Game Theory

9. Mechanisms Without Money

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel, Stefan Wölfl, and Robert Mattmüller

June 28th, 2016

Motivation

Mechanisms without Money

Motivation 1:

- According to Gibbard-Satterthwaite:
  In general, nontrivial social choice functions manipulable.
- One way out: Introduction of money
  (cf. VCG mechanisms)
- Other way out: Restriction of preferences
  (cf. single-peaked preferences; this chapter)

Motivation 2:

- Introduction of central concept from cooperative game theory: the core

Examples:

- House allocation problem
- Stable matchings

House Allocation Problem

House Allocation Problem

- Players $N = \{1, \ldots, n\}$.
- Each player i owns house i.
- Each player i has strict linear preference order $\lhd_i$ over the set of houses.
  Example: $j \lhd_i k$ means player i prefers house k to house j.
- Alternatives A: allocations of houses to players (permutations $\pi \in S_n$ of N).
  Example: $\pi(i) = j$ means player i gets house j.
- Objective: reallocate the houses among the agents "appropriately".

House Allocation Problem

- Note on preference relations:
  - Arbitrary (strict linear) preference orders $\lhd_i$ over houses,
  - but no arbitrary preference orders $\preceq_i$ over A.
- Rather: Player i indifferent between different allocations π_1 and π_2 as long as $\pi_1(i) = \pi_2(i)$.
  Indifference denoted as $\pi_1 \approx_i \pi_2$.
- If player i is not indifferent: $\pi_1 \prec_i \pi_2$ iff $\pi_1(i) \lhd_i \pi_2(i)$.
- Notation: $\pi_1 \preceq_i \pi_2$ iff $\pi_1 \prec_i \pi_2$ or $\pi_1 \approx_i \pi_2$.
- This makes Gibbard-Satterthwaite inapplicable.

House Allocation Problem

- Important new aspect of house allocation problem: players control resources to be allocated.
- Allocation can be subverted by subset of agents breaking away and trading among themselves.
- How to avoid such allocations?
- How to make allocation mechanism non-manipulable?

Notation: For $M \subseteq N$, let

$$A(M) = \{\pi \in A \,|\, \forall i \in M : \pi(i) \in M\}$$

be the set of allocations that can be achieved by the agents in M trading among themselves.

Definition (blocking coalition)

Let $\pi \in A$ be an allocation. A set $M \subseteq N$ is called a blocking coalition for π if there exists a $\pi' \in A(M)$ such that

- $\pi \preceq_i \pi'$ for all $i \in M$ and
- $\pi \prec_i \pi'$ for at least one $i \in M$.

House Allocation Problem

Intuition:

A blocking coalition can receive houses everyone from the coalition likes at least as much as under allocation π, with at least one player being strictly better off, by trading among themselves.

Definition (core)

The set of allocations that is not blocked by any subset of agents is called the core.

Question: Is the core nonempty?

Top Trading Cycle Algorithm (TTCA)

- Algorithm to construct allocation
- Let $G = \langle V, A, c \rangle$ be an arc-colored directed graph where:
  - $V = N$ (i.e., one vertex for each player),
  - $A = V \times V$, and
  - $c : A \rightarrow N$ such that $c(i,j) = k$ if house j is player i's kth ranked choice according to $\lhd_i$.
- Note: Loops (i,i) are allowed. We treat them as cycles of length 0.

Top Trading Cycle Algorithm (TTCA)

Pseudocode:

let $\pi(i) = i$ for all $i \in N$.
while players unaccounted for **do**
  consider subgraph G' of G where each vertex has
    only one outgoing arc: the least-colored one from G.
  identify cycles in G'.
  add corresponding cyclic permutations to π.
  delete players accounted for and incident edges from G.
end while
output π.

Notation:

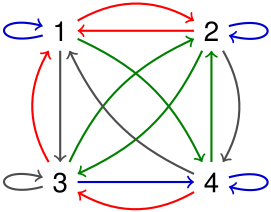
Let N_i be the set of vertices on cycles identified in iteration i.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:



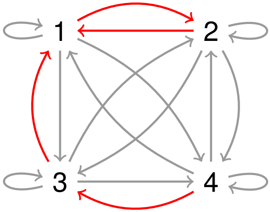
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:



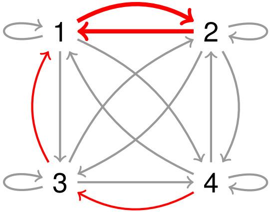
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:



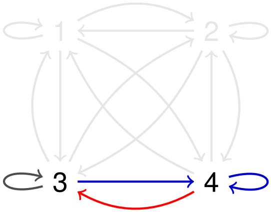
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:



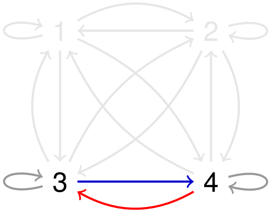
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:



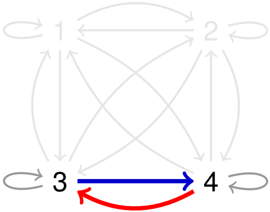
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:



- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Example:

- Player 1: $3 \lhd_1 1 \lhd_1 4 \lhd_1 2$
- Player 2: $4 \lhd_2 2 \lhd_2 3 \lhd_2 1$
- Player 3: $3 \lhd_3 4 \lhd_3 2 \lhd_3 1$
- Player 4: $1 \lhd_4 4 \lhd_4 2 \lhd_4 3$

Corresponding graph:

- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.
- Iteration 2: $\pi(3) = 4$, $\pi(4) = 3$.
- Done: $\pi(1) = 2$, $\pi(2) = 1$, $\pi(3) = 4$, $\pi(4) = 3$.

Top Trading Cycle Algorithm (TTCA)

Theorem

The core of the house allocation problem consists of exactly one matching.

Proof sketch

At most one matching: Show that if a matching is in the core, it must be the one returned by the TTCA.

In TTCA, each player in N_1 receives his favorite house.

Therefore, N_1 would form a blocking coalition to any allocation that does not assign to all of those players the houses they would receive in TTCA.

...

Top Trading Cycle Algorithm (TTCA)

Theorem

The core of the house allocation problem consists of exactly one matching.

Proof sketch

At most one matching: Show that if a matching is in the core, it must be the one returned by the TTCA.

In TTCA, each player in N_1 receives his favorite house.

Therefore, N_1 would form a blocking coalition to any allocation that does not assign to all of those players the houses they would receive in TTCA.

...

Top Trading Cycle Algorithm (TTCA)

Theorem

The core of the house allocation problem consists of exactly one matching.

Proof sketch

At most one matching: Show that if a matching is in the core, it must be the one returned by the TTCA.

In TTCA, each player in N_1 receives his favorite house.

Therefore, N_1 would form a blocking coalition to any allocation that does not assign to all of those players the houses they would receive in TTCA.

...

Proof sketch (ctd.)

That is, any core allocation must assign N_1 to houses as TTCA assigns them.

Argument can be extended inductively to N_k, $2 \leq k \leq n$.

At least one matching: Show that TTCA allocation is in the core, i.e., that there is no other blocking coalition $M \subseteq N$. Homework. □

Top Trading Cycle Algorithm (TTCA)

Proof sketch (ctd.)

That is, any core allocation must assign N_1 to houses as TTCA assigns them.

Argument can be extended inductively to N_k, $2 \leq k \leq n$.

At least one matching: Show that TTCA allocation is in the core, i.e., that there is no other blocking coalition $M \subseteq N$. Homework. □

Top Trading Cycle Algorithm (TTCA)

UNI FREIBURG

Proof sketch (ctd.)

That is, any core allocation must assign N_1 to houses as TTCA assigns them.

Argument can be extended inductively to N_k, $2 \leq k \leq n$.

At least one matching: Show that TTCA allocation is in the core, i.e., that there is no other blocking coalition $M \subseteq N$. Homework. □

Top Trading Cycle Mechanism (TTCM)

Question: What about manipulability?

Definition (top trading cycle mechanism)

The top trading cycle mechanism (TTCM) is the function that, for each profile of preferences, returns the allocation computed by the TTCA.

Theorem

The TTCM cannot be manipulated.

Proof

Homework. □

Stable Matchings

Stable Matchings

Problem statement:

- Given disjoint finite sets M of men and W of women.
- Assume WLOG that $|M| = |W|$ (introduce dummy-men/dummy-women).
- Each $m \in M$ has strict preference ordering $\prec_m$ over W.
- Each $w \in W$ has strict preference ordering $\prec_w$ over M.
- Matching: "appropriate" assignment of men to women such that each man is assigned to at most one woman and vice versa.

Stable Matchings

Note: A group of players can subvert a matching by opting out.

Definition (stability, blocking pair)

A matching is called unstable if there are two men m, m' and two women w, w' such that

- m is matched to w,
- m' is matched to w', and
- $w \prec_m w'$ and $m' \prec_{w'} m$.

The pair $\langle m, w' \rangle$ is called a blocking pair.
A matching that has no blocking pairs is called stable.

Definition (core)

The core of the matching game is the set of all stable matchings.

Stable Matchings

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Two matchings:

- Matching $\{\langle m_1, w_1\rangle, \langle m_2, w_2\rangle, \langle m_3, w_3\rangle\}$
  - unstable ($\langle m_1, w_2\rangle$ is a blocking pair)
- Matching $\{\langle m_1, w_1\rangle, \langle m_3, w_2\rangle, \langle m_2, w_3\rangle\}$
  - stable

Stable Matchings

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Two matchings:

- Matching $\{\langle m_1, w_1\rangle, \langle m_2, w_2\rangle, \langle m_3, w_3\rangle\}$
  - unstable ($\langle m_1, w_2\rangle$ is a blocking pair)
- Matching $\{\langle m_1, w_1\rangle, \langle m_3, w_2\rangle, \langle m_2, w_3\rangle\}$
  - stable

Stable Matchings

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Two matchings:

- Matching $\{\langle m_1, w_1\rangle, \langle m_2, w_2\rangle, \langle m_3, w_3\rangle\}$
  - unstable ($\langle m_1, w_2\rangle$ is a blocking pair)
- Matching $\{\langle m_1, w_1\rangle, \langle m_3, w_2\rangle, \langle m_2, w_3\rangle\}$
  - stable

UNI FREIBURG

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Two matchings:

- Matching $\{\langle m_1, w_1\rangle, \langle m_2, w_2\rangle, \langle m_3, w_3\rangle\}$
  - unstable ($\langle m_1, w_2\rangle$ is a blocking pair)
- Matching $\{\langle m_1, w_1\rangle, \langle m_3, w_2\rangle, \langle m_2, w_3\rangle\}$
  - stable

Stable Matchings

Question: Is there always a stable matching?

Answer: Yes! And it can even be efficiently constructed.

How? Deferred acceptance algorithm!

Definition (deferred acceptance algorithm, male proposals)

1. Each man proposes to his top-ranked choice.
2. Each woman who has received at least one proposal (including tentatively kept one from earlier rounds) tentatively keeps top-ranked proposal and rejects rest.
3. If no man is left rejected, stop.
4. Otherwise, each man who has been rejected proposes to his top-ranked choice among the women who have not rejected him. Then, goto 2.

Deferred Acceptance Algorithm

Note:

- Algorithm has polynomial runtime.
- No man is assigned to more than one woman.
- No woman is assigned to more than one man.
- $\rightsquigarrow$ matching

Deferred Acceptance Algorithm

Motivation

House Allocation Problem

Stable Matchings

Definitions

Deferred Acceptance Algorithm

Properties

Summary

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Deferred acceptance algorithm:

1. m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1.
2. w_1 keeps m_3 and rejects m_2, w_2 keeps m_1.
3. m_2 now proposes to w_3.
4. w_3 keeps m_2.

Deferred Acceptance Algorithm

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Deferred acceptance algorithm:

1. m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1.
2. w_1 keeps m_3 and rejects m_2, w_2 keeps m_1.
3. m_2 now proposes to w_3.
4. w_3 keeps m_2.

Deferred Acceptance Algorithm

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Deferred acceptance algorithm:

1. m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1.
2. w_1 keeps m_3 and rejects m_2, w_2 keeps m_1.
3. m_2 now proposes to w_3.
4. w_3 keeps m_2.

Deferred Acceptance Algorithm

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Deferred acceptance algorithm:

1. m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1.
2. w_1 keeps m_3 and rejects m_2, w_2 keeps m_1.
3. m_2 now proposes to w_3.
4. w_3 keeps m_2.

Deferred Acceptance Algorithm

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Deferred acceptance algorithm:

1. m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1.
2. w_1 keeps m_3 and rejects m_2, w_2 keeps m_1.
3. m_2 now proposes to w_3.
4. w_3 keeps m_2.

Deferred Acceptance Algorithm

Example:

- Man 1: $w_3 \prec_{m_1} w_1 \prec_{m_1} w_2$
- Man 2: $w_2 \prec_{m_2} w_3 \prec_{m_2} w_1$
- Man 3: $w_3 \prec_{m_3} w_2 \prec_{m_3} w_1$
- Woman 1: $m_2 \prec_{w_1} m_3 \prec_{w_1} m_1$
- Woman 2: $m_2 \prec_{w_2} m_1 \prec_{w_2} m_3$
- Woman 3: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

Deferred acceptance algorithm:

1. m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1.
2. w_1 keeps m_3 and rejects m_2, w_2 keeps m_1.
3. m_2 now proposes to w_3.
4. w_3 keeps m_2.

Resulting matching: $\{\langle m_1, w_2\rangle, \langle m_2, w_3\rangle, \langle m_3, w_1\rangle\}$.

Deferred Acceptance Algorithm

Theorem

The deferred acceptance algorithm with male proposals terminates in a stable matching.

Proof

Suppose not.

Then there exists a blocking pair $\langle m_1, w_1 \rangle$ with m_1 matched to some w_2 and w_1 matched to some m_2.

Since $\langle m_1, w_1 \rangle$ is blocking and $w_2 \prec_{m_1} w_1$, in the proposal algorithm, m_1 would have proposed to w_1 before w_2.

Since m_1 was not matched with w_1 by the algorithm, it must be because w_1 received a proposal from a man she ranked higher than m_1. . . .

Deferred Acceptance Algorithm

Theorem

The deferred acceptance algorithm with male proposals terminates in a stable matching.

Proof

Suppose not.

Then there exists a blocking pair $\langle m_1, w_1 \rangle$ with m_1 matched to some w_2 and w_1 matched to some m_2.

Since $\langle m_1, w_1 \rangle$ is blocking and $w_2 \prec_{m_1} w_1$, in the proposal algorithm, m_1 would have proposed to w_1 before w_2.

Since m_1 was not matched with w_1 by the algorithm, it must be because w_1 received a proposal from a man she ranked higher than m_1. ...

Deferred Acceptance Algorithm

Theorem

The deferred acceptance algorithm with male proposals terminates in a stable matching.

Proof

Suppose not.

Then there exists a blocking pair $\langle m_1, w_1 \rangle$ with m_1 matched to some w_2 and w_1 matched to some m_2.

Since $\langle m_1, w_1 \rangle$ is blocking and $w_2 \prec_{m_1} w_1$, in the proposal algorithm, m_1 would have proposed to w_1 before w_2.

Since m_1 was not matched with w_1 by the algorithm, it must be because w_1 received a proposal from a man she ranked higher than m_1. ...

Deferred Acceptance Algorithm

Theorem

The deferred acceptance algorithm with male proposals terminates in a stable matching.

Proof

Suppose not.

Then there exists a blocking pair $\langle m_1, w_1 \rangle$ with m_1 matched to some w_2 and w_1 matched to some m_2.

Since $\langle m_1, w_1 \rangle$ is blocking and $w_2 \prec_{m_1} w_1$, in the proposal algorithm, m_1 would have proposed to w_1 before w_2.

Since m_1 was not matched with w_1 by the algorithm, it must be because w_1 received a proposal from a man she ranked higher than m_1. ...

Proof (ctd.)

Since the algorithm matches her to m_2 it follows that $m_1 \prec_{w_1} m_2$.

This contradicts the fact that $\langle m_1, w_1 \rangle$ is a blocking pair. □

Analogous version where the women propose: outcome would also be a stable matching.

Deferred Acceptance Algorithm

Proof (ctd.)

Since the algorithm matches her to m_2 it follows that $m_1 \prec_{w_1} m_2$.

This contradicts the fact that $\langle m_1, w_1 \rangle$ is a blocking pair. □

Analogous version where the women propose: outcome would also be a stable matching.

Proof (ctd.)

Since the algorithm matches her to m_2 it follows that $m_1 \prec_{w_1} m_2$.

This contradicts the fact that $\langle m_1, w_1 \rangle$ is a blocking pair. □

Analogous version where the women propose: outcome would also be a stable matching.

Deferred Acceptance Algorithm

Denote a matching by μ. The woman assigned to man m in μ is $\mu(m)$, and the man assigned to woman w is $\mu(w)$.

Definition (optimality)

A matching μ is male-optimal if there is no stable matching ν such that $\mu(m) \prec_m \nu(m)$ or $\mu(m) = \nu(m)$ for all $m \in M$ and $\mu(m) \prec_m \nu(m)$ for at least one $m \in M$. Female-optimal: similar.

Theorem

- The stable matching produced by the (fe)male-proposal deferred acceptance algorithm is (fe)male-optimal.
- In general, there is no stable matching that is male-optimal and female-optimal. □

Deferred Acceptance Algorithm

Theorem

The mechanism associated with the (fe)male-proposal algorithm cannot be manipulated by the (fe)males. □

Summary

Summary

- Avoid Gibbard-Satterthwaite by restricting domain of preferences.
- House allocation problem:
  - Solved using top trading cycle algorithm.
  - Algorithm finds unique solution in the core, where no blocking coalition of players has an incentive to break away.
  - The top trading cycle mechanism cannot be manipulated.
- Stable matchings:
  - Solved using deferred acceptance algorithm.
  - Algorithm finds a stable matching in the core, where no blocking pair of players has an incentive to break away.
  - The mechanism associated with the (fe)male-proposal algorithm cannot be manipulated by the (fe)males.