Mechanism design

Lecture 1: Introduction, VCG, single parameter settings

Elias Koutsoupias
Oxford

ADFOCS 2024

A simple mechanism design setting: Single item auction

- A seller has a single item to sell
- There are n potential buyers, aka bidders, players, agents
- Buyer i has value v_i for the item

If bidder i acquires the item at a price p_i, their utility will be $u_i = v_i - p_i$. This is called a quasilinear utility setting.

The essential difficulty of auctions and more generally of mechanism design is that **the values v_i are private**.

Thus, a mechanism must **elicit** these values and compute

- the outcome, i.e., who gets the item
- the payment of each bidder

Examples of single item auctions

- English or Ascending auction
  - price starts at 0 and goes up
  - as the price goes up, bidders drop out
  - last bidder to remain gets the item and pays the current price
- Dutch or Descending auction
  - price starts at infinity and goes down
  - first bidder to accept the price wins the item and pays the price
- Sealed-bid first-price auction
  - bidders submit their bids in sealed envelopes
  - highest bidder gets the item and pays the highest bid
- Sealed-bid second-price auction
  - bidders submit their bids in sealed envelopes
  - highest bidder gets the item and pays the second highest bid

First-price auction $\leftrightarrow$ Dutch auction

Second-price auction $\leftrightarrow$ English auction

First-price auction

- v_i value of bidder i
- b_i bid of bidder i, not necessarily equal to v_i
- p_i payment of bidder i
- $u_i = v_i - p_i$ utility of bidder i

In the **first-price auction**, the item is given to the bidder with the maximum bid, who pays their bid.

An auction induces a game between the bidders. This is usually an incomplete information game.

What do the players know? Two common settings:

- players have complete information; they know the values of all bidders
- Bayesian setting, in which values come from known probability distributions: $v_i \sim F_i$; player i knows v_i and $F_1, \ldots, F_n$.

First-price auction – complete information example

Two bidders with values $v_1 = 4$ and $v_2 = 7$ participate in a first-price auction.

(Let's assume that the bids must be positive integers and in case of a tie, the item is given to bidder 1.)

This defines a 4×7 matrix game. What are its **Nash equilibria**? It has a few of them. For example:

- $(b_1, b_2) = (4, 5)$, which gives utilities $(u_1, u_2) = (0, v_2 - b_2) = (0, 2)$ or
- $(b_1, b_2) = (3, 4)$, which gives utilities $(u_1, u_2) = (0, v_2 - b_2) = (0, 3)$

Is there a **dominant strategy equilibrium**?

First-price auction – Bayesian example

Two bidders with values drawn independently from the $[0, 1]$ uniform distribution.

In the **Bayesian setting**, the appropriate equilibrium concept is **Bayes-Nash equilibrium**, in which deviations do not increase the expected utility.

It can be shown that $(b_1, b_2) = (\frac{v_1}{2}, \frac{v_2}{2})$ is a Bayes-Nash equilibrium.

Dominant strategies and truthful auctions

In auctions and more generally in mechanism design, it is desirable to move beyond Nash equilibria and consider (weakly) dominant-strategy equilibria.

Definition

A mechanism is **truthful** (or incentive compatible) if bidding truthfully is a weakly dominant strategy equilibrium.

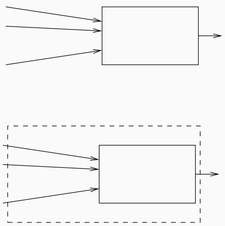
In the jargon of auctions, this is also known as **DSIC** — dominant strategy incentive compatible.

In these lectures, we consider DSIC mechanisms.

Dominant strategies and truthful auctions

We want **truthful bidding**, i.e. $b_i = v_i$, to be a weakly dominant strategy for a few reasons:

- Bidding truthfully remains a dominant strategy even when a bidder has incomplete information about the values of the other bidders
- It makes it easier for bidders to compute their best strategy
- The **revelation principle** works :: every mechanism is equivalent (same allocation, same payments) to a truthful mechanism



Second-price (Vickrey) auction

Are there any interesting truthful auctions?

The first-price auction is not truthful. In particular, the highest bidder has no reason to bid higher than the second highest bid.

Second-price (Vickrey) auction

Theorem

The second-price auction is truthful.

Proof.

The payment for bidder i is $p_i = \max_{j \neq i} b_j$.

If $b_i \geq p_i$ then bidder i wins and gains $u_i = v_i - p_i$. Otherwise $u_i = 0$.

Bidder i selects b_i to maximize their utility:

$$u_i = \max(v_i - p_i, 0).$$

So,

- if $v_i - p_i \geq 0$, the bidder should bid any value greater than p_i; in particular $b_i = v_i$ is optimal
- otherwise the utility will be $u_i = 0$, so by bidding $b_i < p_i$, the bidder will lose the item and achieve utility 0; in particular $b_i = v_i$ is optimal.

Truthfulness in second-price auction

Why is the second-price auction truthful?

- The **payment depends only** on the allocation and the values of the other players
- **The allocation is monotone**: increasing the declared value makes it more likely to get the item

A look at truthfulness

Consider one bidder with value v for an item. Let

- b be the bid, the value that the bidder declares
- $a(b)$ be the probability or fraction that the bidder gets
- $p(b)$ be the payment

- The utility of the bidder is

$$u(b|v) = a(b) \cdot v - p(b)$$

- **For which functions a and p is the mechanism truthful?** That is, when

$$u(v|v) = \sup_b u(b|v)?$$

A look at truthfulness

Theorem

A mechanism is truthful if and only if

- *the utility* $u(v) = u(v|v)$ *of the bidder is a convex function of the private value* v.
- *the probability of getting the item is given by*

$$a(v) = u'(v)$$

Note: no mention of payments!

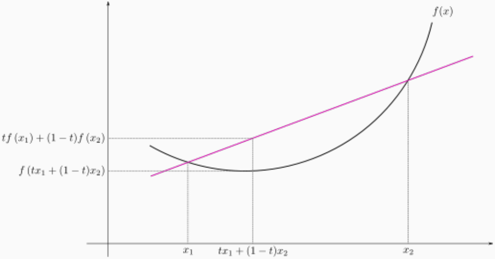
A digression: convexity

Convexity

Definition: A function $f : R^n \to R$ is called convex when

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(\lambda x + (1 - \lambda) y)$$

for every x, y and $\lambda \in [0, 1]$.



The three layers of convexity

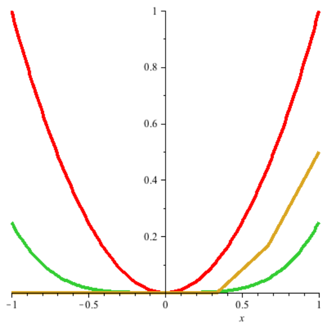
We focus on functions of `one variable`, but everything generalizes appropriately to many variables.

The following are equivalent (for doubly differentiable functions)

1. $f(x)$ is convex
2. $f'(x)$ is monotone (nondecreasing)
3. $f''(x)$ is nonnegative

Examples of convex functions

- x^2
- $\frac{1}{4}x^4$
- $\max\{0, \frac{x}{2} - \frac{1}{6}, x - \frac{1}{2}\}$



Important properties of convex functions

Proposition

For every function g, the function f defined by

$$f(x) = \sup_y \{x \cdot y - g(y)\},$$

is convex.

Proposition

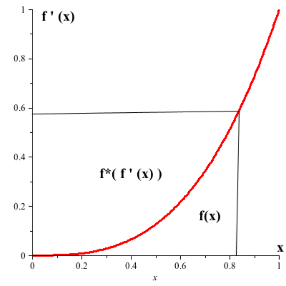
For every convex function f, there exists a function f^* (called the conjugate of f), such that

$$f(x) = \sup_y \{f'(y) \cdot x - f^*(f'(y))\}$$

Conjugate

The conjugate function f^* of a function f is defined by

$$f^*(y) = \sup_x \{x \cdot y - f(x)\}$$



$$y = f'(x)$$

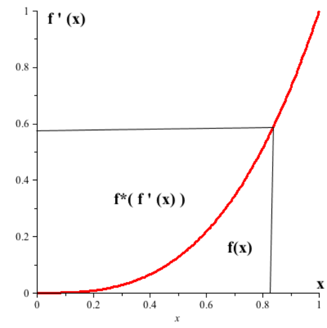
Notice the symmetry

$$f^*(y) = \sup_x \{x \cdot y - f(x)\} \qquad x \leftrightarrow y$$

$$f(x) = \sup_y \{x \cdot y - f^*(x)\} \qquad f \leftrightarrow f^*$$

Example

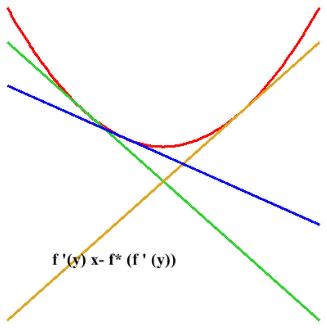
- $f(x) = \frac{1}{4}x^4$
- $f'(x) = x^3$
- $f^{*\prime}(x) = x^{1/3}$
- $f^*(x) = \frac{3}{4}x^{4/3}$
- $f'(x) \cdot x - f^*(f'(x)) = x^3 \cdot x - \frac{3}{4}(x^3)^{4/3} = f(x)$



Supporting hyperplanes

For every convex function f, the conjugate function f^* defines the supporting hyperplanes

$$f(x) = \sup_y \{f'(y) \cdot x - f^*(f'(y))\}$$



Example:

$$f(x) = \frac{1}{4}x^4 = \sup_y \left\{ y^3 \cdot x - \frac{3}{4}y^4 \right\}.$$

A look at truthfulness

Putting everything together for truthful mechanisms:

- the **utility** of the agent is $u(v) = \text{argmax}_b\{a(b) \cdot v - p(b)\}$, which is convex even if the mechanism is not truthful.
- Convexity implies $u(v) = \sup_y\{u'(v) \cdot v - u^*(u'(y))\}$
- The **allocation** is $a(v) = u'(v)$, and
- the **payment** is $p(v) = u^*(u'(v))$, where u^* is the conjugate of u.

We can add a constant to the payment without affecting the argmax: $p(v) = u^*(u'(v)) + \text{const}$.

Back to truthfulness

Truthfulness and monotonicity

Monotonicity for the single-item auction: For any fixed bids of the other bidders, when bidder i increases their bid b_i, the chances of getting the item cannot decrease.

- The second-price auction is truthful and allocates the item to the agent with the **highest value**.
- Is there a truthful mechanism to allocate it to the agent with **median** (or **minimum**) value?
  No, because the allocation function is not monotone.

The general mechanism design framework
domains and objectives

Beyond single-item auctions

- k-unit auction: there are k copies of an item and each bidder wants a single copy.
- combinatorial auctions: there are m items and each bidder i has a private valuation $v_i(S)$ for every subset S of these items.
- general mechanism design setting: there is a set A of possible outcomes and each bidder has a private valuation $v_i(a)$ for every $a \in A$.

General mechanism design setting

Definition (Mechanism design setting)

We can define a mechanism design problem by:

- the set of bidders or agents. Wlog we assume it to be $\{1, 2, \ldots, n\}$.
- the set A of possible outcomes
- sets of valuation functions $V_1 \times \cdots \times V_n$, one for each bidder. Each element $v_i \in V_i$ determines the value of bidder i for each outcome: $v_i : A \to \mathbb{R}$.

Example (Two voters, three candidates)

	Alice	**Bob**	**Carol**
Voter 1	10	18	20
Voter 2	21	18	12

Single-item auction is the general mechanism design setting

Example (Single-item auction)

Suppose that $v_1^*, \ldots, v_n^*$ denote the values of the bidders in a single-item auction. Then

- The set of outcomes is $A = \{1, 2 \ldots, n\}$: which bidder gets the item.
- The valuation functions are of the form

$$v_i(a) = \begin{cases} v_i^* & a = i \\ 0 & \text{otherwise} \end{cases}$$

	O1	**O2**	**O3**
Player 1	v_1	0	0
Player 2	0	v_2	0
Player 3	0	0	v_3

Combinatorial auction

Example (Combinatorial auction)

Suppose that $v_1^*(S), \ldots, v_n^*(S)$ denote the valuation functions of the bidders in a combinatorial auction of m items. Then

- The set of outcomes A contains all allocation functions of m items to n bidders. Such an allocation can be represented by a legal 0-1 matrix $a_{i,j}$, $i \in [n]$, $j \in [m]$, where $a_{i,j} = 1$ if and only if bidder i gets item j.
- The valuation functions are of the form

$$v_i(a) = v_i^*(\{j \colon a_{i,j} = 1\}).$$

Example (3 players, 2 items)

	O1	O2	O3	O4	O5	O6	O7	O8	O9
Agent 1	$v_1^*(12)$	$v_1^*(1)$	$v_1^*(1)$	$v_1^*(2)$	$v_1^*(2)$	0	0	0	0
Agent 2	0	$v_2^*(2)$	0	$v_2^*(1)$	0	$v_2^*(12)$	$v_2^*(1)$	$v_2^*(2)$	0
Agent 3	0	0	$v_3^*(2)$	0	$v_3^*(1)$	0	$v_3^*(2)$	$v_3^*(1)$	$v_3^*(12)$

Mechanisms in the general setting

Definition

Fix a mechanism design setting with n bidders, set of outcomes A, and set of valuation functions $V = V_1 \times \cdots \times V_n$.

A (direct revelation) mechanism consists of two parts:

- a social choice function $f\colon V \to A$
- a vector of payment functions $p = (p_1, \ldots, p_n)$, where $p_i\colon V \to \mathbb{R}$.

- Each bidder i provides its valuation function $v_i \in V_i$.
- The outcome is determined by the social choice function f and the payments of the bidders are determined by the payment functions.
- The utility of bidder i is $u_i(v) = v_i(f(v)) - p_i(v)$.

Definition (Truthful)

A mechanism (f, p) is called **truthful or incentive compatible** if for every player i, every $v \in V$ and every $v_i' \in V_i$:

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

Single-parameter domains

An important special class of mechanism design is the **single-parameter setting**.

- The private value of a bidder i is a single real value, $v_i^* \in \mathbb{R}$
- For every outcome a, the value of the bidder is proportional to v_i^*, i.e., $v_i(a) = \lambda_i(a) v_i^*$, for some λ_i.

Example (Examples)

- Shortest path problem on a graph, where every edge e belongs to some agent who is willing to sell it at a price.
- A multi-unit auction where the value of bidder i for k items is $(2k^2 - 1)v_i^*$.

Vickrey-Clarke-Groves (VCG) mechanism

For a given outcome, the sum of the values of all bidders is called **social welfare**, i.e., the social welfare for outcome a is $\sum_{i\in[n]} v_i(a)$.

The VCG mechanism is a truthful mechanism, which **selects the outcome that maximizes the social welfare**. For example, for the single-item auction, it allocates the item to the bidder with the highest value.

Definition (Vickrey-Clarke-Groves (VCG) mechanisms)

The VCG mechanism has

- $f(v) = \operatorname{argmax}_{a\in A} \sum_{i\in[n]} v_i(a)$
- $p_i(v) = -\sum_{j\neq i} v_j(f(v)) + h_i(v_{-i})$, for some $h_i \colon V_{-i} \to \mathbb{R}$.

The VCG mechanism

Vickrey-Clarke-Groves (VCG) mechanism

The VCG mechanism has

- $f(v) = \text{argmax}_{a \in A} \sum_{i \in [n]} v_i(a)$
- $p_i(v) = -\sum_{j \neq i} v_j(f(v)) + h_i(v_{-i})$, for some $h_i : V_{-i} \to \mathbb{R}$.

Note that the payments of VCG are not completely determined.

Choosing $h_i(v_{-i}) = \max_{a \in A} \sum_{j \neq i} v_j(a)$ is called the **Clarke pivot rule**. With these payments, one can interpret the VCG as the mechanism that each bidder pays their value minus a discount equal to the increase of the social welfare due to their participation in the mechanism.

$$p_i(v) = v_i(f(v)) - \left(\sum_{j \in [n]} v_j(f(v)) - \max_{a \in A} \sum_{j \neq i} v_j(a) \right)$$

Examples of the VCG mechanism

	Alice	**Bob**	**Carol**
Bidder 1	10	18	20
Bidder 2	21	18	12

VCG selects Bob (his column has the maximum sum).

Bidder 1 pays 18-(36-21)=3

Bidder 2 pays 18-(36-20)=2

Examples of the VCG mechanism

Example (Second-price auction)

The second price auction is VCG with Clarke pivot rule.

The social welfare of the VCG outcome is equal to the highest value. For simplicity, assume that $v_1 \geq v_2 \cdots \geq v_n$. Then the social welfare is v_1. If the winner does not participate, the social welfare will drop to v_2, so the winner pays their value (v_1) minus a discount $v_1 - v_2$; so the payment is $v_1 - (v_1 - v_2) = v_2$.

Examples of the VCG mechanism

Example (multi-unit auction)

There are $k \geq 1$ identical units of a good and each bidder wants a single one of them.

VCG (with Clarke pivot rule) will give the k items to the k highest bids and each one of them will pay the $(k+1)$-st highest bid.

Examples of the VCG mechanism

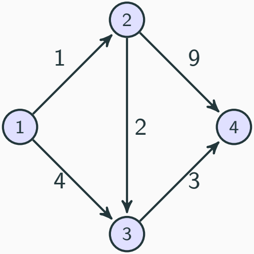
Example (Buying a shortest path)

Given a graph in which every edge is controlled by a different seller, we want to buy a shortest path from some vertex s to some other vertex t.

Note that this is an inverse auction (procurement): the bidders are sellers and the auctioneer a buyer.

The VCG mechanism will select the shortest path. Each bidder will get their value plus the increase in the length of the shortest path when we remove their edge.

Examples of the VCG mechanism



- VCG selects a shortest path P: $P = (1, 2, 3, 4)$
- Edges not in P are paid nothing
- To compute the payment of an edge e on the path P:
  - We remove e and compute a shortest path P_e
  - The payment for edge e is

$$p_e = v_e + \text{length of } P_e - \text{length of } P$$

For example,
  - for edge $[1, 2]$, $P_e = (1, 3, 4)$. The payment is $1 + 7 - 6 = 2$
  - for edge $[2, 3]$, $P_e = (1, 3, 4)$. The payment is $2 + 7 - 6 = 3$

Examples of the VCG mechanism

Example (Public project)

A city wants to undertake a public project with cost C, for example to build a new road. There are n citizens/bidders. Bidder i will get benefit v_i from the project.

The city will undertake the project when the sum of reported values exceeds C, i.e., $\sum_{i \in [n]} b_i \geq C$. The social welfare will be $\sum_{i \in [n]} v_i - C$ if the road is built, and 0 otherwise.

If we use the VCG mechanism, the payment of bidder i will be 0, unless bidder i is critical. A bidder is critical if the total bids of the other bidders is below C, but together with b_i the sum is above C. In this case, the bidder will pay $C - \sum_{j \neq i} b_j$.

Note that this solution has some undesirable properties: it is **not budget-balanced** (i.e., the total of payments is less than C in general), and it is susceptible to collusion (e.g., if two bidders report that their values are C, the project will be built, and they will pay nothing).

Truthfulness of the VCG

Theorem

VCG is truthful.

Proof.

Fix some player i, v_{-i}, and v_i. If b_i is the bid of bidder i, its utility is

$$u_i(b_i, v_{-i}) = v_i(f(b_i, v_{-i})) - p(b_i, v_{-i}).$$

We want to show that $b_i = v_i$ maximizes this expression. By the definition of VCG, $p(b_i, v_{-i}) = -\sum_{j \neq i} v_j(f(b_i, v_{-i})) + h_j(v_{-i})$, so

$$u_i(b_i, v_{-i}) = \sum_{j \in [n]} v_j(f(b_i, v_{-i})) + h_i(v_{-i}).$$

Note that the term with h_i is not affected by b_i, so bidder i wants to select b_i that maximizes the social welfare $\sum_{j \in [n]} v_j(f(b_i, v_{-i}))$. But, by the definition of VCG, this is maximized when $b_i = v_i$. □

It follows from the proof that VCG aligns the interests of all bidders with the objective of the mechanism (i.e., to maximize the social welfare).

Why VCG is not always the answer

VCG is a truthful mechanism that can be applied to **every mechanism design setting**.

Why do we need to search for other mechanisms?

- The computational or communication complexity of VCG may be prohibitive (e.g. combinatorial auctions).
- VCG optimizes the social welfare. But in many cases, the objective may be different.
  - For example, a usual objective is to maximize revenue. The theory of **optimal auctions** tries to maximize revenue when the mechanism designer knows the probability distributions of the values of bidders.
  - Another example: in the **scheduling problem**, there are n machines (bidders) and m tasks, and we want a mechanism to allocate the tasks to minimize the makespan. The objective here (makespan) is different than the social welfare.
- Payments may not be allowed (e.g. in voting), or payments may have to satisfy certain conditions (e.g., in the public project setting, we require **budget balance**).

The related machines scheduling problem

The related machines scheduling problem

There are n machines (agents) with speeds $s_1, \ldots, s_n$ which are private values. There is also a set of tasks T to be scheduled on the machines. The mechanism consists of the allocation function $a(s_1, \ldots, s_n)$ that allocates the tasks to the machines and the payment functions $p_1(s_1, \ldots, s_n), \ldots, p_n(s_1, \ldots, s_n)$.

Greedy algorithm = allocate one-by-one the items to optimize the makespan myopically.

Greedy is not truthful: Take two machines with the first slightly faster than the second and jobs $2, 1+\epsilon, 1+\epsilon$. The first (fast) machine will get the first job, while the second will get the other two tasks.

The related machines scheduling problem

Two things to note:

- each set of tasks defines a different mechanism design setting
- the objective is to minimize the **makespan**, i.e., the time when every task has finished.

Therefore VCG, which maximizes the total welfare ($=$ the sum of completion times), may not be optimal.

Actually, VCG has approximation ratio n (equal to the number of machines).

The related machines scheduling problem

Theorem

There exists a truthful mechanism with optimal makespan.

Algorithm: return the **lexicographically minimum** among the optimal allocations.

Let $w = (w_1, \dots, w_n)$ be the load assigned to the machines. w is lexicographically smaller than w' if there is k such that

$$
\begin{aligned}
w_i &= w'_i && \text{for } i < k \text{, and} \\
w_k &< w'_k
\end{aligned}
$$

This algorithm achieves optimal makespan, but it is not a polynomial time algorithm. However, there is a monotone PTAS.

Selling digital goods

Selling digital goods

Formally, we have n unit-demand bidders with valuations $v_1, \ldots, v_n$ and n identical items.

Suppose that $v_1 \geq v_2 \geq \cdots \geq v_n$ and assume we know them. If we want to maximize revenue with the **same price for all bidders**, we should select $p = v_i$, where v_i maximizes iv_i.

What can we do if we don't know the values?

For every bidder i, a truthful mechanism should make a take-it-or-leave-it offer of some price p_i.

Example (Random Sampling Optimal Pricing (RSOP))

The bidders are uniformly partitioned into two parts, and the optimal single price of each part (i.e., $\text{argmax}_{v_i}\{i \cdot v_i\}$) is offered to the bidders of the other part.

This is a **prior-free mechanism**: the designer does not assume anything about the values.