Algorithmic Game Theory
CoReLab (NTUA)

Lecture 9:

Combinatorial Auctions

Recap (Single Parameter Environment)

Single-parameter environment

A special case of the general mechanism design setting
able to model simple auction formats:

- n bidders.
- Each bidder i has a valuation $v_i \in R$ which is her value "per unit of stuff" she gets.
- A feasible set X . Each element of X is an n −vector $(x_1, \ldots, x_n)$,
  where x_i denotes the "amount of stuff" that player i gets.

Recap (General Model)

General model

- Set $N = \{1, 2, \ldots, n\}$ of agents (players).
- Set Ω of possible outcomes.
- Valuation vector $v = (v_1, \ldots, v_n) \in V$
  where $v_i : \Omega \to R$ is the (private) valuation function of each player.
- Outcome function: $f : V^n \to \Omega$
- Payment vector: $p = (p_1, \ldots, p_n)$ where $p_i : V^n \to R$

Remember: **VCG** guarantees DSIC, Surplus Maximization
for any case included in the general model!

Combinatorial Auctions

Combinatorial Auctions

- n bidders.
- m indivisible goods.
- Bidders have private valuation for each outcome.
- Outcome : $(S_1, \ldots, S_n)$ allocation of the goods to bidders.
- Social welfare : $\sum_{i=1}^{n} v_i(S_i)$.

Simplifying assumptions

- Quasi-linear.
- No externalities.
- Normalization $v_i(\emptyset) = 0$.
- Free disposal $S \subseteq S^*$ implies $v_i(S) \leq v_i(S^*)$.

Combinatorial Auctions
Major Challenges

- Eliciting preferences is of exponential size, namely 2^m bids per bidder. Motivating iterative auctions (English ascending auction) and indirect mechanisms.

- Maximizing the surplus is really a Weighted Set Packing Problem thus NP-complete (Single-minded case reduction to come).

Combinatorial Auctions
VCG? No thanks!

- An attempt to use an allocation that approximates the maximum surplus and couple it with VCG-like payments leads to non-truthful mechanisms.

- VCG's revenue non-monotonicity invites collusion.

Example:

$\mathrm{N} = \{1,2\}$
$M = \{A, B\}$
$v_1(A, B) = 1$
$v_2(A, B) = v_2(A) = 1$

VCG revenue = 1

What if bidder 3 with

$v_3(A, B) = v_3(B) = 1$

joins the auction?

Linear Programming Relaxation
(The winner determination problem)

LPR

Max $\sum_{i \in N, S \subseteq M} x_{i,S} v_i(S_i)$

s.t. $\sum_{i \in N,\, S | j \in S} \ x_{i,S} \leq 1$ for all $j \in M$

$\sum_{S \subseteq M} \ x_{i,S} \leq 1$ for all i $\in N$

$x_{i,S} \geq 0$ for all i $\in N, S \subseteq M$

Linear Programming Relaxation (Dual)

DLPR

Min $\sum_{i \in N} u_i + \sum_{j \in M} p_j$

s.t. $u_i + \sum_{j \in S} p_j \geq v_i(S)$ for all $i \in N,\ S \subseteq M$

$u_i \geq 0, \quad p_j \geq 0$ for all $i \in N,\ j \in M$

Walrasian Equilibrium

Definition :

A set of nonnegative prices $p_1, \ldots, p_m$ and an allocation $S_1, \ldots, S_n$ of the items is **Walrasian Equilibrium** if for every player i, S_i is a demand of bidder i at prices $p_1, \ldots, p_m$ and for any item j that is not allocated we have $p_j = 0$.

Fact :

The existence of **Walrasian Equilibrium** is a sufficient and necessary condition for having an integer optimal solution for the linear programming relaxation.

Upshot:

In environments where **WE** exists efficient allocation can be computed in polynomial time.

Valuation functions

Complementarities : $v(A + B) > v(A) + v(B)$

Subadditive : $v(A + B) \leq v(A) + v(B)$

Submodular

Gross Substitutes

Single-minded (k-minded)

Unit Demand

Combinatorial Auctions
(The single-minded case)

The **allocation problem** among **single-minded bidders**:

Input: (S_i, v_i) for each bidder i.

Output: A subset of winning bids $W = \{1, \dots, n\}$ with maximum social welfare $\sum_{i \in W} v_i$ such that the winners are compatible with each other.

Reduction from **Independent Set** provides NP-completeness!

In fact approximating the **max IS** within a factor of $n^{1-\varepsilon}$ is also NP-complete.
It follow that approximating our problem within a factor better than $\sqrt{m}$ is NP-hard.

Greedy mechanism with $\sqrt{m}$ approximation (Single-minded bidders)

Greedy mechanism

Initialization:

- Reorder the bids such that $v_1/\sqrt{|S_1|} \geq v_2/\sqrt{|S_2|} \geq \ldots \geq v_n/\sqrt{|S_n|}$
- $W \leftarrow \emptyset$

For $\boldsymbol{i = 1 \ldots n}$ if $S_i \cap (\cup_{j \in W} \mathrm{S_j}) = \emptyset$ $\quad W \leftarrow W \cup \{i\}$

Output: Allocation: The set of winners W.

Payments: For each $i \in W$, $p_i = v_i/\sqrt{|S_j|/|S_i|}$
where j is the first player that would had win if player i left the auction.

Query types

- **Value query** : The auctioneer presents a bundle S, the bidder reports his value $v(S)$ for this bundle.

- **Demand query** : The auctioneer presents a vector of item prices $p_1, \ldots, p_m$ and the bidder reports a demand bundle under these prices i.e., some set S that maximizes $v(S) - \sum_{i \in S} p_i$.

Allowing **Value queries** and assuming **submodular valuations** we will present a polynomial, $2-$approximation algorithm for combinatorial auctions.

Non truthful 2-approximation (Submodular functions)

Input: $v_1, v_2, \dots, v_n$ submodular valuations, given as black boxes.

Output: An allocation $S_1, \dots, S_n$ which is 2-approximation to the optimal.

Algorithm

- Set $S_1 = S_2 = \cdots = S_n = 0$
- For $x = 1 \dots m$ do

  Allocate x to bidder j i.e. $S_j = S_j \cup \{x\}$,

  where j is the bidder with the highest marginal value $v_j(x|S_j)$.


Thank You!