Game Theory

8. Mechanism Design

8.1. Introduction and Example

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

Motivation

Objective: Design the rules of the game such that desirable behavior is dominant behavior.

Here: desirable $\approx$

- truthful about one's own preferences +
- contributing to maximizing social welfare

Motivation

Model:

- Strict linear orders $\prec$ contain no information about "by how much" one alternative is preferred.
- Idea: Instead, use money to measure this.
- Use money also for transfers between players "for compensation".

Setting

Given:

- set of alternatives A
- set of n players N
- valuation functions $v_i : A \to \mathbb{R}$ such that $v_i(a)$ denotes the value player i assigns to alternative a

Find:

- a chosen alternative $a \in A$.
- payments $p_i \in \mathbb{R}$ to be paid by player i

Utility of player i: $u_i(a) = v_i(a) - p_i$.

Example: Vickrey Auctions

Second price auctions (aka Vickrey auctions):

- There are n players bidding for a single item.
- Player i's private valuations of item: w_i.
- Desired outcome: Player with highest private valuation wins bid.
- Players should reveal their valuations truthfully.
- Winner i pays price p^* and has utility $w_i - p^*$.
- Non-winners pay nothing and have utility 0.

Example: Vickrey Auctions

UNI FREIBURG

Formally:

- $A = N$
- $v_i(a) = \begin{cases} w_i & \text{if } a = i \\ 0 & \text{else} \end{cases}$
- What about payments? Say player i wins:
  - $p^* = 0$ (winner pays nothing): bad idea, players would manipulate and publicly declare values $w_i' \gg w_i$.
  - $p^* = w_i$ (winner pays his valuation): bad idea, players would manipulate and publicly declare values $w_i' = w_i - \varepsilon$.
  - better: $p^* = \max_{j \neq i} w_j$ (winner pays second highest bid).

Vickrey Auction

Definition (Vickrey Auction)

The winner of the Vickrey Auction (aka second price auction) is the player i with the highest declared value w_i. He has to pay the second highest declared bid $p^* = \max_{j \neq i} w_j$.

Proposition (Vickrey)

Let i be one of the players and w_i his valuation for the item, u_i his utility if he truthfully declares w_i as his valuation of the item, and u_i' his utility if he falsely declares w_i' as his valuation of the item. Then $u_i \geq u_i'$.

Proof

See
`http://en.wikipedia.org/wiki/Vickrey_auction`. □

Summary

- New preference model: with money.
- To ensure truthful revelation of preferences, we need the right payment functions.
- Example: Vickrey auctions.

Game Theory

8. Mechanism Design

8.2. Incentive Compatible Mechanisms

8.2.1. VCG Mechanisms

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

Incentive Compatible Mechanisms

- Idea: Generalization of Vickrey auctions.
- Preferences modeled as functions $v_i : A \to \mathbb{R}$.
- Let V_i be the space of all such functions for player i.
- Unlike for social choice functions: Not only decide about chosen alternative, but also about payments.

Incentive Compatible Mechanisms

Definition (mechanism)

A mechanism $\langle f, p_1, \dots, p_n \rangle$ consists of

- a social choice function $f : V_1 \times \cdots \times V_n \to A$ and
- for each player i, a payment function $p_i : V_1 \times \cdots \times V_n \to \mathbb{R}$.

Definition (incentive compatibility)

A mechanism $\langle f, p_1, \dots, p_n \rangle$ is called incentive compatible if for each player $i = 1, \dots, n$, for all preferences $v_1 \in V_1, \dots, v_n \in V_n$ and for each preference $v_i' \in V_i$,

$$v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(f(v_i', v_{-i})) - p_i(v_i', v_{-i}).$$

VCG Mechanisms

- If $\langle f, p_1, \ldots, p_n \rangle$ is incentive compatible, truthfully declaring ones preference is dominant strategy.
- The Vickrey-Clarke-Groves mechanism is an incentive compatible mechanism that maximizes "social welfare", i.e., the sum over all individual utilities $\sum_{i=1}^{n} v_i(a)$.
- Idea: Reflect other players' utilities in payment functions, align all players' incentives with goal of maximizing social welfare.

VCG Mechanisms

UNI FREIBURG

Definition (Vickrey-Clarke-Groves mechanism)

A mechanism $\langle f, p_1, \dots, p_n \rangle$ is called a Vickrey-Clarke-Groves mechanism (VCG mechanism) if

1. $f(v_1, \dots, v_n) \in \text{argmax}_{a \in A} \sum_{i=1}^{n} v_i(a)$ for all $v_1, \dots, v_n$ and
2. there are functions $h_1, \dots, h_n$ with $h_i : V_{-i} \to \mathbb{R}$ such that $p_i(v_1, \dots, v_n) = h_i(v_{-i}) - \sum_{j \neq i} v_j(f(v_1, \dots, v_n))$ for all $i = 1, \dots, n$ and $v_1, \dots, v_n$.

Note: $h_i(v_{-i})$ independent of player i's declared preference $\Rightarrow$ $h_i(v_{-i}) = c$ constant from player i's perspective.

Utility of player i $= v_i(f(v_1, \dots, v_n)) + \sum_{j \neq i} v_j(f(v_1, \dots, v_n)) - c = \sum_{j=1}^{n} v_j(f(v_1, \dots, v_n)) - c =$ social welfare $- c$.

VCG Mechanisms

Theorem (Vickrey-Clarke-Groves)

Every VCG mechanism is incentive compatible.

Proof.

Let i, v_{-i}, v_i and v_i' be given. Show: Declaring true preference v_i dominates declaring false preference v_i'.

Let $a = f(v_i, v_{-i})$ and $a' = f(v_i', v_{-i})$.

$$\text{Utility player } i = \begin{cases} v_i(a) + \sum_{j \neq i} v_j(a) - h_i(v_{-i}) & \text{if declaring } v_i \\ v_i(a') + \sum_{j \neq i} v_j(a') - h_i(v_{-i}) & \text{if declaring } v_i' \end{cases}$$

Alternative $a = f(v_i, v_{-i})$ maximizes social welfare

$\Rightarrow v_i(a) + \sum_{j \neq i} v_j(a) \geq v_i(a') + \sum_{j \neq i} v_j(a')$.

$\Rightarrow v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(f(v_i', v_{-i})) - p_i(v_i', v_{-i})$. □

Summary

- New preference model: with money.
- VCG mechanisms generalize Vickrey auctions.
- VCG mechanisms are incentive compatible mechanisms maximizing social welfare.

Game Theory

8. Mechanism Design

8.2. Incentive Compatible Mechanisms

8.2.2. Clarke Pivot Functions

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

Clarke Pivot Function

- So far: functions h_i unspecified
  $\rightsquigarrow$ payment functions p_i only partially specified
- One possibility: $h_i(v_{-i}) = 0$ for all h_i and v_{-i}
  Drawback: too much money distributed among players (more that necessary)
- Further requirements:
  - Players should pay at most as much as they value the outcome.
  - Players should only pay, never receive money.

Individual Rationality, Positive Transfers

Definition (individual rationality)

A mechanism is individually rational if all players always get a nonnegative utility, i.e., if for all $i = 1, \dots, n$ and all $v_1, \dots, v_n$,

$$v_i(f(v_1, \dots, v_n)) - p_i(v_1, \dots, v_n) \geq 0.$$

Definition (positive transfers)

A mechanism has no positive transfers if no player is ever paid money, i.e., if for all $i = 1, \dots, n$ and all $v_1, \dots, v_n$,

$$p_i(v_1, \dots, v_n) \geq 0.$$

Clarke Pivot Function

Definition (Clarke pivot function)

The Clarke pivot function is the function

$$h_i(v_{-i}) = \max_{b \in A} \sum_{j \neq i} v_j(b).$$

- This leads to payment functions

$$p_i(v_1, \dots, v_n) = \max_{b \in A} \sum_{j \neq i} v_j(b) - \sum_{j \neq i} v_j(a)$$

  for $a = f(v_1, \dots, v_n)$.
- Player i pays the difference between what the other players could achieve without him and what they achieve with him.
- Each player internalizes the externalities he causes.

Clarke Pivot Function

Example

- Players $N = \{1, 2\}$, alternatives $A = \{a, b\}$.
- Values: $v_1(a) = 10$, $v_1(b) = 2$, $v_2(a) = 9$ and $v_2(b) = 15$.
- Without player 1: b best, since $v_2(b) = 15 > 9 = v_2(a)$.
- With player 1: a best, since
  $v_1(a) + v_2(a) = 10 + 9 = 19 > 17 = 2 + 15 = v_1(b) + v_2(b)$.
- With player 1, other players (i.e., player 2) lose
  $v_2(b) - v_2(a) = 6$ units of utility.

$\Rightarrow$ Clarke pivot function $h_1(v_2) = 15$

$\Rightarrow$ payment function

$$p_1(v_1, \ldots, v_n) = \max_{b \in A} \sum_{j \neq 1} v_j(b) - \sum_{j \neq 1} v_j(a) = 15 - 9 = 6.$$

Clarke Pivot Rule

Lemma (Clarke pivot rule)

A VCG mechanism with Clarke pivot functions has no positive transfers. If $v_i(a) \geq 0$ for all $i = 1, \dots, n$, $v_i \in V_i$ and $a \in A$, then the mechanism is also individually rational.

Proof.

Let $a = f(v_1, \dots, v_n)$ be the alternative maximizing $\sum_{j=1}^{n} v_j(a)$, and b the alternative maximizing $\sum_{j \neq i} v_j(b)$.

Payment function for i: $p_i(v_1, \dots, v_n) = \sum_{j \neq i} v_j(b) - \sum_{j \neq i} v_j(a)$.

Since b maximizes $\sum_{j \neq i} v_j(b)$: $p_i(v_1, \dots, v_n) \geq 0$
($\rightsquigarrow$ no positive transfers).

Utility of player i: $u_i = v_i(a) + \sum_{j \neq i} v_j(a) - \sum_{j \neq i} v_j(b)$.

...

Clarke Pivot Rule

Proof (ctd.)

Individual rationality: Since $v_i(b) \geq 0$,

$$u_i = v_i(a) + \sum_{j \neq i} v_j(a) - \sum_{j \neq i} v_j(b) \geq \sum_{j=1}^{n} v_j(a) - \sum_{j=1}^{n} v_j(b).$$

Since a maximizes $\sum_{j=1}^{n} v_j(a)$,

$$\sum_{j=1}^{n} v_j(a) \geq \sum_{j=1}^{n} v_j(b)$$

and hence $u_i \geq 0$.

Therefore, the mechanism is also individually rational. $\square$

Summary

- Recall: VCG mechanisms are incentive compatible mechanisms maximizing social welfare.
- With Clarke pivot functions:
  - no positive transfers and
  - individual rationality (if nonnegative valuations).

Game Theory

8. Mechanism Design

8.2. Incentive Compatible Mechanisms

8.2.3. Examples

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

Vickrey Auction as a VCG Mechanism

- $A = N$. Valuations: w_i. $v_a(a) = w_a$, $v_i(a) = 0$ ($i \neq a$).
- a maximizes social welfare $\sum_{i=1}^{n} v_i(a)$ iff a maximizes w_a.
- Let $a = f(v_1, \dots, v_n) = \mathrm{argmax}_{j \in A} w_j$ be the highest bidder.
- Payments: $p_i(v_1, \dots, v_n) = \max_{b \in A} \sum_{j \neq i} v_j(b) - \sum_{j \neq i} v_j(a)$.
- But $\max_{b \in A} \sum_{j \neq i} v_j(b) = \max_{b \in A \setminus \{i\}} w_b$.
- Winner pays value of second highest bid:

$$p_a(v_1, \dots, v_n) = \max_{b \in A} \sum_{j \neq a} v_j(b) - \sum_{j \neq a} v_j(a)$$

$$= \max_{b \in A \setminus \{a\}} w_b - 0 = \max_{b \in A \setminus \{a\}} w_b.$$

- Non-winners pay nothing: For $i \neq a$,

$$p_i(v_1, \dots, v_n) = \max_{b \in A} \sum_{j \neq i} v_j(b) - \sum_{j \neq i} v_j(a)$$

$$= \max_{b \in A \setminus \{i\}} w_b - w_a = w_a - w_a = 0.$$

Example: Bilateral Trade

- Seller s offers item he values with $0 \leq w_s \leq 1$.
- Potential buyer b values item with $0 \leq w_b \leq 1$.
- Alternatives $A = \{\textit{no-trade}, \textit{trade}\}$.
- Valuations:

$$v_s(\textit{no-trade}) = 0, \qquad v_s(\textit{trade}) = -w_s,$$
$$v_b(\textit{no-trade}) = 0, \qquad v_b(\textit{trade}) = w_b.$$

- VCG mechanism maximizes $v_s(a) + v_b(a)$.
- We have

$$v_s(\textit{no-trade}) + v_b(\textit{no-trade}) = 0,$$
$$v_s(\textit{trade}) + v_b(\textit{trade}) = w_b - w_s$$

i.e., *trade* maximizes social welfare iff $w_b \geq w_s$.

Example: Bilateral Trade (ctd.)

- Requirement: if *no-trade* is chosen, neither player pays anything:

$$p_s(v_s, v_b) = p_b(v_s, v_b) = 0.$$

- To that end, choose Clarke pivot function for buyer:

$$h_b(v_s) = \max_{a \in A} v_s(a).$$

- For seller: Modify Clarke pivot function by an additive constant and set

$$h_s(v_b) = \max_{a \in A} v_b(a) - w_b.$$

Example: Bilateral Trade (ctd.)

- For alternative *no-trade*,

$$\begin{aligned} p_s(v_s, v_b) &= \max_{a \in A} v_b(a) - w_b - v_b(\textit{no-trade}) \\ &= w_b - w_b - 0 = 0 \quad \text{and} \\ p_b(v_s, v_b) &= \max_{a \in A} v_s(a) - v_s(\textit{no-trade}) \\ &= 0 - 0 = 0. \end{aligned}$$

- For alternative *trade*,

$$\begin{aligned} p_s(v_s, v_b) &= \max_{a \in A} v_b(a) - w_b - v_b(\textit{trade}) \\ &= w_b - w_b - w_b = -w_b \quad \text{and} \\ p_b(v_s, v_b) &= \max_{a \in A} v_s(a) - v_s(\textit{trade}) \\ &= 0 + w_s = w_s. \end{aligned}$$

Example: Bilateral Trade (ctd.)

- Because $w_b \geq w_s$, the seller gets at least as much as the buyer pays, i.e., the mechanism subsidizes the trade.
- Without subsidies, no incentive compatible bilateral trade possible.
- Note: Buyer and seller can exploit the system by colluding.

Example: Public Project

- Project costs C units.
- Each citizen i privately values the project at w_i units.
- Government will undertake project if $\sum_i w_i > C$.
- Alternatives: $A = \{\textit{no-project}, \textit{project}\}$.
- Valuations:

$$v_G(\textit{no-project}) = 0, \qquad v_G(\textit{project}) = -C,$$
$$v_i(\textit{no-project}) = 0, \qquad v_i(\textit{project}) = w_i.$$

- VCG mechanism with Clarke pivot rule: for each citizen i,

$$h_i(v_{-i}) = \max_{a \in A}\left(\sum_{j \neq i} v_j(a) + v_G(a)\right)$$
$$= \begin{cases} \sum_{j \neq i} w_j - C, & \text{if } \sum_{j \neq i} w_j > C \\ 0, & \text{otherwise.} \end{cases}$$

Example: Public Project (ctd.)

- Citizen i pivotal if $\sum_j w_j > C$ and $\sum_{j \neq i} w_j \leq C$.
- Payment function for citizen i:

$$p_i(v_{1..n}, v_G) = h_i(v_{-i}) - \left(\sum_{j \neq i} v_j(f(v_{1..n}, v_G)) + v_G(f(v_{1..n}, v_G)) \right)$$

  - Case 1: Project undertaken, i pivotal:

$$p_i(v_{1..n}, v_G) = 0 - \left(\sum_{j \neq i} w_j - C \right) = C - \sum_{j \neq i} w_j$$

  - Case 2: Project undertaken, i not pivotal:

$$p_i(v_{1..n}, v_G) = \left(\sum_{j \neq i} w_j - C \right) - \left(\sum_{j \neq i} w_j - C \right) = 0$$

  - Case 3: Project not undertaken:

$$p_i(v_{1..n}, v_G) = 0$$

Example: Public Project (ctd.)

- I.e., citizen i pays nonzero amount

$$C - \sum_{j \neq i} w_j$$

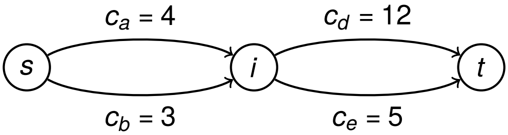
  only if he is pivotal.
- He pays difference between value of project to fellow citizens and cost C, in general less than w_i.
- Generally,

$$\sum_i p_i(\mathit{project}) \leq C$$

  i.e., project has to be subsidized.

Example: Buying a Path in a Network

- Communication network modeled as $G = (V, E)$.
- Each link $e \in E$ owned by different player e.
- Each link $e \in E$ has cost c_e if used.
- Objective: procure communication path from s to t.
- Alternatives: $A = \{\pi \mid \pi \text{ path from } s \text{ to } t\}$.
- Valuations: $v_e(\pi) = -c_e$, if $e \in \pi$, and $v_e(\pi) = 0$, if $e \notin \pi$.
- Maximizing social welfare:
  minimize $\sum_{e \in \pi} c_e$ over all paths π from s to t.
- Example:



Example: Buying a Path in a Network (ctd.)

- For $G = (V, E)$ and $e \in E$ let $G \setminus e = (V, E \setminus \{e\})$.
- VCG mechanism:

$$h_e(v_{-e}) = \max_{\pi' \in G \setminus e} \sum_{e' \in \pi'} -c_{e'}$$

i.e., the cost of the cheapest path from s to t in $G \setminus e$.
(Assume that G is 2-connected, s.t. such π' exists.)

- Payment functions: for chosen path $\pi = f(v_1, \ldots, v_n)$,

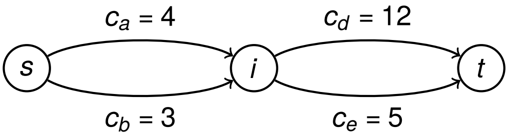
$$p_e(v_1, \ldots, v_n) = h_e(v_{-e}) - \sum_{e \neq e' \in \pi} -c_{e'}.$$

  - Case 1: $e \notin \pi$. Then $p_e(v_1, \ldots, v_n) = 0$.
  - Case 2: $e \in \pi$. Then

$$p_e(v_1, \ldots, v_n) = \max_{\pi' \in G \setminus e} \sum_{e' \in \pi'} -c_{e'} - \sum_{e \neq e' \in \pi} -c_{e'}.$$

Example: Buying a Path in a Network (ctd.)

- Example:



  - Cost along b and e: 8
  - Cost without e: 3
  - Cost of cheapest path without e: 15 (along b and d)
  - Difference is payment: $-15-(-3) = -12$
    I.e., owner of arc e gets payed 12 for using his arc.

- Note: Alternative path after deletion of e does not necessarily differ from original path at only one position. Could be totally different.

Summary

We saw some examples of applications of VCG mechanisms:

- Vickrey Auctions
- Bilateral Trade
- Public Projects
- Buying a Path in a Network

Game Theory

8. Mechanism Design

8.3. Mechanisms without Money

8.3.1. House Allocation

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

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 (Sec. 8.3.1)
- Stable matchings (Sec. 8.3.2)

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?

House Allocation Problem

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
  - $c : A \to N$ 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:

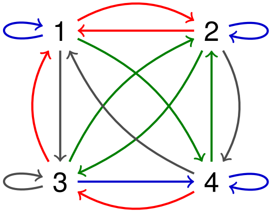
Let N_i be the set of vertices on cycles identified in iteration i.

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:

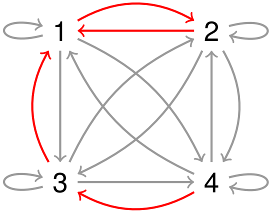


Top Trading Cycle Algorithm (TTCA)

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:

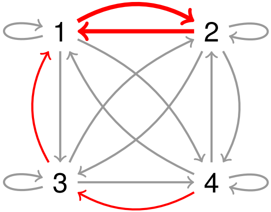


Top Trading Cycle Algorithm (TTCA)

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:



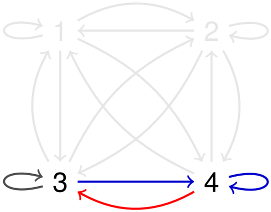
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.

UNI FREIBURG

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:



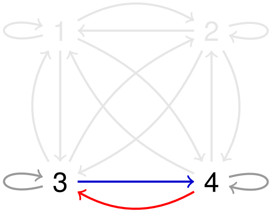
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.

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:



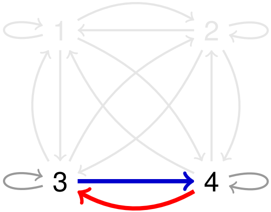
- Iteration 1: $\pi(1) = 2$, $\pi(2) = 1$.

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: $\pi(3) = 4$, $\pi(4) = 3$.

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)

UNI FREIBURG

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)

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. □

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.

Game Theory

8. Mechanism Design

8.3. Mechanisms without Money

8.3.2. Stable Matchings

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

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 (Sec. 8.3.1)
- Stable matchings (Sec. 8.3.2)

Stable Matchings

UNI FREIBURG

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:

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 $\{\langle m_1, w_1\rangle, \langle m_2, w_2\rangle, \langle m_3, w_3\rangle\}$

Stable Matchings

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)

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

Question: Is there always a stable matching?

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

How? Deferred acceptance algorithm!

Deferred Acceptance Algorithm

UNI FREIBURG

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

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:

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.

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.

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.

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

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: $m_2 \prec_{w_3} m_3 \prec_{w_3} m_1$

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

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. $\square$

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.* $\square$

Deferred Acceptance Algorithm

Theorem

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

Note: The mechanism associated with the male-proposal algorithm can be manipulated by the females and vice versa.

(Idea: strategically reject a proposal who then binds your main competitor for your favorite partner in the next round, freeing up that partner for you $\rightsquigarrow$ try this out with our running example!)

Summary

- Avoid Gibbard-Satterthwaite by restricting domain of preferences.
- 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.

Game Theory

8. Mechanism Design

8.4. Combinatorial Auctions

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel and Robert Mattmüller

Summer semester 2020

Combinatorial Auctions

Motivation:

- Multiple items are auctioned concurrently.
- Bidders have preferences for combinations (bundles) of items.
- Items can complement or substitute one another.
  - complement: left and right shoe together.
  - substitute: two right shoes.
- Aim: socially optimal allocation of items to bidders.

Combinatorial Auctions

Applications:

- Spectrum auctions (with combinations of spectrum bands and geographical areas)
- Procurement of transportation services for multiple routes
- …

Notation:

- Items: $G = \{1, \ldots, m\}$
- Bidders: $N = \{1, \ldots, n\}$

Combinatorial Auctions

Definition (valuation)

A valuation is a function $v : 2^G \to \mathbb{R}^+$ with $v(\emptyset) = 0$ and $v(S) \leq v(T)$ for $S \subseteq T \subseteq G$.

- Requirement $v(\emptyset) = 0$ to "normalize" valuations.
- Requirement $v(S) \leq v(T)$ for $S \subseteq T \subseteq G$: monotonicity (or "free disposal").

Let $S, T \subseteq G$ be disjoint.

- S and T are complements to each other if $v(S \cup T) > v(S) + v(T)$.
- S and T are substitutes if $v(S \cup T) < v(S) + v(T)$.

Allocations

Definition (allocation)

An allocation of the items to the bidders is a tuple $\langle S_1, \ldots, S_n \rangle$ with $S_i \subseteq G$ for $i = 1, \ldots, n$ and $S_i \cap S_j = \emptyset$ for $i \neq j$.

The social welfare obtained by an allocation is $\sum_{i=1}^{n} v_i(S_i)$ if $v_1, \ldots, v_n$ are the valuations of the bidders.

An allocation is called socially efficient if it maximizes social welfare among all allocations.

Let A be the set of all allocations.

Winner Determination Problem

Definition (winner determination problem)

Let $v_i : 2^G \to \mathbb{R}^+$, $i = 1, \ldots, n$, be the declared valuations of the bidders. The winner determination problem (WDP) is the problem of finding a socially efficient allocation $a \in A$ for these valuations.

Aim: Develop mechanism for WDP.

Challenges:

- Incentive compatibility
- Complexity of representation and communication of preferences (exponentially many subsets of items!)
- Computational complexity

Special Case: Single-Minded Bidders

Motivation:

- Focus on single-minded bidders: cuts complexity of representation down to polynomial space.
- Idea: single-minded bidder focuses on one bundle, has fixed valuation v^* for that bundle (and its supersets), valuation 0 for all other bundles.

Single-Minded Bidders

Definition (single-minded bidder)

A valuation v is called single-minded if there is a bundle $S^* \subseteq G$ and a value $v^* \in \mathbb{R}^+$ such that

$$v(S) = \begin{cases} v^* & \text{if } S^* \subseteq S \\ 0 & \text{otherwise} \end{cases}$$

A single-minded bid is a pair $\langle S^*, v^* \rangle$.

- Representational complexity: solved.
- Computational complexity: not solved.

Allocation Problem for Single-Minded Bidders

Definition (allocation problem for single-minded bidders)

The allocation problem for single-minded bidders (APSMB) is defined by the following input and output.

- INPUT: Bids $\langle S_i^*, v_i^* \rangle$ for $i = 1, \ldots, n$
- OUTPUT: $W \subseteq \{1, \ldots, n\}$ with $S_i^* \cap S_j^* = \emptyset$ for $i, j \in W$, $i \neq j$ such that $\sum_{i \in W} v_i^*$ is maximized.

Claim: APSMB is NP-complete.

Allocation Problem for Single-Minded Bidders

Since APSMB is an optimization problem, consider the corresponding decision problem:

Definition (allocation problem for single-minded bidders, decision problem)

The decision problem version of APSMB (APSMB-D) is defined by the following input and output.

- INPUT: Bids $\langle S_i^*, v_i^* \rangle$ for $i = 1, \ldots, n$ and $k \in \mathbb{N}$
- OUTPUT: Is there a $W \subseteq \{1, \ldots, n\}$ with $S_i^* \cap S_j^* = \emptyset$ for $i, j \in W$, $i \neq j$ such that $\sum_{i \in W} v_i^* \geq k$?

Theorem

APSMB-D is NP-complete.

APSMB-D is NP-complete

Proof

NP-hardness: reduction from INDEPENDENT-SET.

INDEPENDENT-SET instance:

- undirected graph $\langle V, E\rangle$ and $k_{IS} \in \mathbb{N}$.
- Question: Is there an independent set of size k_{IS} in $\langle V, E\rangle$?

Corresponding APSMB-D instance:

- $k = k_{IS}$, items $G = E$, bidders $N = V$, and
- for each bidder $i \in V$ the bid $\langle S_i^*, v_i^*\rangle$ with $S_i^* = \{e \in E \mid i \in e\}$ and $v_i^* = 1$.
- Question: Is there an allocation with social welfare $\geq k$?
- (Intuitively: Vertices bid for their incident edges.)

APSMB-D is NP-complete

Proof (ctd.)

Since $S_i^* \cap S_j^* = \emptyset$ for $i, j \in W$, $i \neq j$, the set of winners W represents an independent set of cardinality

$$|W| = \sum_{i \in W} v_i^*.$$

Therefore, there is an independent set of cardinality at least k_{IS} iff there is a set of winners W with $\sum_{i \in W} v_i^* \geq k$.
This proves NP-hardness.

APSMB-D $\in$ NP: obvious (guess and verify set of winners).

APSMB-D is NP-complete

Consequences:

- Solving APSMB optimally: too costly.
- Alternatives:
  - approximation algorithm
  - heuristic approach
  - special cases
- Here: approximation algorithm

Definition (approximation factor)

Let $c \geq 1$. An allocation $\langle S_1, \ldots, S_n \rangle$ is a *c-approximation* of an optimal allocation if

$$\sum_{i=1}^{n} v_i(T_i) \leq c \cdot \sum_{i=1}^{n} v_i(S_i)$$

for an optimal allocation $\langle T_1, \ldots, T_n \rangle$.

Proposition

Approximating APSMB within a factor of $c \leq m^{1/2-\varepsilon}$ for any $\varepsilon > 0$ is NP-hard. $\square$

Approximation Algorithms

Best we can still hope for in case of single-minded bidders:

- incentive compatible
- $m^{1/2}$-approximation algorithm
- with polynomial runtime.

Good news:

- Such an algorithm exists!

Mechanism for Single-Minded Bidders

Definition (mechanism for single-minded bidders)

Let V_{sm} be the set of all single-minded bids and A the set of all allocations.

A mechanism for single-minded bidders is a tuple $\langle f, p_1, \dots, p_n \rangle$ consisting of

- a social choice function $f : V_{sm}^n \to A$ and
- payment functions $p_i : V_{sm}^n \to \mathbb{R}$ for all $i = 1, \dots, n$.

Definition (efficient computability)

A mechanism for single-minded bidders is efficiently computable if f and all p_i can be computed in polynomial time.

Definition (incentive compatibility)

A mechanism for single-minded bidders is incentive compatible if

$$v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(f(v_i', v_{-i})) - p_i(v_i', v_{-i})$$

for all $i = 1, \ldots, n$ and all $v_1, \ldots, v_n, v_i' \in V_{sm}$, where $v_i(a) = v_i^*$ if i wins in a (gets the desired bundle), and $v_i(a) = 0$, otherwise.

Mechanism for Single-Minded Bidders

How to build such a mechanism?

- In principle: could use a VCG mechanism.
- Problem with VCG: incentive compatible, but not efficiently computable
  (need to compute social welfare, which is NP-hard)
- Alternative idea: VCG-like mechanism that approximates social welfare
- Problem with alternative: efficiently computable, but not incentive compatible
- Solution: forget VCG, use specific mechanism for single-minded bidders.

Greedy Mechanism for Single-Minded Bidders

Definition (greedy mechanism for single-minded bidders)

The greedy mechanism for single-minded bidders (GMSMB) is defined as follows.

Let the bidders $1, \dots, n$ be ordered such that

$$\frac{v_1^*}{\sqrt{|S_1^*|}} \geq \frac{v_2^*}{\sqrt{|S_2^*|}} \geq \cdots \geq \frac{v_n^*}{\sqrt{|S_n^*|}}.$$

...

Definition (greedy mechanism for single-minded bidders, ctd.)

Let the set $W \subseteq \{1, \dots, n\}$ be procedurally defined by the following pseudocode:

$W \leftarrow \emptyset$
for $i = 1, \dots, n$ **do**
  if $S_i^* \cap \left(\bigcup_{j \in W} S_j^* \right) = \emptyset$ **then**
    $W \leftarrow W \cup \{i\}$
  end if
end for

...

Greedy Mechanism for Single-Minded Bidders

Definition (greedy mechanism for single-minded bidders, ctd.)

Result: allocation a where exactly the bidders in W win.

Payments:

- Case 1: If $i \in W$ and there is a smallest index j such that $S_i^* \cap S_j^* \neq \emptyset$ and for all $k < j$, $k \neq i$, $S_k^* \cap S_j^* = \emptyset$, then

$$p_i(v_1, \ldots, v_n) = \frac{v_j^*}{\sqrt{|S_j^*|/|S_i^*|}},$$

- Case 2: Otherwise,

$$p_i(v_1, \ldots, v_n) = 0.$$

Greedy Mechanism for Single-Minded Bidders

Example

Let $N = \{1,2,3,4\}$ and $G = \{1,\ldots,13\}$.

i	**Package** S_i^*	**Val.** v_i^*	$v_i^* / \sqrt{\lvert S_i^* \rvert}$	**Assignm. order**
1	$\{1,2,3,4,5,6,7,8,9\}$	15		
2	$\{3,4,5,6,7,8,9,12,13\}$	3		
3	$\{1,2,10,11\}$	12		
4	$\{10,11,12,13\}$	8		

Positions in assignment order? Winner set? Assignment? Social welfare of winner set?

Example (ctd.)

Assignments:

1. Bidder 3 gets $\{1, 2, 10, 11\}$.
2. Bidder 1 gets nothing (obj. 1 and 2 already assigned).
3. Bidder 4 gets nothing (obj. 10 and 11 already assigned).
4. Bidder 2 gets the remainder, i.e., $\{3, 4, 5, 6, 7, 8, 9, 12, 13\}$.

Payments:

1. Bidder 3 pays

$$\frac{v_1^*}{\sqrt{|S_1^*|/|S_3^*|}} = \frac{15}{\sqrt{9/4}} = \frac{15}{3/2} = 10.$$

2. Bidders 1, 4 and 2 pay 0.

Greedy Mechanism for Single-Minded Bidders

Example (ctd.)

Therefore:

- Winner set: $W = \{2,3\}$.
  Social welfare: $U = 12 + 3 = 15$.
- Optimal winner set: $W^* = \{1,4\}$.
  Optimal social welfare: $U^* = 15 + 8 = 23$.
- Approximation ratio: $23/15 < 2 < 3 < \sqrt{13} = \sqrt{m}$

Greedy Mechanism for Single-Minded Bidders: Efficient Computability

Theorem

GMSMB is efficiently computable. □

Open questions:

- What about incentive compatibility?
- What about approximation factor of $\sqrt{m}$?

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

To prove incentive compatibility:

- Step 1: Show that GMSMB is monotone.
- Step 2: Show that GMSMB uses critical payments.
- Step 3: Show that in GMSMB losers pay nothing.
- Step 4: Show that every mechanism for single-minded bidders that is monotone, that uses critical payments, and where losers pay nothing is incentive compatible.

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

Definition (monotonicity)

A mechanism for single-minded bidders is monotone if a bidder who wins with bid $\langle S^*, v^* \rangle$ would also win with any bid $\langle S', v' \rangle$ where $S' \subseteq S^*$ and $v' \geq v^*$ (for fixed bids of the other bidders).

Definition (critical payments)

A mechanism for single-minded bidders uses critical payments if a bidder who wins pays the minimal amount necessary for winning, i.e., the infimum of all v' such that $\langle S^*, v' \rangle$ still wins.

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

Lemma

GMSBM is monotone, uses critical payments, and losers pay nothing.

Proof

Monotonicity: Increasing v_i^* or decreasing S_i^* can only move bidder i up in the greedy order, making it easier to win.

Critical payments: Bidder i wins as long as he is before bidder j in the greedy order (if such a j exists). This holds iff

$$\frac{v_i^*}{\sqrt{|S_i^*|}} \geq \frac{v_j^*}{\sqrt{|S_j^*|}} \quad \text{iff} \quad v_i^* \geq \frac{v_j^* \sqrt{|S_i^*|}}{\sqrt{|S_j^*|}} = \frac{v_j^*}{\sqrt{|S_j^*|/|S_i^*|}} = p_i.$$

Losers pay nothing: Obvious. □

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

Lemma

A mechanism for single-minded bidders that is monotone, that uses critical payments, and where losers pay nothing is incentive compatible.

Proof

(A) Truthful bids never lead to negative utility.

- If the declared bid loses, bidder has utility 0.
- If the declared bid wins, he has utility $v^* - p^* \geq 0$, since $v^* \geq p^*$, because p^* is the critical payment, and if the bid wins, the bidder must have (truthfully) bid a value v^* of at least p^*.

...

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

Proof (ctd.)

(B) Truthful bids never lead to lower utility than untruthful bids.
Suppose declaration of untruthful bid $\langle S', v' \rangle$ deviating from truthful bid $\langle S^*, v^* \rangle$.

(B.1) Case 1: untruthful bid is losing or not useful for bidder.
Suppose $\langle S', v' \rangle$ is losing or $S^* \not\subseteq S'$ (bidder does not get the bundle he wants). Then utility ≤ 0 in $\langle S', v' \rangle$, i.e., no improvement over utility when declaring $\langle S^*, v^* \rangle$ (cf. (A)).

(B.2) Case 2: untruthful bid is winning and useful for bidder.
Assume $\langle S', v' \rangle$ is winning and $S^* \subseteq S'$. To show that $\langle S^*, v^* \rangle$ is at least as good a bid as $\langle S', v' \rangle$, show that $\langle S^*, v' \rangle$ is at least as good as $\langle S', v' \rangle$ and that $\langle S^*, v^* \rangle$ is at least as good as $\langle S^*, v' \rangle$.

Proof (ctd.)

- (B.2.a) Lying about desired bundle does not help.

  Show that $\langle S^*, v' \rangle$ is at least as good as $\langle S', v' \rangle$.

  Let p' be the payment for bid $\langle S', v' \rangle$ and p the payment for bid $\langle S^*, v' \rangle$.

  For all $x < p$, $\langle S^*, x \rangle$ is losing, since p is the critical payment for S^*.

  Due to monotonicity, also $\langle S', x \rangle$ is losing for all $x < p$. Hence, the critical payment p' for S' is at least p.

  Thus, $\langle S^*, v' \rangle$ is still winning, if $\langle S', v' \rangle$ was, and leads to the same or even lower payment.

...

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

Proof (ctd.)

- (B.2.b) Lying about valuation does not help.
  Show that $\langle S^*, v^* \rangle$ is at least as good as $\langle S^*, v' \rangle$.
  - (B.2.b.i) Case 1: $\langle S^*, v^* \rangle$ is winning with payment p^*.
    If $v' > p^*$, then $\langle S^*, v' \rangle$ is still winning with the same payment, so there is no incentive to deviate to $\langle S^*, v' \rangle$.

    If $v' \leq p^*$, then $\langle S^*, v' \rangle$ is losing, so there is also no incentive to deviate to $\langle S^*, v' \rangle$.
  - (B.2.b.ii) Case 2: $\langle S^*, v^* \rangle$ is losing.
    Then v^* is less than the critical payment, i.e., the payment p' for a winning bid $\langle S^*, v' \rangle$ would be greater than v^*, making a deviation to $\langle S^*, v' \rangle$ unprofitable. $\square$

Greedy Mechanism for Single-Minded Bidders: Incentive Compatibility

Corollary

The greedy mechanism for single-minded bidders is incentive compatible. □

Open question:

- What about approximation factor of $\sqrt{m}$?

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

In the next proof, we will need the Cauchy-Schwarz inequality:

Theorem (Cauchy-Schwarz inequality)

Let $x_j, y_j \in \mathbb{R}$. Then

$$\sum_j x_j y_j \leq \sqrt{\sum_j x_j^2} \cdot \sqrt{\sum_j y_j^2}.$$

□

Lemma

GMSBM produces a winner set W that induces a social welfare that is at most a factor $\sqrt{m}$ worse than the optimal social welfare.

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

Proof

- Let W^* be a set of winning bidders such that $\sum_{i \in W^*} v_i^*$ is maximal and $S_i^* \cap S_j^* = \emptyset$ for $i, j \in W^*$, $i \neq j$.
- Let W be the result of GMSMB.

Show:

$$\sum_{i \in W^*} v_i^* \leq \sqrt{m} \sum_{i \in W} v_i^* .$$

For $i \in W$ let

$$W_i^* = \{j \in W^* \mid j \geq i \text{ and } S_i^* \cap S_j^* \neq \emptyset\}$$

be the winners in W^* identical with i or not contained in W because of bidder i. …

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

Proof (ctd.)

Since no $j \in W_i^*$ is before i in the greedy ordering, for such j,

$$v_j^* \leq \frac{v_i^*}{\sqrt{|S_i^*|}} \sqrt{|S_j^*|} \qquad \text{and, summing over } j \in W_i^*$$

$$\sum_{j \in W_i^*} v_j^* \leq \frac{v_i^*}{\sqrt{|S_i^*|}} \sum_{j \in W_i^*} \sqrt{|S_j^*|}. \tag{1}$$

With Cauchy-Schwarz for $x_j = 1$ and $y_j = \sqrt{|S_j^*|}$:

$$\sum_{j \in W_i^*} \sqrt{|S_j^*|} \leq \sqrt{\sum_{j \in W_i^*} 1^2} \sqrt{\sum_{j \in W_i^*} |S_j^*|} = \sqrt{|W_i^*|} \sqrt{\sum_{j \in W_i^*} |S_j^*|}. \tag{2}$$

...

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

Proof (ctd.)

For all $j \in W_i^*$, $S_i^* \cap S_j^* \neq \emptyset$, i.e., there is a $g(j) \in S_i^* \cap S_j^*$.

Since W^* induces an allocation, for all $j_1, j_2 \in W_i^*$, $j_1 \neq j_2$,

$$S_{j_1}^* \cap S_{j_2}^* = \emptyset$$

Hence,

$$(S_i^* \cap S_{j_1}^*) \cap (S_i^* \cap S_{j_2}^*) = \emptyset$$

i.e., $g(j_1) \neq g(j_2)$ for $j_1, j_2 \in W_i^*$ with $j_1 \neq j_2$, making g an injective function from W_i^* to S_i^*.

Thus,

$$|W_i^*| \leq |S_i^*|. \tag{3}$$

...

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

Proof (ctd.)

Since W^* induces an allocation and $W_i^* \subseteq W^*$,

$$\sum_{j \in W_i^*} |S_j^*| \leq m. \tag{4}$$

...

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

Proof (ctd.)

Recall inequalities (1), (2), (3), and (4):

$$\sum_{j \in W_i^*} v_j^* \overset{(1)}{\leq} \frac{v_i^*}{\sqrt{|S_i^*|}} \sum_{j \in W_i^*} \sqrt{|S_j^*|}, \qquad |W_i^*| \overset{(3)}{\leq} |S_i^*|,$$

$$\sum_{j \in W_i^*} \sqrt{|S_j^*|} \overset{(2)}{\leq} \sqrt{|W_i^*|} \sqrt{\sum_{j \in W_i^*} |S_j^*|}, \qquad \sum_{j \in W_i^*} |S_j^*| \overset{(4)}{\leq} m.$$

With these, we get (5):

$$\sum_{j \in W_i^*} v_j^* \overset{(1)}{\leq} \frac{v_i^*}{\sqrt{|S_i^*|}} \sum_{j \in W_i^*} \sqrt{|S_j^*|} \overset{(2)}{\leq} \frac{v_i^*}{\sqrt{|S_i^*|}} \sqrt{|W_i^*|} \sqrt{\sum_{j \in W_i^*} |S_j^*|}$$

$$\overset{(3)}{\leq} \frac{v_i^*}{\sqrt{|S_i^*|}} \sqrt{|S_i^*|} \sqrt{\sum_{j \in W_i^*} |S_j^*|} \overset{(4)}{\leq} \frac{v_i^*}{\sqrt{|S_i^*|}} \sqrt{|S_i^*|} \sqrt{m} = \sqrt{m} v_i^*. \qquad \ldots$$

Greedy Mechanism for Single-Minded Bidders: Approximation Factor

Proof (ctd.)

Recall that for $i \in W$,

$$W_i^* = \{j \in W^* \mid j \geq i \text{ and } S_i^* \cap S_j^* \neq \emptyset\}.$$

Let $j \in W^*$.

- If $j \in W$: then by definition, $j \in W_j^*$ (assuming, WLOG, $S_j^* \neq \emptyset$).
- If $j \notin W$: then there must be some $i \in W$ such that $j \geq i$ and $S_i^* \cap S_j^* \neq \emptyset$, i.e., $j \in W_i^*$.

Therefore, for each $j \in W^*$, there is an $i \in W$ such that $j \in W_i^*$:

$$W^* \subseteq \bigcup_{i \in W} W_i^*. \qquad \ldots \tag{6}$$

Proof (ctd.)

Recall (5) and (6):

$$\sum_{j \in W_i^*} v_j^* \overset{(5)}{\leq} \sqrt{m} v_i^*, \qquad W^* \overset{(6)}{\subseteq} \bigcup_{i \in W} W_i^*.$$

With these, we finally obtain the desired estimation

$$\sum_{i \in W^*} v_i^* \overset{(6)}{\leq} \sum_{i \in W} \sum_{j \in W_i^*} v_j^* \overset{(5)}{\leq} \sum_{i \in W} \sqrt{m} v_i^* = \sqrt{m} \sum_{i \in W} v_i^*.$$

Thus, the social welfare of W differs from the optimal social welfare by a factor of at most $\sqrt{m}$. □

Greedy Mechanism for Single-Minded Bidders

The following theorem summarizes the results in this chapter:

Theorem

The greedy mechanism for single-minded bidders is efficiently computable, incentive compatible, and leads to an allocation whose social welfare is a $\sqrt{m}$-approximation of the optimal social welfare. □

Summary

- In combinatorial auctions, bidders bid for bundles of items.
- Exponential space needed just to represent and communicate valuations.
- Therefore: Focus on special case of single-minded bidders (compact representation of valuations).
- Unfortunately, still, optimal allocation NP-hard.
- Solution: approximate optimal allocation.
- Polynomial-time approximation possible for approximation factor no better than $\sqrt{m}$.
- Greedy mechanism for single-minded bidders:
  - achieves $\sqrt{m}$-approximation of social welfare,
  - is efficiently computable, and
  - is incentive compatible.