CPSC 455/555
Combinatorial Auctions, Continued…

Shaili Jain

September 29, 2011

Combinatorial Auction Model

- Set M of m indivisible items that are concurrently auctioned among a set N of n bidders
- Bidders have preferences on bundles of items
- Bidder i has valuation v_i
  - – Monotone: for S µ T, we have v(S) · v(T)
  - – v(;) = 0
- Allocation among the bidders: S_1, …, S_n
- Want to maximize social welfare: $\sum_i v_i(S_i)$

Iterative Auctions: The Query Model

- Consider indirect ways of sending information about the valuation
- Auction protocol repeatedly interacts with different bidders, adaptively elicits enough information about bidder's preferences
- Adaptivity may allow pinpointing; may not require full disclosure
- Can reduce complexity, preserve privacy, etc.

Iterative Auctions: The Query Model

- Think of bidders as oracles and auctioneer repeatedly queries the oracles
- Want computational efficiency, both in number of queries and in internal computations
- Efficiency means polynomial running time in m and n

Types of Queries

- Value Query:
  - – Auctioneer presents a bundle S
  - – The bidder reports his value $v(S)$ for this bundle
- Demand Query (with item prices):
  - – Auctioneer gives a vector of item prices: $p_1, \ldots, p_m$
  - – The bidder reports a demand bundle under these prices, i.e. a set S that maximizes $v(S) - \sum_{i2S} p_i$

Value vs. Demand Queries

- Lemma: A value query may be simulated by mt demand queries, where t is the number of bits of precision in the representation of a bundle’s value.

- Marginal value query:
  - Auctioneer presents bundle S and j 2 M – S
  - Bidder gives v(j|S) = v(S [{j}) – v(S)

Value vs. Demand Queries

- How to simulate a marginal value query using a demand query?
- For all i 2 S, set $p_i = 0$
- For all i 2 M – S – {j}, set $p_i = 1$
- Run binary search on p_j
- Need up to m marginal value queries to simulate a value query

Value vs. Demand Queries

- Lemma: An exponential number of value queries may be required for simulating a single demand query.

- Part of your homework…
- Consider two agents
- Use the fact that there are exponentially many sets of size m/2

An IP Formulation

Let $x_{i,S}$ = 1 if agent i gets S, $x_{i,S}$ = 0 otherwise

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

$$\text{s.t.} \ \sum_{i \in N, S | j \in S} x_{i,S} \leq 1 \quad \forall j \in M$$

$$\sum_{S \subseteq M} x_{i,S} \leq 1 \quad \forall i \in N$$

$$x_{i,S} \in \{0,1\} \quad \forall i \in N, S \subseteq M$$

LP Relaxation

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

$$\text{s.t. } \textstyle\sum_{i\in N, S | j\in S} x_{i,S} \leq 1 \quad \forall j \in M$$

$$\textstyle\sum_{S\subseteq M} x_{i,S} \leq 1 \quad \forall i \in N$$

$$x_{i,S} \geq 0 \quad \forall i \in N, S \subseteq M$$

The Dual

$$\min \sum_{i2N} u_i + \sum_{j2M} p_j$$

s.t. $u_i + \sum_{j2S} p_j$ ¸ $v_i(S)$ 8 i 2 N, S µ M

u_i ¸ 0, p_j ¸ 0 8 i 2 N, j 2 M

Using demand queries…

- Use demand queries to solve the linear programming relaxation efficiently
- Solve the dual using the Ellipsoid method
- Dual is polynomial in number of variables, exponential in the number of constraints
- Ellipsoid algorithm is polynomial provided that a “separation oracle” is given
- Show how to implement the separation oracle via a single demand query to each agent

Using demand queries…

- Theorem: LPR can be solved in polynomial time (in n, m, and the number of bits of precision t) using only demand queries with item prices

Proof

- “separation oracle” either confirms possible solution is feasible or returns constraint that is violated
- Consider a possible solution to the dual, e.g. set of u_i and p_j
- Rewrite the constraints as u_i ¸ $v_i(S) - \sum_{j2S} p_j$
- A demand query to bidder i with prices p_j reveals the set S that maximizes the RHS

Proof Continued

- Query each bidder i for his demand D_i under prices p_j
- Check only n constraints: $u_i + \sum_{j2D_i} p_j$ ¸ $v_i(D_i)$

Proof Continued

- Now need to show how the primal is solved
- In solving the dual, we encountered a polynomial number of constraints
- Can remove all other constraints
- Now take the dual of the "reduced dual"
- Has a polynomial number of variables, has the same solution as the original primal

Walrasian Equilibrium

- Given a set of prices, the demand of each bidder is the bundle that maximizes her utility

- More formally…

- For given v_i and $p_1, …, p_m$, a bundle T is called a demand of bidder i if for every other S µ M, we have: $v_i(S) - \sum_{j2S} p_j \cdot v_i(T) - \sum_{j2T} p_j$

Walrasian Equilibrium

- Set of “market-clearing” prices where every bidder receives a bundle in his demand set
- Unallocated items have price of 0
- More formally…
- A set p^*_1, …, p^*_m and an allocation S^*_1, …, S^*_m is a Walrasian equilibrium if for every i, S^*_i is a demand of bidder i at prices p^*_1, …, p^*_m and for any item j not allocated, we have $p^*_j = 0$

An Example

- 2 players, Alice and Bob
- 2 items, {a, b}
- Alice has value 2 for every nonempty set of items
- Bob has value 3 for the whole bundle {a,b} and 0 for any of the singletons

- What is the optimal allocation?

An Example

- Optimal allocation: Both items to Bob
- In a Walrasian equilibrium, Alice must demand the empty set
- Therefore, the price of each item must be at least 2
- The price of whole bundle must be at least 4
- Bob will not demand this bundle

Walrasian Equilibrium

- Walrasian equilibrium, if they exist, are economically efficient
- "First Welfare Theorem"
- Welfare in a Walrasian equilibrium is maximal even if the items are divisible
- If a Walrasian equilibrium exists, then the optimal solution to the linear program relaxation will be integral

Walrasian Equilibrium

- The existence of an integral optimum to the linear programming relaxation is a sufficient condition for the existence of a Walrasian equilibrium

- “Second Welfare Theorem”

References

- This material was from section 11.3 and 11.5 in the AGT book
- For a good reference on LP-duality, look at "Approximation Algorithms" by Vijay Vazirani
- Questions? shaili.jain@yale.edu