Multi-item Auctions

Kate Larson

Cheriton School of Computer Science
University of Waterloo

Outline

Auctions

Sometimes we want to sell/allocate multiple items. How do we do this?

- Sequential Auctions
- Simultaneous Auctions
- Re-auctioning schemes
- Combinatorial Auctions

Sequential Auctions

- Relations between goods?
- Exposure problem
- Strategic bidding
- Inefficiencies

Simultaneous Auctions

- Inefficiencies
- Not clear what the equilibria look like

Combinatorial Auctions

- Sell all items at once (in a single round)
- Agents place bids on *bundles* of items
  - $1.00 for coffee and a donut
  - $1.00 for coffee and a donut OR $1.50 for a tea and cake, but I do not want both

Bidders' Perspective: "What can I bid on?"

An agent will have a valuation function over bundles

$$v_i : 2^X \rightarrow \mathbb{R}$$

where X is the set of all items being auctioned.
There are certain properties of valuation functions:

- No externalities: the valuation function depends only on the set of goods the agent is allocated
- Free disposal: If $S \subseteq T$ then $v_i(S) \leq v_i(T)$
- Nothing-for-nothing: $v_i(\emptyset) = 0$

Properties of Valuation Functions

Other properties describe how a valuation of one good is affected by the presence or absence of another.

- Complementarities: if $v(S \cup T) \geq v(S) + v(T)$ then S, T are complementary.
- Substitutes: $S, T \subseteq X$, $S \cap T = \emptyset$, $v(S \cup T) < v(S) + v(T)$ then S, T are substitutes

Bidding Languages

The bidder must communicate its valuation information to the auctioneer. The *bidding language* used determines what information can be communicated.

- Naive approach:

Properties

- Conciseness
- Expressiveness
- "Natural"
- Tractable for the auctioneer to process

Possible Languages

Atomic bids:

- (S, p) where S is the subset and p is the maximum price the bidder is willing to pay for S
  - Assume there is an implicit AND between items in S
  - If $S = \{a, b, c\}$ $(S, \$10)$ is interpreted as "I will pay up to \$10 if I get a AND b AND c."

Strengths? Weaknesses?

Possible Languages

OR bids:

- $(S_1, p_1) \vee (S_2, p_2) \vee \ldots$
- $v(S) = \max \sum_{i \in W} p_i$ where W is a collection of items $S_i, \cap S_j = \emptyset$

Strengths? Weaknesses?

Possible Languages

XOR bids:

- (S_1, p_1)XOR(S_2, p_2)XOR...
- $v(S) = \max_{i|S_i \subseteq S} p_i$ that is the bidder is willing to accept at most one, but not more than one of the bundles.

Strengths? Weaknesses?

Theorem

XOR is fully expressive.

Possible Languages

XOR bids:

- (S_1, p_1)XOR(S_2, p_2)XOR...
- $v(S) = \max_{i|S_i \subseteq S} p_i$ that is the bidder is willing to accept at most one, but not more than one of the bundles.

Strengths? Weaknesses?

Theorem

XOR is fully expressive.

Conciseness

We measure the size of a bid in terms of atomic bids in it. Consider additive valuation $v(S) = |S|$. Assume we have m items in total.

- Atomic: (not possible)
- OR in size m
- XOR in size 2^m

Combinations of Operations

It is possible to use combinations of operations

- OR-of-XOR: submit an arbitrary number of XOR bids
- XOR-of-OR: submit an arbitrary number of OR bids and get at most one

Sometimes these languages are concise and sometimes not. XOR-of-OR tends to be less "natural".

Combinations of Operations

It is possible to use combinations of operations

- OR-of-XOR: submit an arbitrary number of XOR bids
- XOR-of-OR: submit an arbitrary number of OR bids and get at most one

Sometimes these languages are concise and sometimes not. XOR-of-OR tends to be less "natural".

OR*

We can simulate the XOR language by using OR*.

- Main idea: introduce "dummy" bids
- Example: ({coffee, donut, Dummy}, \$1.00) OR ({ tea, cake, Dummy}, \$1.50) forces exclusivity.

Anything we can express in any other language, can be expressed in OR*

- OR-of-XOR in size s then OR* in size s with s dummy bids
- XOR-of-OR in size s then OR* in size s with s^2 dummy bids

OR*

We can simulate the XOR language by using OR*.

- Main idea: introduce "dummy" bids
- Example: ({coffee, donut, Dummy}, \$1.00) OR ({ tea, cake, Dummy}, \$1.50) forces exclusivity.

Anything we can express in any other language, can be expressed in OR*

- OR-of-XOR in size s then OR* in size s with s dummy bids
- XOR-of-OR in size s then OR* in size s with s^2 dummy bids

Auctioneer's Perspective

What sort of auction mechanism should be used?

Generalized Vickrey Auction (GVA) (a VCG mechanism):

Find allocation S^* that is feasible and maximizes the sum of total bids

$$S^* = \arg\max_S \sum_i v_i(S)$$

Payments

$$p_i = \sum_{j \neq i} v_j(S') - \sum_{j \neq i} v_j(S^*)$$

where S' is the optimal allocation if agent i did not exist.

Auctioneer's Perspective

What sort of auction mechanism should be used?
Generalized Vickrey Auction (GVA) (a VCG mechanism):
Find allocation S^* that is feasible and maximizes the sum of total bids

$$S^* = \arg\max_S \sum_i v_i(S)$$

Payments

$$p_i = \sum_{j \neq i} v_j(S') - \sum_{j \neq i} v_j(S^*)$$

where S' is the optimal allocation if agent i did not exist.

Example

2 items, x, y and three bidders

- $v_1(\{x, y\}) = 100$
- $v_2(\{x\}) = 75$
- $v_3(\{y\}) = 40$
- all other valuations are 0

Let's look at the auctioneer's problem in more detail

$$\max \sum_{i \in N} \sum_{S \subseteq X} v_i(S) a_{S,i}$$

such that

$$\sum_{S \subseteq X, j \in S} \sum_{i \in N} a_{S,i} \leq 1 \ \forall j \in X$$

$$\sum_{S \subseteq X} a_{S,i} \leq 1 \ \forall i \in N$$

$$a_{S,i} \in \{0, 1\} \ \forall S \subseteq X, i \in N$$

What can be done?

1. Find special cases for which there are poly-time solutions
2. Heuristics and approximations
3. "Brute force"

Restricting Classes of Preferences

- $|S| \leq 2$
- Contiguous bids (geographical interpretations)
- Tree-based bids: every 2 sets are either disjoint or one is a subset of another

Using heuristics and approximations

Idea: Just replace the winner-determination algorithm in the GVA with an approximation

Result: Any reasonable VCG-based mechanism for combinatorial auctions is not truthful, unless it uses a computationally intractable optimal allocation algorithm. [Nisan and Ronen, 2000]

"Reasonable": If an item is valued by only a single agent, then that agent gets the item.

Using heuristics and approximations

Idea: Just replace the winner-determination algorithm in the GVA with an approximation

Result: Any reasonable VCG-based mechanism for combinatorial auctions is not truthful, unless it uses a computationally intractable optimal allocation algorithm. [Nisan and Ronen, 2000]

"Reasonable": If an item is valued by only a single agent, then that agent gets the item.

Using heuristics and approximations

Idea: Just replace the winner-determination algorithm in the GVA with an approximation

Result: Any reasonable VCG-based mechanism for combinatorial auctions is not truthful, unless it uses a computationally intractable optimal allocation algorithm. [Nisan and Ronen, 2000]

"Reasonable": If an item is valued by only a single agent, then that agent gets the item.

We need to resort to non-VCG mechanisms

Example: A randomized mechanism for single-minded bidders

- Let K_j be the number of units available for item j. Define $K'_j = \lfloor (1-\epsilon) \rfloor$. Solve

$$\max \sum_i v_i x_i$$

such that

$$\sum_{i|j \in S_i} x_i \leq K'_j \ \forall j$$

$$0 \leq x_i \leq 1$$

This finds the optimal fractional allocation under the constraint that at most K'_j are sold of each item.

Randomized Example Continued

- Round variable x_j to 1 with probability x_j, 0 otherwise
- Select all agents i with $x_i = 1$ and for whom constraints for all items in their bundle S_i is satisfied
- Drop each agent with some additional probability θ_i

This is poly-time (linear program) and is truthful in expectation, as well as truthful with high probability. The approximation is $1 + O(\epsilon)$.

Randomized Example Continued

- Round variable x_j to 1 with probability x_j, 0 otherwise
- Select all agents i with $x_i = 1$ and for whom constraints for all items in their bundle S_i is satisfied
- Drop each agent with some additional probability θ_i

This is poly-time (linear program) and is truthful in expectation, as well as truthful with high probability. The approximation is $1 + O(\epsilon)$.

Brute Force

- Formulate the problem as an AI-style search (IDA*)
  - Works very well
  - Heuristics for
    - node ordering
    - sophisticated pre-processing
    - careful definition of the search space

Iterative Combinatorial Auctions

Iterative (indirect) approaches can simplify

- Winner determination problem
- Preference elicitation problem for bidders

An example of an iterative combinatorial auction (iBundle)

Init: All bundles have price 0

Bid: Agent indicate which bundles they want (they must bid higher than current price to be considered)

Winner determination: Optimal, but the idea is that you are working with a smaller set of bids

An example of an iterative combinatorial auction (iBundle)

Init: All bundles have price 0

Bid: Agent indicate which bundles they want (they must bid higher than current price to be considered)

Winner determination: Optimal, but the idea is that you are working with a smaller set of bids

An example of an iterative combinatorial auction (iBundle)

Init: All bundles have price 0

Bid: Agent indicate which bundles they want (they must bid higher than current price to be considered)

Winner determination: Optimal, but the idea is that you are working with a smaller set of bids

iBundle continued

Price update: Define $p_i(S)$ to be the price for S by i, and let "unhappy agents" be the agents who have not been allocated a bundle.

$$p^{t+1}(S) = \max\left[p^t(S), \max_{i \in \text{unhappy}} p_i(S) + \epsilon\right]$$

Termination: End the auction when all agents are happy, or all submit the same bids in the sequential rounds.

Agents do not have dominant strategies, but instead the myopic best response is to bid truthfully.

iBundle continued

Price update: Define $p_i(S)$ to be the price for S by i, and let "unhappy agents" be the agents who have not been allocated a bundle.

$$p^{t+1}(S) = \max\left[p^t(S), \max_{i \in \text{unhappy}} p_i(S) + \epsilon\right]$$

Termination: End the auction when all agents are happy, or all submit the same bids in the sequential rounds.

Agents do not have dominant strategies, but instead the myopic best response is to bid truthfully.

iBundle continued

Price update: Define $p_i(S)$ to be the price for S by i, and let "unhappy agents" be the agents who have not been allocated a bundle.

$$p^{t+1}(S) = \max\left[p^t(S), \max_{i \in \text{unhappy}} p_i(S) + \epsilon\right]$$

Termination: End the auction when all agents are happy, or all submit the same bids in the sequential rounds.

Agents do not have dominant strategies, but instead the myopic best response is to bid truthfully.

Conclusion

- Auctioning multiple items is much more complex that a single item
- Computational issues really come to the forefront