Combinatorial Auctions

Noam Nisan

Hebrew University, Jerusalem

Talk Structure

- Introduction
- The IP formulation and Walrasian Equilibrium
- Bidding Languages and Winner Determination
- Incentive Compatibility and Computational Hardness
- Iterative Auctions and Communication Complexity

Introduction

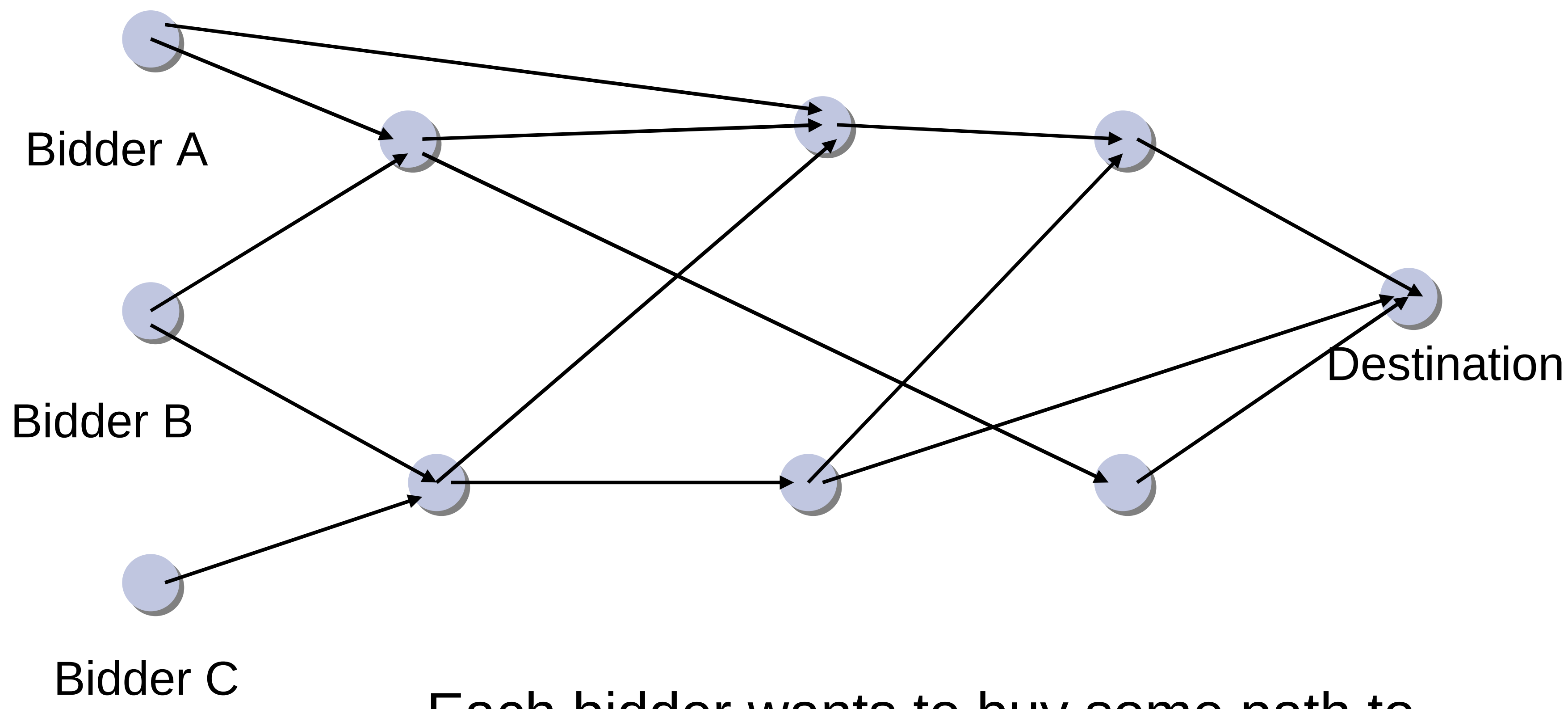
Combinatorial Auctions

- N indivisible non-identical items for sale
- m bidders compete for subsets of these items
- Each bidder i has a valuation for each set of items: $v_i(S)$ = value that i assigns to acquiring the set S
  - v_i is non-decreasing ("free disposal")
  - $v_i(\emptyset) = 0$
- **Objective:** Find a partition $(S_1 \ldots S_m)$ of $\{1..N\}$ that maximizes the social welfare: $\sum_i v_i(S_i)$
- **Issues:** communication, allocation, strategies

Complements and Substitutes

- $v_j()$ may have *complements*: $v_j(S \cup T) > v_j(S)+v_j(T)$ for some S and T.
  - Extreme case: "single-minded bid" -- will only pay for a complete package -- pay p for the set S but pay nothing for anything else
- $v_j()$ may have *substitutes*: $v_j(S \cup T) < v_j(S)+v_j(T)$ for some disjoint S and T.
  - Extreme case: "unit demand bid" -- will pay for at most a single item – the price may depend on the item

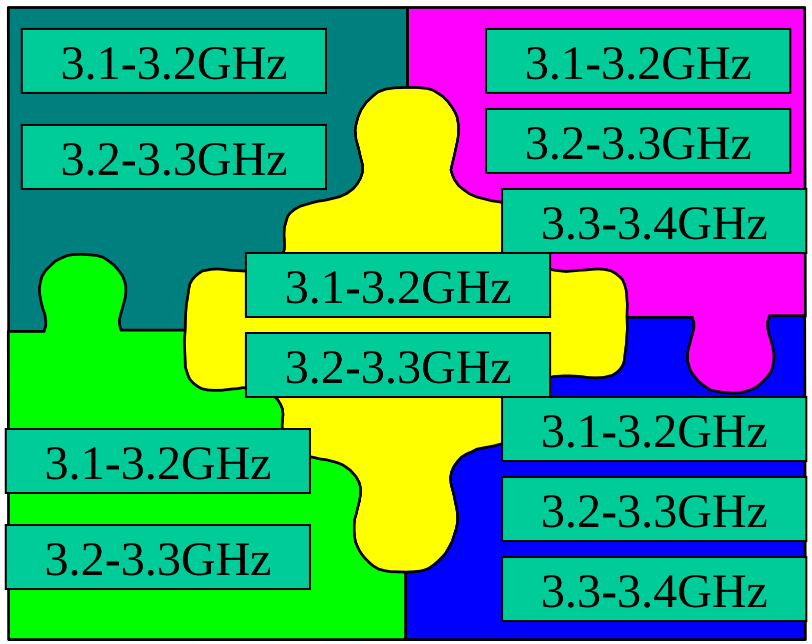
Routing as a Combinatorial Auction



- Each bidder wants to buy some path to the destination
- Each link is an item

The FCC Spectrum Auctions

- The FCC auctions spectrum licenses for many geographic regions and various frequency bands
- These auctions have raised billions of dollars
- The value of a license to a bidder depends on the other licenses it holds
- Currently licenses are sold in a simultaneous auction
- USA Congress mandated that the next spectrum auction be made combinatorial.



The IP formulation and Walrasian Equibilria

An Integer Programming Formulation

Maximize:

$$\sum_{i,S} x_{i,S} v_i(S)$$

Subject to:

- For each item *j*:

$$\sum_{i, j \in S} x_{i,S} \le 1$$

- For each bidder *i*:

$$\sum_{S} x_{i,S} \le 1$$

- For each *i, S*:

$$x_{i,S} \in \{0,1\}$$

Linear Programming Relaxation and its Dual

The Primal LP

Maximize:

$$\sum_{i,S} x_{i,S} v_i(S)$$

Subject to:

- For each item *j*:

$$\sum_{i,\, j \in S} x_{i,S} \le 1$$

- For each bidder *i*:

$$\sum_{S} x_{i,S} \le 1$$

- For each *i, S*:

$$x_{i,S} \ge 0$$

The Dual LP

Minimize:

$$\sum_{j} p_j + \sum_{i} u_i$$

Subject to:

- For each *i, S*:

$$u_i + \sum_{j \in S} p_j \ge v_i(S)$$

- For each *i, j*: $p_j \ge 0, \qquad u_i \ge 0$

Walrasian Equilibrium

Definition: The demand $D=D_i(p_1 \ldots p_N)$ of player i at item prices $p_1 \ldots p_N$ is the bundle of items that maximizes

$$v_i(D) - \sum_{j \in D} p_j$$

Definition: A vector of prices $p_1 \ldots p_N$ and an allocation $D_1 \ldots D_m$ is called a Walrasian equilibrium if for every bidder i, $D_i=D_i(p_1 \ldots p_N)$.

First Welfare Theorem: For any Walrasian equilibrium, $D_1 \ldots D_m$ is an optimal allocation.

Walrasian Equilibrium and the LP

First Welfare Theorem: For any Walrasian equilibrium, $D_1 \ldots D_m$ is an optimal allocation.

Proof: For any other allocation $S_1 \ldots S_m$:

$$\sum_i (v_i(S_i) - \sum_{j \in S_i} p_j) \leq \sum_i (v_i(D_i) - \sum_{j \in D_i} p_j)$$

Note: Optimality is also over fractional allocations.

Corollary: the LP has integral solutions.

Theorem: The LP has integral solutions iff a Walrasian equilibrium exists.

Proof (➔): Dual prices + complementary slackness

Bidding and Winner Determination

Bidding Languages

Question: How do we represent v_i?

- An exponential length vector is not practical

A bidding Language is a syntactic representation of valuations.

- Should be expressive
- Should be simple

Basic Bidding Languages

Bundle bid: {a,b} : 5

- Pay 5 for any bundle that contains {a,b} (and 0 otherwise)

XOR bid: {a,b}:5 XOR {c:d}:6

- Accept any of these bundles, but not both (*v(abcd)=6*)

OR bid: {a,b}:5 OR {c,d}:6

- Accept any of them or both (*v(abcd)=11*)
- Notice that for *v=({a,b}:5 OR {a,c}:6),* we have *v({abc})=6*

OR/XOR formula: ({a,b}:5 OR {c,d}:6) XOR {e}:5

- Recursive definition of (v OR u) and of (v XOR u)

Dummy Items

OR* language: OR language, but allow bidders to invent worthless dummy items

Idea: Express XOR using OR.

- ({a,b}:5 XOR {c,d}:6) ➔ ({a,b,z}:5 OR {c,d,z}:6)

Theorem: Any OR/XOR formula of size *s* may be converted to an OR* formula of size *s* with at most s^2 dummy items.

Proof:

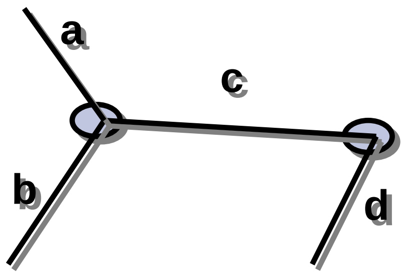
- Recursively translate sub-formulae into OR* form
- Use idea above (add dummy items) whenever connective is XOR
- At the end leave only essential dummy items

Corollary: Any winner determination algorithm that works for single bundle bids works also for OR/XOR formulae.

Computational Hardness

Theorem: it is NPC to find optimal solution in a CA with even single bundle bids for each player.

Proof: reduction form "independent set":



➔ {a,b,c}:1 ; {c,d}:1

Corollary: Approximation to within a factor of $N^{1/2-\varepsilon}$ is NPC

In Practice: Use IP solvers or special purpose software. Experimentally, CAs with 10s-100s of items can be solved exactly and 100s-1000s of items approximately.

Known Solvable Special Cases

- Unit-demand bids and their generalization "substitutes" bids give integral LP solutions.
- Hierarchal bids and their generalization "linear-order" bids give integral LP solutions.

- CAs with Sub-modular bids can be approximated to within a factor of 2
- CAs with Complement-free bids can be approximated to within a logarithmic factor
- CAs with *k*-duplicates of each item can be approximated to within a $\sim N^{1/k}$ factor

Incentive Compatibility

Incentive Compatibility

Definition: A mechanism is incentive compatible if truth is a dominating strategy for all players.

For combinatorial auctions it means that for any v_i, $v`_i$, v_{-i}:

$$v_i(S_i) - p_i \geq v_i(S`_i) - p`_i$$

- S_i and p_i are *i*'s allocation and payment on input (v_i, v_{-i})
- $S`_i$ and $p`_i$ are *i*'s allocation and payment on input $(v`_i, v_{-i})$

The Problem

- This is an Auction – players are selfish!
- How do we convince the players to reveal their valuations?
- Standard trick: use VCG prices
  - Charge each player the difference between the social welfare of the others and what it would have been without him
- Problem: We can not compute optimal allocations
  - Using the “VCG” calculation on the best algorithm we have will not be incentive compatible
  - THM (Nisan&Ronen): ever.

- **Open Problem**: Find any incentive compatible allocation algorithm that is “somewhat reasonable”.

The LOS Mechanism

Major Assumption: each bidder is single minded: (v_i, S_i)

Mechanism:

- Order the bids by decreasing value of $v_i / \sqrt{|S_i|}$
- Greedily allocate sets that do not contain previously allocated items
- For each winner *i* find the first loser *j* that lost just because of *i* and charge the price $v_j \sqrt{|S_i|} / \sqrt{|S_j|}$

Theorem: This Mechanism

1. Is incentive Compatible
2. Gives a $O(\sqrt{N})$ approximation

Proof of incentive compatibility

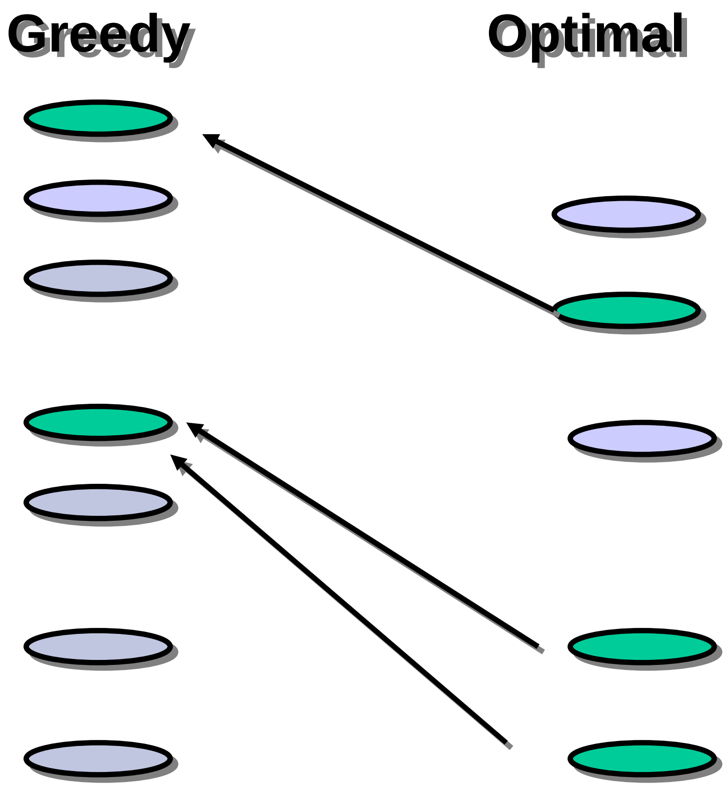
Lemma: A mechanism for single-minded bidders is IC iff:

1. It is monotone. I.e. *(v,S)* wins ➔ *(u,T)* wins for every $u \geq v$, $T \subseteq S$
2. The price is the critical value (smallest v that will still win)

Proof (if):

- Lying about *v* will not help since anyway I pay the smallest *v* that will win. If the truth will make me loose, then I really don't want to win, since my payment will be higher than my real value.
- Lying about *S* will not help: if the lie does not contain *S*, I will get no value from the bundle won. If it does contain *S*, I can only pay more.

Proof of Approximation Factor



Facts:

- The bids (u,T) assigned to each Greedy winner (v,S) are disjoint → at most |S| bids and $\sum T_j \le N$
- For each bid (u,T) assigned to a greedy winner (v,S): $u \le v\sqrt{T}/\sqrt{S}$.

Corollary:

$$\sum_{\text{bids assigned to } (v,S)} u_j \le v\sqrt{N}$$

Proof:

$$\sum u_j \le (v/\sqrt{S}) \sum \sqrt{T_j} \le$$
$$\le (v/\sqrt{S}) \sqrt{S}\sqrt{N}$$

Iterative Auctions and Communication

A Natural Tatonnement Process

Demange&Gale&Sotomayor

Initialize item prices: $p_1=0, p_2=0, \ldots, p_N=0$

Repeat:

For each bidder i query $D_i(p_1 \ldots p_N)$

For each item j that is demanded by more than a single bidder do

$p_j \leftarrow p_j+\varepsilon$

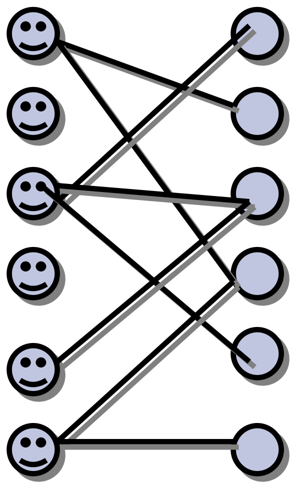
Until: all items are demanded by at most a single bidder

Theorem: If all valuations are "substitutes" then this finds an allocation that is εN close to optimal. Kelso&Crawford

Proof: An ε-Walrasian Equilibrium is reached

Definition: A valuation is "substitutes" if raising prices of some items can not decrease demand for the other items.

An "auction" algorithm for bipartite matching



Algorithm:

- Initialize all item prices $p_j=0$
- Repeat:
  - Some player i that does not hold any item j "takes" neighbor j with minimal p_j *(as long as $p_j < 1$)*
  - $p_j = p_j + \varepsilon$ (where $\varepsilon \cong 1/n$)

Runtime Analysis:

- (Assume n=|items|=|bidders|)
- At most $n*n$ iterations of the loop
- Each loop iteration can be implemented in $O(n)$ time
- ➔ A cubic algorithm for bipartite matching

The General Case

Theorem: In the worst case, an exponential (in *N*) number of queries to bidders are needed (even for 2 bidders, even with 0/1 valuations, and even without requiring any type of equilibrium) Nisan&Segal

Proof:

Definition: *Answers(A,v_1,v_2)* Is the sequence of answers for the queries that algorithm *A* asked on bids v_1, v_2

- For every valuation *v* of bidder 1, with $v(\{1 \ldots N\}) = 1$, we will define a valuation v^* for bidder 2, and show,

Main Lemma: If *A* always finds optimal allocations, then for every $u \neq v$, we have Answers(A,v,v*) $\neq$ Answers(A,u,u*).

Corollary: (number of answers) $\geq$ (number of valuations) ➔ (total length of answer sequence) $\geq$ (representation length of valuation) = exponential ➔ number of queries is exponential.

Proof of Main Lemma

Definition: $v^*(S)=1-v(S^c)$

Facts:

- For every v, S, we have $v(S)+v^*(S^c) = 1$
- For every $u(S)>v(S)$ we have $u(S)+v^*(S^c) > 1$
  $v(S)+u^*(S^c) < 1$

Cut-and-paste lemma: If *Answers(A,u,u*)=Answers(A,v,v*)* then also *Answers(A,u,v*)=Answers(A,v,u*)*

- Proof: the query answers will keep being the same in all 4 cases

Corollary: If $u \neq v$, then Answers(A,u,u*) $\neq$ Answers(A,v,v*)

- Proof: the optimal S for (u,v^*) is sub-optimal for (v,u^*)

Remarks on Lower Bound

- A general communication lower bound
- Applies also for “non-deterministic” communication
- Applies even if valuations of sets are 0,1
- When valuations are continues the message space must have exponential dimension
- May be viewed as a lower bound for number of “prices”

Open issues

- Bidding and Winner determination – in practice
- Interesting Special cases with good approximations
- How good are Iterative Auctions?
- Incentive Compatible auctions – possible?
- Generalizations to 2-sided CAs?