Disjoint Stable Matchings

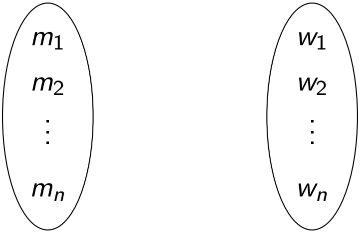
Aadityan Ganesh, Vishwa Prakash H.V., **Prajakta Nimbhorkar**, Geevarghese Philip

Chennai Mathematical Institute

Indo-Spanish Pre-Conference School on Algorithms and Combinatorics 2025

Matching in Bipartite Graphs

Bipartite Graph $G = (V, E)$



Set of men M   Set of women W

Matching

$M \subseteq E$ such that $\forall (m_i, w_j), (m_k, w_\ell) \in M \; i \neq k \iff j \neq \ell$ thus edges in a matching are disjoint.

Stable Matching Setup

Bipartite graph with preference ordering

For each man $m \in M$, there is a *preference ordering* π_m, which is a permutation over neighbors of m in G.
Similarly there is π_w for each $w \in W$, which is a permutation over neighbors of w in G.

Example for a complete graph:

$$\pi_{m_1} = \langle w_4, w_3, \cdots, w_n \rangle \qquad \pi_{w_1} = \langle m_1, m_3, \cdots, m_n \rangle$$

$$\pi_{m_2} = \langle w_1, w_6, \cdots, w_9 \rangle \qquad \pi_{w_2} = \langle m_5, m_2, \cdots, m_9 \rangle$$

$$\vdots \qquad \vdots$$

$$\pi_{m_n} = \langle w_5, w_7, \cdots, w_n \rangle \qquad \pi_{w_n} = \langle m_5, m_7, \cdots, m_n \rangle$$

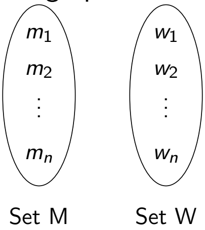
Stable Matching Setup

Bipartite graph with preference ordering

For each man $m \in M$, there is a *preference ordering* π_m, which is a permutation over neighbors of m in G.
Similarly there is π_w for each $w \in W$, which is a permutation over neighbors of w in G.

Example for a complete graph:

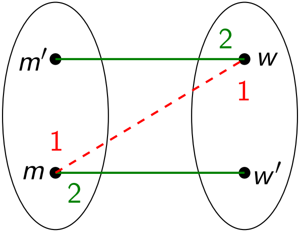
$$\pi_{m_1} = \langle w_4, w_3, \cdots, w_n \rangle$$
$$\pi_{m_2} = \langle w_1, w_6, \cdots, w_9 \rangle$$
$$\vdots$$
$$\pi_{m_n} = \langle w_5, w_7, \cdots, w_n \rangle$$



$$\pi_{w_1} = \langle m_1, m_3, \cdots, m_n \rangle$$
$$\pi_{w_2} = \langle m_5, m_2, \cdots, m_9 \rangle$$
$$\vdots$$
$$\pi_{w_n} = \langle m_5, m_7, \cdots, m_n \rangle$$

Blocking pair

A matching N is *blocked* by $(m, w) \notin N$ if $(m, w'), (m', w) \in N$ but $\pi_m(w) > \pi_m(w')$ *and* $\pi_w(m) > \pi_w(m')$.
Both m, w prefer each other more than their *partner* in N.



The Stable matching problem

Stable matching

A matching N is *stable* if it is *not* blocked by any pair (m, w).

This is the classical Stable Marriage Problem

M: set of men, W: set of women
A stable matching is a pairing of men and women so that no man-woman pair leaves their assigned partner and pairs up with each other.

The Stable matching problem

Stable matching

A matching N is *stable* if it is *not* blocked by any pair (m, w).

This is the classical Stable Marriage Problem

M: set of men, W: set of women

A stable matching is a pairing of men and women so that no man-woman pair leaves their assigned partner and pairs up with each other.

The Stable matching problem

Stable matching

A matching N is *stable* if it is *not* blocked by any pair (m, w).

This is the classical Stable Marriage Problem

M: set of men, W: set of women
A stable matching is a pairing of men and women so that no man-woman pair leaves their assigned partner and pairs up with each other.

Motivation

- Practical: Models real-world situations like college admissions, assignment of medical interns to hospitals etc.
- Mathematical: Set of stable matchings has an interesting, rich structure.

Recognition (excerpt from Wikipedia)

Shapley and Roth were awarded the 2012 Nobel Memorial Prize in Economic Sciences "for the theory of stable allocations and the practice of market design". Gale had died in 2008, making him ineligible for the prize.

Example

$$M = \{m_1, m_2, m_3\},\ W = \{w_1, w_2, w_3\}$$

π_{m_1} :	w_2	w_3	w_1
π_{m_2} :	w_3	w_1	w_2
π_{m_3} :	w_2	w_1	w_3

π_{w_1} :	m_1	m_2	m_3
π_{w_2} :	m_2	m_1	m_3
π_{w_3} :	m_3	m_2	m_1

Two stable matchings: red and green

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

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

Purple matching is blocked by (m_2, w_3).

Example

$$M = \{m_1, m_2, m_3\},\ W = \{w_1, w_2, w_3\}$$

π_{m_1} :	w_2	w_3	w_1
π_{m_2} :	w_3	w_1	w_2
π_{m_3} :	w_2	w_1	w_3

π_{w_1} :	m_1	m_2	m_3
π_{w_2} :	m_2	m_1	m_3
π_{w_3} :	m_3	m_2	m_1

Two stable matchings: red and green

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

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

Purple matching is blocked by (m_2, w_3).

Example

$$M = \{m_1, m_2, m_3\},\ W = \{w_1, w_2, w_3\}$$

π_{m_1} :	w_2	w_3	w_1
π_{m_2} :	w_3	w_1	w_2
π_{m_3} :	w_2	w_1	w_3

π_{w_1} :	m_1	m_2	m_3
π_{w_2} :	m_2	m_1	m_3
π_{w_3} :	m_3	m_2	m_1

Two stable matchings: red and green

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

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

Purple matching is blocked by (m_2, w_3).

Properties of stable matchings[1, 2, 3]

1. Existence: Every bipartite graph with preference ordering has at least one stable matching.
2. Computation: A stable matching can be efficiently computed by the Gale-Shapley algorithm.
3. Optimality: There is a unique stable matching that is simultaneously best for all $m \in M$, and simultaneously worst for all $w \in W$, (called M-optimal and W-pessimal stable matching). Similarly there is a W-optimal and M-pessimal stable matching.
4. Structure: Stable matchings in a bipartite graph form a distributive lattice under a suitably defined partial order. Conversely, for every finite distributive lattice L, there exists a stable matching instance whose lattice of stable matchings is isomorphic to L.

Stable Matchings

Stable Matching

A matching with no blocking pair

Checking stability: $O(n^2)$

Stable and Fixed Pairs

Stable Pair

A pair (m, w) is called as a *stable pair* if m and w are partners in at least one stable matching.

Fixed Pair

A pair (m, w) is called as a *fixed pair* if m and w are partners in at all stable matchings.

Stable and Fixed Pairs

Stable Pair

A pair (m, w) is called as a *stable pair* if m and w are partners in at least one stable matching.

Fixed Pair

A pair (m, w) is called as a *fixed pair* if m and w are partners in at all stable matchings.

Gale-Shapley Algorithm

Algorithm Gale-Shapley

```
procedure FIND STABLE MATCHING(M)
    assign each person to be free
    while some man m is free do
        w ← first woman on m's list to whom m hasn't proposed
        if w is free then
            assign m and w to be engaged to each other
        else
            if w prefers m to her current matched partner m′ then
                assign m and w to be engaged and m′ to be free
            else
                w rejects m                                    ▷ m remains free
            end if
        end if
    end while
    return Stable matching consisting of n engaged pairs
end procedure
```

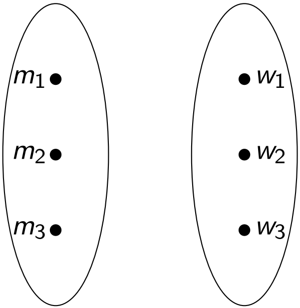
Gale-Shapley Algorithm

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



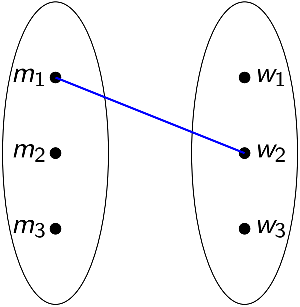
Gale-Shapley Algorithm

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



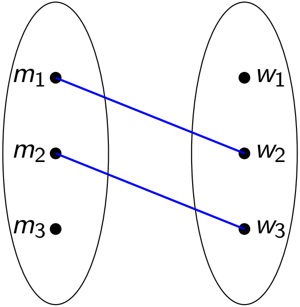
Gale-Shapley Algorithm

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



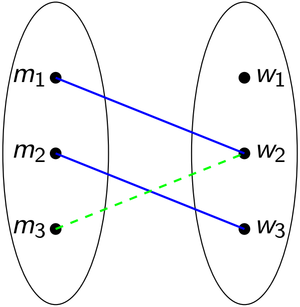
Gale-Shapley Algorithm

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



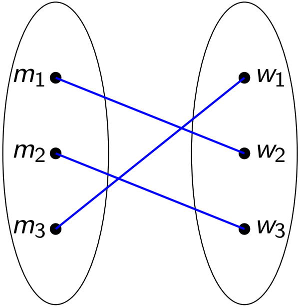
Gale-Shapley Algorithm

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



Gale-Shapley Algorithm - Key Results

1. Every stable matching instance has at least one stable matching.
2. All possible executions of the Gale-Shapley algorithm yield the same result.
3. If men propose, we get the *"Man-optimal"* stable matching.

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

   Woman-optimal: Every woman is matched with her best partner among all stable partners.
5. The man-optimal stable matching is *woman-pessimal*, and vice-versa.
6. All stable matchings match the same set of vertices.

Gale-Shapley Algorithm - Key Results

1. Every stable matching instance has at least one stable matching.
2. All possible executions of the Gale-Shapley algorithm yield the same result.
3. If men propose, we get the *"Man-optimal"* stable matching.

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

   Woman-optimal: Every woman is matched with her best partner among all stable partners.
5. The man-optimal stable matching is *woman-pessimal*, and vice-versa.
6. All stable matchings match the same set of vertices.

Gale-Shapley Algorithm - Key Results

1. Every stable matching instance has at least one stable matching.
2. All possible executions of the Gale-Shapley algorithm yield the same result.
3. If men propose, we get the *"Man-optimal"* stable matching.

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

   Woman-optimal: Every woman is matched with her best partner among all stable partners.
5. The man-optimal stable matching is *woman-pessimal*, and vice-versa.
6. All stable matchings match the same set of vertices.

Gale-Shapley Algorithm - Key Results

1. Every stable matching instance has at least one stable matching.
2. All possible executions of the Gale-Shapley algorithm yield the same result.
3. If men propose, we get the *"Man-optimal"* stable matching.

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

   Woman-optimal: Every woman is matched with her best partner among all stable partners.
5. The man-optimal stable matching is *woman-pessimal*, and vice-versa.
6. All stable matchings match the same set of vertices.

Gale-Shapley Algorithm - Key Results

1. Every stable matching instance has at least one stable matching.
2. All possible executions of the Gale-Shapley algorithm yield the same result.
3. If men propose, we get the *"Man-optimal"* stable matching.

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

   Woman-optimal: Every woman is matched with her best partner among all stable partners.
5. The man-optimal stable matching is *woman-pessimal*, and vice-versa.
6. All stable matchings match the same set of vertices.

Gale-Shapley Algorithm - Key Results

1. Every stable matching instance has at least one stable matching.
2. All possible executions of the Gale-Shapley algorithm yield the same result.
3. If men propose, we get the *"Man-optimal"* stable matching.

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

   Woman-optimal: Every woman is matched with her best partner among all stable partners.
5. The man-optimal stable matching is *woman-pessimal*, and vice-versa.
6. All stable matchings match the same set of vertices.

The partial order

A vertex $x \in M \cup W$ is said to *prefer* a matching P to a matching Q if $\exists y \neq z$ such that $(x, y) \in P, (x, z) \in Q$ and $\pi_x(y) > \pi_x(z)$.

Domination

A stable matching P is said to *dominate* a stable matching Q, written $P \preceq Q$, if for every $m \in M$, either m prefers P to Q or has the same partner in both P and Q.

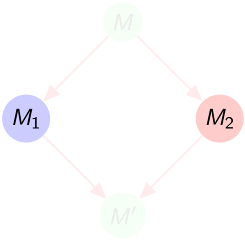
The partial order

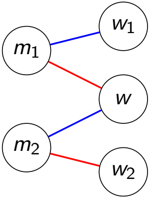
A vertex $x \in M \cup W$ is said to *prefer* a matching P to a matching Q if $\exists y \neq z$ such that $(x, y) \in P, (x, z) \in Q$ and $\pi_x(y) > \pi_x(z)$.

Domination

A stable matching P is said to *dominate* a stable matching Q, written $P \preceq Q$, if for every $m \in M$, either m prefers P to Q or has the same partner in both P and Q.

Meet and Join





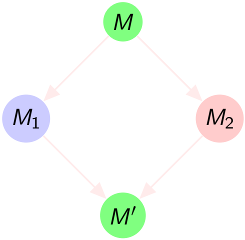
w can't be the best partner of both m_1, m_2!

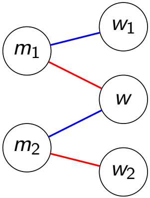
$$M' = \{(m,w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$
$$M'' = \{(m,w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

$p_{M_1}(m)$ and $p_{M_2}(m)$ are partners of m in M_1 and M_2 respectively.

M and M' are stable matchings!

Meet and Join





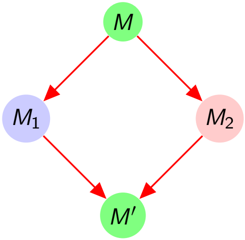
w can't be the best partner of both m_1, m_2!

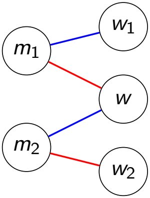
$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$
$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

$p_{M_1}(m)$ and $p_{M_2}(m)$ are partners of m in M_1 and M_2 respectively.

M and M' are stable matchings!

Meet and Join





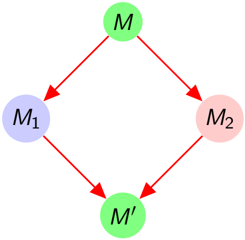
w can't be the best partner of both m_1, m_2!

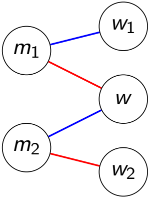
$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$
$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

$p_{M_1}(m)$ and $p_{M_2}(m)$ are partners of m in M_1 and M_2 respectively.

M and M' are stable matchings!

Meet and Join





w can't be the best partner of both m_1, m_2!

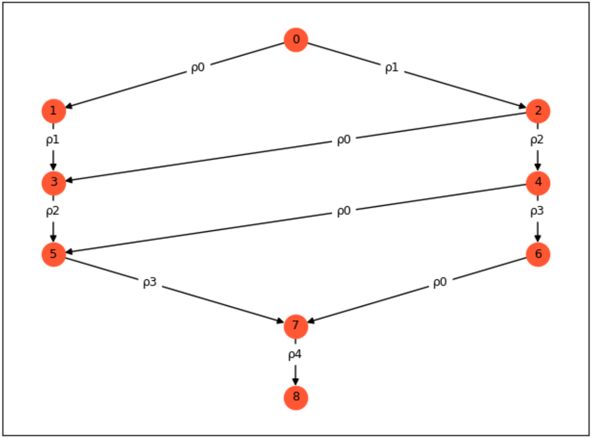
$$M' = \{(m, w) \mid w = best(p_{M_1}(m), p_{M_2}(m))\}$$
$$M'' = \{(m, w) \mid w = worst(p_{M_1}(m), p_{M_2}(m))\}$$

$p_{M_1}(m)$ and $p_{M_2}(m)$ are partners of m in M_1 and M_2 respectively.

> M and M' are stable matchings!

The Lattice Structure

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



Our problem and result

Our problem

Give an algorithm for a largest collection of *pairwise disjoint stable matchings*.

Our result

A largest pairwise disjoint collection is a longest chain in the lattice. It can be found in linear time.

Our problem and result

Our problem

Give an algorithm for a largest collection of *pairwise disjoint stable matchings*.

Our result

A largest pairwise disjoint collection is a longest chain in the lattice. It can be found in linear time.

When does a non-trivial set of disjoint stable matchings exist?

Let P, Q be the M-optimal and the W-optimal stable matchings.
If $N \cap N' \neq \emptyset$, then every $(m, w) \in P \cap Q$ is in every stable matching.

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

Necessary condition

To have a non-trivial set of disjoint stable matchings, it is necessary that $P \cap Q = \emptyset$.

Proof idea

Theorem 1 (Teo-Sethuraman 1998[4])

Given a set S of stable matchigns, assigning k-th best partner to each $m \in M$ among all partners in S, gives a stable matching.

Let $P_1, P_2, \cdots, P_k$ be a largest set of disjoint stable matchings
Define $Q_i = \{(m, w) \mid w \text{ is } i\text{th best partner of m in } P_1, \ldots, P_k\}$.

Implication of Theorem 1

. $Q_1, \ldots, Q_k$ are all stable matchings, pairwise disjoint and form a chain.

Proof idea

Theorem 1 (Teo-Sethuraman 1998[4])

Given a set S of stable matchigns, assigning k-th best partner to each $m \in M$ among all partners in S, gives a stable matching.

Let $P_1, P_2, \cdots, P_k$ be a largest set of disjoint stable matchings
Define $Q_i = \{(m, w) \mid w \text{ is } i\text{th best partner of m in } P_1, \ldots, P_k\}$.

Implication of Theorem 1

. $Q_1, \ldots, Q_k$ are all stable matchings, pairwise disjoint and form a chain.

Proof idea continued

What do we get?

For any set of k pairwise-disjoint stable matchings, $\exists$ k-length chain of pairwise disjoint stable matchings.

Does it give us an algorithm?

Yes, just a modification of the Gale-Shapley algorithm!

Proof idea continued

What do we get?

For any set of k pairwise-disjoint stable matchings, $\exists$ k-length chain of pairwise disjoint stable matchings.

Does it give us an algorithm?

Yes, just a modification of the Gale-Shapley algorithm!

Disjoint Stable Matching Algorithm

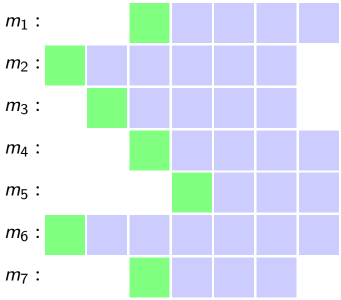
Step 1: Start with the men-proposing GS algorithm.

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

Men's preference list

Disjoint Stable Matching Algorithm

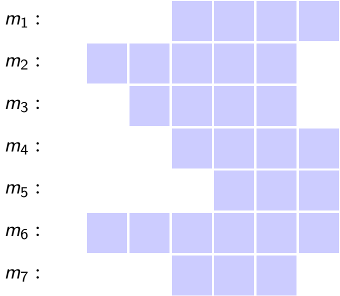
Step 2: Get the first stable matching M_1.



Men's Preference

Disjoint Stable Matching Algorithm

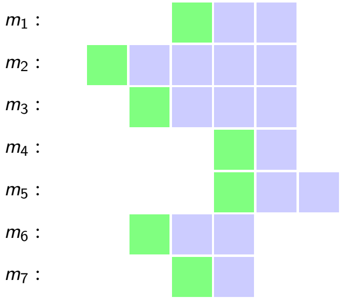
Step 3: Delete the matching edges. Continue GS algorithm.



Men's Preference

Disjoint Stable Matching Algorithm

Step 4: Get the next matching M_2. Delete it and repeat until the instance is empty.



Men's Preference

Termination and Time Complexity

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

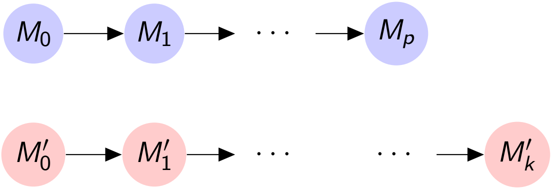
For the same reason, the running time of the algorithm is $\mathbf{O}(\mathbf{n^2})$.

Longest Chain of Disjoint Stable matchings

Correctness

The algorithm gives the longest chain of disjoint stable matchings.

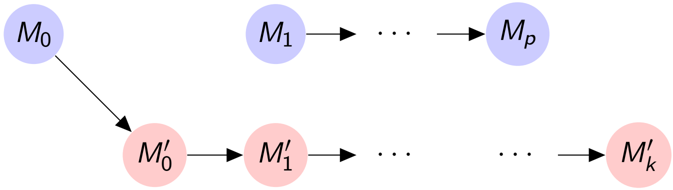
Proof:



Correctness

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

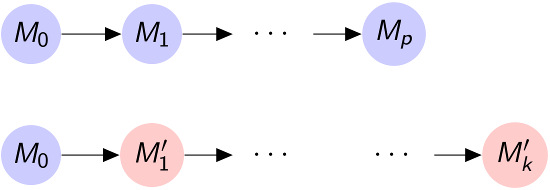


Longest Chain of Disjoint Stable matchings

Correctness

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

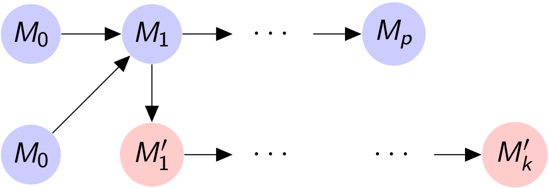


Longest Chain of Disjoint Stable matchings

Correctness

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

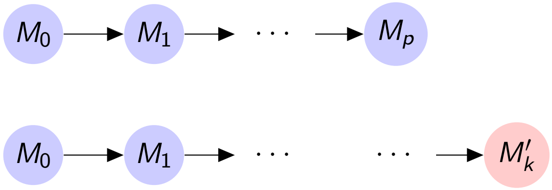


Longest Chain of Disjoint Stable matchings

Correctness

The algorithm gives the longest chain of disjoint stable matchings.

Proof:

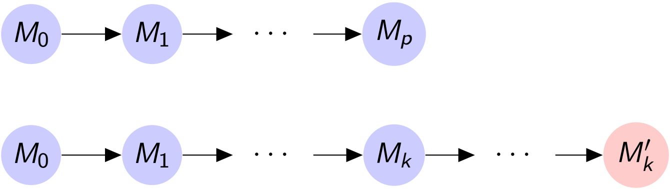


Longest Chain of Disjoint Stable matchings

Correctness

The algorithm gives the longest chain of disjoint stable matchings.

Proof:



Random Instance

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

Lemma 2

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

Corollary 3

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 2

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 3

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 2

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 3

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

Future Work

- Disjoint Stable Matchings in the Stable Roommates problem.
- When disjoint stable matchings do not exist, minimize pairwise intersection.

References

D. Gale and L. S. Shapley.
College admissions and the stability of marriage.
The American Mathematical Monthly, 69(1):9–15, 1962.

Dan Gusfield and Robert W. Irving.
The Stable marriage problem - structure and algorithms.
Foundations of computing series. MIT Press, 1989.

Donald Knuth.
Mariages stables et leurs relations avec d'autres problèmes combinatoires : introduction à l'analyse mathèmatique des algorithmes.
Presses de l'Universitè de Montrèal, Montrèal, 1976.

Chung-Piaw Teo and Jay Sethuraman.
The geometry of fractional stable matchings and its applications.
Mathematics of Operations Research, 23(4):874–891, 1998.