CS 598: AI Methods for Market Design

Lecture 8: Matching Markets

Xintong Wang

Spring 2024

Outline

- One-sided matching
- Two-sided matching
- Kidney-paired donation
- Project discussion

Outline

- One-sided matching
- Two-sided matching
- Kidney-paired donation
- Project discussion

One-Sided Matching

- Agents have strict preferences on items
- Items (indivisible) do not have preferences on agents
- No item is assigned more than once

One-Sided Matching

- Agents have strict preferences on items
- Items (indivisible) do not have preferences on agents
- No item is assigned more than once

Examples: assign classrooms to courses, dorm rooms to students, tasks to volunteers

Example: The Draw

The mechanism that assigns students to dorms

Example: The Draw

The mechanism that assigns students to dorms:

1. Each student submits a ranked list, ordering dorms from most to least preferred
2. Each student is assigned a number in {1,2,…,N}
3. For i = 1, 2, …, N:

   Student i is assigned to her favorite choice among options that are still available

Example: The Draw

The mechanism that assigns students to dorms:

1. Each student submits a ranked list, ordering dorms from most to least preferred
2. Each student is assigned a number in {1,2,…,N}
3. For i = 1, 2, …, N:

   Student i is assigned to her favorite choice among options that are still available

Is the Draw a good mechanism?

Example: The Draw

Strategy proof

Pareto optimal

Example: The Draw

Strategy proof: the property of a mechanism that being truthful is always the best strategy, i.e., lying about your preferences cannot make you better off

Pareto optimal: the property of an outcome that you can't make anyone better off without making someone else worse off

Example: The Draw

The Draw is strategy-proof and Pareto optimal.

Example: The Draw

The Draw is strategy-proof and Pareto optimal.

Proof (strategy-proof):

(1) The report of agent i will not affect agents before her (i.e., agents with better priority).

(2) By truthful report, agent i will receive the most preferred item of those still available.

Example: The Draw

The Draw is strategy-proof and Pareto optimal.

Proof (Pareto optimal):

Prove by induction and contradiction. Assume there's an assignment X' that Pareto dominates current X.

(1) Base: i=0, both empty assignment $X'_{(0)} = X_{(0)}$

(2) Inductive hypothesis: the first i-1 students are assigned identically in X' and X

(3) Inductive step: in X', student i must also get her favorite option among those remaining, so X'=X.

Serial Dictatorship

The mechanism used in the Draw is called serial dictatorship

Serial dictatorship

- Order the agents
- In this order, allow each agent to dictate their favorite feasible option

Serial Dictatorship

The mechanism used in the Draw is called serial dictatorship

Serial dictatorship

- Order the agents
- In this order, allow each agent to dictate their favorite feasible option

How about fairness?

Outline

- One-sided matching
- Two-sided matching
- Kidney-paired donation
- Project discussion

Two-Sided Matching

- Two sets of agents, with each member in one set having strict preferences over each member of the other
- A matching: each agent is assigned to at most one agent on the other side

Examples: college admissions, medical students to residencies, job market, dating apps…

Example: College Admission

What do you think about current system?

Things to consider as an applicant:

- How many colleges to apply to?
- Should I apply for early admission?
- Should I accept an offer or wait for my waitlist?

Example: College Admission

What do you think about current system?

Things to consider as an applicant:

- How many colleges to apply to?
- Should I apply for early admission?
- Should I accept an offer or wait for my waitlist?

Things to consider as a college admission officer:

- How can I get good students?
- How can I get the right number of students?

Example: National Resident Matching Program (NRMP)

- 1900-1945: matching in an ad hoc, decentralized way
  - In 1945, residency offers are extended to medical students by the end of their first year!
  - Unraveling: make offers early to get strong candidates

Example: National Resident Matching Program (NRMP)

- 1900-1945: matching in an ad hoc, decentralized way
  - In 1945, residency offers are extended to medical students by the end of their first year!
  - Unraveling: make offers early to get strong candidates
- 1945: release admission decision on the same date, early in the final year of medical school
  - First choice declines, and all other good candidates accept offers from other programs
  - Exploding offers

Example: National Resident Matching Program (NRMP)

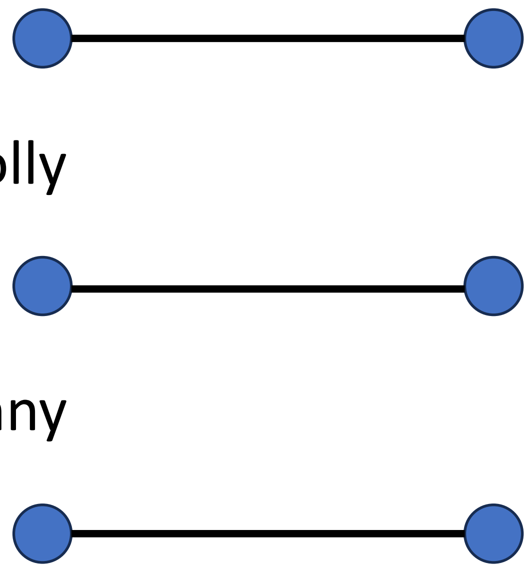
- 1900-1945: matching in an ad hoc, decentralized way
  - In 1945, residency offers are extended to medical students by the end of their first year!
  - Unraveling: make offers early to get strong candidates
- 1945: release admission decision on the same date, early in the final year of medical school
  - First choice declines, and all other good candidates accept offers from other programs
  - Exploding offers
- 1952 until today: a centralized matching algorithm

Illustrative Example

Boys		Girls
Jake		Claire
Jenny > Claire > Holly		Jake > Ray > Ed
Ed		Jenny
Claire > Holly > Jenny		Ray > Jake > Ed
Ray		Holly
Claire > Jenny > Holly		Jake > Ray > Ed

Illustrative Example

Boys | Girls



Jake — Claire

Jenny > Claire > Holly | Jake > Ray > Ed

Ed — Jenny

Claire > Holly > Jenny | Ray > Jake > Ed

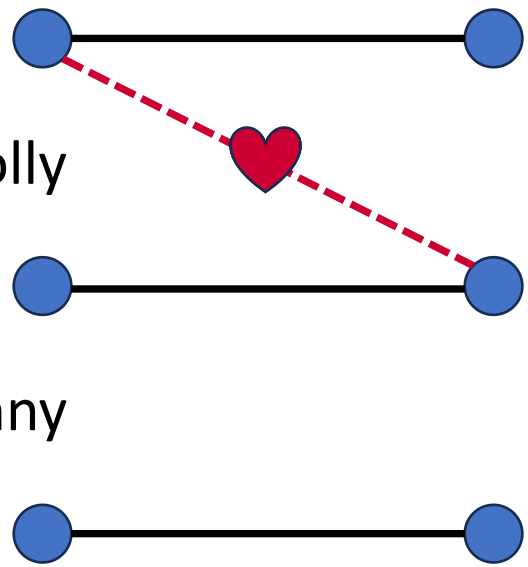
Ray — Holly

Claire > Jenny > Holly | Jake > Ray > Ed

What is the problem with the current matching?

Illustrative Example

Boys		Girls
Jake		Claire
Jenny > Claire > Holly		Jake > Ray > Ed
Ed		Jenny
Claire > Holly > Jenny		Ray > Jake > Ed
Ray		Holly
Claire > Jenny > Holly		Jake > Ray > Ed



What is the problem with the current matching?

Stable Matching

- A matching: each agent assigned to at most one agent on the other side

- A stable matching is a matching with **no blocking pair**
- A blocking pair: two agents who prefer each other to their assigned role in the matching

Stable Matching

- Do stable matchings exist?
- Are they easy to find?
- Are stable matching unique?
- Does stability matter?

We'll study these questions through the Gale-Shapley (1962) *deferred acceptance* (DA) algorithm

Boy-Proposing Deferred Acceptance

The boy-proposing DA proceeds in rounds:

- (Round 1)

Each boy proposes to their most preferred girl.

Each girl *tentatively* accepts the most preferred proposal and rejects the rest.

- (Round r > 1)

Each boy whose proposal was rejected in the previous round makes a proposal to their *next* most preferred girl.

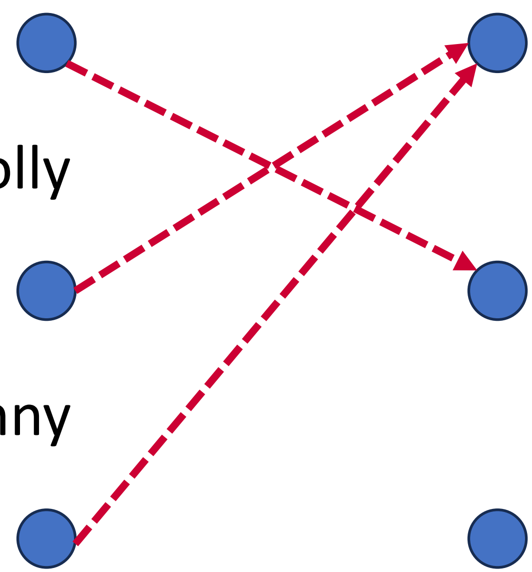
Each girl who has received a new proposal *tentatively* accepts the most preferred proposal so far and rejects the rest.

Terminates when no new proposals are made, and tentative matches become final.

Boy-Proposing Deferred Acceptance

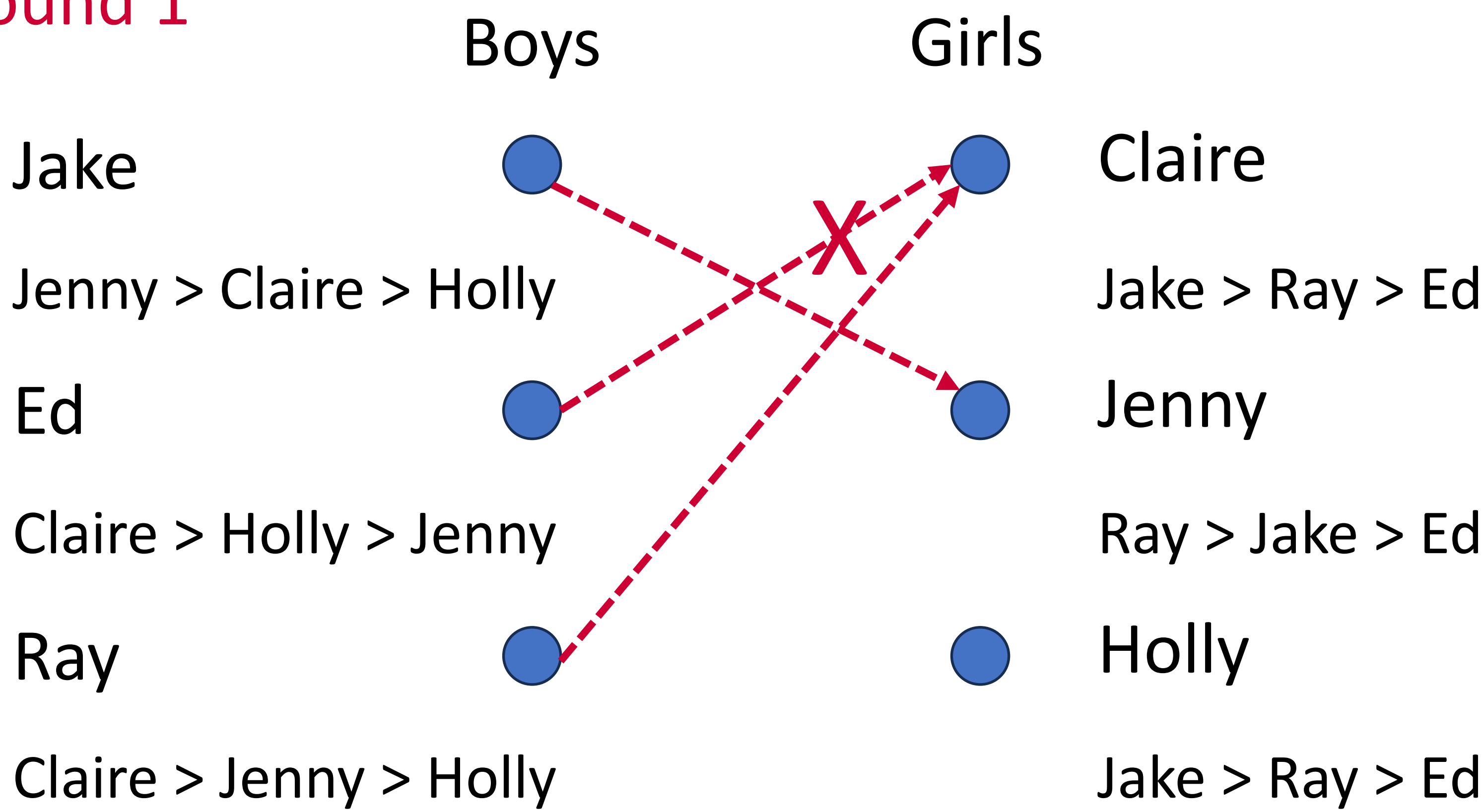
Round 1

Boys		Girls	
Jake	Jenny > Claire > Holly	Claire	Jake > Ray > Ed
Ed	Claire > Holly > Jenny	Jenny	Ray > Jake > Ed
Ray	Claire > Jenny > Holly	Holly	Jake > Ray > Ed



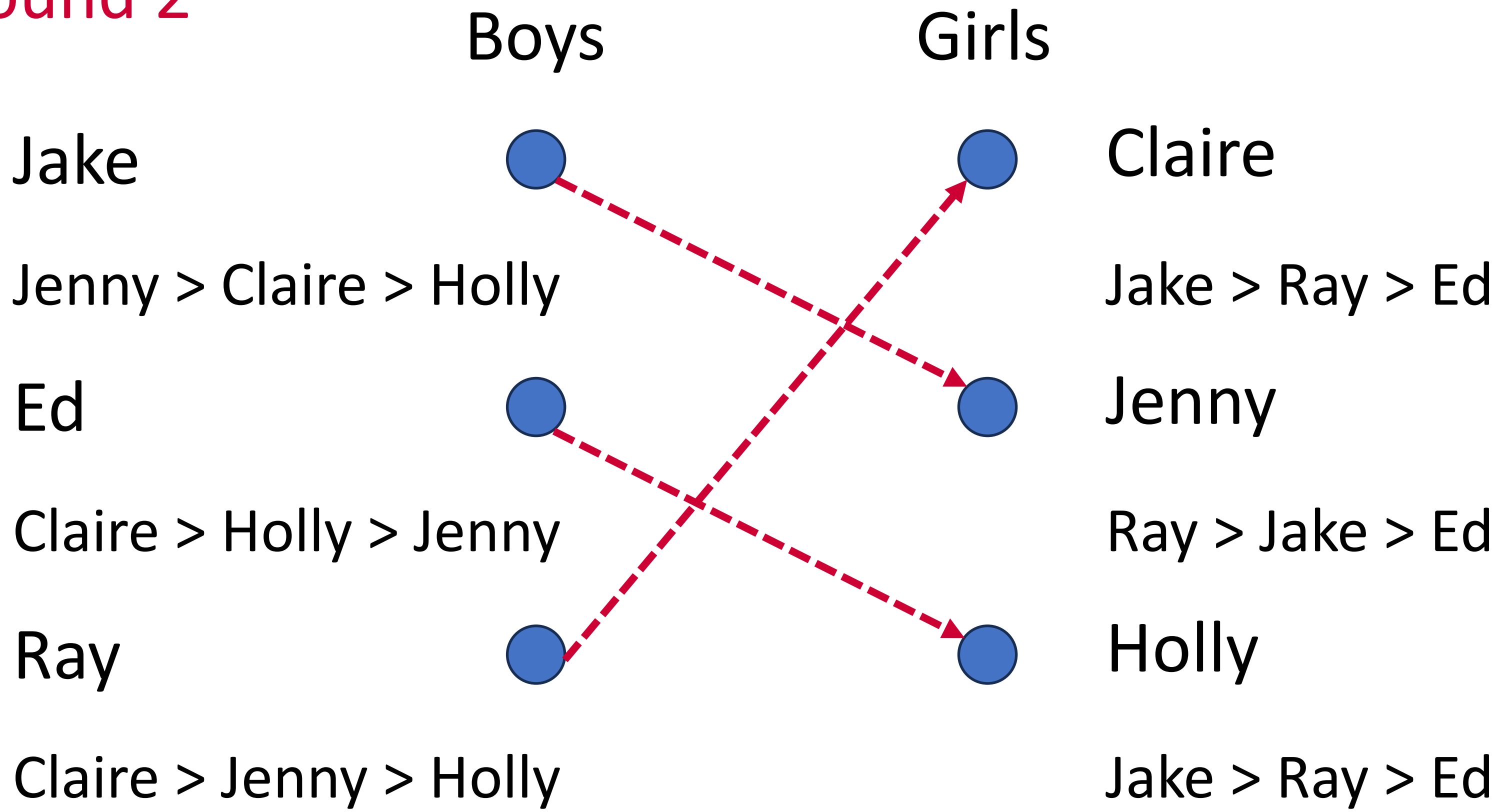
Boy-Proposing Deferred Acceptance

Round 1



Boy-Proposing Deferred Acceptance

Round 2



Boy-Proposing Deferred Acceptance

Final

Boys | Girls

Jake

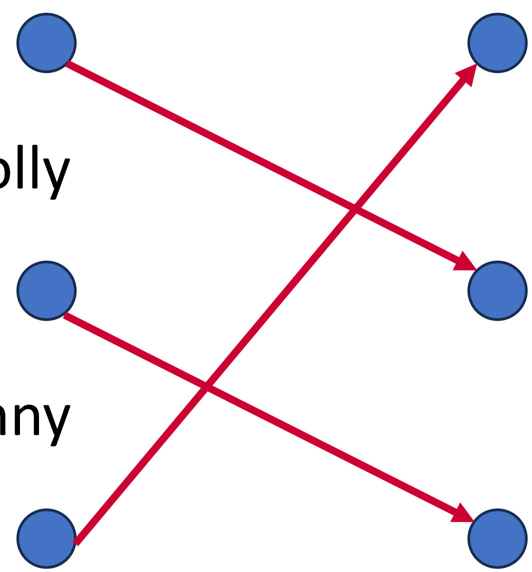
Jenny > Claire > Holly

Ed

Claire > Holly > Jenny

Ray

Claire > Jenny > Holly



Claire

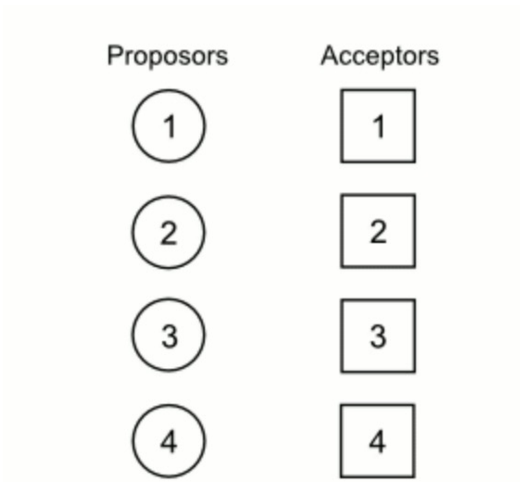
Jake > Ray > Ed

Jenny

Ray > Jake > Ed

Holly

Jake > Ray > Ed



Preferences

□ → ○

Acceptor Table

□1	1	3	2	4
□2	3	4	1	2
□3	4	2	3	1
□4	3	2	1	4

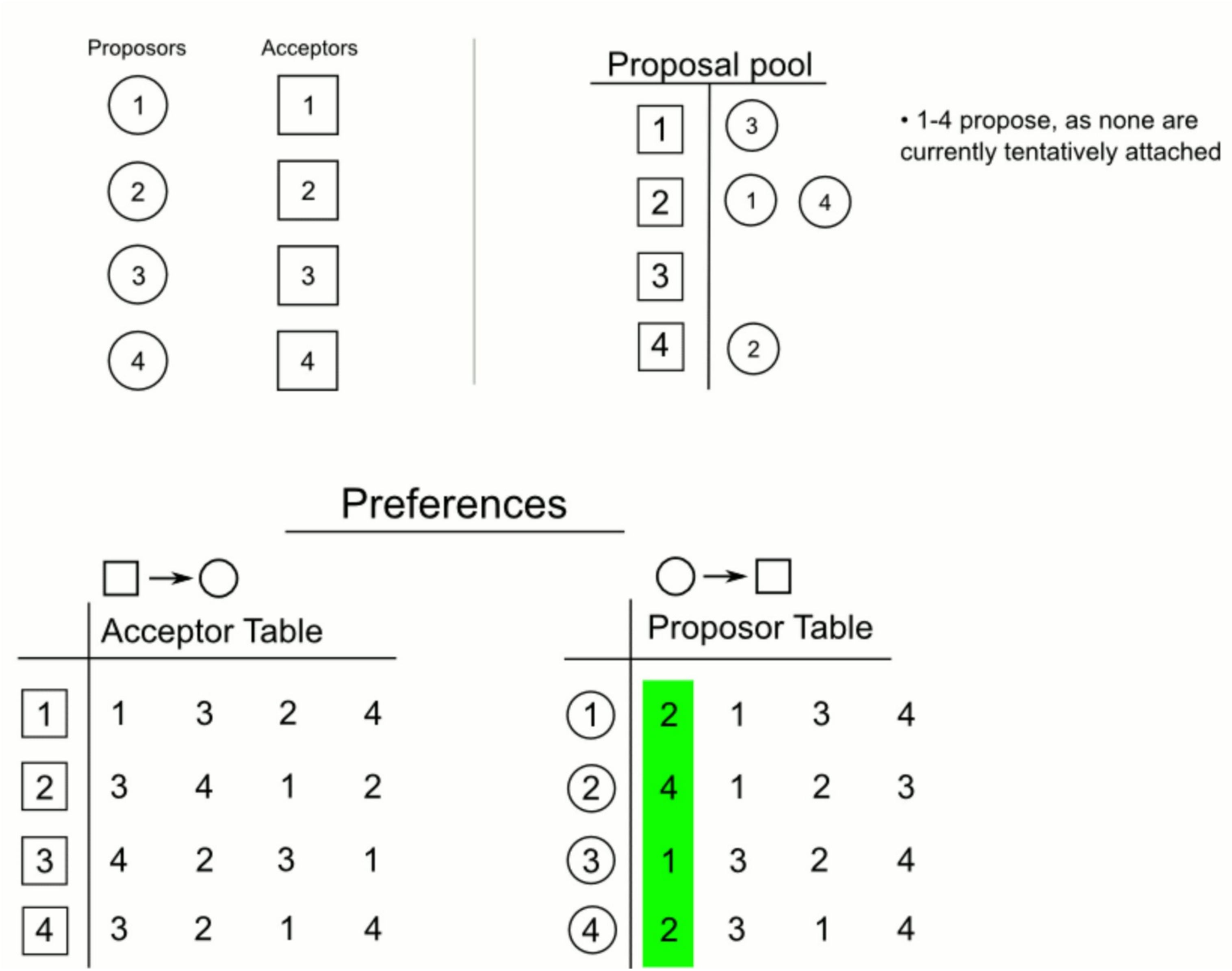
○ → □

Proposor Table

○1	2	1	3	4
○2	4	1	2	3
○3	1	3	2	4
○4	2	3	1	4

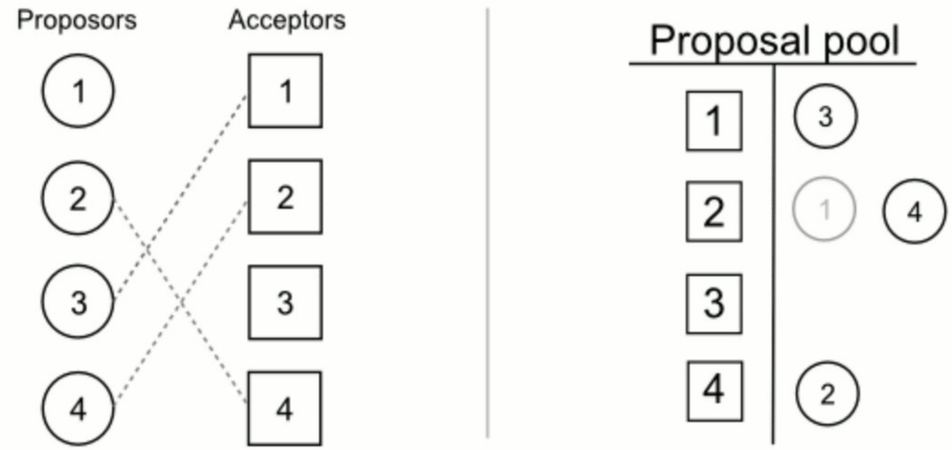
Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Round 1



Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Round 1



- 1 accepts 3's proposal — no better offer.
- 2 accepts 4's proposal as 4 is more prefereable to 1.
- 3 recieves no offer.
- 4 accepts 2's proposal — no better offer.

Preferences

□ → ○ Acceptor Table

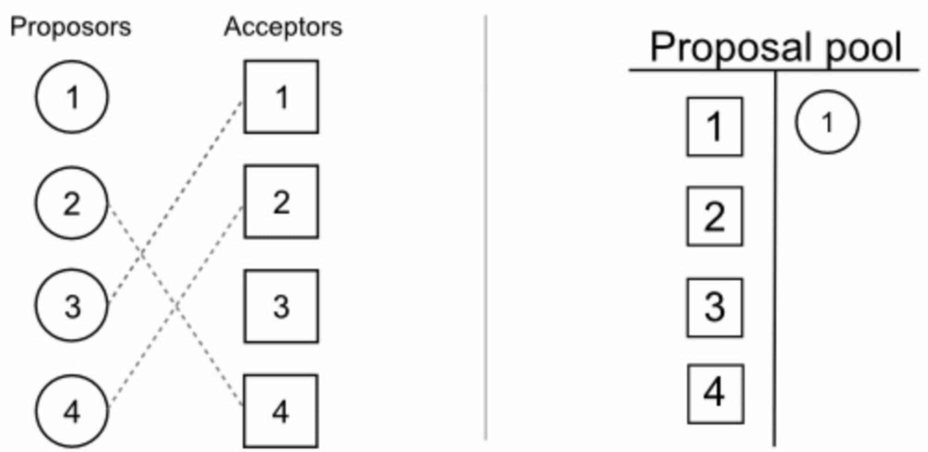
1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

○ → □ Proposor Table

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Round 2



- 1, the only un-attached membe makes its offer to 1, its first preference not previously proposed to.

Preferences

□ → ○ Acceptor Table

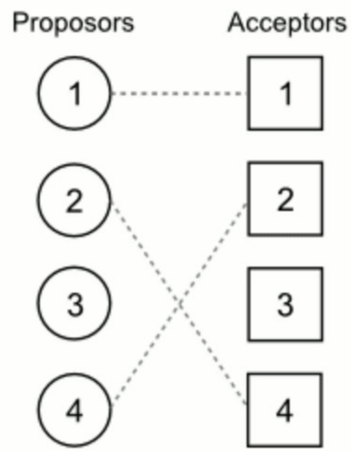
1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

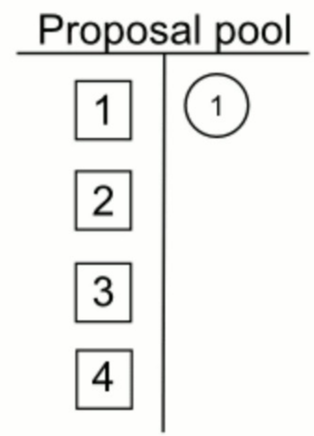
○ → □ Proposor Table

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Round 2





- 1 drops 3's propsal in favour of as this is higher in its preference table. 3 returns to the proposal pool.

Preferences

Acceptor Table (□ → ○)

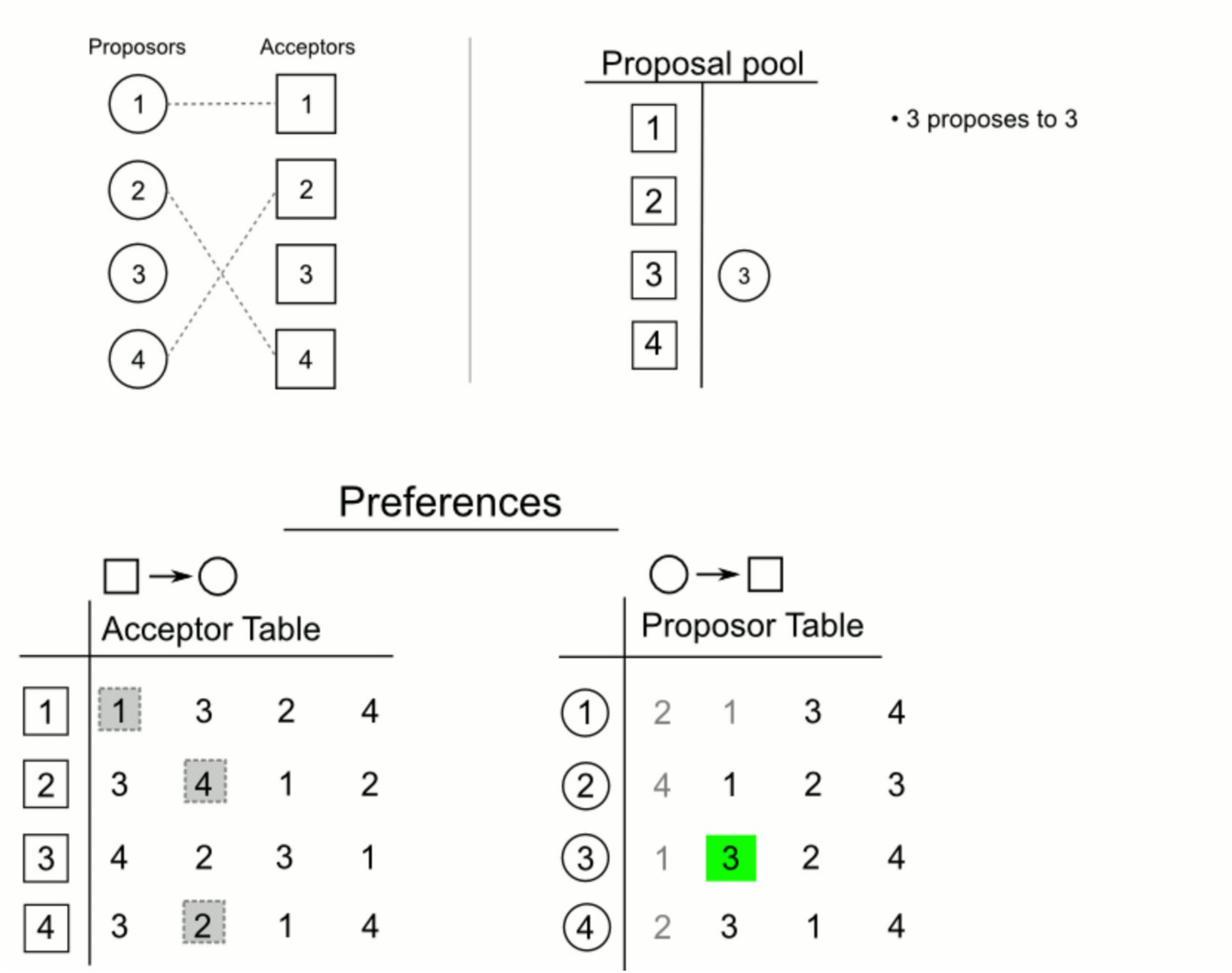
1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

Proposor Table (○ → □)

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

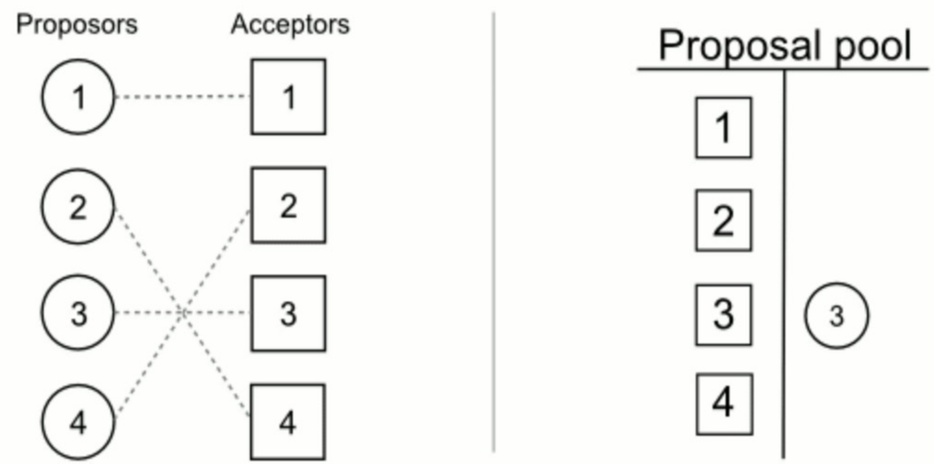
Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Round 3



Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Round 3



- 3 accepts 3, not having a better offer

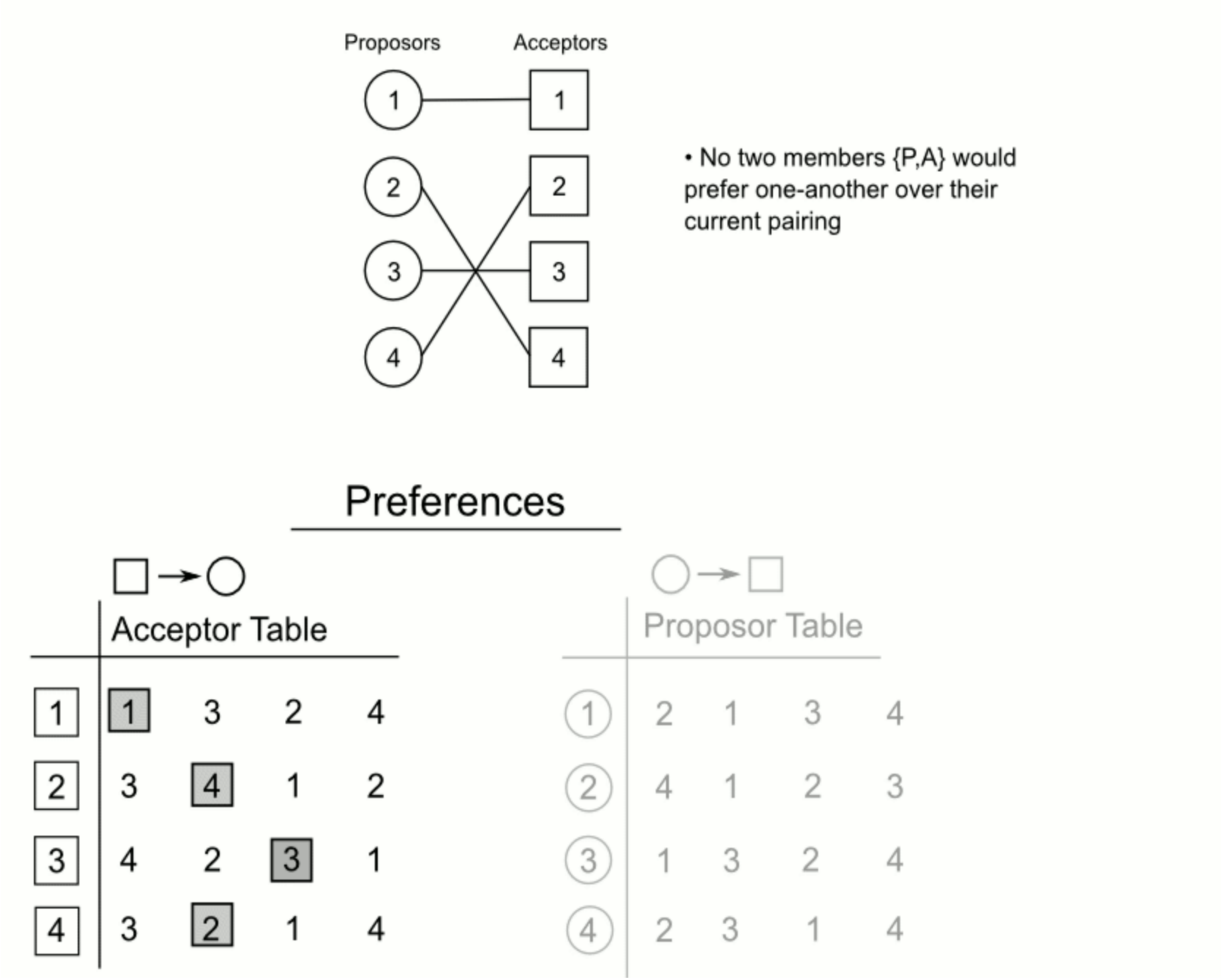
Preferences

Acceptor Table (□ → ○)				
1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

Proposor Table (○ → □)				
1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Final



Source: https://en.wikipedia.org/wiki/Gale-Shapley_algorithm

Boy-Proposing Deferred Acceptance

The boy-proposing DA proceeds in rounds:

- (Round 1)

Each boy proposes to their most preferred girl.

Each girl *tentatively* accepts the most preferred proposal and rejects the rest.

- (Round r > 1)

Each boy whose proposal was rejected in the previous round makes a proposal to their *next* most preferred girl.

Each girl who has received a new proposal *tentatively* accepts the most preferred proposal so far and rejects the rest.

Terminates when no new proposals are made, and tentative matches become final.

Stable Matching

- Do stable matchings exist?
- Are they easy to find?
- Does stability matter?
- Are stable matching unique?

We'll study these questions through the Gale-Shapley (1962) *deferred acceptance* (DA) algorithm

Analysis: Boy-Proposing DA

Fact 0:

Each girl is matched with a weakly more preferred boy across each round.

Intuition:

By design, girls only accept a new offer if it is better than the current offer they hold (if any).

Analysis: Termination of DA

Fact 1:

Deferred acceptance terminates.

Intuition:

In any round $r > 1$, at least one proposal was rejected in the previous round.

No proposal is repeated and there is a finite number of proposals.

Analysis: Existence of Stable Matching

Fact 2:

The boy-proposing DA algorithm terminates with a stable matching

Proof by contradiction:

- Suppose (b, g’) is a *blocking pair* in current DA matching with {(b, g), (b’, g’), …}
- Because b prefers g’ to g, then b must have proposed to g’ before g
- Because g’ is paired with b’, then g’ prefers b’ to b
- So (b, g’) is not a blocking pair to {(b, g), (b’, g’), …}

Analysis: Computation of DA

Fact 3:

The DA algorithm runs in $O(mn)$ rounds for m boys and n girls.

Intuition:

Each boy makes proposals in order of their strict preferences, and keeps track of the girls who have rejected them.

This requires at most n constant-time updates for each boy.

Stable Matching

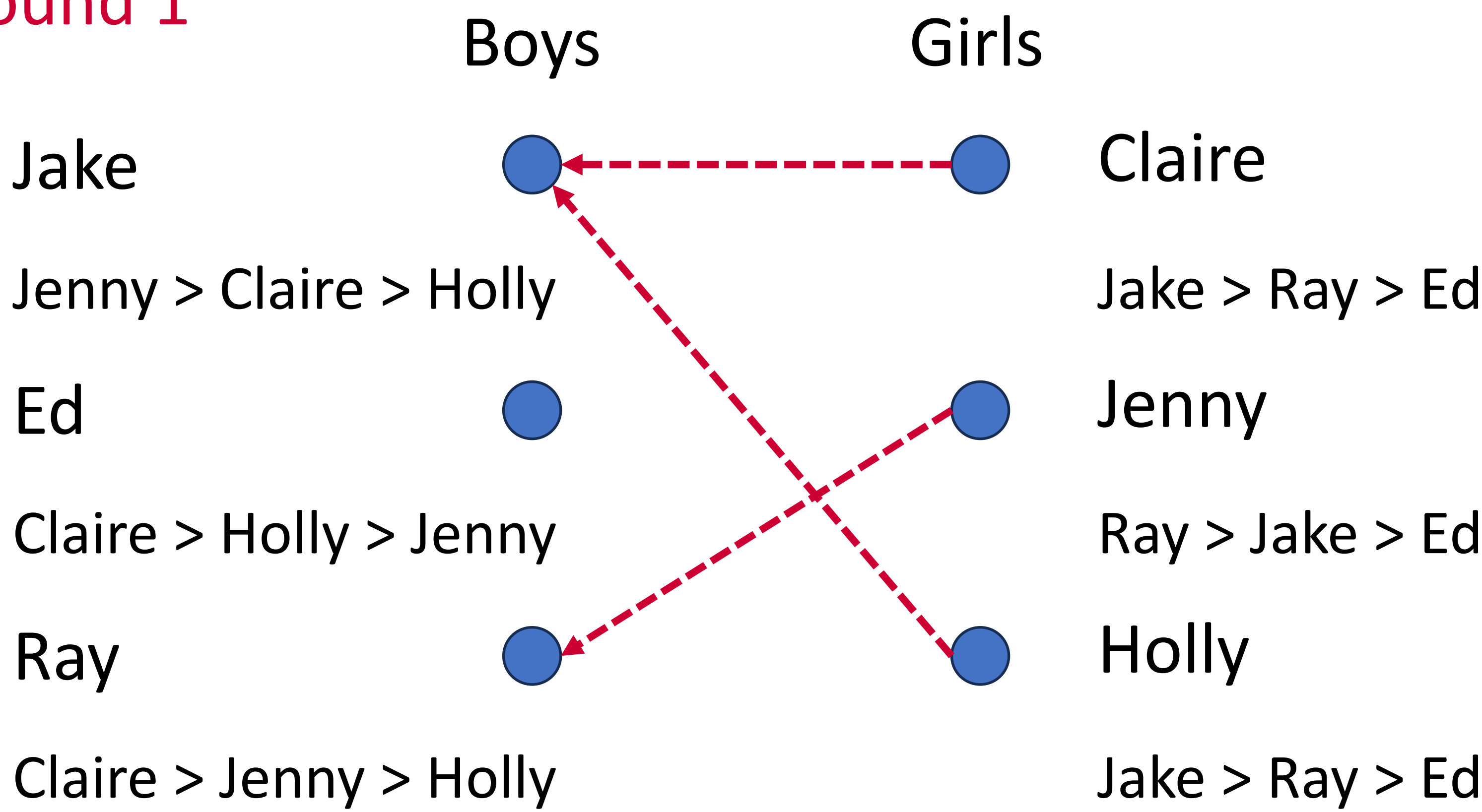
- Do stable matchings exist? YES
- Are they easy to find? YES
- Are stable matching unique?
- Does stability matter?

We'll study these questions through the Gale-Shapley (1962) *deferred acceptance* (DA) algorithm

Girl-Proposing Deferred Acceptance

Girl-Proposing Deferred Acceptance

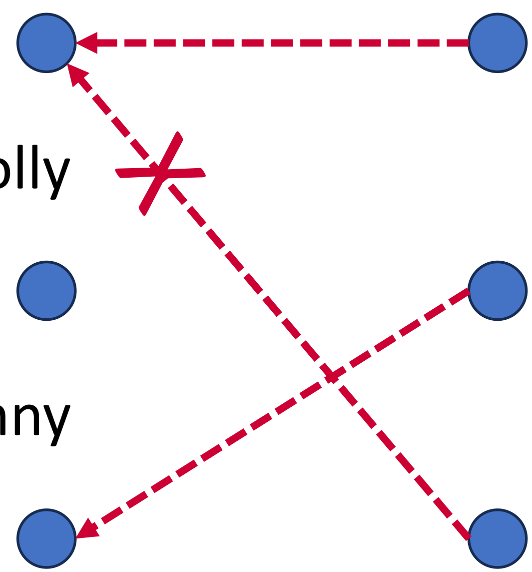
Round 1



Girl-Proposing Deferred Acceptance

Round 1

Boys | Girls



Jake

Jenny > Claire > Holly

Ed

Claire > Holly > Jenny

Ray

Claire > Jenny > Holly

Claire

Jake > Ray > Ed

Jenny

Ray > Jake > Ed

Holly

Jake > Ray > Ed

Girl-Proposing Deferred Acceptance

Round 2

Boys

Girls

Jake

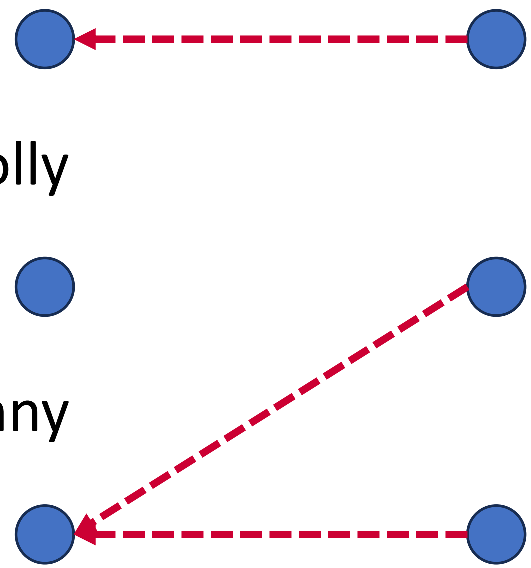
Jenny > Claire > Holly

Ed

Claire > Holly > Jenny

Ray

Claire > Jenny > Holly



Claire

Jake > Ray > Ed

Jenny

Ray > Jake > Ed

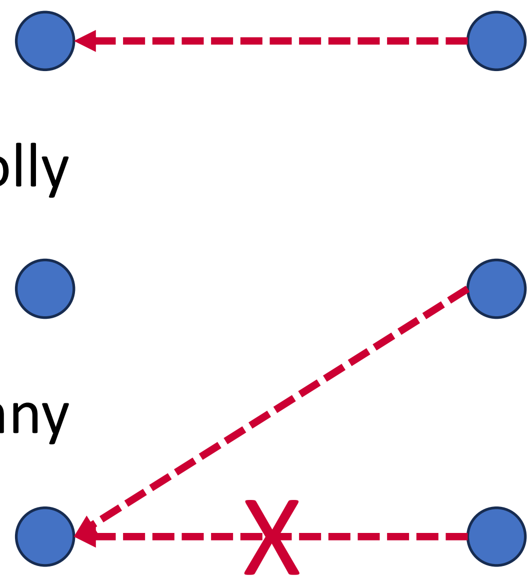
Holly

Jake > Ray > Ed

Girl-Proposing Deferred Acceptance

Round 2

Boys | Girls



Jake

Jenny > Claire > Holly

Ed

Claire > Holly > Jenny

Ray

Claire > Jenny > Holly

Claire

Jake > Ray > Ed

Jenny

Ray > Jake > Ed

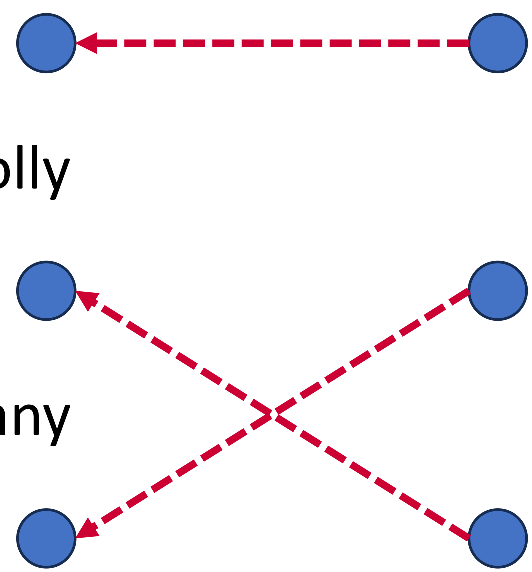
Holly

Jake > Ray > Ed

Girl-Proposing Deferred Acceptance

Round 3

Boys | Girls



Jake — Jenny > Claire > Holly

Ed — Claire > Holly > Jenny

Ray — Claire > Jenny > Holly

Claire — Jake > Ray > Ed

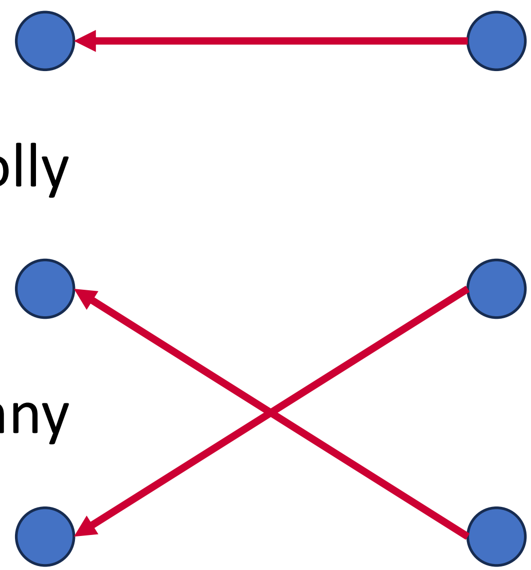
Jenny — Ray > Jake > Ed

Holly — Jake > Ray > Ed

Girl-Proposing Deferred Acceptance

Final

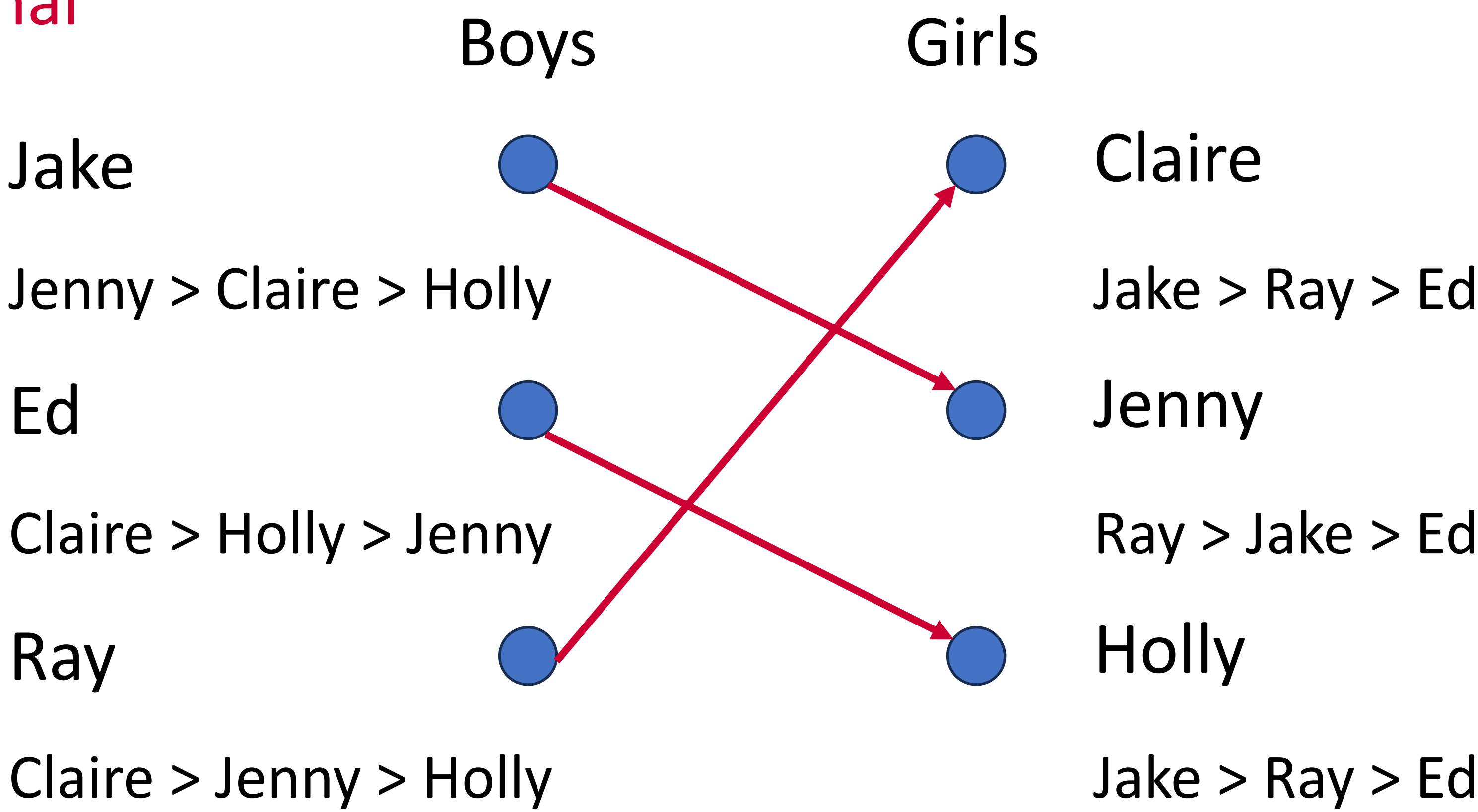
Boys | Girls



Boys	Girls
Jake	Claire
Jenny > Claire > Holly	Jake > Ray > Ed
Ed	Jenny
Claire > Holly > Jenny	Ray > Jake > Ed
Ray	Holly
Claire > Jenny > Holly	Jake > Ray > Ed

Boy-Proposing Deferred Acceptance

Final



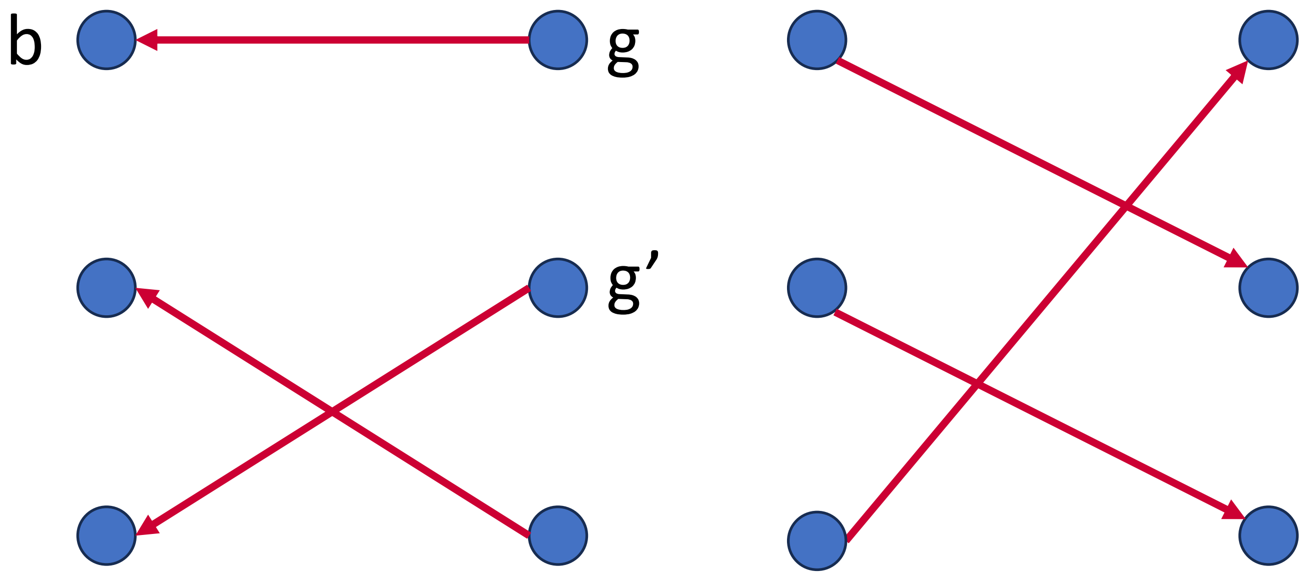
Stable Matching

- Do stable matchings exist? YES
- Are they easy to find? YES
- Are stable matching unique? NO, then who propose
- Does stability matter?

We’ll study these questions through the Gale-Shapley (1962) *deferred acceptance* (DA) algorithm

Achievable Outcomes

Girl g is achievable for b if b and g match in *some* stable matching.



E.g., {g, g'} are achievable for b

Strategic Analysis: Who Propose

Given truthful reports, in boy-proposing DA:

1. Each boy matches with his most preferred, achievable girl
2. Each girl is matched to her least preferred, achievable boy

And vice versa for girl-proposing DA

Strategic Analysis: Who Propose

Given truthful reports, in boy-proposing DA:

1. Each boy matches with his most preferred, achievable girl

Proof by contradiction:

- Assume b is rejected by his most preferred, achievable g who is in favor of b’
- By achievable outcome, exists {(b, g), (b’, g’)} for some g’
- Since b’ prefers g, (b’, g) is a blocking pair. Not stable.

Strategic Analysis: Who Propose

Given truthful reports, in boy-proposing DA:

2. Each girl is matched to her least preferred, achievable boy

Prove by contradiction:

- Given (b, g), assume b' is more preferred than b, then b will be rejected
- Boy b will not be an achievable boy for g

Strategic Analysis: Who Propose

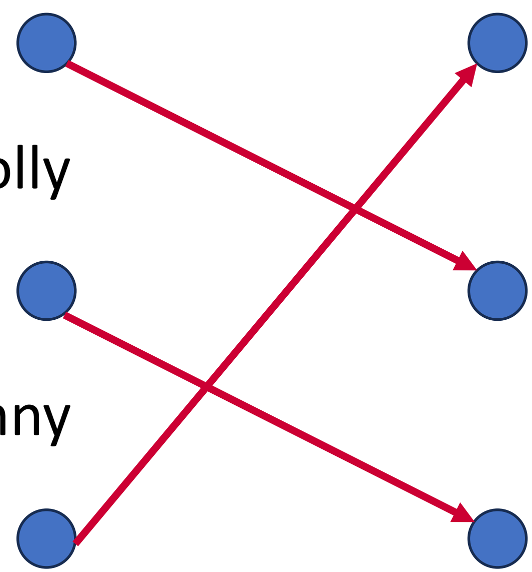
- Is truthful reporting a dominant strategy for proposers?
- YES! Proof Sketch:
  - If truthful, boy b is matched to his most-preferred, achievable girl
  - You cannot do better

Strategic Analysis: Who Propose

- Is truthful reporting a dominant strategy for proposers?
- YES! Proof Sketch:
  - If truthful, boy b is matched to his most-preferred, achievable girl
  - You cannot do better
- Is truthful reporting a dominant strategy for acceptors?
- NO! Let's look at an example…

Strategic Analysis: Who Propose

Boys		Girls	
Jake	Jenny > Claire > Holly	Claire	Jake > Ray > Ed
Ed	Claire > Holly > Jenny	Jenny	Ray > Jake > Ed
Ray	Claire > Jenny > Holly	Holly	Jake > Ray > Ed

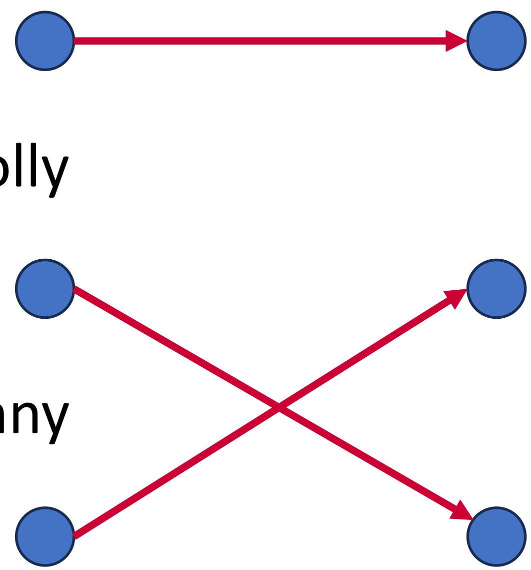


Strategic Analysis: Who Propose

Boys	Girls
Jake	Claire
Jenny > Claire > Holly	Jake > Ed > Ray ~~Jake > Ray > Ed~~
Ed	Jenny
Claire > Holly > Jenny	Ray > Jake > Ed
Ray	Holly
Claire > Jenny > Holly	Jake > Ray > Ed

Strategic Analysis: Who Propose

Boys Girls



Jake

Jenny > Claire > Holly

Ed

Claire > Holly > Jenny

Ray

Claire > Jenny > Holly

Claire

Jake > Ed > Ray

~~Jake > Ray > Ed~~

Jenny

Ray > Jake > Ed

Holly

Jake > Ray > Ed

Strategic Analysis: Who Propose

- Is truthful reporting a dominant strategy for proposers?
- YES! Proof Sketch:
  - If truthful, boy b is matched to his most-preferred, achievable girl
  - You cannot do better
- Is truthful reporting a dominant strategy for acceptors?
- NO! No matching mechanism is stable and (fully) strategy-proof ☹

Real-World Matching Markets

Hospital-proposing

Student-proposing (w/ two-body problem)

Market	Stable	Still in use (stopped unraveling)
U.S. NRMP ('52,'98)	yes	yes
Edinburgh ('69)	yes	yes
Cardiff	yes	yes
Birmingham	no	no
Edinburgh ('67-'69)	no	no
Newcastle	no	no
Sheffield	no	no
Cambridge	*no*	*yes*
London Hospital	*no*	*yes*
U.S. medical specialties	yes	yes (∼30 markets, *1 failure*)
U.S. Osteopaths (<'94)	no	no
U.S. Osteopaths (≥'94)	yes	yes

Stable Matching

- Do stable matchings exist? YES
- Are they easy to find? YES
- Are stable matching unique? NO, then who propose
- Does stability matter? YES

We'll study these questions through the Gale-Shapley (1962) *deferred acceptance* (DA) algorithm

Outline

- One-sided matching
- Two-sided matching
- Kidney-paired donation
- Project discussion

Kidney-Paired Donation

- Kidney failure is a serious medical problem
- Preferred treatment: kidney transplant
  - Cadaver kidneys or live kidney donation
  - Match based on blood-type and tissue-type compatibility

Kidney-Paired Donation

Waiting list candidates as of 03/07/2024

88,942 people

are waiting for a kidney transplant in the US.

In 2023,

39,680 patients received cadaver kidneys

6,950 patients received living donor kidneys

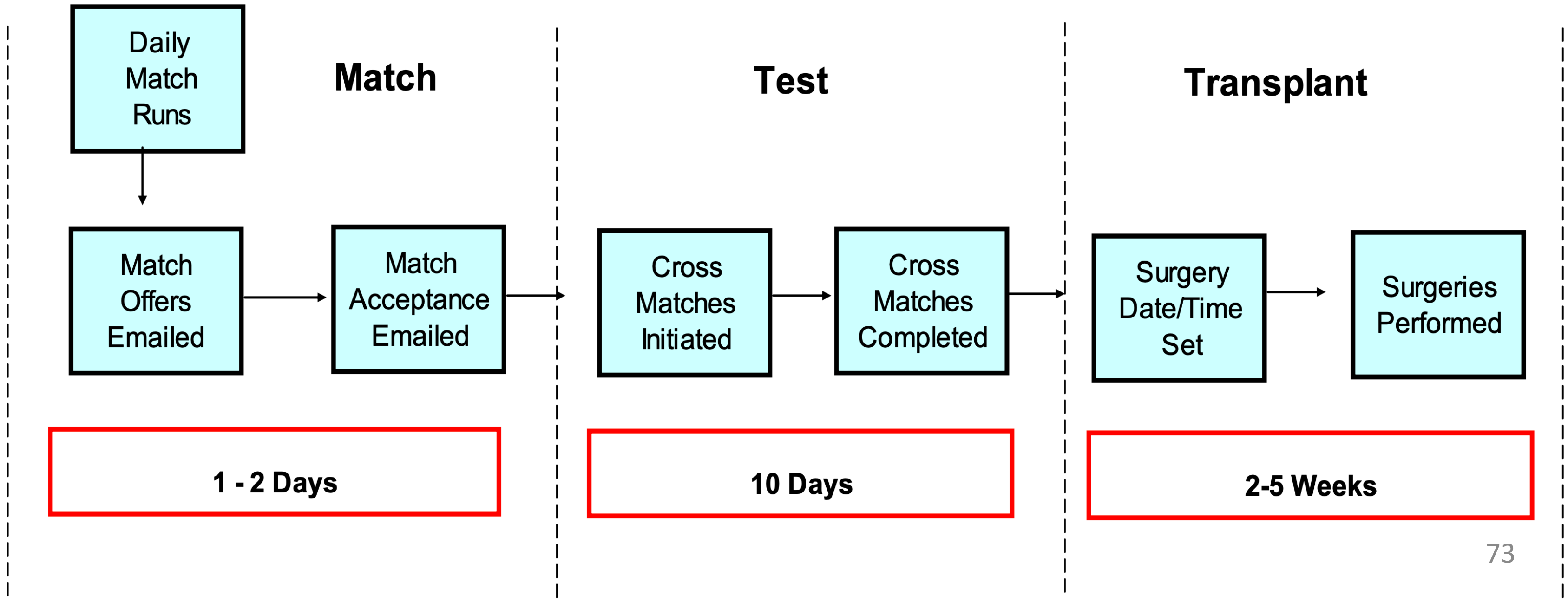
https://optn.transplant.hrsa.gov/data/view-data-reports/national-data/#

Kidney-Paired Donation

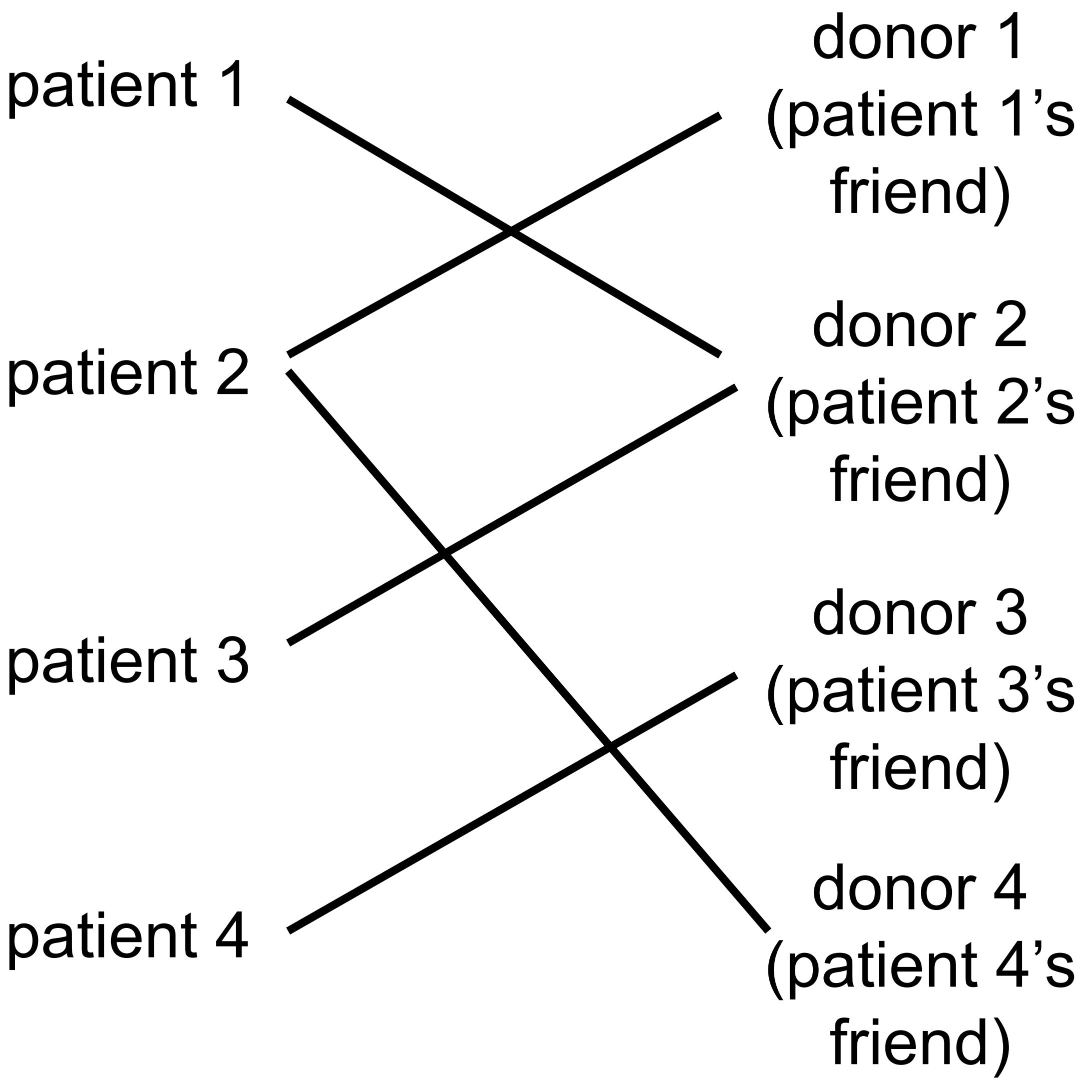
- Incompatible pairs arrive at the matching market
  - (Donor, Patient)
  - Participate in swaps or cycles
  - E.g., (Sick with blood type A, Healthy with blood type B)
    (Sick with blood type B, Healthy with blood type A)

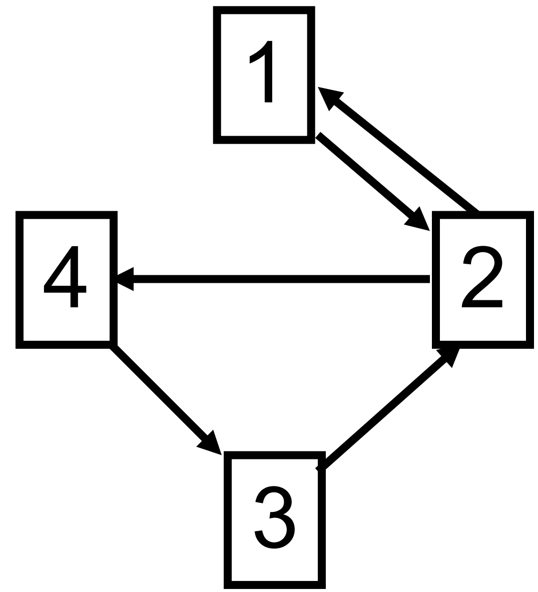
Kidney-Paired Donation

- How is the matching different?
  - 0/1 preferences: either compatible or not
  - Constraints: transplants at the same time, limit cycle size
  - A weighted objective: medical priorities



Matching Representations



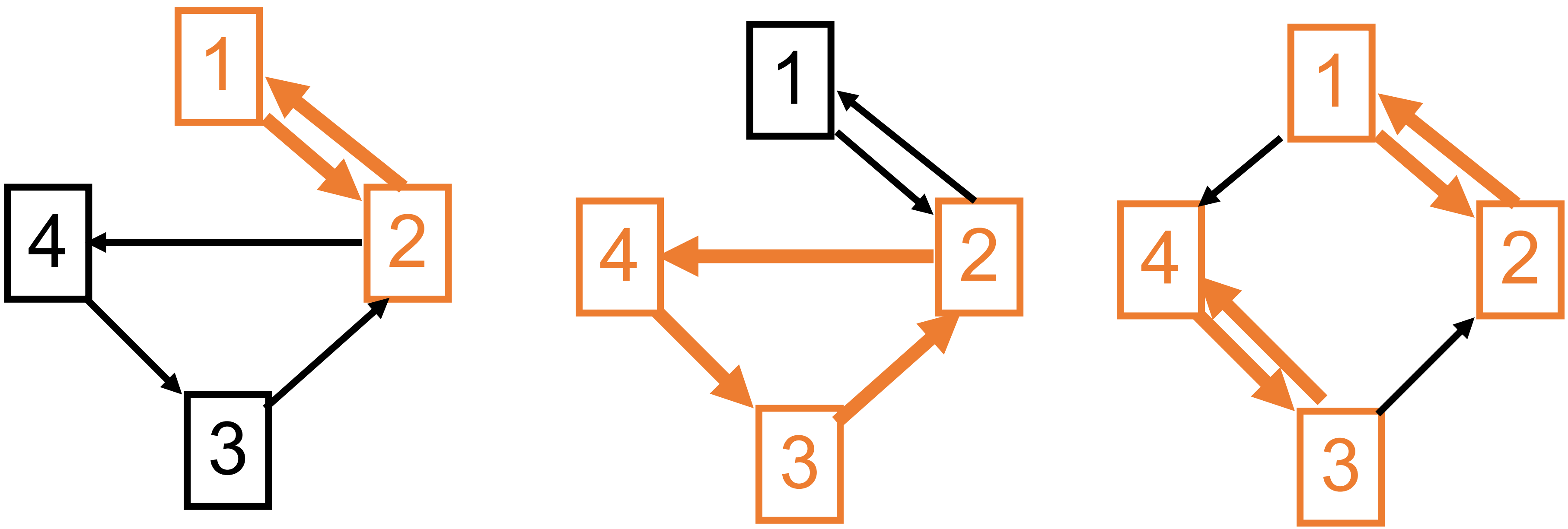


edge from i to j:
patient i wants
donor j's kidney

(Source: Conitzer)

Market Clearing Problem

- Try to cover as many vertices as possible with (vertex-)disjoint cycles of length at most k

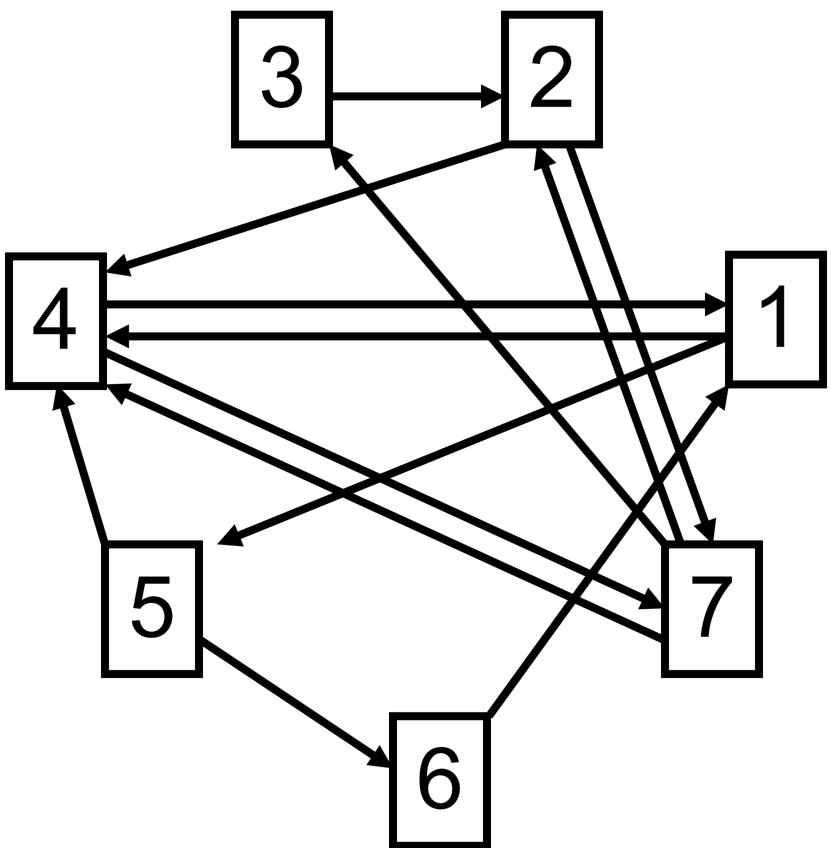


k=2 k=3 k=2,3

(Source: Conitzer)

Market Clearing Problem

- Try to cover as many vertices as possible with (vertex-)disjoint cycles of length at most k

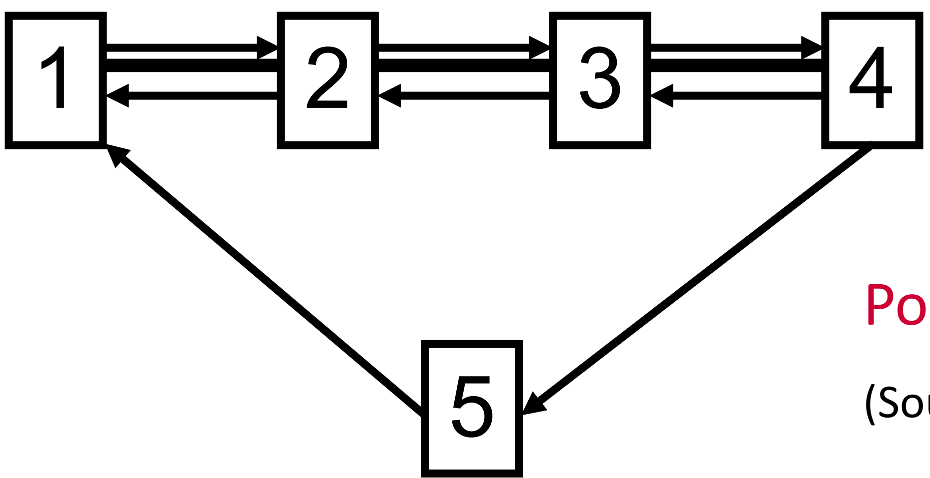


(Source: Conitzer)

Market Clearing Problem (k=2)

- Try to cover as many vertices as possible with (vertex-)disjoint cycles of length at most k

1. If edges go both directions, replace by an undirected edge
2. Remove other edges
3. Maximum matching problem (max #edges with every vertex incident on at most one edge)



Polynomial time

(Source: Conitzer)

Market Clearing Problem (k=∞)

$$\max_z \sum_{(i,j)\in E} z_{ij}$$

1 if i gets j's kidney, 0 otherwise

$$\text{s.t.} \sum_j z_{ij} \le 1, \quad \text{for all } j$$

j gives a most one kidney

$$\sum_j z_{ij} = \sum_j z_{ji}, \quad \text{for all } i$$

received by i = # given by i

$$z_{ij} \ge 0, \quad \text{for all } i,\, j$$

No need to force integer! Polynomial time

Market Clearing Problem (general k)

- For each cycle c of length at most k, make a binary variable x_c
  - 1 if all edges on this cycle are used, 0 otherwise
- Integer programming

$$\max_{X_c} \quad \sum_{c \in C} |c| x_c$$

$$\text{s.t.} \quad \sum_{c \in C, v \in c} x_c \leq 1 \quad \forall v \in V$$

every vertex in at most one used cycle

$$x_c \in \{0, 1\} \quad \forall c \in C$$

NP-hard!

Announcements

- Two paper presentations next week
  - Peer evaluation for paper presentation
- HW1 is due today 11:59pm!
- Discuss final project guidelines. Feel free to discuss your idea with me during office hour
- Class survey instead of pre-class CQ