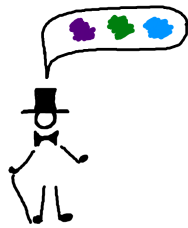
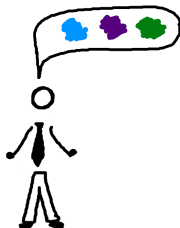
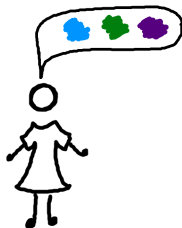


# House allocation with existing tenants and the stable roommates problem

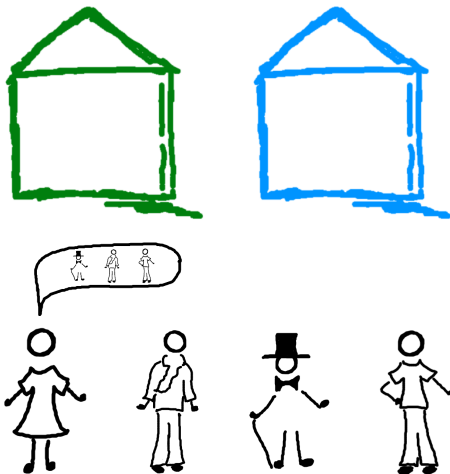
Katharina Schaar

Seminar  
Economics and Computation

5th June 2013



What is a good way of assigning applicants to houses?



What is a good method for matching roommates?

- 1 House allocation with existing tenants
  - Deficiency of algorithms typically used
  - Top trading cycle mechanism
  
- 2 The stable roommates problem
  - Irving's algorithm

- ▶ Atila Abdulkadiroğlu and Tayfun Sönmez  
House Allocation with Existing Tenants.  
*Journal of Economic Theory*, 88, 233-260 (1999).
- ▶ Robert W. Irving.  
An Efficient Algorithm for the “Stable Roommates” Problem.  
*Journal of Algorithms* 6, 577-595 (1985).

- 1 House allocation with existing tenants
  - Deficiency of algorithms typically used
  - Top trading cycle mechanism
  
- 2 The stable roommates problem
  - Irving's algorithm

## Some notation

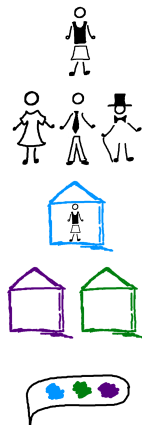
Set of existing tenants  $I_E$

Set of new applicants  $I_N$

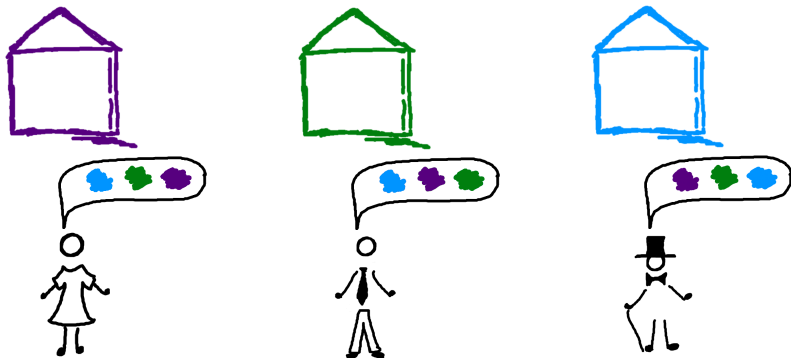
Set of occupied houses  $H_O$

Set of vacant houses  $H_V$

Preference relations  $P_i$  where  $i \in I_E \cup I_N$

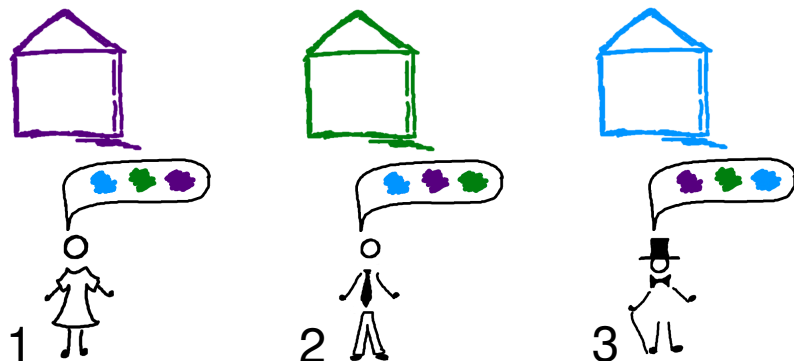


## Deficiency of algorithms typically used



**Random serial dictatorship:**

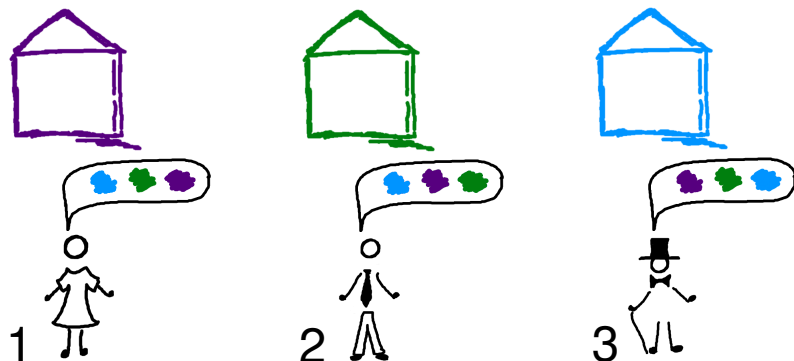




### Random serial dictatorship:

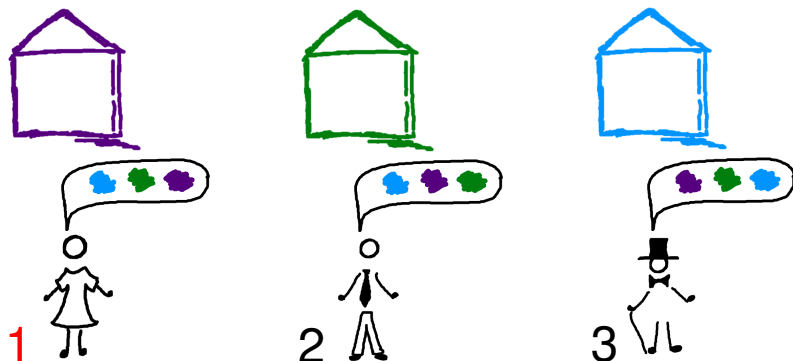
- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)

## Deficiency of algorithms typically used



### Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$



### Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$

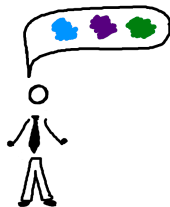


## Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$



2

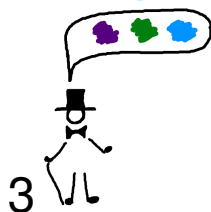


3



## Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$



## Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$



## Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$



### Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$





### Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$
- Stop when all houses are occupied, i. e. when  $H_V = \emptyset$



### Random serial dictatorship:

- Assign ordering to applicants  $I_N$  (randomly or according to some hierarchy)
- In this order: Assign each applicant  $i \in I_N$  his/her top choice among remaining houses  $H_V$
- Stop when all houses are occupied, i. e. when  $H_V = \emptyset$

## Short recap



## Short recap



**Pareto efficient**



**Individually rational**

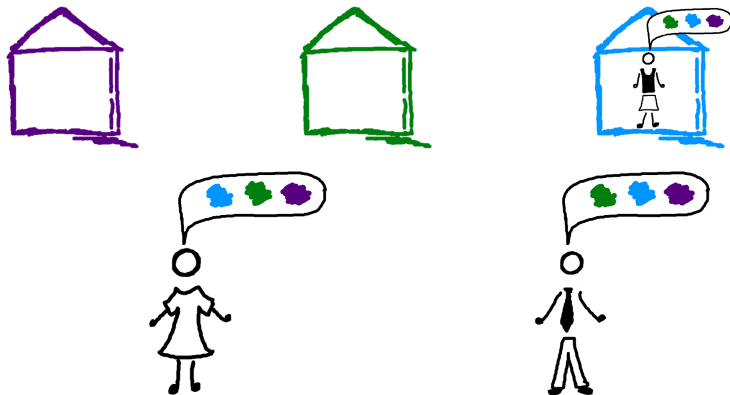


**Strategy-proof**



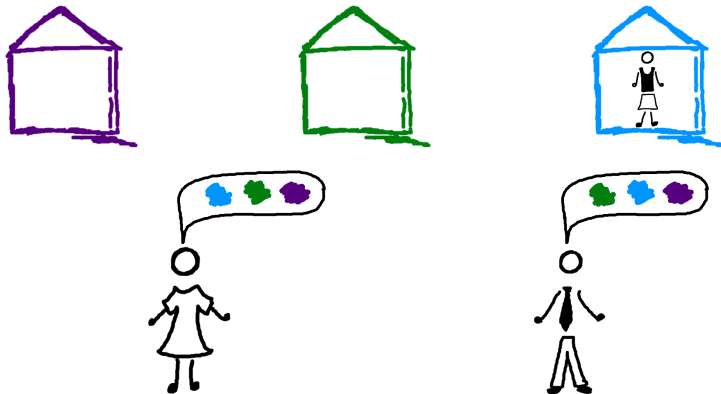
These three properties hold because there were NO existing tenants!

## Deficiency of algorithms typically used



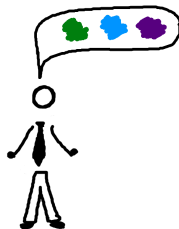
Typically existing tenants have to give up their current houses to apply for a new one.

## Deficiency of algorithms typically used



Typically existing tenants have to give up their current houses to apply for a new one.

## Deficiency of algorithms typically used



Typically existing tenants have to give up their current houses to apply for a new one.

## Deficiency of algorithms typically used



Typically existing tenants have to give up their current houses to apply for a new one.

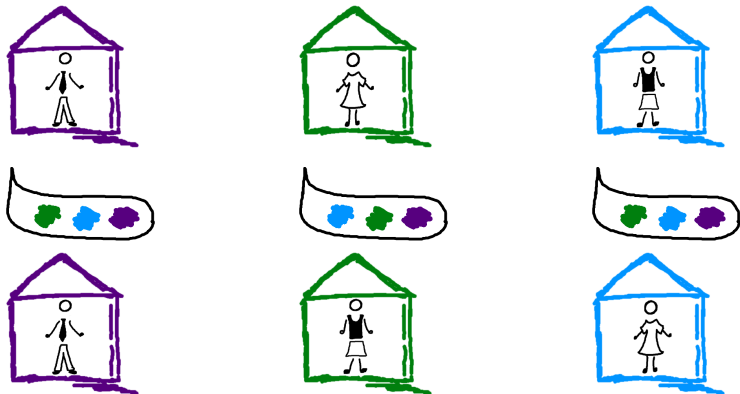


## Deficiency of algorithms typically used



Typically existing tenants have to give up their current houses to apply for a new one.

## Deficiency of algorithms typically used



Typically existing tenants have to give up their current houses to apply for a new one.






**Using RSD with existing tenants can cause non-Pareto-optimal outcomes!**

- Pareto efficient algorithms for *no* existing tenants might not be Pareto efficient for situations *with* existing tenants.
- In fact, most of the algorithms used in practice are not Pareto efficient if there are already tenants.
- Abdulkadiroğlu and Sönmez provide an algorithm that is Pareto efficient, individually rational, strategy-proof and can deal with existing tenants.

## The Algorithm

- 1 Assign ordering to all agents  $i \in I_E \cup I_N$ .
- 2 Each agent is assigned an arrow pointing to his/her favourite house among the remaining ones.
- 3 Each house is assigned an arrow pointing to...
  - ... the agent with the highest priority, if the house is vacant.
  - ... its current tenant, if the house is occupied.
- 4 There will be at least one cycle (if all sets are finite).
- 5 Pick a cycle and assign each agent in that cycle the house he/she points to.
- 6 Repeat until there are no more vacant houses or no more applicants.

## Top trading cycles mechanism

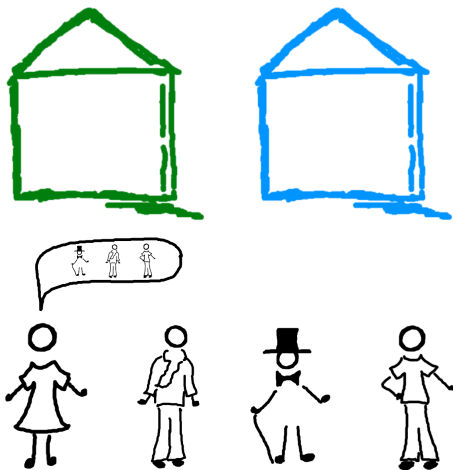
Agent	Current house	Preferences over houses
 $i_1$	1	2 6 5 1 4 3 7 0
 $i_2$	2	7 1 6 5 4 3 2 0
 $i_3$	3	2 1 4 7 3 6 5 0
 $i_4$	4	2 4 3 6 1 7 5 0
 $i_5$		4 3 7 1 2 5 6 0

The top trading cycle algorithm...

- ... is Pareto efficient,
- ... individually rational,
- ... strategy-proof and
- ... can easily accomodate seniorities.

- 1 House allocation with existing tenants
  - Deficiency of algorithms typically used
  - Top trading cycle mechanism
  
- 2 The stable roommates problem
  - Irving's algorithm

## The stable roommates problem



What is a good method for matching roommates?



## Connection to stable marriage problem



Set of women



Set of men

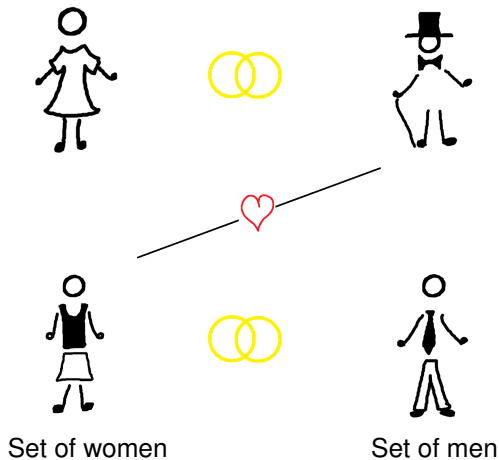
## Connection to stable marriage problem



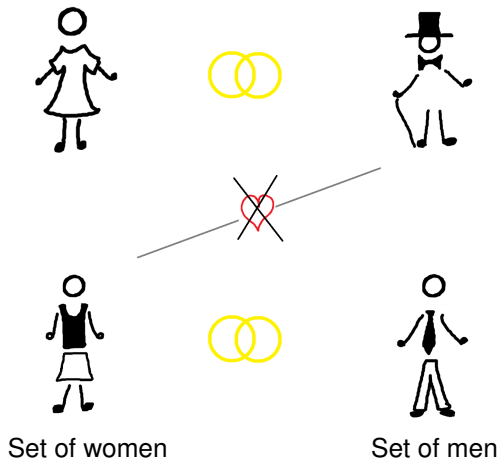
Set of women

Set of men

## Connection to stable marriage problem



## Connection to stable marriage problem



### **Irving's algorithm**

Determines if a stable matching exists and if so finds one.

**Input:** Preference lists of all participants over all other participants

**Phase 1:** Proposing

**Intermediate Step:** First reduction of preference lists

**Phase 2:** Reduction via all-or-nothing cycles







**Output:** Stable matching if there exists one,  
otherwise "No stable matching possible"

It runs in polynomial time.

## Phase 1 - Proposing

- 1 If  $x$  receives a proposal from  $y$  then...
  - ... he rejects it, if he already holds a better proposal,
  - ... otherwise he holds it for consideration.
- 2 Every participant proposes to the others in the order of his preference list.
- 3 A participant  $x$  does not propose while someone is considering the proposal from  $x$ .
- 4 The algorithm stops, if one person is rejected by everybody ("There is no stable matching").
- 5 This phase terminates when everybody holds a proposal.

**Phase 1 - Proposing**

Considers proposal from	Agent	Preference list/Proposes to
		
		
		
		
		
		

**Phase 1 - Proposing**

Considers proposal from	Agent	Preference list/Proposes to
	1	
	2	
	3	
	4	
	5	
	6	



**Phase 1 - Proposing**

Considers proposal from	Agent	Preference list/Proposes to
	1	
	2	
	3	
	4	
	5	
	6	

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	<b>1</b>	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3



## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3



## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3



## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
1	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3



## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
2	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3



## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3



## Phase 1 - Proposing

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

- 1 When a participant  $x$  receives a proposal from  $y$  that he considers, he deletes everybody he likes less than  $y$  from his preference list.
- 2 If  $z$  is deleted from  $x$ 's preference list,  $z$  for his part deletes  $x$  from his preference list.

**Intermediate step: First reduction of preference lists**

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3



## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3



## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

## Intermediate step: First reduction of preference lists

Considers proposal from	Agent	Preference list/Proposes to				
6	1	4	6	2	5	3
4	2	6	3	5	1	4
2	3	4	5	1	6	2
5	4	2	6	5	1	3
3	5	4	2	3	6	1
1	6	5	1	4	2	3

**Intermediate step: First reduction of preference lists**

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
4	2	3 5 4
2	3	5 2
5	4	2 5
3	5	4 2 3
1	6	1

## Phase 2: Reduction via all-or-nothing cycles

- 1 Find a cyclic sequence of distinct agents  $a_1, \dots, a_r$  such that the second person in  $a_i$ 's reduced preference list is the first one in  $a_{i+1}$ 's.  
→ all-or-nothing cycle
- 2 Make all agents holding proposals from agents in the cycle reject those proposals.
- 3 Terminate when all preference lists are reduced to only one person or return "There is no stable matching", if a preference list becomes empty.
- 4 If the algorithm terminates, then the preference lists specify the stable matching.

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
4	2	3 5 4
2	3	5 2
5	4	2 5
3	5	4 2 3
1	6	1

**Current Cycle:**

**Phase 2: Reduction via all-or-nothing cycles**

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
4	2	3 5 4
2	3	5 2
5	4	2 5
3	5	4 2 3
1	6	1

**Current Cycle: 3-4**

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
	2	3 5 4
2	3	5 2
5	4	2 5
	5	4 2 3
1	6	1

**Current Cycle: 3-4**



## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
	2	3 5 4
2	3	5 2
5	4	2 5
	5	4 2 3
1	6	1

**Current Cycle:** 3-4

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3 5 4
2	3	5 2
5	4	2 5
	5	4 2 3
1	6	1

Current Cycle: 3-4

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3 5 4
2	3	5 2
5	4	2 5
	5	4 2 3
1	6	1

Current Cycle: 3-4

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3 5 4
2	3	5 2
5	4	2 5
	5	4 2 3
1	6	1

Current Cycle: 3-4

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3
2	3	5    2
5	4	5
	5	4    3
1	6	1

**Current Cycle: 3-4**

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3
2	3	5 2
5	4	5
	5	4 3
1	6	1

Current Cycle: 3-4

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3
2	3	5 2
5	4	5
4	5	4 3
1	6	1

**Current Cycle:** 3-4

## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3
2	3	5 2
5	4	5
4	5	4 3
1	6	1

Current Cycle: 3-4



## Phase 2: Reduction via all-or-nothing cycles

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3
2	3	5 2
5	4	5
4	5	4 3
1	6	1

**Current Cycle: 3-4**

**Phase 2: Reduction via all-or-nothing cycles**

Considers proposal from	Agent	Preference list/Proposes to
6	1	6
3	2	3
2	3	2
5	4	5
4	5	4
1	6	1

**Current Cycle:**

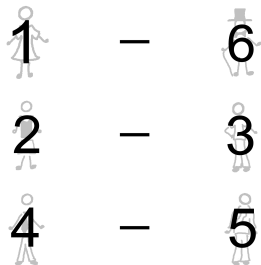
**Return stable matching:**

1 - 6

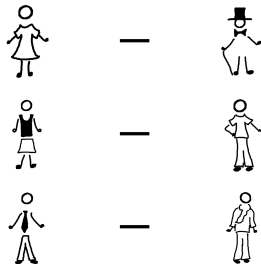
2 - 3

4 - 5

**Return stable matching:**



**Return stable matching:**



# Conclusion

