Algorithmic Game Theory

LECTURE 9

Main Topics Covered:

Mechanism Design with Payment Constraints

- Budget Constraints
- The Uniform-Price Multi-Unit Auction
- The Clinching Auction
- Mechanism Design without Money

Budget Constraints

- In many mechanisms there are constraints on the payments charged by the mechanism

- A budget constraint limits the amount of money an agent can pay

- To incorporate the budget constraint, the utility is now redefined as follows:

$$\begin{array}{ll} v_i(\omega) - p_i & if \;\; p_i \leq B_i \\ -\infty & if \; p_i > B_i \end{array}$$

  where ω is the outcome and B_i is the budget, p_i is payment

- A new auction format is now needed to accommodate the budget constraint in utility !

Uniform-Price Multi-Unit Auctions (1/2)

- Multi-Unit Auctions →
  - *m* identical items
  - Each bidder has a private valuation v_i for each item he/she gets
  - Each bidder wants as many items as possible
    Thus bidder i obtains value k.v_i from k items
  - Each bidder has a *public* budget B_i
  - Multi-unit auctions are single-parameter environments
- Some definitions:
  - Demand of a bidder

Suppose supply is 'm' number of items

Demand of bidder i ($D_i(p)$) = $min\left\{floor\left(\frac{B_i}{p}\right), m\right\}$ $if\ p < v_i$

= 0 $if\ p > v_i$

Uniform-Price Multi-Unit Auctions (2/2)

- Definitions continued
  - Aggregate Demand

  For a price *p* different from all bidders' valuations, we define the aggregate demand by

  $$A(p) = \sum_{i=1}^{n} D_i(p)$$

  In general we define the limits of A(p) from below and above as follows:

  $A^{-}(p) = lim_{q \uparrow p} \sum_{i=1}^{n} D_i(q)$ and $A^{+}(p) = lim_{q \downarrow p} \sum_{i=1}^{n} D_i(q)$

- Uniform-price auction picks price *p* that that equalizes supply and aggregate demand
  Gives every bidder his/her demanded number of items at a price of p each

The Uniform-Price Auction

1. Let p equalize supply and aggregate demand, meaning $A^{-}(p) \geq m \geq A^{+}(p)$.
2. Award $D_i(p)$ items to each bidder i, each at the price p. Define demands $D_i(p)$ for bidders i with $v_i = p$ so that all m items are allocated.

Uniform-Price Auction - Disadvantages

- Although, the uniform-price auction respects the bidders' budgets, it has a major drawback
- Uniform-Price Auction is not DSIC
- It is vulnerable to the problem of Demand Reduction [Lecture 8]
- Therefore, we need to modify the payment and allocation rules of uniform-price auction to make it DSIC !
- This is where the "Clinching Auction" comes into play

Clinching Auctions (1/2)

- The items are sold at increasing prices
- In addition to the current price (p), the auction keeps track of the current supply s (initially m) and the residual budget $\widehat{B_i}$ (initially B_i) of each bidder i
- The residual demand $\widehat{D_i}(p)$ of bidder i at price p != v_i is defined w.r.t the residual budget and supply:

$$\widehat{D_i}(p) = min\left\{floor\left(\frac{\widehat{B_i}}{p}\right), m\right\} \quad if\ p < v_i$$
$$= 0\ if\ p > v_i$$

- The clinching auction iteratively raises the current price p and the bidder i *clinches* some items at price p whenever they are uncontested
- Different items are sold in different iterations
- The auction continues until all items have been allocated

Clinching Auctions (2/2)

- Merits of Clinching Auction
  - No Demand Reduction
  - Clinching Auction is Feasible

    The clinching auction always stops, allocates exactly m items and charges payments that are at most bidders’ budget
  - Clinching Auction is DSIC

    The clinching auction for bidders with public budgets is DSIC

Mechanism Design Without Money

- Mechanism Design without Money is relevant for designing and understanding methods for voting, organ donation and school choice
- Example:
  - House Allocation Problem:

    There are n agents, each owning a house. Each agent's preferences are represented by a total ordering over the n houses.

    Here, reallocation of the houses is done by the Top Trading Cycle (TTC) Algorithm

Top Trading Cycle (TTC) Algorithm

initialize N to the set of all agents
while $N \neq \emptyset$ **do**
  form the directed graph G with vertex set N and edge set $\{(i, \ell)$:
   i's favorite house within N is owned by $\ell\}$
  compute the directed cycles $C_1, \ldots, C_h$ of G^3
  `// self-loops count as directed cycles`
  `// cycles are disjoint`
  for each edge (i, ℓ) of each cycle $C_1, \ldots, C_h$ **do**
    reallocate ℓ's house to agent i
  remove the agents of $C_1, \ldots, C_h$ from N

References

- Twenty Lectures on Algorithmic Game Theory by Tim Roughgarden, 2016, Cambridge University Press
- https://www.youtube.com/playlist?list=PLEGCF-WLh2RJBqmxvZ0_ie-mleCFhi2N4

THANK YOU