MECHANISM DESIGN

Auction Theory

Different kinds of auctions:

- Sealed bid: Submit bids privately, winner decided at the end
- English: Seller starts with a low price and keeps increasing it.
- Dutch: Seller starts with a high price and keeps decreasing it.

· Typical auctions. Eg: IPL
· Upper bounds can be used as stoppers ⇒ second price auction

· Lower bounds can be used as stoppers ⇒ First price auction

Why auctions?

- Sell an object / Buy a service
- Price discovery mechanism. Eg: art
- Perishable goods that need to be sold quickly. Eg: Flowers, fruits, etc.

Sealed Bid Auctions

- 1 seller, 1 object for sale.
- n bidders, each bidder has a value $v_i \geq 0$ for the object.
- Process:
  - Bidders choose a bid (b_i) and submit in a sealed envelop.
  - Auctioneer opens the envelops. Selects the highest bidder as the winner.
- Two main flavors:
  1) 1st price auction
  2) 2nd price auction

First Price Auction

- Winner pays what he bids
- N- bidders

  $(v_i)_{i \in N}$ — valuation ; $(b_i)_{i \in N}$ — bid

  $$u_i(b) = \begin{cases} \boxed{v_i - b_i} & \text{if } i = \text{argsmax } b \\ 0 & \text{otherwise} \end{cases}$$

- For a tuple $b = (b_1 \ldots b_n) \in \mathbb{R}^n$, denote the least l s.t. $b_l = \max_{k \in \{1 \ldots n\}} b_k$ by **argsmax b**

Characterization Theorem 1 : Consider the game associated with the first price auction with valuation v. Then,

b is a Nash Equilibrium iff for $i = \text{argsmax } b$

1. $b_i \leq v_i$ // i does not suffer the winner's curse $\xrightarrow{\text{OTHERWISE}}$ Better to bid $b_i \leftarrow v_i \rightarrow 0$ v/s -ve utility
2. $\max_{j \neq i} v_j \leq b_i$ // winner submitted a sufficiently high bid $\xrightarrow{\text{OTHERWISE}}$ Others have incentive to deviate: $b_j \in (b_i, v_j)$
3. $b_i = \max_{j \neq i} b_j$ // another player submitted the same bid $\xrightarrow{\text{OTHERWISE}}$ $b_i \leftarrow b_j$ is a better strategy

Eg: $v = (1, 8, 6, 3)$;
$b = (1, 6, 6, 3)$, $b = (1, 6, 5, 6) \longrightarrow$ NE

Second Price Auction

$$u_i(b) = \begin{cases} \boxed{v_i - \max_{j \neq i} b_j} & \text{if } i = \text{argsmax } b \\ 0 & \text{otherwise} \end{cases}$$

- When the two highest bids co-incide, the utilities are the same as in first price auction.
- Winner's curse is possible: When $v_i < b_i$ and some other player bids in (v_i, b_i)

Characterization Theorem 2: For the game associated with second price auction, the tuple of bids b is a Nash Equilibrium iff for $i = \text{argsmax}\ b$:

1. $\max_{j \neq i} v_j \leq b_i$ — // Sufficiently high bid $\xrightarrow{\text{OTHERWISE}}$ j has incentive to deviate: specifically, $b_j \in (b_i, v_j)$ provides better utility
2. $\max_{j \neq i} b_j \leq v_i$ — // Valuation is sufficiently high $\xrightarrow{\text{OTHERWISE}}$ i obtains -ve utility; better to make a lower bid (indifferent in $[0, v_i]$)

Ex 1: Truthful bidding constitutes an NE.
- Let $i = \text{argsmax}\ v$; i wins under truthful bidding

Ex 2: Consider the profile where i bids v_i and all other players bid $0 \longrightarrow$ NE

Ex 3: Suppose the two highest valuations are v_i & v_j with $0 < v_i < v_j$ and $i < j$
- Consider the bid vector where $b_i = v_j$, $b_j = v_i$, rest $0 \longrightarrow$ NE
- Then, i wins and $u_i(b) = 0$
- For j to win, he has to bid above $b_i = v_j$

Incentive Compatibility

General form of an auction:
- Fix the payment function pay_i

$$u_i(b) = \begin{cases} v_i - pay_i(b) & \text{if } i = \text{argsmax}\ b \\ 0 & \text{otherwise} \end{cases}$$

- Note: pay_i cannot / does not depend on v_i, since the auctioneer only gets to know b.

Only for the sake of analysis do we assume oracles for / knowledge of v.

- Denote the resulting game by $G_{pay.v}$
- A sealed bid auction with payments $pay_1, \ldots, pay_n$ is incentive compatible if for all sequences v of player valuations, for each bidder i, his valuation v_i is a dominant strategy in the corresponding game $G_{pay.v}$

Mechanism Design

- Identify the required outcome.

  Given an outcome, how should the game be designed in order to achieve that outcome?

Decision Problem

- Set of decisions — D
- Set of players — $N = \{1, 2, \ldots, n\}$
- Set of types Θ_i for each player i
- An initial utility function

  $v_i : D \times \Theta_i \longrightarrow \mathbb{R}$
- Notation: $\Theta = \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n$

  $\Theta_{-i} = \Theta_1 \times \ldots \times \Theta_{i-1} \times \Theta_{i+1} \times \ldots \times \Theta_n$
- Decision rule $f : \Theta \longrightarrow D$
- A decision problem $(D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$

* The process of obtaining a decision (with a central authority):

1. Each player has a type $\theta_i \in \Theta_i$

2. Each player informs the central authority his type $\theta'_i \in \Theta_i$

- Tuple of declared types:
$$\theta' = (\theta'_1, \theta'_2, \ldots, \theta'_n)$$

3. Central authority takes a decision $d := f(\theta')$ and informs all players
4. The initial utility for player i is $v_i(d, \theta_i)$

- The central authority only gets to know the declared types θ' and not real types θ — private to each player.

Ex: Sealed bid auction: single object for sale
- $D = \{1, \ldots, n\}$
- For all $i \in N$, $\Theta_i = \mathbb{R}_+$, $\theta_i \in \Theta_i$ is player i's valuation of the object
- $v_i(d, \theta_i) = \begin{cases} \theta_i, & \text{if } d = i \\ 0, & \text{otherwise} \end{cases}$
- $f(\theta') = \text{argsmax } \theta'$

Ex: Public project problem
- $D = \{0, 1\}$ · $\Theta_i = \mathbb{R}_+$
- $v_i(d, \theta_i) = d(\theta_i - c/n)$
- $f(\theta') = \begin{cases} 1 & \text{if } \sum_{i=1}^{i=n} \theta'_i \geq c \\ 0 & \text{otherwise} \end{cases}$

Ex: Reverse sealed bid auction

The bidders are "selling" a service
↪ The cheapest one is selected

- $D = \{1, \ldots, n\}$ · $\Theta_i = \mathbb{R}_-$
- $f(\theta') = \text{argsmin } \theta'$

- $v_i(d, \theta_i) = \begin{cases} \theta_i & \text{if } d = i \\ 0 & \text{otherwise} \end{cases}$

Efficiency

A decision rule f is efficient if for all $\theta \in \Theta$ and $d' \in D$

$$\sum_{i=1}^{i=n} v_i(f(\theta), \theta_i) \geq \sum_{i=1}^{i=n} v_i(d', \theta_i)$$

- For all $\theta \in \Theta$, $f(\theta)$ — the decision that maximises the initial social welfare $\sum_{i=1}^{i=n} v_i(d, \theta_i)$
- That is, $f(\theta) \in \underset{d \in D}{\operatorname{argmax}} \sum_{i=1}^{i=n} v_i(d, \theta_i)$

Direct Mechanism

- Use tax to prevent strategic manipulation
- $(D, (\Theta_i)_{i \in N}, (v_i)_{i \in N}, f)$ is modified as follows:
    - Set of decisions $D \times \mathbb{R}^n$
    - Decision rule $(f, t) : \Theta \longrightarrow D \times \mathbb{R}^n$ where $t : \Theta \longrightarrow \mathbb{R}^n$ and
      $(f, t)(\theta) = (f(\theta), t(\theta))$
    - Final utility $u_i : D \times \mathbb{R}^n \times \Theta_i \longrightarrow \mathbb{R}$ defined as
      $$u_i(d, t_1, \ldots, t_n, \theta_i) = v_i(d, \theta_i) + t_i$$

Direct mechanism is a tuple
$(D \times \mathbb{R}^n, (\Theta_i), (v_i), (f, t))$

When the true type of player i is θ_i and the announced type is θ_i',

$$u_i((f, t)(\theta_i', \theta_{-i}'), \theta_i) = v_i(f(\theta_i', \theta_{-i}'), \theta_i) + t_i(\theta_i', \theta_{-i}')$$

where θ_{-i}' is the announced types of the other players.

- Given θ' of announced types, $(t_1(\theta'), \ldots, t_n(\theta'))$ is the vector of payments, players have to make.
  If $t_i(\theta) \geq 0$, player i receives from the central authority, $|t_i(\theta')|$.
  If $t_i(\theta) < 0$, player i pays the central authority, $|t_i(\theta')|$.

Incentive Compatibility

A direct mechanism is incentive compatible if for all $\theta \in \Theta$, $i \in \{1, \ldots, n\}$ and $\theta_i' \in \Theta_i$,

$$u_i((f,t)(\theta_i, \theta_{-i}'), \theta_i) \geq u_i((f,t)(\theta_i', \theta_{-i}'), \theta_i)$$

Groves Mechanism

Direct mechanism with tax function $t = (t_1, \ldots, t_n)$ where $\forall\ i \in \{1, \ldots, n\}$,

- $t_i : \Theta \longrightarrow \mathbb{R}$ is defined by
  $$t_i(\theta') = g_i(\theta') + h_i(\theta_{-i}')$$
- $g_i(\theta') = \sum_{j \neq i} v_j(f(\theta'), \theta_j)$
- $h_i : \Theta_{-i} \longrightarrow \mathbb{R}$ is any arbitrary function

- Note: $v_i(f(\theta), \theta_i) + g_i(\theta) = \sum_{j \in N} v_j(f(\theta), \theta_j)$ — the initial social welfare.
  Final social welfare: $\sum_{j \in N} u_j((f,t)(\theta), \theta_j)$

- Theorem. Let $(D, (\Theta_i)_{i \in N}, (v_i)_{i \in N}, f)$ be a decision problem with an efficient decision rule f. Then, every Groves' mechanism is incentive compatible.

- A direct mechanism is feasible if, for all θ', $\sum_{i=1}^{i=n} t_i(\theta') \le 0$
  (i.e. can be realised without external financing)
- For all θ', if $\sum_{i=1}^{i=n} t_i(\theta') = 0$ — budget balanced

Pivotal Mechanism [Clarke]

— a special case

$$h_i(\theta'_{-i}) = -\max_{d \in D} \sum_{j \neq i} v_j(d, \theta'_j)$$

$$-\max_{d \in D} \Big[\underbrace{\sum_{j \neq i} v_j(d, \theta'_j)}_{\text{What } j \text{ could have gotten had } i \text{ not been there}} - \underbrace{\sum_{j \neq i} v_j(f(\theta'), \theta'_j)}_{\text{What } j \text{ gets now that } i \text{ is participating in the mechanism}} \Big]$$

If i was not participating, what would things have been like? Essentially, make i pay for the damage done to the society.

$$\Rightarrow t_i(\theta') = \sum_{j \neq i} v_j(f(\theta'), \theta'_j) - \max_{d \in D} \sum_{j \neq i} v_j(d, \theta'_j)$$

Note: $\because \forall i \in N$, $t_i(\theta') \le 0$

— Pivotal Mechanism is feasible

Ex: Sealed bid auction

$D = \{1, \ldots, n\}$, $\Theta_i = \mathbb{R}_{\ge 0}$, $f(\theta') = \operatorname{argsmax} \theta'$

$$v_i(d, \theta_i) = \begin{cases} \theta_i & \text{if } d = i \\ 0 & \text{otherwise} \end{cases}$$

$$t_i(\theta') = \begin{cases} -\max_{j \neq i} \theta'_j & \text{if } d = i \\ 0 & \text{otherwise} \end{cases}$$

$t_i = v_i + h_i$, $h_i := -\max_{j \neq i} \theta'_j$

Q. Can we modify second price auction to yield a larger social welfare?

Perhaps, a 2^{nd} PA as a price discovery mechanism, with limited taxes and $\therefore$ better social welfare?

Currently, final SW in 2^{nd} PA is

$$\sum_{j=1}^{j=n} u_j((f, t)(\theta'), \theta_j) = \theta_i - \max_{j \neq i} \theta'_j$$

where $i = \operatorname{argsmax} \theta'$

Notation: For a sequence $\theta \in \mathbb{R}^n$, let θ^* denote the re-ordering of θ from largest to smallest.

Ex: $\theta = (1, 5, 2, 4, 0)$

$\Rightarrow \theta^* = (5, 4, 2, 1, 0)$

$\theta_{-2} = (1, 2, 4, 0)$

$(\theta_{-2})^* = (4, 2, 1, 0)$

$(\theta_{-2})^*_2 = 2$

Bailey-Cavallo Mechanism

- Let $t_i'(\theta')$ be the tax under 2^{nd} PA, then $t_i(\theta') = t_i'(\theta') + h_i'(\theta_{-i})$
  where $h_i'(\theta_{-i}) = (\theta_{-i})^*_2 / n$
  Note: This gives a Groves mechanism since,
  $$t_i(\theta') = \sum_{j \neq i} v_j(f(\theta'), \theta_j')) + (h_i + h_i')(\theta_{-i}')$$

- Let π be a permutation of $1, \ldots, n$ s.t. the i^{th} highest bid is by player $\pi(i)$
- So, the object is sold to $\pi(1)$. Then,
  $(\theta_{-i})^*_2 = \theta^*_3$ for $i \in \{\pi(1), \pi(2)\}$
  $(\theta_{-i})^*_2 = \theta^*_2$ for $i \in N \setminus \{\pi(1), \pi(2)\}$

  Thus, in Bailey-Cavallo mechanism, $\pi(1)$ makes payments:
  - θ^*_3 / n to player $\pi(2)$
  - θ^*_2 / n to player $\pi(3), \ldots, \pi(n)$
  - $\theta^*_2 - 2\dfrac{\theta^*_3}{n} - (n-2)\dfrac{\theta^*_2}{n} = \dfrac{2}{n}(\theta^*_2 - \theta^*_3)$

    to the central authority.

Note: If the central authority is included in the SW, then the value of SW is θ_i

Ex: Assume that players reveal their true type:

Player	Type	Tax	u_i
A	18	7	7
B	21	6	6
C	24	-15	9

- 2^{nd} PA:
  SW: $0 + 0 + (24-21)$
  $= 3$
- Bailey-Cavallo mechanism:
  $\theta_2^* = 21, \ \theta_3^* = 18$
  C pays:
  - $18/3 = 6$ to player B
  - $21/3 = 7$ to player A
  - $18/3 = 6$ to himself
  - $21 - 19 = 2$ to the CA

  SW: $(6) + (7) + (24 - 21 + 6)$
  $= (24) - (2) = 22$

STABLE MATCHING

Q. [Gale & Shapley]
Can one design an admission process, or a job-recruiting process that is self-enforcing?

Q. Given a set of preferences among employers and applicants, can we assign applicants to employers so that for every employer E, and every applicant A who is not scheduled to work for E, at least one of the following two hold:
- E prefers every one of its accepted

applicants to A
- A prefers her current situation over working for employer E
- If this holds, then the outcome is stable
- If a pair (A, E) exists for which neither of the two conditions hold, it is called a blocking pair.

Matching

- PS:
  - A set of n men $M = \{m_1, \dots, m_n\}$
  - A set of n women $W = \{w_1, \dots, w_n\}$
- A matching S is a set of ordered pairs each from $M \times W$ such that each member of M and each member of W appears in at most one pair in S.
- A perfect matching S' is a matching with the property that each member of M and each member of W appear in exactly one pair in S'.

* Each man $m \in M$ ranks all the women (without ties).

$w \succ_m w'$ — m prefers w to w'

Analogously, each woman also ranks all the men (without ties).

Q. Given the preference ordering and a perfect matching, what can go wrong?

$\exists$ (m, w) and (m', w') in S s.t.

$w' >_m w$ and $m >_{w'} m'$

→ We would like to find a stable matching — a perfect matching without instabilities.

* A matching is unstable if there are two men m and m' and two women w and w' such that,
- m is matched to w
- m' is matched to w'
- $w' >_m w$ and $m >_{w'} m'$

The pair (m, w') is called a blocking pair.

— A matching is stable if it is perfect and has no blocking pair.

Ex: $M = \{m, m'\}$ and $W = \{w, w'\}$

1. $w >_m w'$ $\quad w >_{m'} w'$
   $m >_w m'$ $\quad m >_{w'} m'$

   Stable Matching:
   (m, w), (m', w')
   (Unique)

2. $w >_m w'$ $\quad w' >_{m'} w$
   $m >_w m'$ $\quad m >_{w'} m'$

   Stable Matching:
   (m, w), (m', w')
   (m', w), (m, w')

Q. Does there exist a stable matching for every set of preference orderings? ✓

Q. Given a set of preference orderings, can we efficiently construct a stable matching if there is one? ✓

- Main idea — temporary matching, have the pair (m, w) enter a temporary matching (engaged).

Gale - Shapley Algorithm

Initially all $m \in M$ and $w \in W$ are free

```
WHILE (there is an m who is free and
       hasn't proposed to every woman)
DO
    Choose such a man m
    Let w be the first woman in m's
    preference to whom m has
    not yet proposed
    IF w is free THEN
        (m, w) become engaged
    ELSE
        w is currently engaged to m'
        IF w prefers m' to m THEN
            m remains free
        ELSE
            w prefers m to m'
            (m, w) become engaged
            m' becomes free
        END
    END
END
```

Observations:

- The set of engaged pairs form a matching
- A woman w remains engaged from the point at which she receives her

first proposal and the sequence of partners she is engaged to gets better (w.r.t. w's preference).

- The sequence of women to whom m proposes gets worse (w.r.t. m's preference).
- The algorithm terminates after at most n^2 iterations.
  - → $P(t)$: Set of pairs (m, w) such that m has proposed to w by the end of iteration t

$$|P(t)| < |P(t+1)| \le n^2$$

Correctness

Lemma 1. If m is free at some point in the execution of the algorithm, then there is a w to whom he has not proposed yet.

Lemma 2. The set S returned at termination is a perfect matching.

Theorem. Let S be the matching returned by the execution of the algorithm. S is a stable matching.

Proof. Suppose S is not stable. Then, there is a blocking pair, i.e., $\exists\ (m, w)$ and (m', w') in S s.t. $w' >_m w$ & $m >_{w'} m'$.

In the execution that produced S, m's last proposal is to w.
Q: Did m propose to w'?
No: w must occur higher up in m's preference than w'.
But, $w' >_m w$ $\Rightarrow\Leftarrow$
Yes: m was rejected by w' in favour of some m'', and w' prefers m'' to m.
m' is the final partner of w'.
So, either $m' = m''$ or w' prefers m' to m''. Either way,
$m' \geq_{w'} m'' >_{w'} m$ $\Rightarrow\Leftarrow$

$\square$

Properties

Best valid partner

- A woman w is a valid partner of m if there is a stable matching that contains the pair (m, w).
- w is the best valid partner of m if w is a valid partner of m and there is no $w' >_m w$ which is a valid partner of m.
- Notation: $best(m)$ — best valid partner of m

Theorem: Let S be the output of the Gale-Shapley algorithm. In the stable matching S, each m is paired with the best valid partner.

Theorem: In S, each w is paired with the worst valid partner.

Incentive Compatibility

Q. In the male optimal algorithm, do women have an incentive to lie? Yes.

M – A: $X > Y > Z$, B: $X > Z > Y$, C: $Y > X > Z$
F – X: $C > A > B$, Y: $A > C$, Z: Any
Truthful
X: $C > B > A$
Lie
$AY, CX, BZ >_X AX, BZ, CY$

Q. Is there any incentive compatible (or truthful) mechanism? for both men and women

A. No [Roth]

Q. When can people benefit from lying?

A. Only if they have more than one stable match.

* Correlated preference lists –

If everyone has the same preference list, then everybody has a unique stable partner.

· If preference orderings are independent random permutations, almost every person has more than one stable partner [Knuth]

One-Sided Market

[Shapley-Scarf House allocation problem]

· Set of agents $N = \{1, \ldots, n\}$
· Set of items $H = \{h_1, \ldots, h_n\}$
($|H| \geq |N|$ in general)

- Each agent has a preference order over H and is allocated a single item from H.
- A matching is a bijection $\pi : N \to H$
- A pair (i, j) is a **blocking pair** in π if $\exists\ \pi' \in M$ where
  - $\pi(i) = \pi'(j)$ and $\pi(j) = \pi'(i)$
  - $\pi(k) = \pi'(k) \quad \forall\ k \in N \setminus \{i, j\}$
  - $\pi'(i) >_i \pi(i)$ and $\pi'(j) >_j \pi(j)$

π is **stable** if there are no blocking pairs in π. → 2-stable

Q. Is there always a stable outcome in the house allocation problem? Yes

Serial Dictatorship

- Fix any order of agents $i_1, \ldots, i_n$.
- Greedy approach:
  - Take a player in turn according to the ordering above.
  - Assign the most-preferred item on the preference list.
- Outcome depends on the order by which applicants are processed.

Q. What if the players have an initial allocation? [Housing market / One-sided market]
Starting with an arbitrary allocation, can we have a "good" reallocation

of items? Yes.

* **Core Stable**: An allocation where no coalition of players is able to improve the utilities of all the players involved by exchanging items.

· **Blocking Coalition**: For $\pi \in M$, $X \subseteq N$ is a blocking coalition if $\exists\ \pi'$ and a bijection $\mu : X \to X$ s.t. for all $i \in X$,

$\pi(i) \neq \pi'(i)$, $\pi'(i) = \pi(\mu(i))$, $\pi'(i) >_i \pi(i)$

· π is **core stable** if there are no blocking coalitions in π.

Top-Trading Cycle Algorithm

Theorem. A core-stable outcome always exists in one-sided markets.

Solution. Top trading cycle algorithm [Gale].

Algorithm:

- Let π^* be an arbitrary initial allocation.
- S1: Given an allocation π, create a graph on the item set as follows:

  $x \to y$, if y is the most-preferred item for agent $\pi^{-1}(x)$ from the items in the graph at the moment.

- S2: For any cycle $(x_1, x_2, \ldots, x_k, x_1)$, assign x_i to $\pi^{-1}(x_{i-1})$ $(x_{1-1} = x_k)$. Remove these items.
- Repeat S1 and S2 until there are no items remaining in the graph.

Termination: TTC terminates since at least one item is removed in each round.

$\because$ self-loops are allowed to exist

Correctness:

Theorem. Let π^* be the matching generated by TTC. π^* is core stable.

Proof. Let $C_1, C_2, \ldots, C_t$ be the sequence of cycles processed in TTC.

All agents in C_1 are receiving their most preferred item. So, no agent in C_1 has an incentive to deviate.

... All agents in C_i are receiving their most preferred items left in round i.

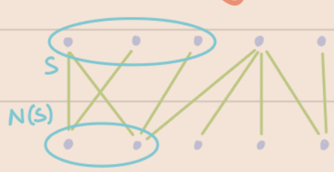
Bipartite Matching

- $G = (V, E)$ is **bipartite** if $V = A \cup B$ s.t. $A \cap B = \phi$ and $E \subseteq A \times B$
- **Matching** in G is a set of edges $M \subseteq E$ s.t. each node appears in at most one edge in M.
- M is a perfect matching if each node appears in exactly one edge in M.
- Problem: Given an arbitrary G, find a matching of maximum size.
- If $|A| = |B| = n$ then there is a perfect matching iff the maximum matching has size n.

Q. It is easy to demonstrate a perfect matching when it exists — indicate the edges.
But, how do you show that a bipartite graph has no perfect matching?

Constricted Sets

- Let $G = (V, E)$ be a bipartite graph such that $V = A \cup B$
- For any $S \subseteq B$ let $N(S) = \{ u \in A \mid \exists\, w \in S \text{ s.t. } (u, w) \in E \}$, the neighbours of set S.
- A set $S \subseteq B$ is constricted if $|S| > |N(S)|$



Theorem. If a bipartite graph has no perfect matching, then it must contain a constricted set.

MARKET CLEARING PRICES

The Setting

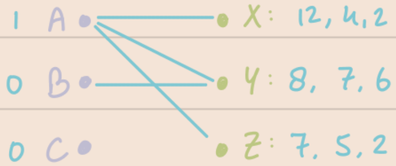
- We have a collection of sellers each with a house and a collection of buyers, each who want to buy a house ($|sellers| = |buyers|$)
- Each buyer has a valuation for each house, v_{ij} — valuation that buyer j has for seller i's house
- Each valuation is a non-negative integer.

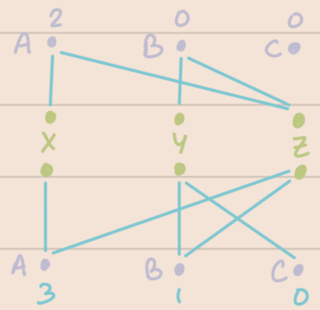
Payoff

- Each seller i puts his house for a price $p_i > 0$
- If buyer j buys the house from seller i, the payoff is: $v_{ij} - p_i$
- Given the price for each house, to maximize his payoff, buyer j will buy the house for which $v_{ij} - p_i$ is maximized.

- If $v_{ij} - p_i$ is maximized in a tie between multiple sellers, then the buyer j can choose any one of them.
- If $v_{ij} - p_i$ is negative for every choice of seller i, then buyer j prefers not to buy any house.
- The seller(s) that maximizes the payoff for buyer j is (are) called the **preferred seller(s)** for buyer j (provided the payoff is not negative).

- For a set of prices, the **preferred seller graph** is the graph obtained by constructing an edge between each buyer and his preferred seller(s).





Market Clearing Prices

- A set of prices is **market clearing** if the resulting preferred seller graph has a perfect matching.

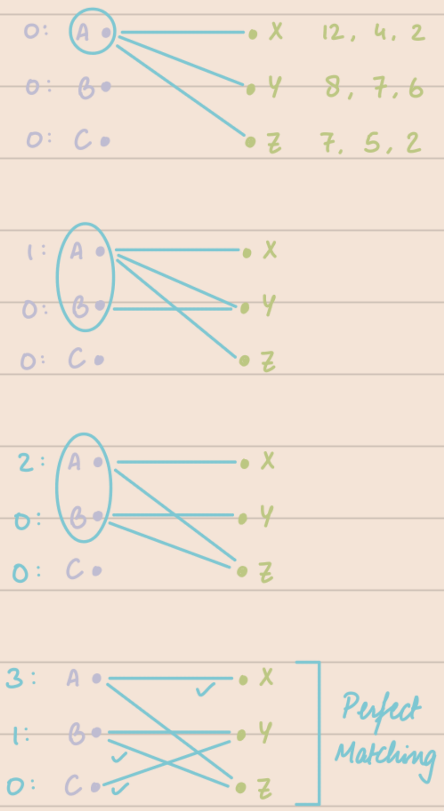
Theorem. For any set of buyer valuations, there exists a set of market clearing prices.

- We show the existence of MCPs by describing an auction procedure that constructs these prices.

- Features of the auction procedure:
  - multiple objects are being auctioned
  - multiple buyers have different valuations for the objects.
- This auction procedure was descibed by Demange, Gale and Sotomayer.

The Auction

- Initially, all sellers set their price to 0.
- Construct the preferred seller graph and check if there is a perfect matching.
- If there is a perfect matching then the prices are market clearing.
- If not, find a constricted set of buyers S and their neighbours N(S).



- Each seller in N(S) simultaneously raises his price by one unit.
- Reduction step. If we reach a state where all prices are strictly greater than 0, then we reduce the prices by subtracting the smallest price p from each price. This drops the

lowest price to 0.

- Begin the next round of the auction, using the new prices.

Correctness

- The only way that the auction procedure can come to an end is if it reaches a set of MCPs. Otherwise, the rounds continue.
- So, if we can show that the auction must come to an end for any set of buyer valuations, then it follows that the MCPs always exist.
- We define a potential function which keeps track of a progress measure in each round of the auction.

The Potential

- For any set of prices, the potential of a buyer is the maximum payoff he gets from any seller.
- For any set of prices, the potential of a seller is the current price he receives.
- Potential of the auction: Sum of the potential of all participants — buyers and sellers.

How does the potential of the auction change as we run the auction?

- At the beginning, all sellers have potential 0. Each buyer has a potential equal to his maximum valuation for any house.
  Potential of the auction — some $P_0 \geq 0$
- At the start of each round, everyone has potential at least 0:
  - Sellers → at least 0 ∵ prices at least 0
  - Price reduction → lowest price 0
    → Each buyer has payoff at least as much as his valuation for this house
    → Potential at least 0
- ∴ Potential of auction is at least 0 at the start of each round.
- Reduction of prices does not change the potential of the auction — Potential changes only when the prices change.
- When the sellers in $N(S)$ all raise their prices by one unit, each of these sellers' potential goes up by one unit.
  The potential of each buyer in S goes down by one unit ∵ all their preferred houses get more expensive.

- Since $|S| > |N(S)|$, the potential of the auction decreases by at least one unit.
- The potential of the auction:
  - starts at some value $P_0 \geq 0$
  - decreases by at least one unit in each round of the auction
  - cannot drop below 0
- Thus, the algorithm terminates
$\Rightarrow$ MCPs always exist.

SPONSORED SEARCH

Ex:

CTR (Click-Through Rate)	Slots	Advertiser	RPC (Rate per Click)	Valuation (CTR · RPC)
10	A	X	3	30, 15, 6
5	B	Y	2	20, 10, 4
2	C	Z	1	10, 5, 2

What buyer x gets to know

What search Engine gets to know

- If the valuations are known to the search engine, easy to generate the MCPs.

Ex: A (13) : X , B (3) : Y , C (0) : Z

Q. Price-setting mechanism when the search engine does not know the valuations?

Auction: Advertisers report their RPC. Highly Confidential

- Analogue of 2^{nd} PA for multiple slots.

Generalised 2^{nd} PA

- Each advertiser j, announce a bid b_j — price it is willing to pay per click.
- Each slot i is awarded to the i^{th} highest bidder at a price per click equal to the $(i+1)^{th}$ highest bid.
- Bids per click — $b_1\ b_2\ b_3 \ldots$ $\xrightarrow[\text{descending}]{}$

  Slot i : $r_i b_{i+1}$

Ex:

CTR	Slots	Adv.	RPC	Valuation
10	A	X	7	70, 28, 0
4	B	Y	6	60, 24, 0
0	C	Z	1	10, 4, 0

- Suppose advertisers bid truthfully, then

  $A \rightarrow X$: price = 6 / click

  $B \rightarrow Y$: price = 1 / click

  Payment for X : $6 \times 10 = 60$

  Utility for X : $70 - 60 = 10$

  Utility for Y : $24 - 4 = 20$

Q. Is GSP incentive compatible?

- Suppose X bids 5 instead of 7.

  Utility for X(B): $28 - 4 = 24 > 10$

Conclusion: GSP is not incentive compatible.

- Equilibria in GSP:

  (X, Y, Z) bid (5, 4, 2) is an NE

that maximizes social welfare.
(X, Y, Z) bid (3, 5, 1) is another NE but it does not maximize social welfare.

- Is there a different mechanism we can consider? Yes, recall Pivotal mechanism.
$$t_i(\theta) = \sum_{j \neq i} v_j(f(\theta), \theta_j) - \max_{d \in D} \sum_{j \neq i} v_j(d, \theta_j)$$

VCG Pricing

- Assignment: Maximize total valuation
- V^S_B — maximum total valuation over all possible perfect matchings.

S: Sellers
B: Buyers

$P_{ij} = V^S_{B-j} - V^{S-i}_{B-j}$ / $tax_{ij} = V^{S-i}_{B-j} - V^S_{B-j}$

V^S_{B-j} — maximum total valuation when buyer j is not present
V^{S-i}_{B-j} — maximum total valuation when seller i and buyer j are not present

- Note: MCP's — posted prices
  VCG — personalised; depends on item & buyer.

Ex:

CTR	Slots	Adv.	RPC	Valuation
10	A	X	3	30, 15, 6
5	B	Y	2	20, 10, 4
2	C	Z	1	10, 5, 2

Assignment: (A, X), (B, Y), (C, Z)

Tax for X = (10 + 2) − (20 + 5) = −13
Utility for X = 30 − 13 = 17
Tax for Y = (30 + 2) − (30 + 4) = −2
Tax for Z = (30 + 10) − (30 + 10) = 0

VCG – the general case

- Set of players N
- Set of alternatives A
- Player i has valuation $v_i(a)$ for each $a \in A$
- Goal: Select an alternative a^* that maximizes total valuation $\sum_{i \in N} v_i(a^*)$
- Strategy for a player i is $b_i : A \to \mathbb{R}$
- Auction process chooses a^* that maximizes $\sum_{i \in N} b_i(a^*)$
- Player i has to pay

$$p_i(b) = \underbrace{\max_{a \in A} \sum_{j \neq i} b_j(a)}_{\text{Social welfare without } i} - \underbrace{\sum_{j \neq i} b_j(a^*)}_{\text{Social welfare of others when } i \text{ participates}}$$

$$t_i(b) = -p_i(b)$$

Combinatorial Auction

Items

	A	B	C	AB	BC	CA	ABC
P_1	3	4	0	9	6	5	14
P_2	4	1	3	8	6	7	9
P_3	3	2	7	4	11	8	15
P_4	0	4	2	6	8	4	1

Bidders

- VCG allocation will maximize initial SW:
  $P_1 \leftarrow \{A, B\}$, $P_2 \leftarrow \phi$, $P_3 \leftarrow \{C\}$, $P_4 \leftarrow \phi$
  with initial SW = 16
- To compute tax, need to find maximum SW had the agent not been there.
- P_1:
  - Allocation: $P_2 \leftarrow \phi$, $P_3 \leftarrow \{A, B, C\}$, $P_4 \leftarrow \phi$
  - Tax: $(0 + 7 + 0) - (0 + 15 + 0) = -8$

  P_3:
  - Allocation: $P_1 \leftarrow \{A, B, C\}$, $P_2 \leftarrow \phi$, $P_4 \leftarrow \phi$
  - Tax: $(9 + 0 + 0) - (14 + 0 + 0) = -5$

  Tax for P_2 and P_4 is 0 since they were not allotted anything, so the $\because$ optimal $\because$ allocation remains the same even in their absence.