4. Social choice & mechanism design

Basic setting

Goal of social choice: *aggregate preferences of a group of agents*

A social choice problem consists of:

- a finite set $P = \{1, \ldots, n\}$ of agents
- a finite set O of outcomes (alternatives, candidates)
- for each player i, a preference ordering $\succ_i$ over O
  - must be a total order on O (use L for set of total orders on O)
  - $o_1 \succ_i o_2$ means agent i prefers o_1 to o_2

A social welfare function is a function $f : L^n \rightarrow L$.

A social choice function is a function $F : L^n \rightarrow O$.

Examples of voting rules

Scoring rules are based on a vector $(c_1, \ldots, c_n)$ of constants, where c_i represents the points a candidate receives for being ranked ith. A candidate wins if he gets the most points.

Some examples of scoring rules:

- Plurality: we use the vector $(1, 0, \ldots, 0)$
  - winner is the candidate who is ranked first most often
- Veto: we use $(1, 1, \ldots, 1, 0)$
  - winner is the candidate who is ranked last least often
- Borda: we use $(m-1, m-2, \ldots, 0)$

Examples of voting rules

Plurality with runoff: first use plurality to select top two candidates, then whichever is ranked higher than the other by more voters, wins

Single transferable vote (STV, instant runoff): candidate with lowest plurality score is removed; if you voted for that candidate, your vote transfers to the next (live) candidate on your list; repeat until a single candidate remains.

Approval: instead of giving a ranking, each voter divides the candidates into two sets, those he approves, and those he doesn't. The candidate with the most approvals wins.

Examples of voting rules

A pairwise election between candidates a and b consists in comparing how many voters rank a above b and how many prefer b to a.

Some voting rules based on pairwise elections:

- Copeland give two points to a candidate for each pairwise election he wins, one point for a tie, candidate with most points wins
- Simpson: choose candidate whose worst result in pairwise election is the best
- Pairwise elimination: pair up the candidates, those who lose are removed, repeat until only one candidate (like in sports tournaments)

Condorcet condition

A candidate is a Condorcet winner if she wins all pairwise elections.

Note: sometimes there is no Condorcet winner.

$$a \succ b \succ c \qquad b \succ c \succ a \qquad c \succ a \succ b$$

A voting rule satisfies the Condorcet condition if it always selects a Condorcet winner whenever one exists.

Plurality does not satisfy the Condorcet condition!

499 agents: $a \succ b \succ c$
3 agents: $b \succ c \succ a$
498 agents: $c \succ b \succ a$

Condorcet winner: b

Plurality winner: a

Sensitivity to losing candidate

Consider the following preference profile:

35 agents: $a \succ c \succ b$
33 agents: $b \succ a \succ c$
32 agents: $c \succ b \succ a$

Result with plurality?
Result with Borda?

Sensitivity to losing candidate

Consider the following preference profile:

35 agents: $a \succ c \succ b$
33 agents: $b \succ a \succ c$
32 agents: $c \succ b \succ a$

Result with plurality? a
Result with Borda?

Sensitivity to losing candidate

Consider the following preference profile:

35 agents: $a \succ c \succ b$
33 agents: $b \succ a \succ c$
32 agents: $c \succ b \succ a$

Result with plurality? a
Result with Borda?

Now suppose c drops out of the race, leaving only a and b:

35 agents: $a \succ b$
65 agents: $b \succ a$

Result with plurality?
Result with Borda?

Sensitivity to losing candidate

Consider the following preference profile:

35 agents: $a \succ c \succ b$
33 agents: $b \succ a \succ c$
32 agents: $c \succ b \succ a$

Result with plurality?
Result with Borda?

a

Now suppose c drops out of the race, leaving only a and b:

35 agents: $a \succ b$
65 agents: $b \succ a$

Result with plurality?
Result with Borda?

b

Sensitivity to the agenda setter

We'll consider the same preference profile:

35 agents: $a \succ c \succ b$
33 agents: $b \succ a \succ c$
32 agents: $c \succ b \succ a$

But now we use pairwise elimination.

Result if start by pairing a and b?

Result if start by pairing b and c?

Result if start by pairing a and c?

Sensitivity to the agenda setter

We'll consider the same preference profile:

35 agents: $a \succ c \succ b$
33 agents: $b \succ a \succ c$
32 agents: $c \succ b \succ a$

But now we use pairwise elimination.

Result if start by pairing a and b? c

Result if start by pairing b and c? a

Result if start by pairing a and c? b

The agenda setter can determine the outcome!

Sensitivity to the agenda setter

Another preference profile:

1 agent: $a \succ b \succ d \succ c$
1 agent: $b \succ d \succ c \succ a$
1 agent: $c \succ a \succ b \succ d$

We use pairwise elimination with ordering a, b, c, d.

Result?

Sensitivity to the agenda setter

Another preference profile:

1 agent: $a \succ b \succ d \succ c$
1 agent: $b \succ d \succ c \succ a$
1 agent: $c \succ a \succ b \succ d$

We use pairwise elimination with ordering a, b, c, d.

Result? d

Sensitivity to the agenda setter

Another preference profile:

1 agent: $a \succ b \succ d \succ c$
1 agent: $b \succ d \succ c \succ a$
1 agent: $c \succ a \succ b \succ d$

We use pairwise elimination with ordering a, b, c, d.

Result? d

But every agent strictly prefers b to d!

Properties of social welfare functions

Pareto efficiency (PE): A social welfare function F is Pareto-efficient if for any $o_1, o_2 \in O$, if $o_1 \succ_i o_2$ for all i, then $o_1 \succ_F o_2$.

Independence of irrelevant alternatives (IIA): A social welfare function F satisfies IIA if, for any $o_1, o_2 \in O$ and any $[\succ_i'], [\succ_i''] \in L^n$:

If $\forall i (o_1 \succ' o_2 \Leftrightarrow o_1 \succ'' o_2)$, then $o_1 \succ_{F([\succ_i'])} o_2 \Leftrightarrow o_1 \succ_{F([\succ_i''])} o_2$.

Nondictatorship: A social welfare function F is non-dictatorial if there does not exist $i \in P$ such that $o_1 \succ_i o_2 \Rightarrow o_1 \succ_F o_2$ for all $o_1, o_2 \in O$.

Arrow's impossibility theorem

Theorem

If $|O| \geq 3$, any social welfare function that is Pareto-efficient and satisfies IIA must be a dictatorship.

A very important, but rather negative result.

Idea: maybe we can do better for social choice functions

Properties of social choice functions

Weak Pareto efficiency (PE): A social choice function f is weak Pareto-efficient if, for any preference profile $[\succ] \in L^n$, if there exists $o_1, o_2 \in O$ such that $o_1 \succ_i o_2$ for all i, then $f([\succ]) \neq o_2$.

Monotonicity: A social choice function f is monotonic if, for any preference profile $[\succ_i] \in L^n$ with $f([\succ_i]) = o$, then for any other preference profile $[\succ_i']$ such that $\forall i \forall o'$, $o \succ_i' o'$ if $o \succ_i o'$, it must be the case that $f([\succ_i']) = o$.

Nondictatorship: A social choice function f is non-dictatorial if there does not exist $i \in P$ such that f always selects i's most preferred candidate.

Muller-Satterthwaite theorem

Theorem

If $|O| \geq 3$, any social choice function that is weakly Pareto-efficient and monotonic is dictatorial.

So negative result holds also for social choice functions.

Case of plurality

Plurality rule is weakly Pareto-efficient and non-dictatorial.

It follows from the previous theorem that plurality is not monotonic.

Consider the following preference profile:

3 agents: $a \succ b \succ c$
2 agents: $b \succ c \succ a$
2 agents: $c \succ b \succ a$

Plurality selects a

Case of plurality

Plurality rule is Pareto-efficient and non-dictatorial.

It follows from the previous theorem that plurality is not monotonic.

Consider the following preference profile:

3 agents: $a \succ b \succ c$
2 agents: $b \succ c \succ a$
2 agents: $c \succ b \succ a$

Plurality selects a

Now move c behind a in the last ranking:

3 agents: $a \succ b \succ c$
2 agents: $b \succ c \succ a$
2 agents: $b \succ a \succ c$

Plurality selects b!

Manipulability

So far we have assumed that voters report their true preferences.

However, sometimes a voter can improve the outcome by lying about her preferences.

Example: Borda

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ b \succ c \succ d$

Manipulability

So far we have assumed that voters report their true preferences.

However, sometimes a voter can improve the outcome by lying about her preferences.

Example: Borda

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ b \succ c \succ d$

a: 7, b: 8

Manipulability

So far we have assumed that voters report their true preferences.

However, sometimes a voter can improve the outcome by lying about her preferences.

Example: Borda

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ b \succ c \succ d$

a: 7, b: 8

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ c \succ d \succ b$

Manipulability

So far we have assumed that voters report their true preferences.

However, sometimes a voter can improve the outcome by lying about her preferences.

Example: Borda

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ b \succ c \succ d$

a: 7, b: 8

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ c \succ d \succ b$

a: 7, b: 6

Manipulability

So far we have assumed that voters report their true preferences.

However, sometimes a voter can improve the outcome by lying about her preferences.

Example: Borda

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ b \succ c \succ d$

a: 7, b: 8

2 agents: $b \succ a \succ c \succ d$
1 agent: $a \succ c \succ d \succ b$

a: 7, b: 6

A voting rule is called strategy-proof is no agent can benefit by lying about her preferences.

Gibbard-Satterthwaite theorem

A social choice function f is onto if for each candidate $o \in O$ there is a preference profile $[\succ_i]$ such that $f([\succ_i]) = o$.

Theorem

If $|O| \geq 3$, any social choice function that is onto and strategy-proof is dictatorial.

In other words, any voting rule which is not dictatorial and doesn't preclude some candidate winning can be manipulated.

Special case: single-peaked preferences

Suppose the candidates are ordered $o_1, \ldots, o_n$.

An agent's preference relation $\succ$ is single-peaked if there exists a candidate o^* such that candidates closer to o^* are preferred to candidates further from o^*:

$$o^* = o_k \quad \Rightarrow \quad \begin{array}{l} \forall\, 1 \leq i < j \leq k : o_j \succ o_i \\ \forall\, k \leq i < j \leq n : o_i \succ o_j \end{array}$$

Suppose all agents have single-peaked preferences, and they tell us their preferred option ("peak").

Social choice function which selects the median peak:

- satisfies the Condorcet condition
- non-manipulable (even though non-dictatorial and onto)

Computational issues in voting

Complexity of winner determination

For most voting rules, result computable in polytime.

For some rules (e.g. Slater ranking, Dogson's rule, Kemeny), determining the result can be NP-hard.

Complexity of manipulation

STV is NP-hard to manipulate, and other rules (e.g. plurality, Borda) can be slightly modified to get NP-hardness.

But also some empirical studies showing manipulation is easy in typical cases.

Mechanism design

Basic idea: social choice + game theory

As in social choice, we want to select an outcome based on the preferences of a group of agents.

However, the agents are rational, so might lie to us to try to obtain a better outcome.

Want to design a mechanism (game) such that it is in the best interest of the agents to tell the truth.

Gibbard-Satterthwaite theorem tells us this is not possible in general.

However, there are different settings in which this theorem does not apply, and for which positive results exist.

Examples of mechanism design problems

Auctions

Single-item auction: suppose we have an item we wish to sell, but we do not know what price the agents are willing to pay for it. If we want to sell the item to the agent who values it most, what should we do?

Matching problems

Stable marriage problem: there are n men and n women, and each person ranks all the members of the opposite sex in order of preference. We need to find a way of matching up the men and women so that no pair would prefer to be together rather than with their assigned partner. How can we accomplish this?

As the semester is almost over, our discussion of mechanism design will focus mostly on auctions.

Single-item auctions

Basic setting:

- one seller /auctioneer
- many bidders / buyers
- a single item up for sale
- value v_i of the item for each bidder i
- utility of $v_i - p$ if i wins and pays price p, and otherwise 0

We consider different auction protocols for determining who gets the item and for what price.

English auctions

Also known as ascending bid auctions.

Protocol:

- auction carried out interactively in real time
- bidding starts at reservation price and proceeds in rounds
- at each round, a bidder can propose a bid which is higher than the current one
- auction ends when no new bids are placed
- winner is the last bidder to place a bid, pays his bid

Commonly used for selling art and antiques.

English auctions

Bidders in English auctions have a dominant strategy:

keep placing (minimal) higher bids until you win or
the current price is higher than your valuation

Why dominant?

- if the current price is higher than your valuation, then you might lose utility by placing a bid, but you lose nothing if you don't bid

- if the current price is lower than your valuation, then you might gain utility by placing a bid, but you won't gain anything if you don't bid

Nice property: bidder with highest valuation wins
(assuming all bidders use their dominant strategy)

Dutch auctions

Also known as descending bid auctions.

Protocol:

- auction carried out interactively in real time
- the seller starts by announcing a very high price
- the price is lowered little by little until someone accepts
- this person is the winner, and pays the accepted price

Name comes from fact that this type of auction is used for selling flowers in the Netherlands.

Dutch auctions

What is the best strategy in a Dutch auction?

Ideally, we would like to accept at the lowest possible price, but how long after our valuation is called should we wait?

The problem is of course that the best time to accept depends on the valuations of the other players.

So there can be no dominant strategy for this type of auction.

Note: bidder with highest valuation may not win

First-price sealed-bid auctions (FPSB)

Protocol:

- bidders submit their "sealed bids" to the seller
- the winner is the buyer who submits the highest bid
- the winner pays his bid

What is a good strategy in such an auction?

Ideally, we would like to submit the lowest bid which will guarantee us a win.

The problem again is that we need to know the valuations of the other bidders, so no dominant strategy.

Vickrey auctions

Named after economist William Vickrey.

Also known as second-price sealed-bid auctions.

Protocol:

- bidders submit their “sealed bids” to the seller
- the winner is the buyer who submits the highest bid
- the winner pays the second highest bid

Vickrey auctions

Bidders in Vickrey auctions have a dominant strategy:

bid your valuation

Why dominant?

- if you bid higher than your valuation, then you might have to pay too much for the item

- if you bid lower than your valuation, then you might lose when you could have won, and if you do win, you pay the same as if you bid your true valuation

Nice property: bidder with highest valuation wins

Relationships between protocols

First-price sealed bid auctions can be seen as simulations of Dutch auctions.

During a Dutch auction, no information on other bidders' valuations until someone accepts. So the strategy of a bidder is just the price at which he will accept, which corresponds to the sealed-bid in FPSB.

Likewise, Vickrey auctions can be seen as simulations of English auctions.

In an English auction, the bidder with highest valuation wins, as in the Vickrey auction, and she pays the price at which the second highest bidder drops out, which corresponds to that bidder's valuation.

Which auction is best for the seller?

So far, we assumed the seller is interested in attributing the object to the buyer who values it the most.

But what if he just wants to make as much money as possible?

In general, not clear which auction is best:

- in English/Vickrey, seller only gets second-highest bid
- in Dutch/FPSB, bidders will usually submit a bid lower than their actual valuations

<u>Vickrey's Revenue Equivalence Theorem:</u>

Under some (rather strong) assumptions on the bidders' valuations, all four auction protocols give the same expected revenue.

Combinatorial auctions

We now consider a more general setting where the auctioneer may sell multiple items:

- one seller /auctioneer
- set B of bidders / buyers
- set G of items for sale
- a value $v_i(S)$ for each bidder i and each $S \in 2^G$
- utility of $v_i(S) - p$ for i if receives S and pays p

Note: we assume $v_i(\emptyset) = 0$ and $v_i(S) \leq v_i(S')$ whenever $S \subseteq S'$

Combinatorial auctions

Definition

An allocation of G among n bidders is a sequence of sets $S_1, S_2, \ldots, S_n$ such that $\cup_i S_i = G$ and $S_i \cap S_j = \emptyset$ for all $i \neq j$.

Bidders submit a (potentially untrue) valuation function to the seller, who then must choose an allocation of goods to the bidders and the payments to be made.

Combinatorial auction protocol = allocation rule + pricing rule

Vickrey auctions, revisited

Goal: a combinatorial auction protocol which works like Vickrey's.

In the Vickrey auction:

- allocation rule = give item to bidder who values it most
- pricing rule = pay bid minus some discount

Discount is difference between highest and second-highest bid.

Another way to see it is the difference in value attained with the bidder and without the bidder:

- with bidder, total value is highest bid
- without bidder, total value is second-highest bid

VCG mechanism

The Vickrey-Clarke-Groves mechanism generalizes the Vickrey auction to the case of multiple goods.

Basic idea:

- allocation rule = choose allocation which yields highest value
- pricing rule = pay bid minus difference in value caused by bidder

  Informally: $b_i - (maxval - maxval_{-i})$

To introduce VCG more formally, we need some notation.

Some terminology and notation

Use v_i for i's valuation, and $\hat{v}_i$ for i's reported valuation.

We use X to refer to the set of (feasible) allocations.

If $x \in X$, then $x(i)$ is the set of goods which x assigns to i.

Maximal value allocation:

$$x^* \in \text{argmax}_{x \in X} \sum_j \hat{v}_j(x(j))$$

Maximal value allocation if i not present:

$$x^*_{-i} \in \text{argmax}_{x \in X_{-i}} \sum_{j \neq i} \hat{v}_j(x_{-i}(j))$$

Pricing rule for VCG mechanism

Let x^* be a maximal value allocation, and for each bidder i, let x^*_{-i} be a maximal value allocation without i.

Then the price p_i for bidder i is defined as follows:

$$\begin{aligned} p_i &= \hat{v}_i(x^*(i)) - \left(\sum_j \hat{v}_j(x^*(j)) - \sum_{j \neq i} \hat{v}_j(x^*_{-i}(j)) \right) \\ &= \sum_{j \neq i} \hat{v}_j(x^*_{-i}(j)) - \sum_{j \neq i} \hat{v}_j(x^*(j)) \end{aligned}$$

Another way to see this: a bidder pays the amount of value the group loses by his participation

VCG example

Suppose the items a and b are for sale, and there are three bidders who report the following valuations:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 30, \hat{v}_1(\{b\}) = 10, \hat{v}_1(\{a,b\}) = 40$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 0, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a,b\}) = 50$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 10, \hat{v}_3(\{b\}) = 30, \hat{v}_3(\{a,b\}) = 35$

VCG example

Suppose the items a and b are for sale, and there are three bidders who report the following valuations:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 30, \hat{v}_1(\{b\}) = 10, \hat{v}_1(\{a,b\}) = 40$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 0, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a,b\}) = 50$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 10, \hat{v}_3(\{b\}) = 30, \hat{v}_3(\{a,b\}) = 35$

We first compute the optimal allocations:

Optimal allocation for the group: give a to 1, b to 3 (value of 60)
Optimal allocation without 1: give a and b to 2 (value 50)
Optimal allocation without 2: same as for whole group (value 60)
Optimal allocation without 3: give a and b to 2 (value 50)

So 1 gets a, 3 gets b, and the payments are as follows:

$$p_1 = 30 - (60 - 50) = 20 \qquad p_2 = 0 - (60 - 60) = 0$$

$$p_3 = 30 - (60 - 50) = 20$$

Properties of the VCG mechanism

The VCG mechanism has the property we were hoping for:

- strategy-proofness: truthful bidding is dominant strategy

It also has the following essential property:

- weak budget balance: bidders pay, not the auctioneer

It does however suffer some disadvantages:

- low or possibly even no revenue to the auctioneer
- non-monotonicity: sometimes revenue decreases unexpectedly
- possibilities for cheating (collusion, false-name bidding)
- high computational complexity

VCG and truthful bidding

Theorem:
In a VCG mechanism, truthful bidding is a dominant strategy.

Proof: Consider bidder i, and suppose x^*, x^*_{-i} as before.
Let $h_i = \sum_{j \neq i} \hat{v}_j(x^*_{-i}(j))$, and remark that h_i does not depend on i's reported valuation. We have:

$$p_i = h_i - \sum_{j \neq i} \hat{v}_j(x^*(j))$$

and hence i's utility can be expressed as

$$v_i(x^*(i)) - p_i = v_i(x^*(i)) + \sum_{j \neq i} \hat{v}_j(x^*(j)) - h_i$$

So i should choose $\hat{v}_i$ to maximize $v_i(x^*(i)) + \sum_{j \neq i} \hat{v}_j(x^*(j))$.

But seller chooses x^* to maximize

$$\sum_j \hat{v}_j(x^*(j)) = \hat{v}_i(x^*(i)) + \sum_{j \neq i} \hat{v}_j(x^*(j))$$

so setting $\hat{v}_i = v_i$ (i.e. telling the truth) is best strategy.

Weak budget balance

Weak budget balance: sum of payments is non-negative

This property holds for VCG because the absence of a bidder cannot decrease the value for the rest of the group.

However, this only is true because we assume bidders are always happy to accept additional goods.

<u>Example:</u>

Bidder 1: $v_1(\emptyset) = 0, v_1(\{a\}) = 90, v_1(\{b\}) = 10, v_1(\{a,b\}) = 10$
Bidder 2: $v_2(\emptyset) = 0, v_2(\{a\}) = 20, v_2(\{b\}) = 30, v_2(\{a,b\}) = 50$

VCG gives payments $p_1 = 20$ and $p_2 = -90$.

Lack of revenue

Consider the following reported valuations:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a, b\}) = 2$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 2, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a, b\}) = 2$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 0, \hat{v}_3(\{b\}) = 2, \hat{v}_3(\{a, b\}) = 2$

We compute the optimal allocations and payments:

Optimal allocation for the group: give a to 2, b to 3 (value of 4)
Optimal allocation without 1: give a to 2, b to 3 (value of 4)
Optimal allocation without 2: give a and b to 1 (value 2)
Optimal allocation without 3: give a and b to 1 (value 2)

$$p_1 = 0 - (4 - 4) = 0$$
$$p_2 = 2 - (4 - 2) = 0$$
$$p_3 = 2 - (4 - 2) = 0$$

$\Leftarrow$ nobody pays!!

Non-monotonicity

Consider the same bids as on previous slide:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a,b\}) = 2$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 2, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a,b\}) = 2$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 0, \hat{v}_3(\{b\}) = 2, \hat{v}_3(\{a,b\}) = 2$

What happens if bidder 3 is no longer present?

What if bidder 3 bids 1 for b (or a, b) instead of 2?

Non-monotonicity

Consider the same bids as on previous slide:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a, b\}) = 2$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 2, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a, b\}) = 2$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 0, \hat{v}_3(\{b\}) = 2, \hat{v}_3(\{a, b\}) = 2$

What happens if bidder 3 is no longer present?

What if bidder 3 bids 1 for b (or a, b) instead of 2?

In both cases, positive revenue to the auctioneer

So increasing the number of bidders or the size of bids does not necessarily lead to greater revenue.

Collusion

If bidders work together, they can obtain a better outcome by lying about their valuations.

Suppose the bidders make the following truthful bids:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a,b\}) = 4$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 1, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a,b\}) = 1$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 0, \hat{v}_3(\{b\}) = 1, \hat{v}_3(\{a,b\}) = 1$

Bidder 1 gets both objects and pays $4 - (4 - 2) = 2$

Now suppose bidders 2 and 3 both lie about their valuations:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a,b\}) = 4$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 4, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a,b\}) = 4$
Bidder 3: $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 0, \hat{v}_3(\{b\}) = 4, \hat{v}_3(\{a,b\}) = 4$

Bidder 2 gets a, bidder 3 gets b, both pay $4 - (8 - 4) = 0$ (nothing!)

False-name bidding

Another way for bidders to manipulation the VCG mechanism is to place multiple bids under fictitious names.

Consider the following example, where bidder 1 wins both goods:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a, b\}) = 4$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 1, \hat{v}_2(\{b\}) = 1, \hat{v}_2(\{a, b\}) = 2$

Suppose bidder 2 creates a fake bidder 2' with following bids:

Bidder 1: $\hat{v}_1(\emptyset) = 0, \hat{v}_1(\{a\}) = 0, \hat{v}_1(\{b\}) = 0, \hat{v}_1(\{a, b\}) = 4$
Bidder 2: $\hat{v}_2(\emptyset) = 0, \hat{v}_2(\{a\}) = 4, \hat{v}_2(\{b\}) = 0, \hat{v}_2(\{a, b\}) = 4$
Bidder 2': $\hat{v}_3(\emptyset) = 0, \hat{v}_3(\{a\}) = 0, \hat{v}_3(\{b\}) = 4, \hat{v}_3(\{a, b\}) = 4$

Bidder 2 gets both items and pays nothing!!

This problem is especially relevant in electronic auctions.

Computational complexity

VCG mechanism requires us to compute an optimal allocation of the goods, as well as optimal allocations with each bidder removed.

So if there are N bidders, we must find N+1 optimal allocations.

Unfortunately, this problem is computationally difficult:

Theorem:

The problem of finding an optimal allocation is NP-hard.

Note: cannot just use an approximately optimal allocation, since then the resulting mechanism might not be strategy-proof

References

We unfortunately didn't get to cover much social choice and mechanism design, so if you're interested, here are some references to learn more:

- chapters 9-12 of AGT book (link to pdf on website)
- chapter 9-10 of MAS book (link to pdf on website)
- "Computational social choice" course by Ulle Endriss (UVA)
  - I used lectures 13 & 14 for preparing part on auctions, but the course covers many more topics
  - slides and video of lectures available at:

    http://staff.science.uva.nl/~ulle/teaching/comsoc/2009/