Locally Stable Roommate Problem

Élise Vandomme

EURO 2024, Copenhagen

HEC LIÈGE
Management School - Liège Université

Joint work

Joint work with two colleagues from HEC Liege (Belgium):

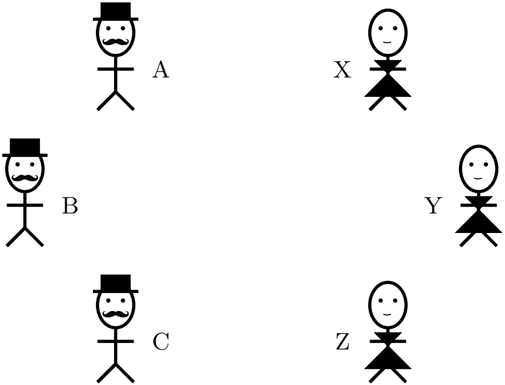
Yves Crama
Professor

Marie Baratto
PhD looking for a postdoc

Disclaimer : This is a work in progress.

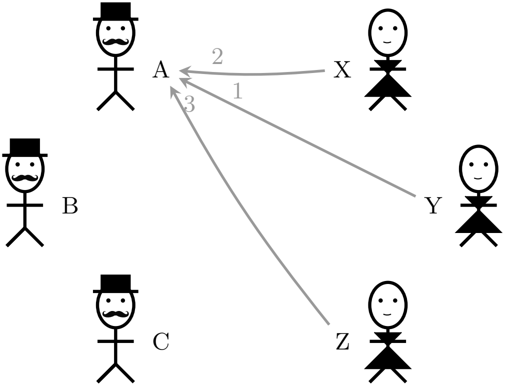
Stable Marriage Problem (SMP)

- Introduced by Gale and Shapley in 1962
- Aim : to match men and women based on their preferences for all members of the opposite gender



Stable Marriage Problem (SMP)

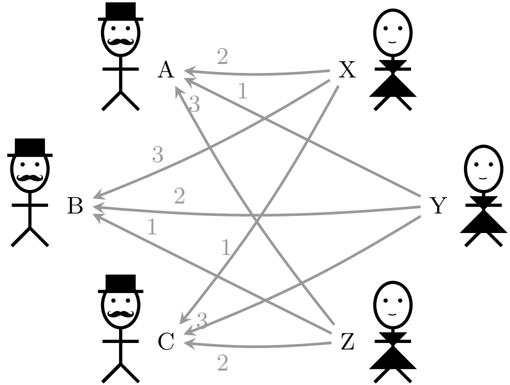
- Introduced by Gale and Shapley in 1962
- Aim : to match men and women based on their preferences for all members of the opposite gender



A: Y X Z

Stable Marriage Problem (SMP)

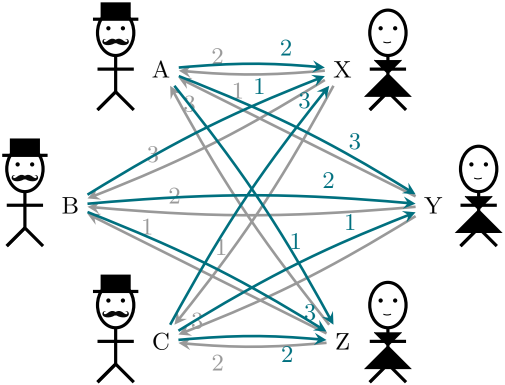
- Introduced by Gale and Shapley in 1962
- Aim : to match men and women based on their preferences for all members of the opposite gender



A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y

Stable Marriage Problem (SMP)

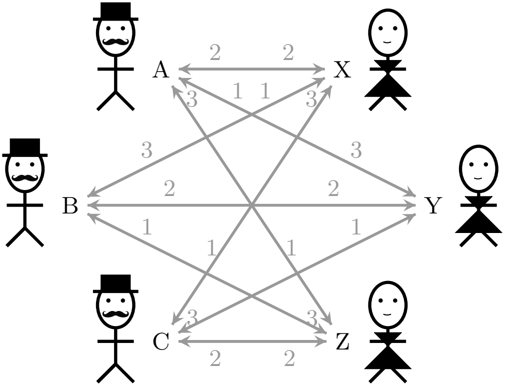
- Introduced by Gale and Shapley in 1962
- Aim : to match men and women based on their preferences for all members of the opposite gender



A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y
X:	B	A	C
Y:	C	B	A
Z:	A	C	B

Stable Marriage Problem (SMP)

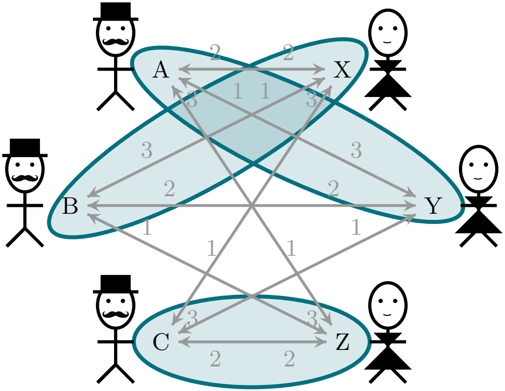
- A matching is stable if there are no blocking pairs.
- A blocking pair is a pair of unmatched individuals who prefer each other to their partners in the matching.



A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y
X:	B	A	C
Y:	C	B	A
Z:	A	C	B

Stable Marriage Problem (SMP)

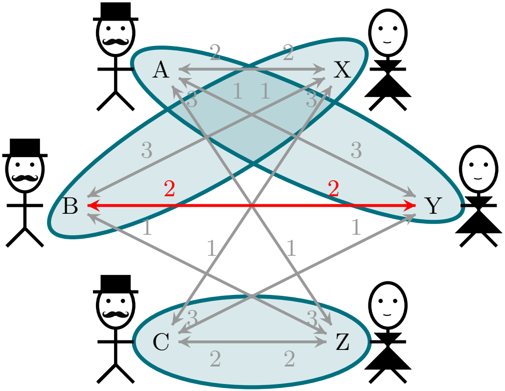
- A matching is stable if there are no blocking pairs.
- A blocking pair is a pair of unmatched individuals who prefer each other to their partners in the matching.



A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y
X:	B	A	C
Y:	C	B	A
Z:	A	C	B

Stable Marriage Problem (SMP)

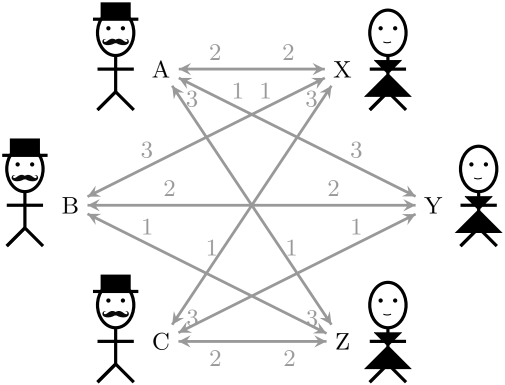
- A matching is stable if there are no blocking pairs.
- A blocking pair is a pair of unmatched individuals who prefer each other to their partners in the matching.



A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y
X:	B	A	C
Y:	C	B	A
Z:	A	C	B

Stable Marriage Problem (SMP)

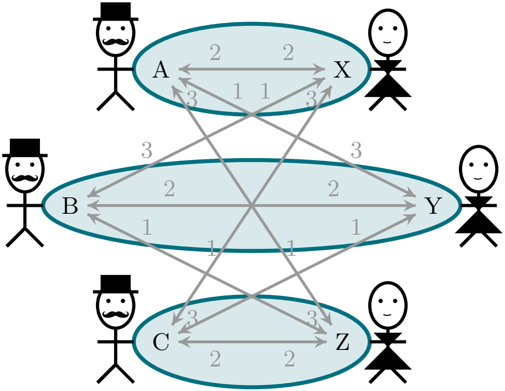
- A matching is stable if there are no blocking pairs.
- A blocking pair is a pair of unmatched individuals who prefer each other to their partners in the matching.



A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y
X:	B	A	C
Y:	C	B	A
Z:	A	C	B

Stable Marriage Problem (SMP)

- A matching is stable if there are no blocking pairs.
- A blocking pair is a pair of unmatched individuals who prefer each other to their partners in the matching.

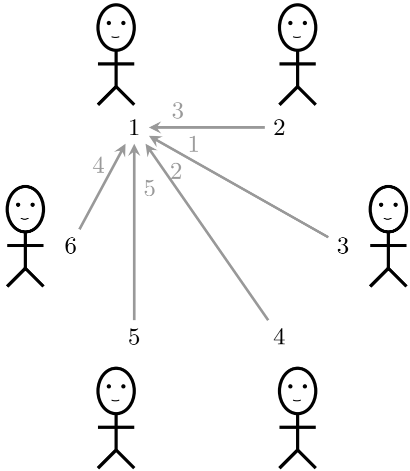


A:	Y	X	Z
B:	Z	Y	X
C:	X	Z	Y

X:	B	A	C
Y:	C	B	A
Z:	A	C	B

Stable Roommate Problem (SRP)

- Generalization of SMP to non-bipartite model
- Each individual ranks all the others in order of preference.



1:	3	4	2	6	5
2:	6	5	4	1	3
3:	2	4	5	1	6
4:	5	2	3	6	1
5:	3	1	2	4	6
6:	5	1	3	4	2

- Aim : to find a stable matching

SMP vs SRP

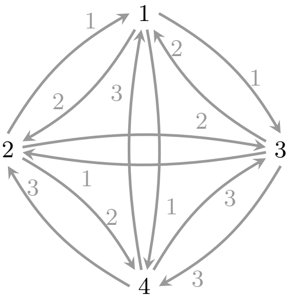
	SMP	**SRP**
Graphs	Complete bipartite	Complete

SMP vs SRP

	SMP	**SRP**
Graphs	Complete bipartite	Complete
$\exists$ stable matching?	Always	Not always

SMP vs SRP

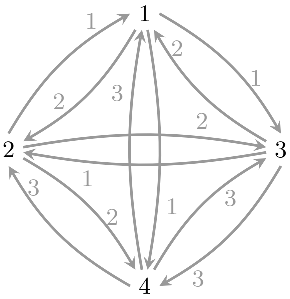
	SMP	**SRP**
Graphs	Complete bipartite	Complete
∃ stable matching?	Always	Not always



No stable matching

SMP vs SRP

	SMP	**SRP**
Graphs	Complete bipartite	Complete
∃ stable matching?	Always	Not always
Find one (if any)	Polynomial	Polynomial
	Gale-Shapley (1962)	Irving (1985)



No stable matching

Stability in Kidney Exchange Programs

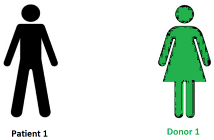
Patient with a serious kidney disease may resort to:

- Dialysis
- Transplant from a deceased donor
- Transplant from a willing donor

Stability in Kidney Exchange Programs

Patient with a serious kidney disease may resort to:

- Dialysis
- Transplant from a deceased donor
- **Transplant from a willing donor**



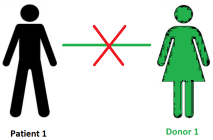
Stability in Kidney Exchange Programs

Patient with a serious kidney disease may resort to:

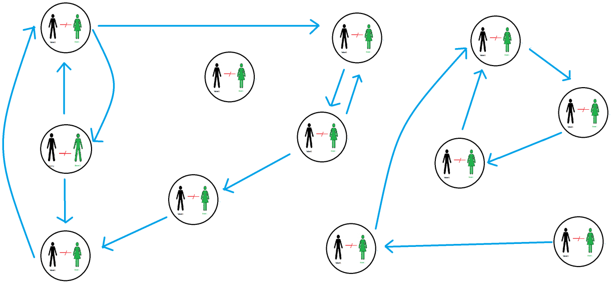
- Dialysis
- Transplant from a deceased donor
- **Transplant from a willing donor**

Patient might not be compatible with the donor: e.g.,

- Blood incompatibility
- Tissue type incompatibility



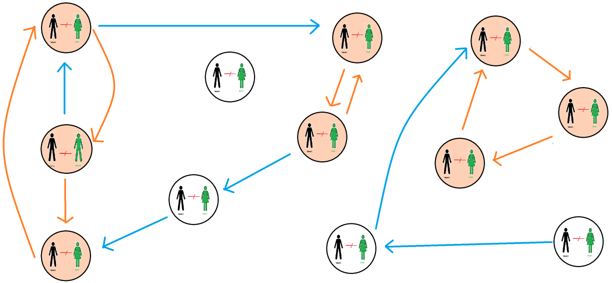
Compatibility graph



G=(V,A) where:

- $V = \{1, ..., n\}$ set of vertices, consisting of all patient-donor pairs.
- A, the set of arcs, designating compatibilities between the vertices. Two vertices i and j are connected by arc (i, j) if the donor in pair i is compatible with the patient in pair j.

Possible exchanges

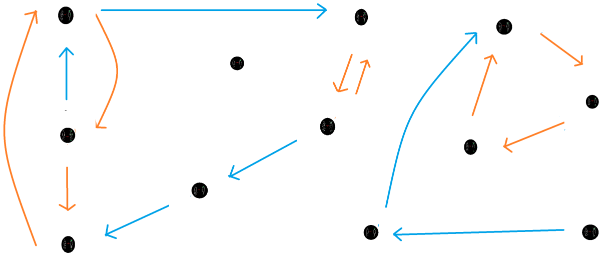


Definition

An exchange is a set of disjoint cycles in the directed graph such that every cycle length does not exceed a given limit K.

- Aim: to **maximize the number of patients transplanted**
- When $K = 2$, an exchange is a matching.

Possible exchanges

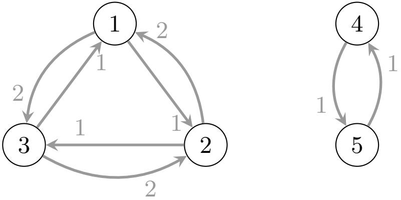


Definition

An exchange is a set of disjoint cycles in the directed graph such that every cycle length does not exceed a given limit K.

- Aim: to **maximize the number of patients transplanted**
- When $K = 2$, an exchange is a matching.

Stable Exchange (SE)

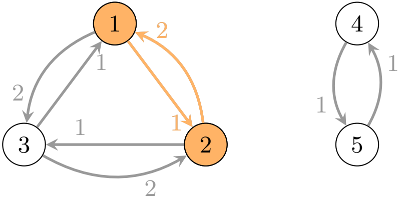


Definition

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

Stable Exchange (SE)

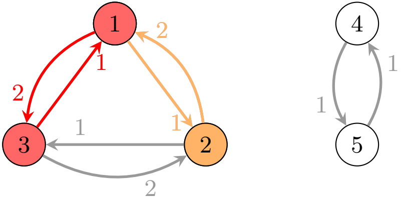


Definition

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

Stable Exchange (SE)

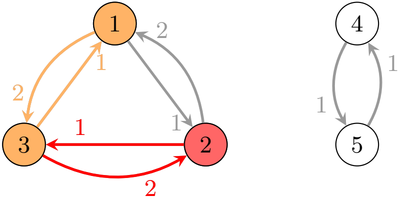


Definition

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

Stable Exchange (SE)

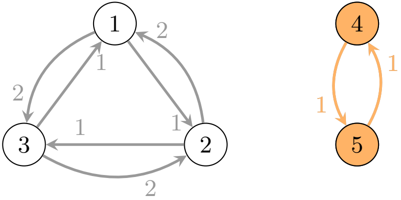


Definition

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

Stable Exchange (SE)

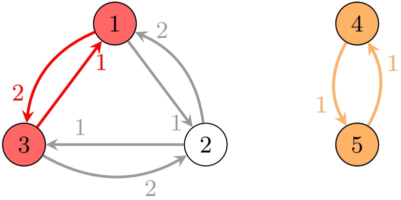


Definition

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

Stable Exchange (SE)

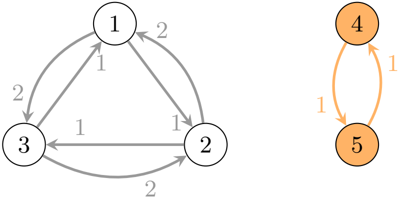


Definition

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

Locally Stable Exchange (LSE)



Definition (Baratto–Crama–Pedroso–Viana, accepted)

Given a directed graph $G = (V, A)$,

- an exchange $\mathcal{M}$ is locally stable if no blocking cycle u exists for $\mathcal{M}$.
- A cycle u is locally blocking for an exchange $\mathcal{M}$ if it is not included in $\mathcal{M}$, it intersects $\mathcal{M}$ and for every vertex $i \in V(u)$, i prefers u to $\mathcal{M}$.
- Vertex i prefers the cycle u to the exchange $\mathcal{M}$ if either
  - $i \notin V(\mathcal{M})$, or
  - $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

SE vs LSE

A stable matching is maximum if it has the largest possible size among all stable matchings. And similarly for maximum locally stable matchings.

- **SE** problem: What is the maximum size of a stable matching?
  ($K = 2$, 72 don't have a stable matching out of 600 tested - 12%)

- **LSE** problem: What is the maximum size of a non-empty locally stable matching?
  ($K = 2$, 1 out of 600 tested has a solution of cardinality zero - 0.2%)

  - For 50 instances with $V \approx 200$,
    45 out of 50 have a stable exchange;
    50 out of 50 have a locally stable exchange > 0;
    45 instances max. stable exchange = max. locally stable exchange

SE vs LSE

A stable matching is maximum if it has the largest possible size among all stable matchings. And similarly for maximum locally stable matchings.

- **SE** problem: What is the maximum size of a stable matching?
  ($K = 2$, 72 don't have a stable matching out of 600 tested - 12%)

- **LSE** problem: What is the maximum size of a non-empty locally stable matching?
  ($K = 2$, 1 out of 600 tested has a solution of cardinality zero - 0.2%)
  - For 50 instances with $V \approx 200$,
    45 out of 50 have a stable exchange;
    50 out of 50 have a locally stable exchange > 0;
    45 instances max. stable exchange = max. locally stable exchange

Work in Progress:

- Computing a maximum locally stable exch. for $K \geq 3$ is NP-hard.

- Computing a maximum $\underbrace{\text{locally stable exch. for } K = 2}_{\text{locally stable roommate problem}}$ is polynomial.

Locally Stable Roommate Problem (LSRP)

Proposition

If M is a stable matching and M' is a locally stable matching, then $V(M') \subseteq V(M)$ and $|M'| \leq |M|$.

Proposition

If a graph has a stable matching, then

1. all its stable matchings cover the same set of vertices, and
2. all its stable matchings are maximum locally stable.

A locally stable matching is maximal if it is not included (edge-wise) in any other locally stable matching.

Proposition

All maximal locally stable matchings cover the same set of vertices and hence, they have the same size.

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists

```
procedure Phase1($T$ : table of preference lists)



end procedure
```

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)




end procedure
```

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure Phase1($T$ : table of preference lists)
    assign each person to be free        free vs semi-engaged
end procedure
```

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free            } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list    } $x$ proposes to $y$



    end while
end procedure
```

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists
so that no deleted pair can be included in a locally stable matching.

```
procedure Phase1($T$ : table of preference lists)
    assign each person to be free                              } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list              } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do  } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                   engaged to $y$
        end for
    end while
end procedure
```

Irving's algorithm - Phase 1

Idea:

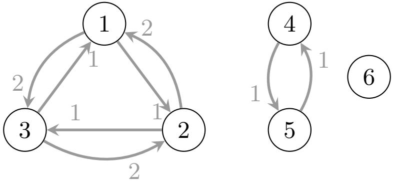
- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list
        for each person $z$ ranked below $x$ in $y$'s list do
            delete the pair $\{y, z\}$ from $T$
        end for
    end while
end procedure
```

free vs semi-engaged

x proposes to y

x becomes semi-engaged to y

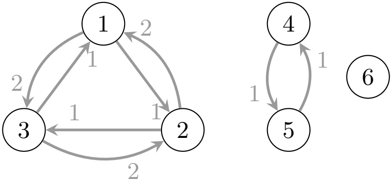


Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure Phase1($T$ : table of preference lists)
    assign each person to be free                            } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list            } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do    } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                    engaged to $y$
        end for
    end while
end procedure
```



1:	3	2
2:	1	3
3:	2	1
4:	5	
5:	4	
6:		

Irving's algorithm - Phase 1

Idea:

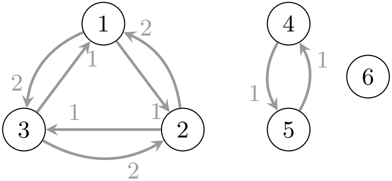
- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure Phase1($T$ : table of preference lists)
    assign each person to be free
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list
        for each person $z$ ranked below $x$ in $y$'s list do
            delete the pair $\{y, z\}$ from $T$
        end for
    end while
end procedure
```

free vs semi-engaged

x proposes to y

x becomes semi-engaged to y



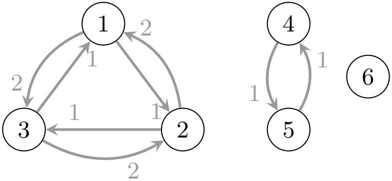
1:	3	2	4	5	6
2:	1	3	4	5	6
3:	2	1	4	5	6
4:	5	1	2	3	6
5:	4	1	2	3	6
6:	1	2	3	4	5

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free                            } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list            } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do    } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                    engaged to $y$
        end for
    end while
end procedure
```



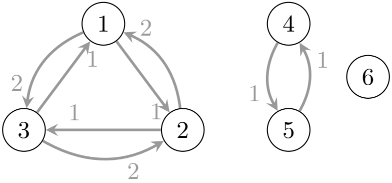
1:	3	2	4	5	6
2:	1	3	4	5	6
3:	2	1	4	5	6
4:	5	1	2	3	6
5:	4	1	2	3	6
6:	1	2	3	4	5

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free                         } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list              } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do   } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                   engaged to $y$
        end for
    end while
end procedure
```



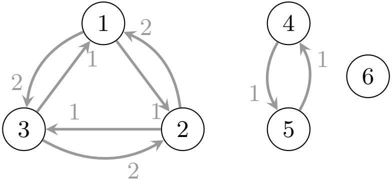
1:	3	2	4	5	6
2:	1	3	4	5	6
3:	2	$\boxed{1}$	4	5	6
4:	5	1	2	3	6
5:	4	1	2	3	6
6:	1	2	3	4	5

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free                        } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list               } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do   } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                   } engaged to $y$
        end for
    end while
end procedure
```



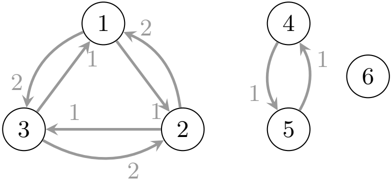
1:	3	2	4	5	6
2:	1	3	4	5	6
3:	2	$\boxed{1}$	~~4~~	~~5~~	~~6~~
4:	5	1	2	~~3~~	6
5:	4	1	2	~~3~~	6
6:	1	2	~~3~~	4	5

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free                              } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list                   } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do    } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                    } engaged to $y$
        end for
    end while
end procedure
```



1:	3	2	4	5	6
2:	1	3	4	5	6
3:	2	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	5	1	2	$\cancel{3}$	6
5:	4	1	2	$\cancel{3}$	6
6:	1	2	$\cancel{3}$	4	5

Irving's algorithm - Phase 1

Idea:

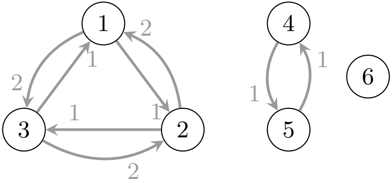
- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list
        for each person $z$ ranked below $x$ in $y$'s list do
            delete the pair $\{y, z\}$ from $T$
        end for
    end while
end procedure
```

free vs semi-engaged

x proposes to y

x becomes semi-engaged to y



1:	③	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	①	3	4	5	6
3:	2	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	5	$\cancel{1}$	2	$\cancel{3}$	6
5:	4	$\cancel{1}$	2	$\cancel{3}$	6
6:	$\cancel{1}$	2	$\cancel{3}$	4	5

Irving's algorithm - Phase 1

Idea:

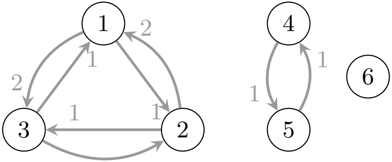
- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure Phase1($T$ : table of preference lists)
    assign each person to be free
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list
        for each person $z$ ranked below $x$ in $y$'s list do
            delete the pair $\{y, z\}$ from $T$
        end for
    end while
end procedure
```

free vs semi-engaged

x proposes to y

x becomes semi-engaged to y



1:	3	$\boxed{2}$	~~4~~	~~5~~	~~6~~
2:	1	3	4	5	6
3:	2	$\boxed{1}$	~~4~~	~~5~~	~~6~~
4:	5	~~1~~	2	~~3~~	6
5:	4	~~1~~	2	~~3~~	6
6:	~~1~~	2	~~3~~	4	5

Irving's algorithm - Phase 1

Idea:

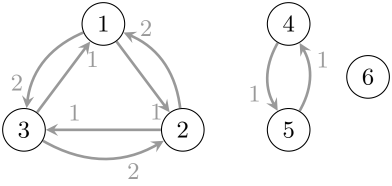
- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure Phase1($T$ : table of preference lists)
    assign each person to be free
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list
        for each person $z$ ranked below $x$ in $y$'s list do
            delete the pair $\{y, z\}$ from $T$
        end for
    end while
end procedure
```

free vs semi-engaged

x proposes to y

x becomes semi-engaged to y



1:	③	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	①	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	②	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	5	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	6
5:	4	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	6
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	4	5

Irving's algorithm - Phase 1

Idea:

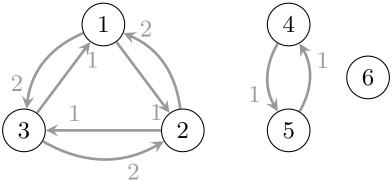
- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list
        for each person $z$ ranked below $x$ in $y$'s list do
            delete the pair $\{y, z\}$ from $T$
        end for
    end while
end procedure
```

} free vs semi-engaged

} x proposes to y

} x becomes semi-engaged to y



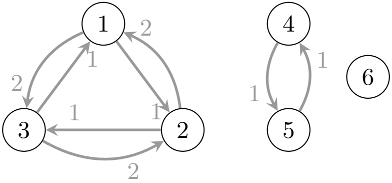
1:	$\textcircled{3}$	$\boxed{2}$	~~4~~	~~5~~	~~6~~
2:	$\textcircled{1}$	$\boxed{3}$	~~4~~	~~5~~	~~6~~
3:	$\textcircled{2}$	$\boxed{1}$	~~4~~	~~5~~	~~6~~
4:	$\textcircled{5}$	~~1~~	~~2~~	~~3~~	6
5:	$\boxed{4}$	~~1~~	~~2~~	~~3~~	~~6~~
6:	~~1~~	~~2~~	~~3~~	4	~~5~~

Irving's algorithm - Phase 1

Idea:

- Successive deletion of entries in the preference lists so that no deleted pair can be included in a locally stable matching.

```
procedure PHASE1($T$ : table of preference lists)
    assign each person to be free                              } free vs semi-engaged
    while some free person $x$ has a nonempty list do
        $y \leftarrow$ first person on $x$'s list             } $x$ proposes to $y$
        for each person $z$ ranked below $x$ in $y$'s list do   } $x$ becomes semi-
            delete the pair $\{y, z\}$ from $T$                   engaged to $y$
        end for
    end while
end procedure
```



1:	3	$\boxed{2}$	~~4~~	~~5~~	~~6~~
2:	1	$\boxed{3}$	~~4~~	~~5~~	~~6~~
3:	2	$\boxed{1}$	~~4~~	~~5~~	~~6~~
4:	$\boxed{5}$	~~1~~	~~2~~	~~3~~	~~6~~
5:	$\boxed{4}$	~~1~~	~~2~~	~~3~~	~~6~~
6:	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~

Irving's algorithm - Phase 1 - comments

- For a given instance of the problem, all possible executions of phase 1 of the algorithm yield the same reduced preference table (Gusfield and Irving 1989)
- If, at any stage of phase 1, x proposes to y, then in a locally stable matching M
  1. x cannot have a better partner than y;
  2. if $y \in V(M)$, y cannot have a worse partner than x.
- If T is the phase-1 table for a roommate instance, then
  1. $y = \mathrm{first}_T(x)$ if and only if $x = \mathrm{last}_T(y)$;
  2. if the edge $\{x, y\}$ is absent from T then x and y cannot be partners in a locally stable matching.

Irving's algorithm - Phase 1

Suppose that T is the phase-1 table for a roommates instance. Then

- if all persons have lists of size 1, there is a stable matching (and therefore a locally stable one) of size n.
- if one person has an empty list, there is no stable matching, but there could be a non-empty locally stable matching.
- if all lists have at least 1 element and some at least 2, then there could exist a non-empty locally stable matching and/or a stable one.

Irving's algorithm - Phase 1

Suppose that T is the phase-1 table for a roommates instance. Then

- if all persons have lists of size 1, there is a stable matching (and therefore a locally stable one) of size n.
- if one person has an empty list, there is no stable matching, but there could be a non-empty locally stable matching.
- if all lists have at least 1 element and some at least 2, then there could exist a non-empty locally stable matching and/or a stable one.

1:	$\textcircled{3}$	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	$\textcircled{1}$	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	$\textcircled{2}$	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\textcircled{\boxed{5}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\textcircled{\boxed{4}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \ldots, (x_{r-1}, y_{r-1})$$

such that $y_i = \mathrm{first}_T(x_i), y_{i+1} = \mathrm{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \dots, (x_{r-1}, y_{r-1})$$

such that $y_i = \text{first}_T(x_i), y_{i+1} = \text{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

1:	$\textcircled{3}$	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	$\textcircled{1}$	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	$\textcircled{2}$	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\textcircled{\boxed{5}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\textcircled{\boxed{4}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \dots, (x_{r-1}, y_{r-1})$$

such that $y_i = \text{first}_T(x_i), y_{i+1} = \text{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

1:	$\textcircled{3}$	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	$\textcircled{1}$	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	$\textcircled{2}$	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\textcircled{\boxed{5}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\textcircled{\boxed{4}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

(1,3),(,2)

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \ldots, (x_{r-1}, y_{r-1})$$

such that $y_i = \mathrm{first}_T(x_i), y_{i+1} = \mathrm{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

1:	③	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	①	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	②	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\boxed{⑤}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\boxed{④}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

(1,3),(3,2)

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \dots, (x_{r-1}, y_{r-1})$$

such that $y_i = \text{first}_T(x_i), y_{i+1} = \text{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

1:	③	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	①	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	②	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\boxed{⑤}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\boxed{④}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

(1,3),(3,2),(,1)

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \dots, (x_{r-1}, y_{r-1})$$

such that $y_i = \text{first}_T(x_i), y_{i+1} = \text{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

1:	③	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	①	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	②	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\boxed{⑤}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\boxed{④}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

(1,3),(3,2),(2,1)

Irving's algorithm - Phase 2

Idea: Further reduction of the table T using rotations.

Definition (Irving 1985)

For a given table T, a sequence

$$\rho = (x_0, y_0), \dots, (x_{r-1}, y_{r-1})$$

such that $y_i = \text{first}_T(x_i), y_{i+1} = \text{second}_T(x_i)$ for all i (taken modulo r), is called a *rotation* exposed in T.

1:	$\textcircled{3}$	$\boxed{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	$\textcircled{1}$	$\boxed{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	$\textcircled{2}$	$\boxed{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	$\textcircled{\boxed{5}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	$\textcircled{\boxed{4}}$	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

(1,3),(3,2),(2,1)

- If T is a table in which some list contains at least two entries, then there is at least one rotation exposed in T.

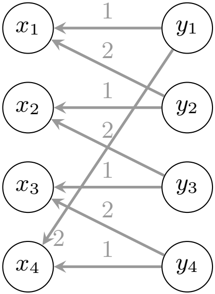
Irving's algorithm - Phase 2

How to use it ?

Conjecture

Let $\rho = (x_0, y_0), \dots, (x_{r-1}, y_{r-1})$ be a rotation exposed in a table T.

1. In any locally stable matching embedded in T, either x_i and y_i are partners for all values of i or for no value of i.
2. If there is a locally stable matching in which x_i and y_i are partners, then there is another one in which they are not.



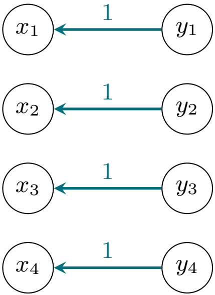
Irving's algorithm - Phase 2

How to use it ?

Conjecture

Let $\rho = (x_0, y_0), \ldots, (x_{r-1}, y_{r-1})$ be a rotation exposed in a table T.

1. In any locally stable matching embedded in T, either x_i and y_i are partners for all values of i or for no value of i.
2. If there is a locally stable matching in which x_i and y_i are partners, then there is another one in which they are not.



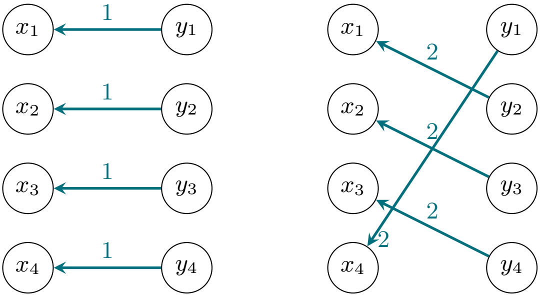
Irving's algorithm - Phase 2

How to use it ?

Conjecture

Let $\rho = (x_0, y_0), \ldots, (x_{r-1}, y_{r-1})$ be a rotation exposed in a table T.

1. In any locally stable matching embedded in T, either x_i and y_i are partners for all values of i or for no value of i.
2. If there is a locally stable matching in which x_i and y_i are partners, then there is another one in which they are not.



Irving's algorithm - Conclusion

In the example, (1,3),(3,2),(2,1) is a rotation:

1:	3	2	~~4~~	~~5~~	~~6~~
2:	1	3	~~4~~	~~5~~	~~6~~
3:	2	1	~~4~~	~~5~~	~~6~~
4:	5	~~1~~	~~2~~	~~3~~	~~6~~
5:	4	~~1~~	~~2~~	~~3~~	~~6~~
6:	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~

Irving's algorithm - Conclusion

In the example, (1,3),(3,2),(2,1) is a rotation:

1:	~~3~~	2	~~4~~	~~5~~	~~6~~
2:	~~1~~	3	~~4~~	~~5~~	~~6~~
3:	~~2~~	1	~~4~~	~~5~~	~~6~~
4:	5	~~1~~	~~2~~	~~3~~	~~6~~
5:	4	~~1~~	~~2~~	~~3~~	~~6~~
6:	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~

Irving's algorithm - Conclusion

In the example, (1,3),(3,2),(2,1) is a rotation:

1:	~~3~~	~~2~~	~~4~~	~~5~~	~~6~~
2:	~~1~~	~~3~~	~~4~~	~~5~~	~~6~~
3:	~~2~~	~~1~~	~~4~~	~~5~~	~~6~~
4:	5	~~1~~	~~2~~	~~3~~	~~6~~
5:	4	~~1~~	~~2~~	~~3~~	~~6~~
6:	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~

Irving's algorithm - Conclusion

In the example, (1,3),(3,2),(2,1) is a rotation:

1:	$\cancel{3}$	$\cancel{2}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
2:	$\cancel{1}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
3:	$\cancel{2}$	$\cancel{1}$	$\cancel{4}$	$\cancel{5}$	$\cancel{6}$
4:	5	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
5:	4	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{6}$
6:	$\cancel{1}$	$\cancel{2}$	$\cancel{3}$	$\cancel{4}$	$\cancel{5}$

Conjecture

At the end of Phase 2,

- if the table contains only empty lists, there is no stable matching and only the trivial empty locally stable matching.
- if the table contains only empty lists and lists with one element, there is no stable matching and the nonempty lists specify a maximum locally stable matching.
- if the table contains only lists with one element, they specify a stable matching which is also a maximum locally stable matching.

Future work

When graphs are incomplete, we think the same reasoning applies.

Future work

When graphs are incomplete, we think the same reasoning applies.

When ties are allowed in the preference lists,

- SMP : Deciding whether a bipartite graph has a stable marriage is polynomial.
- SRP : Deciding whether a graph has a stable matching is NP-complete.
- LSRP : What is the complexity of deciding whether a graph has a nonempty **locally** stable matching?

Future work

When graphs are incomplete, we think the same reasoning applies.

When ties are allowed in the preference lists,

- SMP : Deciding whether a bipartite graph has a stable marriage is polynomial.
- SRP : Deciding whether a graph has a stable matching is NP-complete.
- LSRP : What is the complexity of deciding whether a graph has a nonempty **locally** stable matching?

Only a partial answer for now.

Proposition

When ties are allowed in the preference lists, deciding whether a graph has a locally stable **perfect** matching is NP-complete.