge $(m, w) \in E \setminus M$ such that:

$$w \succ_m M(m) \quad \text{and} \quad m \succ_w M(w)$$

That is, m and w prefer each other over their respective partners in M.



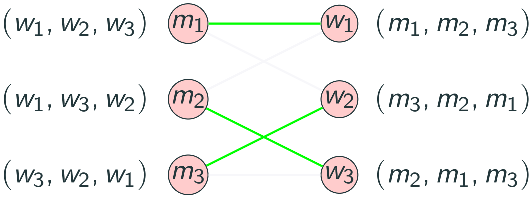
A stable matching always exists (Gale and Shapley, 1962) and can be found in linear time.

Stability

A matching M is said to be stable if there is no edge $(m, w) \in E \backslash M$ such that:

$$w \succ_m M(m) \quad \text{and} \quad m \succ_w M(w)$$

That is, m and w prefer each other over their respective partners in M.



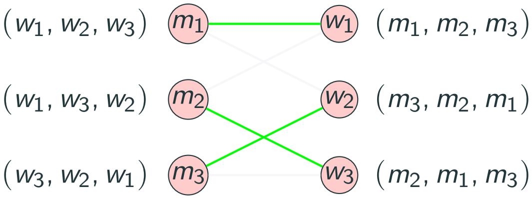
A stable matching always exists (Gale and Shapley, 1962) and can be found in linear time.

Stability

A matching M is said to be stable if there is no edge $(m, w) \in E \setminus M$ such that:

$$w \succ_m M(m) \quad \text{and} \quad m \succ_w M(w)$$

That is, m and w prefer each other over their respective partners in M.



A stable matching always exists (Gale and Shapley, 1962) and can be found in linear time.

Gale and Shapley Algorithm

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *"Man-optimal"* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *"Woman-optimal"* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Gale and Shapley Algorithm

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *"Man-optimal"* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *"Woman-optimal"* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Gale and Shapley Algorithm

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *"Man-optimal"* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *"Woman-optimal"* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Gale and Shapley Algorithm

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *"Man-optimal"* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *"Woman-optimal"* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *"Man-optimal"* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *"Woman-optimal"* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Gale and Shapley Algorithm

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *"Man-optimal"* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *"Woman-optimal"* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Gale and Shapley Algorithm

Unmatched men propose. Women accept or reject based on their preference list.

Key Results:

1. All possible execution of the Gale-Shapley algorithm yields the same result.
2. It results in *“Man-optimal”* stable matching.

   Man-optimal: Every man is matched with his most favored partner among all stable partners.
3. Reversing roles, i.e, women proposing, results in *“Woman-optimal”* stable matching.

   Woman-optimal: Every woman is matched with her most favored partner among all stable partners.
4. The man-optimal stable matching is *woman-pessimal*, and vice-versa.

Extended Gale-Shapley Algorithm

Extended Gale-Shapley(EGS) algorithm is very similar to the Gale-Shapley algorithm except - EGS modifies the input preference list.

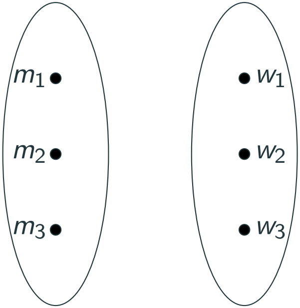
Run of Extended GS Algorithm

m_1 :	w_2	w_3	w_1
m_2 :	w_3	w_1	w_2
m_3 :	w_2	w_1	w_3

Men's Preference

w_1 :	m_1	m_2	m_3
w_2 :	m_2	m_1	m_3
w_3 :	m_3	m_2	m_1

Women's Preference



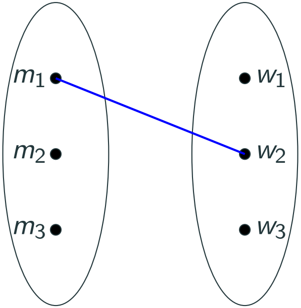
Run of Extended GS Algorithm

m_1 :	w_2	w_3	w_1
m_2 :	w_3	w_1	w_2
m_3 :	w_2	w_1	w_3

Men's Preference

w_1 :	m_1	m_2	m_3
w_2 :	m_2	m_1	m_3
w_3 :	m_3	m_2	m_1

Women's Preference



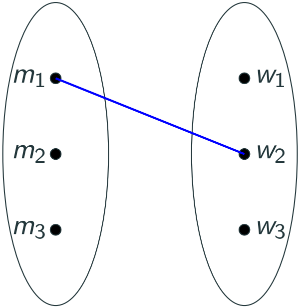
Run of Extended GS Algorithm

m_1 :	w_2	w_3	w_1
m_2 :	w_3	w_1	w_2
m_3 :		w_1	w_3

Men's Preference

w_1 :	m_1	m_2	m_3
w_2 :	m_2	m_1	
w_3 :	m_3	m_2	m_1

Women's Preference



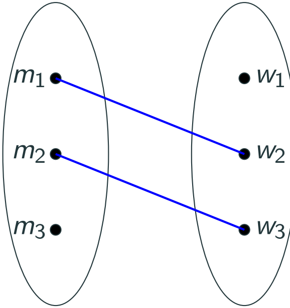
Run of Extended GS Algorithm

m_1 :	w_2	w_3	w_1
m_2 :	w_3	w_1	w_2
m_3 :		w_1	w_3

Men's Preference

w_1 :	m_1	m_2	m_3
w_2 :	m_2	m_1	
w_3 :	m_3	m_2	m_1

Women's Preference



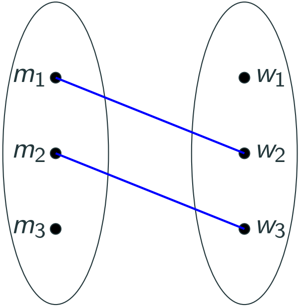
Run of Extended GS Algorithm

m_1 :	w_2		w_1
m_2 :	w_3	w_1	w_2
m_3 :		w_1	w_3

Men's Preference

w_1 :	m_1	m_2	m_3
w_2 :	m_2	m_1	
w_3 :	m_3	m_2	

Women's Preference



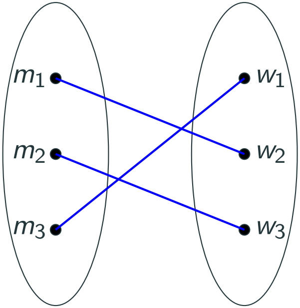
Run of Extended GS Algorithm

m_1 :	w_2		w_1
m_2 :	w_3	w_1	w_2
m_3 :		w_1	w_3

Men's Preference

w_1 :	m_1	m_2	m_3
w_2 :	m_2	m_1	
w_3 :	m_3	m_2	

Women's Preference



A person x is said to *prefer* a matching M to a matching M' if x prefers $p_M(x)$ to $p_{M'}(x)$.

Domination

A stable matching M is said to *dominate* a stable matching M', written $M \preceq M'$, if every man has at least as good a partner in M as he has in M' i.e., every man either prefers M to M' or is indifferent between them. *M strictly dominates M'* ($M \prec M'$) if $M \preceq M'$ and $M \cap M' = \varnothing$.

m_1 :				
m_2 :				
m_3 :				
m_4 :				

Men's Preference

A person x is said to *prefer* a matching M to a matching M' if x prefers $p_M(x)$ to $p_{M'}(x)$.

Domination

A stable matching M is said to *dominate* a stable matching M', written $M \preceq M'$, if every man has at least as good a partner in M as he has in M'.i.e., every man either prefers M to M' or is indifferent between them. M *strictly dominates* $M'(M \prec M')$ if $M \preceq M'$ and $M \cap M' = \varnothing$.

m_1 :				
m_2 :				
m_3 :				
m_4 :				

Men's Preference

The Lattice Structure

A person x is said to *prefer* a matching M to a matching M' if x prefers $p_M(x)$ to $p_{M'}(x)$.

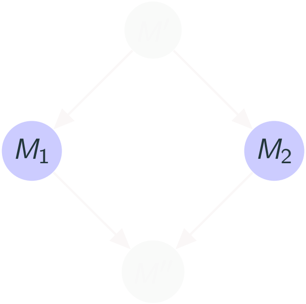
Domination

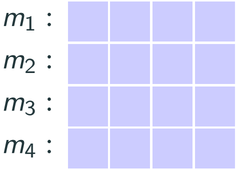
A stable matching M is said to *dominate* a stable matching M', written $M \preceq M'$, if every man has at least as good a partner in M as he has in M'.i.e., every man either prefers M to M' or is indifferent between them. *M strictly dominates M'* $(M \prec M')$ if $M \preceq M'$ and $M \cap M' = \varnothing$.

m_1 :				
m_2 :				
m_3 :				
m_4 :				

Men's Preference

Meet and Join



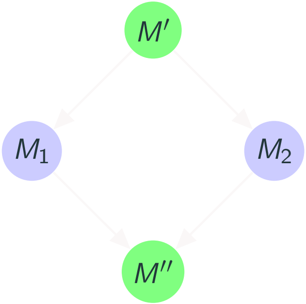


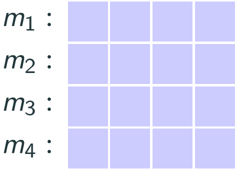
Men's Preference

$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$

$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

Meet and Join



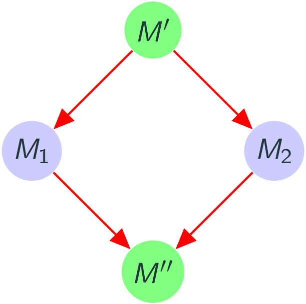


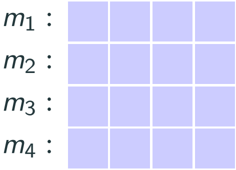
Men's Preference

$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$

$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

Meet and Join



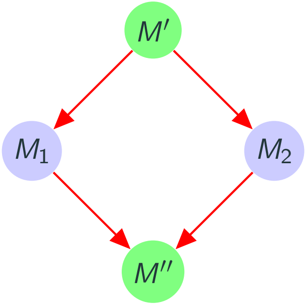


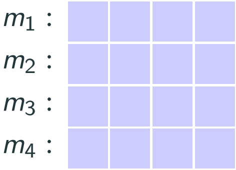
Men's Preference

$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$

$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

Meet and Join





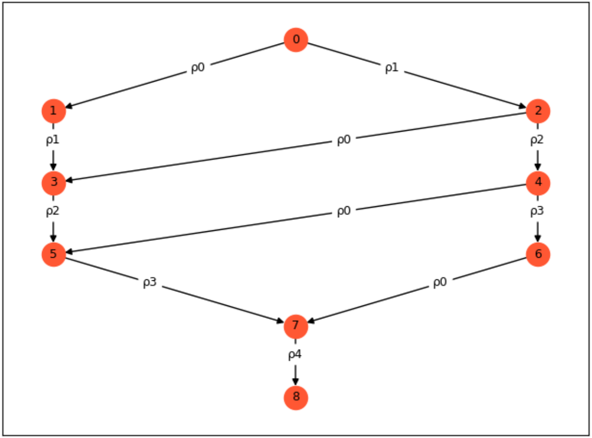
Men's Preference

$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$

$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

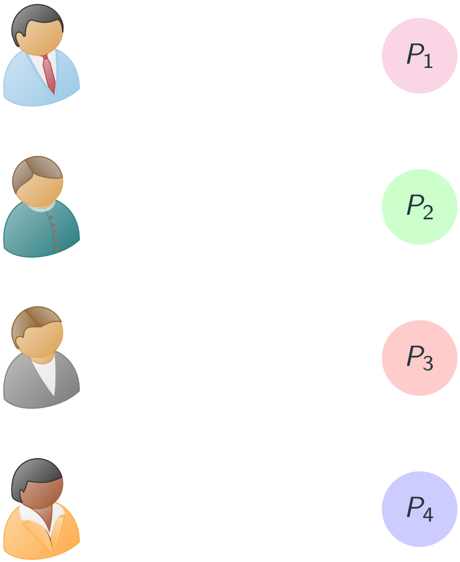
The Lattice Structure

Set of all stable matchings form a distributive lattice under the *Domination* domination.

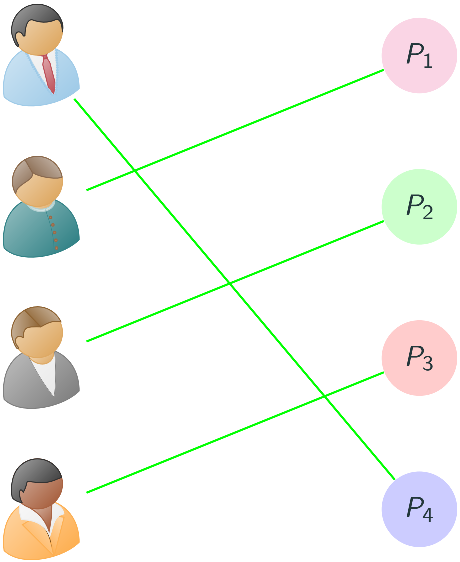


Disjoint Stable Matchings

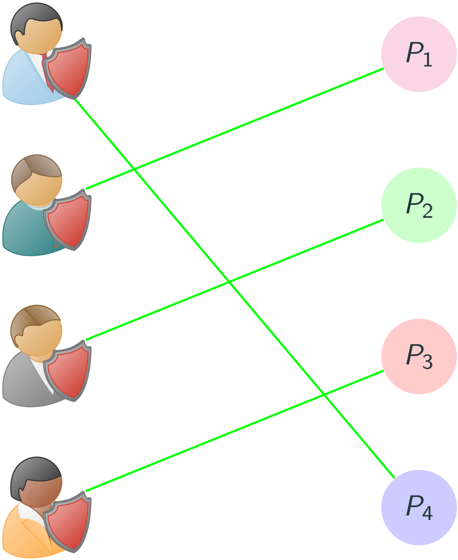
Why do we need them?



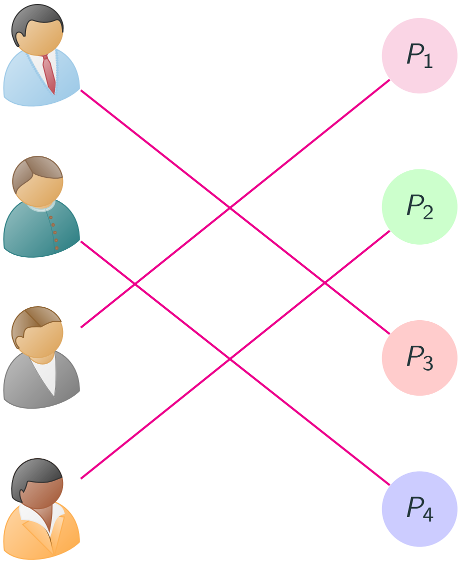
Why do we need them?



Why do we need them?



Why do we need them?



Problem Statement

For a given marriage instance, find a largest set S of disjoint stable matchings.

Existence of Disjoint Stable Matchings

Does there exist a marriage matching instances with disjoint stable matchings?

$m_1 : w_1, w_2, w_3$	$w_1 : m_2, m_3, m_1$
$m_2 : w_2, w_3, w_1$	$w_2 : m_3, m_1, m_2$
$m_3 : w_3, w_1, w_2$	$w_3 : m_1, m_2, m_3$

Existence of Disjoint Stable Matchings

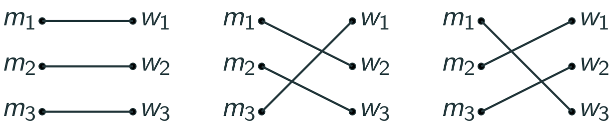
Does there exist a marriage matching instances with disjoint stable matchings?

$$
\begin{array}{ll}
m_1 : w_1, w_2, w_3 & \quad w_1 : m_2, m_3, m_1 \\
m_2 : w_2, w_3, w_1 & \quad w_2 : m_3, m_1, m_2 \\
m_3 : w_3, w_1, w_2 & \quad w_3 : m_1, m_2, m_3
\end{array}
$$

Existence of Disjoint Stable Matchings

Does there exist a marriage matching instances with disjoint stable matchings?

$$
\begin{array}{ll}
m_1 : w_1, w_2, w_3 & w_1 : m_2, m_3, m_1 \\
m_2 : w_2, w_3, w_1 & w_2 : m_3, m_1, m_2 \\
m_3 : w_3, w_1, w_2 & w_3 : m_1, m_2, m_3
\end{array}
$$



Necessary Condition

If the man-optimal and the woman-optimal stable matchings share a common edge (m, w), then (m, w) is in every stable matching.

This is because w is both the best stable partner and the worst stable partner of m.

So, to have disjoint stable matchings, man-optimal and woman-optimal matchings must be disjoint.

Algorithm: Disjoint Stable Matchings

- Input: Marriage instance G, Empty set S.
- $X \leftarrow \textsc{Extended}\mathrm{GS}(G)$
- While $X \cap M_Z = \varnothing$
  - $S \leftarrow S \cup X$
  - Delete X from G
  - $X \leftarrow \textsc{Extended}\mathrm{GS}(G)$

Algorithm: Disjoint Stable Matchings

- Input: Marriage instance G, Empty set S.
- $X \leftarrow \textsc{Extended}\mathrm{GS}(G)$
- While $X \cap M_Z = \varnothing$
    - $S \leftarrow S \cup X$
    - Delete X from G
    - $X \leftarrow \textsc{Extended}\mathrm{GS}(G)$

Algorithm: Disjoint Stable Matchings

- Input: Marriage instance G, Empty set S.
- $X \leftarrow \textsc{ExtendedGS}(G)$
- While $X \cap M_Z = \varnothing$
  - $S \leftarrow S \cup X$
  - Delete X from G
  - $X \leftarrow \textsc{ExtendedGS}(G)$

Algorithm: Disjoint Stable Matchings

- Input: Marriage instance G, Empty set S.
- $X \leftarrow \textsc{Extended}\mathrm{GS}(G)$
- While $X \cap M_Z = \varnothing$
  - $S \leftarrow S \cup X$
  - Delete X from G
  - $X \leftarrow \textsc{Extended}\mathrm{GS}(G)$

Algorithm: Disjoint Stable Matchings

- Input: Marriage instance G, Empty set S.
- $X \leftarrow \textsc{ExtendedGS}(G)$
- While $X \cap M_Z = \varnothing$
  - $S \leftarrow S \cup X$
  - Delete X from G
  - $X \leftarrow \textsc{ExtendedGS}(G)$

Algorithm: Disjoint Stable Matchings

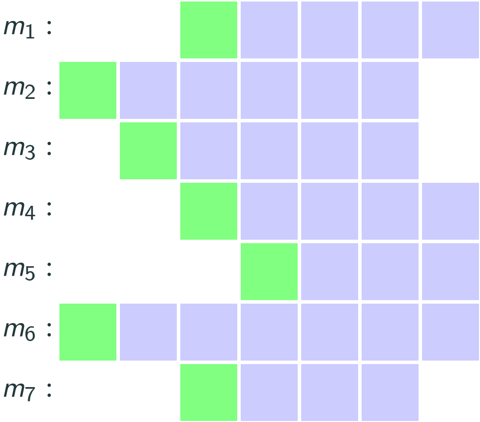
- Input: Marriage instance G, Empty set S.
- $X \leftarrow \textsc{ExtendedGS}(G)$
- While $X \cap M_Z = \varnothing$
  - $S \leftarrow S \cup X$
  - Delete X from G
  - $X \leftarrow \textsc{ExtendedGS}(G)$

Run of Disjoint GS Algorithm

m_1 :							
m_2 :							
m_3 :							
m_4 :							
m_5 :							
m_6 :							
m_7 :							

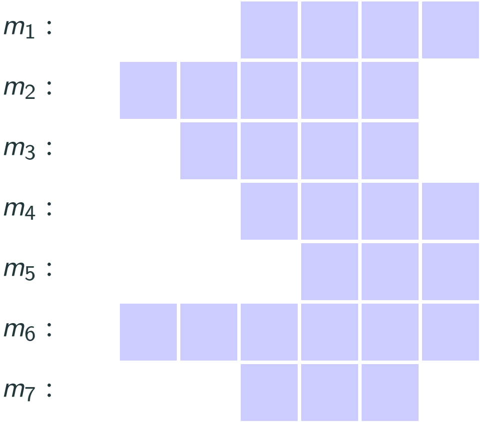
Men's preference list

Run of Disjoint GS Algorithm



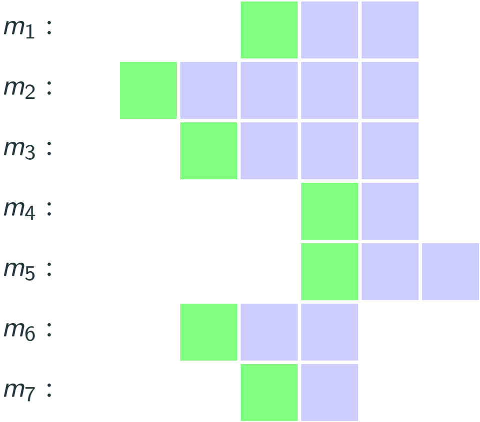
Men's Preference

Run of Disjoint GS Algorithm



Men's Preference

Run of Disjoint GS Algorithm



Men's Preference

Termination and Time Complexity

In every iteration, we delete at least one entry from the preference list. As the size of preference list is $2n^2$, the algorithm **terminates**.

For the same reason, the running time of the algorithm is $\mathbf{O}(\mathbf{n^2})$.

Disjoint Stable Matchings

Lemma 1

Each M_i in the set $S = \{M_0, M_1, \cdots, M_k\}$ is a perfect matching.

Note: It does not come freely from Extended GS!

It only guarantees one-one.

Disjoint Stable Matchings

Lemma 1

Each M_i in the set $S = \{M_0, M_1, \cdots, M_k\}$ is a perfect matching.

Note: It does not come freely from Extended GS!

It only guarantees one-one.

Disjoint Stable Matchings

Lemma 2

All the matchings in the set S are stable matchings.

Lemma 3

If $M_0, M_1, \cdots, M_k$ are the matchings discovered by the algorithm in this order, then $M_0 \prec M_1 \prec \cdots \prec M_k$.

Disjoint Stable Matchings

Lemma 2

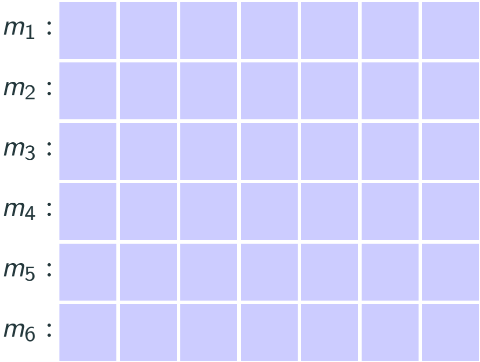
All the matchings in the set S are stable matchings.

Lemma 3

If $M_0, M_1, \cdots, M_k$ are the matchings discovered by the algorithm in this order, then $M_0 \prec M_1 \prec \cdots \prec M_k$.

Lemma 4

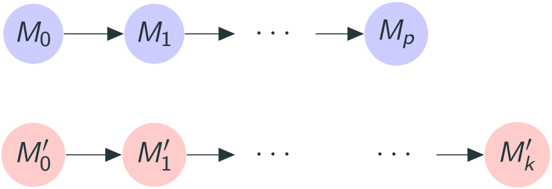
In any arbitrary execution E of the algorithm, for any man m, $p_{M_i}(m)$ is the best stable partner of m when, for **every** *man, stable partners from $M_0, M_1, \cdots, M_{i-1}$ are disallowed.*



Lemma 5

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

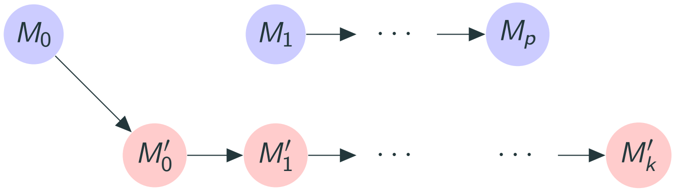


Longest Chain of Disjoint Stable matchings

Lemma 5

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

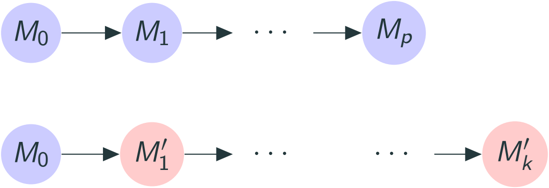


Longest Chain of Disjoint Stable matchings

Lemma 5

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

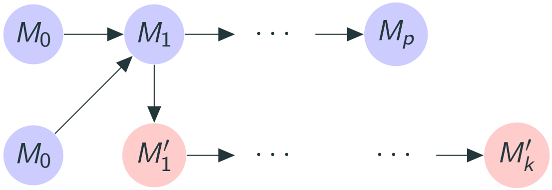


Longest Chain of Disjoint Stable matchings

Lemma 5

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

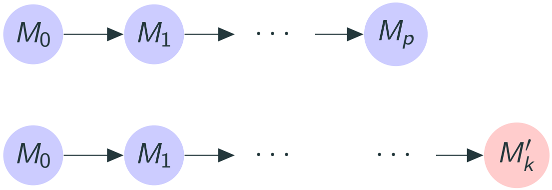


Longest Chain of Disjoint Stable matchings

Lemma 5

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

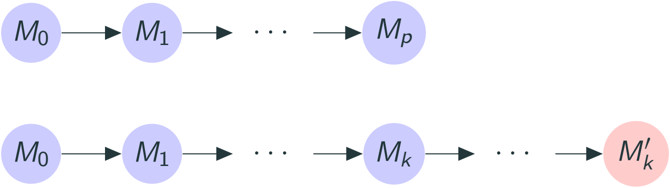


Longest Chain of Disjoint Stable matchings

Lemma 5

The algorithm gives the longest chain of disjoint stable matchings.

Proof:



Disjoint Stable Matching

Theorem 6 (Teo, C.-P. and Sethuraman, J. (1998))

Let $S = \{M_1, M_2, \cdots, M_k\}$ *be a set of stable matchings for a particular stable matchings instance. For each man* m*, let* S_m *be the sorted multiset* $\{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}$*, sorted according to the preference order of* m*. For every* $i \in \{1, 2, \cdots, k\}$ *let* $M_i' = \{(m, w) | m \in \mathcal{M}$ *and* w *is the* i^{th} *woman in* $S_m\}$*. Then for each* $i \in \{1, 2, \cdots, k\}$*,* M_i' *is a stable matching.*

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M_i' = \{ (m, w) \mid w \text{ is the i-th women in the } \textbf{sorted multiset} \ \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\} \}$$

Disjoint Stable Matching

Theorem 6 (Teo, C.-P. and Sethuraman, J. (1998))

Let $S = \{M_1, M_2, \cdots, M_k\}$ be a set of stable matchings for a particular stable matchings instance. For each man m, let S_m be the sorted multiset $\{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}$, sorted according to the preference order of m. For every $i \in \{1, 2, \cdots, k\}$ let $M_i' = \{(m, w) | m \in \mathcal{M}$ and w is the i^{th} woman in $S_m\}$. Then for each $i \in \{1, 2, \cdots, k\}$, M_i' is a stable matching.

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M_i' = \{(m, w) \mid w \text{ is the i-th women in the } \textbf{sorted multiset } \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}\}$$

Disjoint Stable Matching

Theorem 6 (Teo, C.-P. and Sethuraman, J. (1998))

Let $S = \{M_1, M_2, \cdots, M_k\}$ be a set of stable matchings for a particular stable matchings instance. For each man m, let S_m be the sorted multiset $\{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}$, sorted according to the preference order of m. For every $i \in \{1, 2, \cdots, k\}$ let $M'_i = \{(m, w) | m \in \mathcal{M}$ and w is the i^{th} woman in $S_m\}$. Then for each $i \in \{1, 2, \cdots, k\}$, M'_i is a stable matching.

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M'_i = \{(m, w) \mid w \text{ is the i-th women in the } \textbf{sorted multiset } \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}\}$$

Disjoint Chain

Corollary 7

Let $M_1, \ldots, M_k$ and $M'_1, \ldots, M'_k$ be as defined in 6. If $M_1, \ldots, M_k$ are pairwise disjoint, then $M'_1, \ldots, M'_k$ form a k-length chain of disjoint stable matchings.

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M'_i = \{(m, w) \mid w \text{ is the i-th women in the } \textbf{sorted set} \; \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}\}$$

Disjoint Chain

Corollary 7

Let $M_1, \ldots, M_k$ and $M'_1, \ldots, M'_k$ be as defined in 6. If $M_1, \ldots, M_k$ are pairwise disjoint, then $M'_1, \ldots, M'_k$ form a k-length chain of disjoint stable matchings.

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M'_i = \{(m, w) \mid w \text{ is the i-th women in the } \textbf{sorted set} \; \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}\}$$

Disjoint Chain

Corollary 7

Let $M_1, \ldots, M_k$ and $M'_1, \ldots, M'_k$ be as defined in 6. If $M_1, \ldots, M_k$ are pairwise disjoint, then $M'_1, \ldots, M'_k$ form a k-length chain of disjoint stable matchings.

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M'_i = \{(m, w) \mid w \text{ is the i-th women in the } \textbf{sorted set} \; \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}\,\}$$

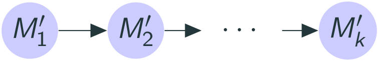
Disjoint Chain

Corollary 7

Let $M_1, \ldots, M_k$ and $M'_1, \ldots, M'_k$ be as defined in 6. If $M_1, \ldots, M_k$ are pairwise disjoint, then $M'_1, \ldots, M'_k$ form a k-length chain of disjoint stable matchings.

Given stable matchings $M_1, M_2, \cdots, M_k$,

$$M'_i = \{(m, w) \mid w \text{ is the i-th women in the } \textbf{sorted set} \; \{p_{M_1}(m), p_{M_2}(m), \cdots, p_{M_k}(m)\}\}$$



Maximum Size Set of Disjoint Stable Matchings

Theorem 8

For a given stable marriage instance, the algorithm gives the maximum size set of disjoint stable matchings.

Enumeration

- Our algorithm gives one of the largest sets of disjoint stable matching.
- Are there multiple solutions to the problem?

Yes!

- Our algorithm gives one of the largest sets of disjoint stable matching.
- Are there multiple solutions to the problem?

Yes!

- Our algorithm gives one of the largest sets of disjoint stable matching.
- Are there multiple solutions to the problem?

Yes!

(w_4, w_1, w_3, w_2) m_1 — w_1 (m_2, m_1, m_3, m_4)

(w_4, w_2, w_3, w_1) m_2 — w_2 (m_1, m_3, m_2, m_4)

(w_1, w_3, w_2, w_4) m_3 — w_3 (m_4, m_2, m_3, m_1)

(w_1, w_4, w_2, w_3) m_4 — w_4 (m_3, m_4, m_2, m_1)

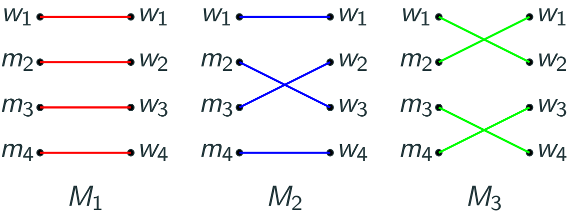
$S_1 = \{M_1, M_3\}$ and $S_2 = \{M_2, M_3\}$

(w_4, w_1, w_3, w_2) m_1 — w_1 (m_2, m_1, m_3, m_4)

(w_4, w_2, w_3, w_1) m_2 — w_2 (m_1, m_3, m_2, m_4)

(w_1, w_3, w_2, w_4) m_3 — w_3 (m_4, m_2, m_3, m_1)

(w_1, w_4, w_2, w_3) m_4 — w_4 (m_3, m_4, m_2, m_1)



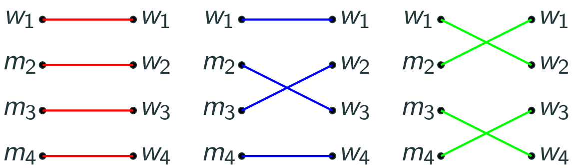
$S_1 = \{M_1, M_3\}$ and $S_2 = \{M_2, M_3\}$

(w_4, w_1, w_3, w_2) m_1 | w_1 (m_2, m_1, m_3, m_4)

(w_4, w_2, w_3, w_1) m_2 | w_2 (m_1, m_3, m_2, m_4)

(w_1, w_3, w_2, w_4) m_3 | w_3 (m_4, m_2, m_3, m_1)

(w_1, w_4, w_2, w_3) m_4 | w_4 (m_3, m_4, m_2, m_1)



M_1 M_2 M_3

$S_1 = \{M_1, M_3\}$ and $S_2 = \{M_2, M_3\}$

Enumeration Algorithm

Enumerating all maximum length chains of disjoint stable matchings:

Given a marriage instance, we run our algorithm once in men-proposing settings and and once more in women-proposing setting to get the following chains of disjoint stable matchings.

Enumeration Algorithm

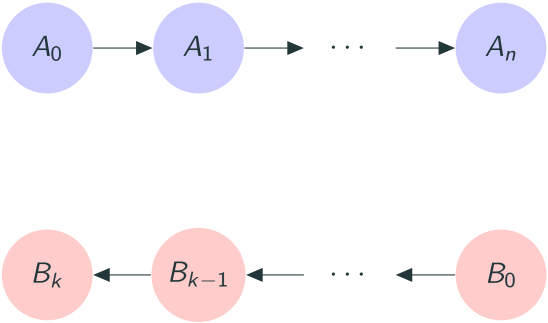
Enumerating all maximum length chains of disjoint stable matchings:

Given a marriage instance, we run our algorithm once in men-proposing settings and and once more in women-proposing setting to get the following chains of disjoint stable matchings.

Enumeration Algorithm

Enumerating all maximum length chains of disjoint stable matchings:

Given a marriage instance, we run our algorithm once in men-proposing settings and and once more in women-proposing setting to get the following chains of disjoint stable matchings.

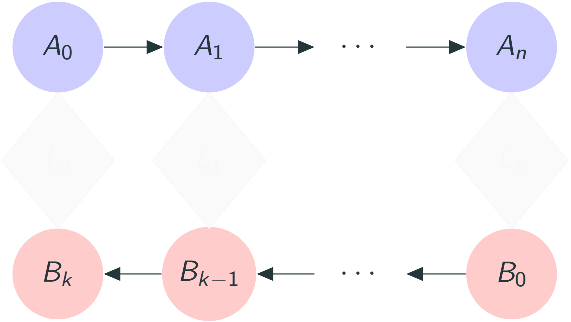


Enumeration Algorithm

We konw that, between any two stable matchings M_1, M_2 such that $M_1 \preceq M_2$, we can easily construct the sublattice of all the stable matchings between M_1 and M_2.

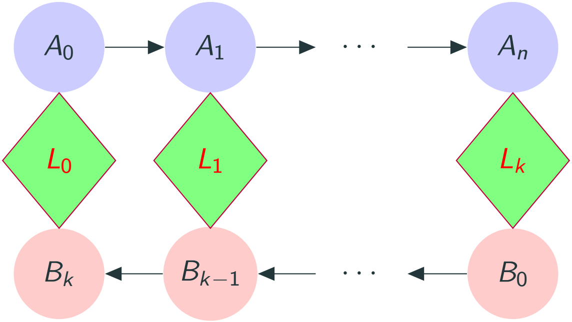
Enumeration Algorithm

We konw that, between any two stable matchings M_1, M_2 such that $M_1 \preceq M_2$, we can easily construct the sublattice of all the stable matchings between M_1 and M_2.



Enumeration Algorithm

We konw that, between any two stable matchings M_1, M_2 such that $M_1 \preceq M_2$, we can easily construct the sublattice of all the stable matchings between M_1 and M_2.

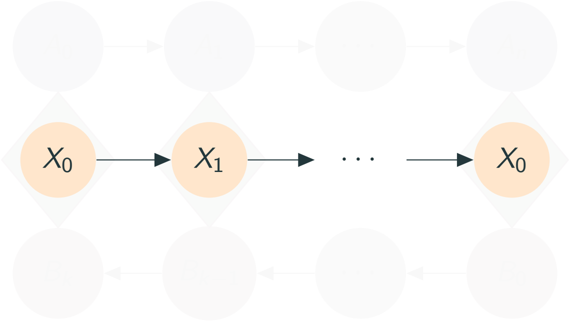


Enumeration Algorithm

Let $X = \{X_0, \cdots X_k\}$ be a maximum-length chain of disjoint stable matchings i.e. $X_0 \prec X_1 \prec \cdots \prec X_k$. We note the following property of the matchings in X.

Lemma 9

For $0 \leq i \leq k$, $A_i \preceq X_i \preceq B_{k-i}$

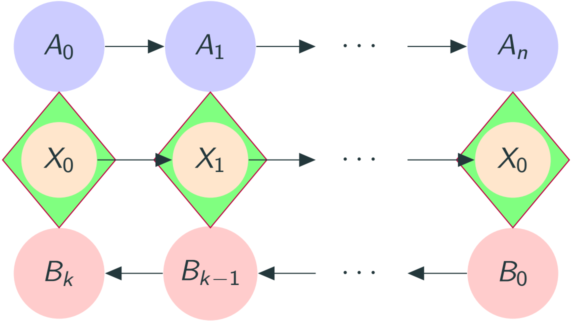


Enumeration Algorithm

Let $X = \{X_0, \cdots X_k\}$ be a maximum-length chain of disjoint stable matchings i.e. $X_0 \prec X_1 \prec \cdots \prec X_k$. We note the following property of the matchings in X.

Lemma 9

For $0 \leq i \leq k$, $A_i \preceq X_i \preceq B_{k-i}$



Enumeration Algorithm

With the help of lemma 9, we use *branching technique* to enumerate all possible max-length chains of disjoint stable matchings in *polynomial delay*.

Random Instance

We analyze the number of maximum-length chains of disjoint stable matchings in a random stable matchings instance with complete lists.

Lemma 10

The probability of the number of maximum size chains of disjoint stable matchings exceeding $(\frac{n}{\ln n})^{\ln n}$ *is at most* $O(\frac{(\ln n)^2}{n^2})$.

Corollary 11

The enumeration algorithm terminates in $O(n^4 + n^{2\ln n + 2})$ *time with probability* 1 *as* $n \to \infty$.

Random Instance

We analyze the number of maximum-length chains of disjoint stable matchings in a random stable matchings instance with complete lists.

Lemma 10

The probability of the number of maximum size chains of disjoint stable matchings exceeding $(\frac{n}{\ln n})^{\ln n}$ *is at most* $O(\frac{(\ln n)^2}{n^2})$.

Corollary 11

The enumeration algorithm terminates in $O(n^4 + n^{2 \ln n + 2})$ *time with probability 1 as* $n \to \infty$.

Random Instance

We analyze the number of maximum-length chains of disjoint stable matchings in a random stable matchings instance with complete lists.

Lemma 10

The probability of the number of maximum size chains of disjoint stable matchings exceeding $(\frac{n}{\ln n})^{\ln n}$ *is at most* $O(\frac{(\ln n)^2}{n^2})$.

Corollary 11

The enumeration algorithm terminates in $O(n^4 + n^{2 \ln n + 2})$ *time with probability* 1 *as* $n \to \infty$.

Thank You!