Approximation Algorithms for Combinatorial Auctions with Complement-Free Bidders

Speaker: Michael Schapira

Joint work with Shahar Dobzinski & Noam Nisan

Talk Structure

- ➔Combinatorial Auctions
- Log(m)-approximation for CF auctions
- An incentive compatible O($m^{1/2}$)-approximation of CF auctions using value queries.
- 2-approximation for XOS auctions
- A lower bound of e/(e-1)-ε for XOS auctions

Combinatorial Auctions

- *A set M of* items for sale. |M|=m.
- *n* bidders, each bidder *i* has a valuation function v_i:2^M->R^+.

  Common assumptions:
  - Normalization: $v_i(\emptyset)=0$
  - Free disposal: $S \subseteq T$ ➔ $v_i(T) \geq v_i(S)$
- **Goal**: find a partition $S_1,\ldots,S_n$ such that social welfare $\Sigma v_i(S_i)$ is maximized

Combinatorial Auctions

- **Problem 1**: finding an optimal allocation is NP-hard.
- **Problem 2**: valuation length is exponential in *m*.
- **Problem 3**: how can we be certain that the bidders do not lie ? (incentive compatibility)

Combinatorial Auctions

- We are interested in algorithms that based on the reported valuations $\{v_i\}_i$ output an allocation which is an approximation to the optimal social welfare.

- We require the algorithms to be polynomial in m and n. That is, **the algorithms must run in sub-linear (polylogarithmic) time**.

- We explore the achievable approximation factors.

Access Models

<u>How can we access the input</u> ?

- One possibility: bidding languages.
- The “black box” approach: each bidder is represented by an oracle which can answer certain queries.

Access Models

- Common types of queries:
  - **Value**: given a bundle *S*, return *v(S)*.
  - **Demand**: given a vector of prices ($p_1,\ldots, p_m$) return the bundle *S* that maximizes $v(S)-\Sigma_{j\in S}p_j$.
  - **General**: any possible type of query (the comunication model).
- Demand queries are strictly more powerful than value queries (Blumrosen-Nisan, Dobzinski-Schapira)

Known Results

- Finding an optimal solution requires exponential communication. Nisan-Segal
- Finding an $O(m^{1/2-\varepsilon})$-approximation requires exponential communication. Nisan-Segal.

  (this result holds for every possible type of oracle)
- Using demand oracles, a matching upper bound of $O(m^{1/2})$ exists (Blumrosen-Nisan).

- **Better results might be obtained by restricting the classes of valuations.**

The Hierarchy of CF Valuations

OXS ⊆ GS ⊆ SM ⊂ **XOS** ⊂ **CF**

Lehmann, Lehmann, Nisan

- **Complement-Free**: ***v(S∪T) ≤ v(S) + v(T).***

- **XOS**: XOR of ORs of singletons
    - Example: (A:2 **OR** B:2) **XOR** (A:3)

- **Submodular**: ***v(S∪T) + v(S ∩T) ≤ v(S) + v(T).***
    - 2-approximation by LLN.

- **GS**: (Gross) Substitutes, **OXS**: OR of XORs of singletons
    - Solvable in polynomial time (LP and Maximum Weighted Matching respectively)

Talk Structure

- Combinatorial Auctions
- ➔Log(m)-approximation for CF auctions
- An incentive compatible O($m^{1/2}$)-approximation CF auctions using value queries.
- 2-approximation for XOS auctions
- A lower bound of e/(e-1)-ε for XOS auctions

Intuition

- We will allow the auctioneer to allocate k duplicates from each item.
- Each bidder is still interested in at most one copy of each item (so valuations are kept the same).
- Using the assumption that all valuations are CF, we will find an approximation to the original auction, based on the k-duplicates allocation.

The Algorithm – Step 1

- Solve the linear relaxation of the problem:

  Maximize: $\Sigma_{i,S} x_{i,S} v_i(S)$

  Subject To:

  - For each item *j*: $\Sigma_{i,S|j\in S} x_{i,S} \leq 1$
  - For each bidder *i*: $\Sigma_S x_{i,S} \leq 1$
  - For each *i,S*: $x_{i,S} \geq 0$

- Despite the exponential number of variables, the LP relaxation may still be solved in polynomial time using demand oracles.(Nisan-Segal).
- ***OPT*=$\Sigma_{i,S} x_{i,S} v_i(S)$*** is an upper bound for the value of the optimal integral allocation.

The Algorithm – Step 2

- Use *randomized rounding* to build a “pre-allocation” $S_1,..,S_n$:
  - Each item *j* appears at most k=$O(\log(m))$ times in $\{S_i\}_i$.
  - **$\Sigma_i v_i(S_i) \geq OPT^*/2$.**

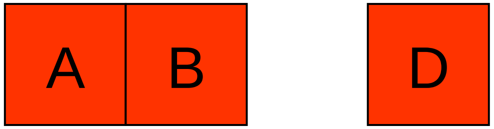
- Randomized Rounding: For each bidder *i*, let S_i be the bundle *S* with probability $x_{i,S}$, and the empty set with probability $1-\Sigma_S x_{i,S}$.
  - The expected value of $v_i(S_i)$ is $\Sigma_S x_{i,S} v_i(S)$
- We use the Chernoff bound to show that such “pre-allocation” is built with high probability.

The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \leq i < i' \leq n$, $1 \leq t \leq t' \leq k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.

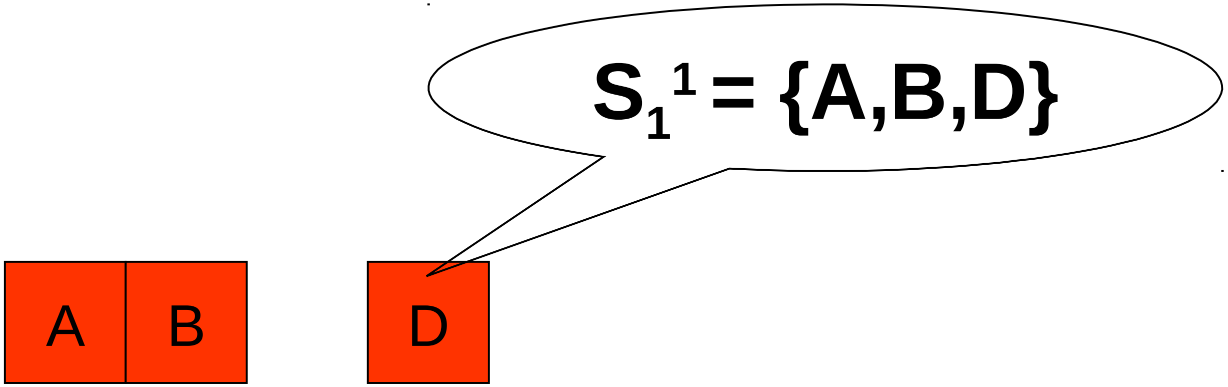
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \leq i < i' \leq n$, $1 \leq t \leq t' \leq k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



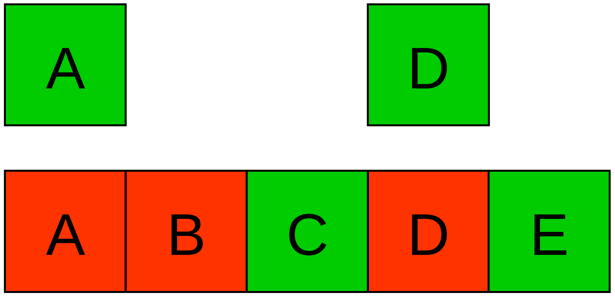
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \leq i < i' \leq n$, $1 \leq t \leq t' \leq k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



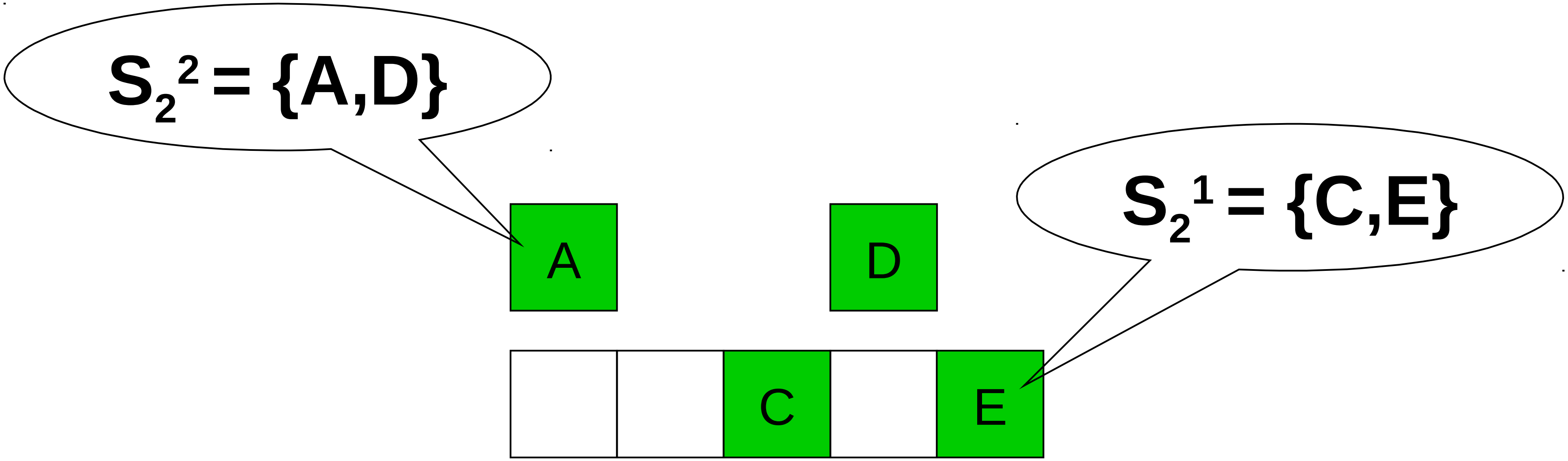
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \le i < i' \le n$, $1 \le t \le t' \le k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



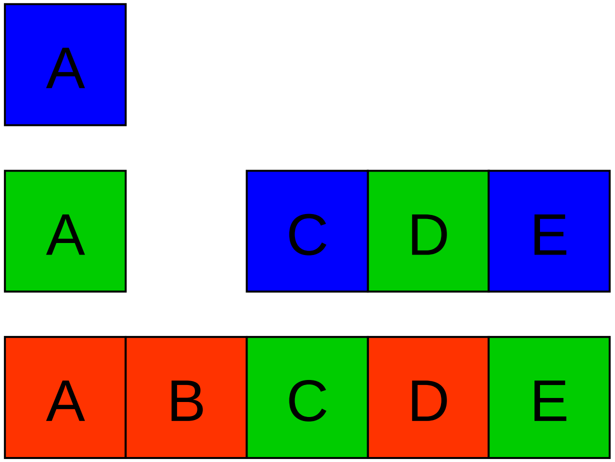
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \le i < i' \le n$, $1 \le t \le t' \le k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



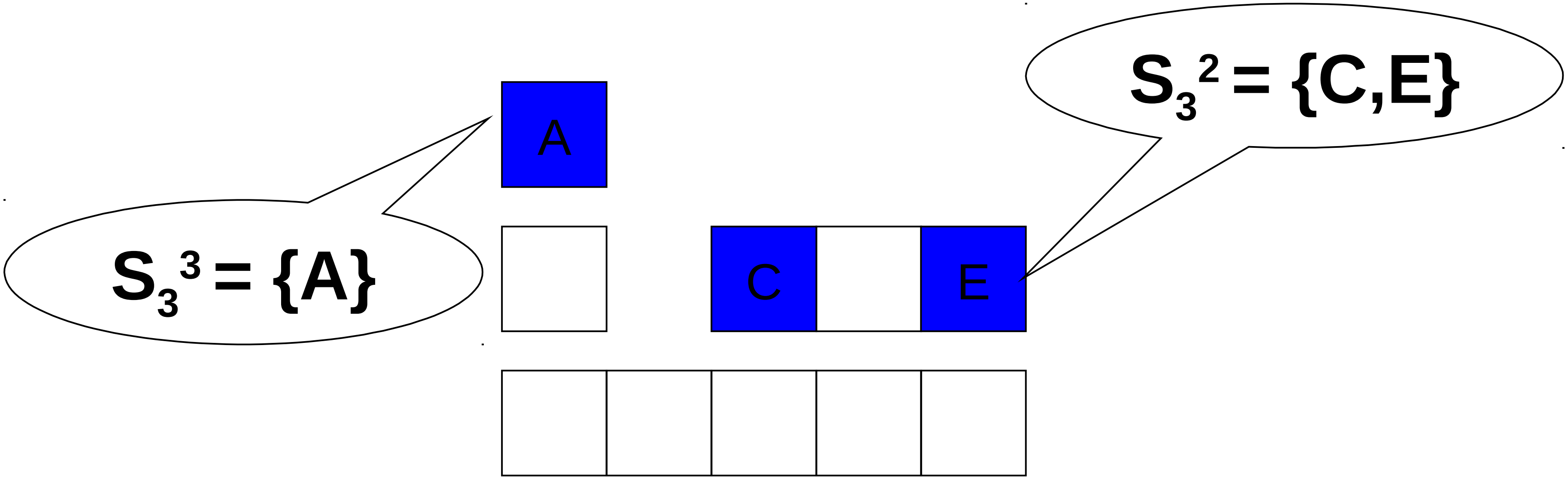
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \le i < i' \le n$, $1 \le t \le t' \le k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



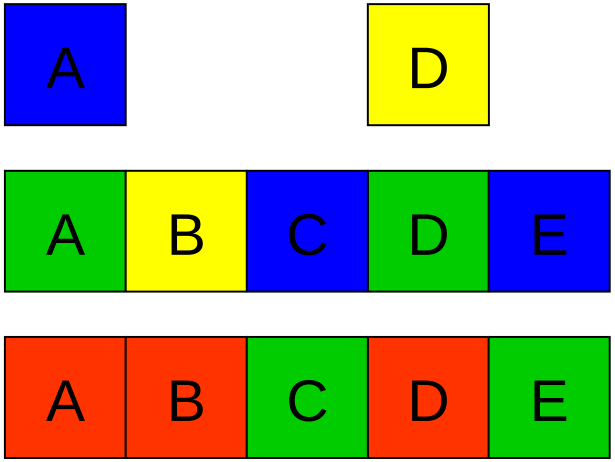
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \le i < i' \le n$, $1 \le t \le t' \le k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



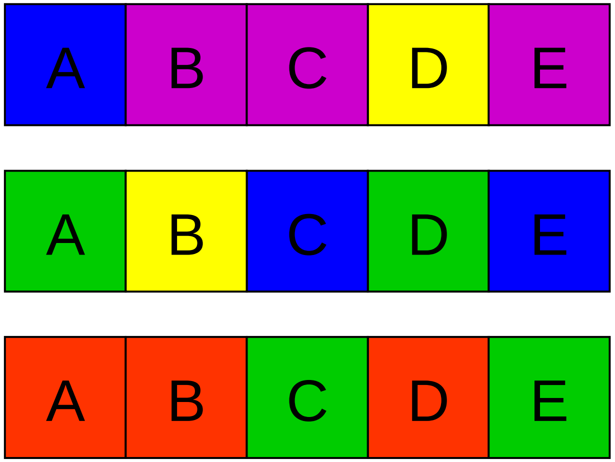
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \le i < i' \le n$, $1 \le t \le t' \le k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



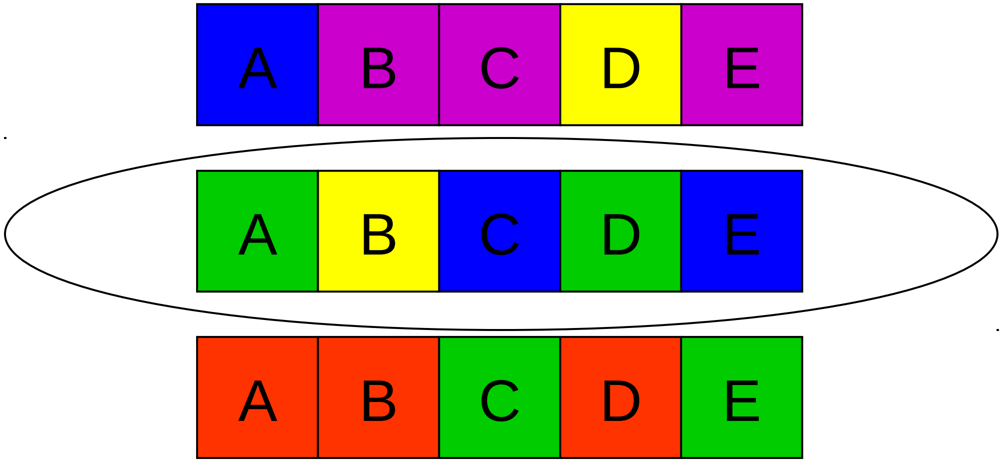
The Algorithm – Step 3

- For each bidder *i*, partition S_i into a disjoint union $S_i = S_i^1 \cup .. \cup S_i^k$ such that for each $1 \le i < i' \le n$, $1 \le t \le t' \le k$, $S_i^t \cap S_{i'}^{t'} = \emptyset$.



The Algorithm – Step 4

- Find the *t* maximizes $\Sigma_i v_i(S_i^t)$
- Return the allocation (S_1^t,...,S_n^t).



- All valuations are CF so:

□$\Sigma_t\Sigma_i v_i(S_i^t) = \Sigma_i\Sigma_t v_i(S_i^t) \geq \Sigma_i v_i(S_i) \geq OPT^*/2$

➔ For the *t* that maximizes $\Sigma_i v_i(S_i^t)$, it holds that:

$\boldsymbol{\Sigma_i v_i(S_i^t) \geq (\Sigma_i v_i(S_i))/k \geq OPT^*/2k = OPT^*/O(\log(m))}$.

A Communication Lower Bound of 2-ε for CF Valuations

Theorem: *Exponential communication is required for approximating the optimal allocation among CF bidders to any factor less than 2.*

Proof: *A simple reduction from the general case.*

Talk Structure

- Combinatorial Auctions
- Log(m)-approximation for CF auctions
- ➔An incentive compatible O($m^{1/2}$)-approximation of CF auctions using value queries.
- 2-approximation for XOS auctions
- A lower bound of e/(e-1)-ε for XOS auctions

Incentive Compatibility & VCG Prices

- We want an algorithm that is **truthful** (incentive compatible). I.e. we require that the dominant strategy of each of the bidders would be to reveal true information.

- **VCG** is the only general technique known for making auctions incentive compatible (if bidders are not single-minded):
    - Each bidder *i* pays: $\Sigma_{k\neq i} v_k(O^{-i}) - \Sigma_{k\neq i} v_k(O^{i})$

      O^{i} is the optimal allocation, O^{-i} the optimal allocation of the auction without the *i*'th bidder.

Incentive Compatibility & VCG Prices

- Problem: VCG requires an optimal allocation!

- Finding an optimal allocation requires exponential communication and is computationally intractable.

- Approximations do not suffice (Nisan-Ronen).

VCG on a Subset of the Range

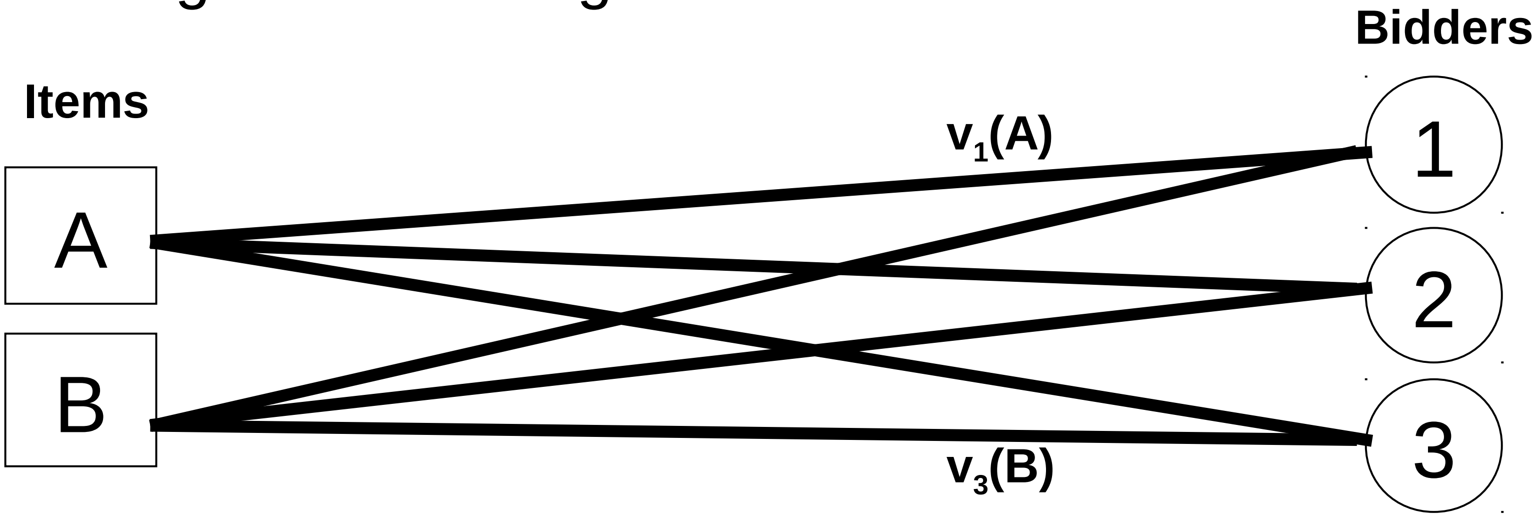
- Our solution: limit the set of possible allocations.
    - We will let each bidder to get at most one item, or we'll allocate all items to a single bidder.
- Optimal solution in the set can be found in polynomial time ➔ VCG prices can be computed ➔ incentive compatibility.
- We still need to prove that we achieve an approximation.

The Algorithm

- Ask each bidder *i* for $v_i(M)$, and for $v_i(j)$, for each item *j*.

  (We have used only value queries)

- Construct a bipartite graph and find the maximum weighted matching *P*.



  - can be done in polynomial time (Tarjan).

The Algorithm (Cont.)

- Let *i* be the bidder that maximizes $v_i(M)$.
- If $v_i(M)>|P|$
  - Allocate all items to *i*.
- else
  - Allocate according to *P*.
- Let each bidder pay his VCG price (in respect to the restricted set).

Proof of the Approximation Ratio

Theorem: If all valuations are CF, the algorithm provides an $O(m^{1/2})$-approximation.

Proof: Let $OPT=(T_1,..,T_k,Q_1,...,Q_l)$, where for each T_i, $|T_i|>m^{1/2}$, and for each Q_i, $|Q_i|\leq m^{1/2}$. $|OPT|= \Sigma_i v_i(T_i) + \Sigma_i v_i(Q_i)$

Case 1: $\Sigma_i v_i(T_i) > \Sigma_i v_i(Q_i)$

(“large” bundles contribute most of the social welfare)

➔ $\Sigma_i v_i(T_i) > |OPT|/2$

At most $m^{1/2}$ bidders get at least $m^{1/2}$ items in OPT.

➔ For the bidder *i* the bidder *i* that maximizes $v_i(M)$,

$\mathbf{v_i(M) > |OPT|/2m^{1/2}}$.

Case 2: $\Sigma_i v_i(Q_i) \geq \Sigma_i v_i(T_i)$

(“small” bundles contribute most of the social welfare)

➔ $\Sigma_i v_i(Q_i) \geq |OPT|/2$

For each bidder *i*, there is an item c_i, such that: $v_i(c_i) > v_i(Q_i) / m^{1/2}$.

(The CF property ensures that the sum of the values is larger than the value of the whole bundle)

$\{c_i\}_i$ is an allocation which assigns at most one item to each bidder:

$\mathbf{|P| \geq \Sigma_i v_i(c_i) \geq |OPT|/2m^{1/2}}$.

Talk Structure

- Combinatorial Auctions
- Log(m)-approximation for CF auctions
- An incentive compatible *$O(m^{1/2})$*-approximation CF auction
- ➔2-approximation for XOS auctions
- A lower bound of e/(e-1)-ε for XOS auctions

Definition of XOS

- XOS: XOR of ORs of Singletons.
- Singleton valuation (x:p)
  - $v(S) = \begin{cases} p & x \in S \\ 0 & \text{otherwise} \end{cases}$
- Example: (A:2 **OR** B:2) **XOR** (A:3)

XOS Properties

- The strongest bidding language syntactically restricted to represent only complement-free valuations.
- Can describe all submodular valuations (and also some non-submodular valuations)
- Can describe interesting NPC problems (Max-k-Cover, SAT).

Supporting Prices

Definition: $p_1,\ldots,p_m$ supports the bundle *S* in v if:

- $v(S) = \Sigma_{j\in S} p_j$
- $v(T) \geq \Sigma_{j\in T} p_j$ for all $T \subseteq S$

Claim: a valuation is XOS iff every bundle *S* has supporting prices.

Proof:

- ➔ There is a clause that maximizes the value of a bundle *S*. The prices in this clause are the supporting prices.
- 🡐 Take the prices of each bundle, and build a clause.

Algorithm-Example

Items: {A, B, C, D, E}. 3 bidders.

- Price vector: p_0=(0,0,0,0,0)

  v_1: (A:1 **OR** B:1 **OR** C:1) **XOR** (C:2)

  Bidder 1 gets his demand: {A,B,C}.

Algorithm-Example

Items: {A, B, C, D, E}. 3 bidders.

- Price vector: p_0=(0,0,0,0,0)
  v_1: (A:1 **OR** B:1 **OR** C:1) **XOR** (C:2)
  Bidder 1 gets his demand: {A,B,C}.

- Price vector: p_1=(1,1,1,0,0)
  v_2: (A:1 OR B:1 **OR** C:9) **XOR** (D:2 **OR** E:2)
  Bidder 2 gets his demand: {C}

Algorithm-Example

Items: {A, B, C, D, E}. 3 bidders.

- Price vector: p_0=(0,0,0,0,0)
  v_1: (A:1 **OR** B:1 **OR** C:1) **XOR** (C:2)
  Bidder 1 gets his demand: {A,B,C}.
- Price vector: p_1=(1,1,1,0,0)
  v_2: (A:1 OR B:1 **OR** C:9) **XOR** (D:2 **OR** E:2)
  Bidder 2 gets his demand: {C}
- Price vector: p_2=(1,1,9,0,0)
  v_3: (C:10 **OR** D:1 **OR** E:2)
  Bidder 3 gets his demand: {C,D,E}

Final allocation: {A,B} to bidder 1, {C,D,E} to bidder 3.

The Algorithm

- Input: *n* bidders, for each we are given a demand oracle and a supporting prices oracle.

- Init: $p_1=\ldots=p_m=0$.
- For each bidder $i=1..n$
  - Let S_i be the demand of the i'th bidder at prices $p_1, \ldots, p_m$.
  - For all $i' < i$ take away from $S_{i'}$ any items from S_i.
  - Let $q_1,\ldots,q_m$ be the supporting prices for S_i in v_i.
  - For all $j \in S_i$ update $p_j = q_j$.

Proof

- To prove the approximation ratio, we will need these two simple lemmas:

Lemma: The total social welfare generated by the algorithm is at least Σp_j.

Lemma: The optimal social welfare is at most $2\Sigma p_j$.

Proof – Lemma 1

Lemma: The total social welfare generated by the algorithm is at least Σp_j.

Proof:

- Each bidder *i* got a bundle T_i at stage *i*.
- At the end of the algorithm, he holds $A_i \subseteq T_i$.
- The supporting prices guarantee that:

$$v_i(A_i) \geq \Sigma_{j \in Ai} p_j$$

Proof – Lemma 2

Lemma: The optimal social welfare is at most $2\Sigma p_j$.

Proof:

- Let $O_1,\ldots,O_n$ be the optimal allocation. Let $p_{i,j}$ be the price of the j'th item at the i'th stage.
- Each bidder i ask for the bundle that maximizes his demand at the i'th stage:

$$v_i(O_i)-\Sigma_{j\in Oi}\, p_{i,j} \leq \Sigma_j\, p_{i,j} - \Sigma_j\, p_{(i-1),j}$$

- Since the prices are non-decreasing:

$$v_i\,(O_i\,)-\Sigma_{j\in Oi}\, p_{n,j} \leq \Sigma_j\, p_{i,j} - \Sigma_j\, p_{(i-1),j}$$

- Summing up on both sides:

$$\Rightarrow \Sigma_i\, v_i(O_i\,)-\Sigma_i\Sigma_{j\in Oi}\, p_{n,j} \leq \Sigma_i\,(\Sigma_j\, p_{i,j} - \Sigma_j p_{(i-1),j})$$

$$\Rightarrow \Sigma_i\, v_i(O_i\,)-\Sigma_j\, p_{n,j} \leq \Sigma_j\, p_{n,j}$$

$$\Rightarrow \Sigma_i\, v_i(O_i\,) \leq 2\Sigma_j\, p_{n,j}$$

Talk Structure

- Combinatorial Auctions
- Log(m)-approximation for CF auctions
- An incentive compatible *$O(m^{1/2})$*-approximation of CF auctions using value queries.
- 2-approximation for XOS auctions
- → A lower bound of e/(e-1)-ε for XOS auctions

XOS Lower Bounds:

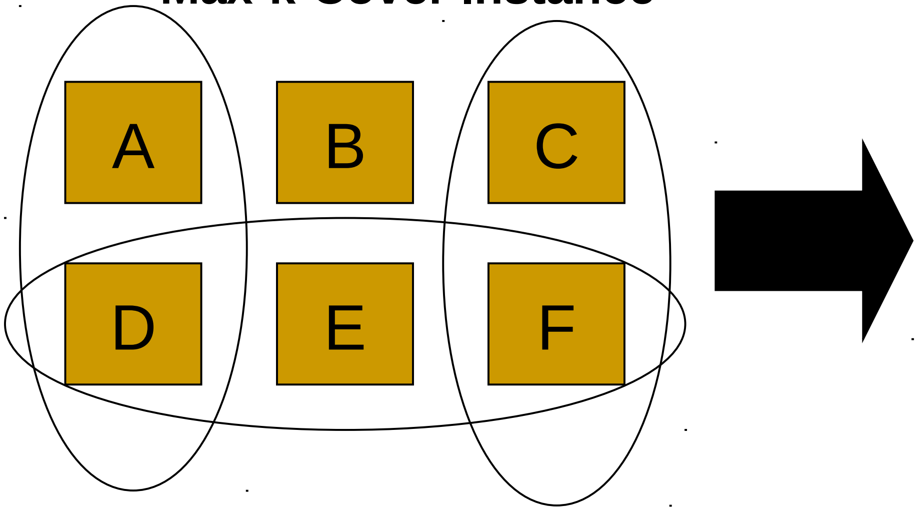
- We show two lower bounds:
  - A communication lower bound of e/(e-1)-ε for the "black box" approach.
  - An NP-Hardness result of e/(e-1)-ε for the case that the input is given in XOS format (bidding language).
- We now prove the second of these results.

Max-k-Cover

- We will show a polynomial time reduction from Max-k-Cover.
- Max-k-Cover definition:
  - Input: a set of $|M|=m$ items, t subsets $S_i \subseteq M$, an integer k.
  - Goal: Find k subsets such that the number of items in their union, $|\cup S_i|$, is maximized.
- **Theorem**: approximating Max-k-Cover within a factor of e/(e-1) is NP-hard (Feige).

The Reduction

Max-k-Cover Instance



XOS Auction with k bidders

v_1: (A:1 **OR** D:1) **XOR** (C:1 **OR** F:1)
XOR (D:1 **OR** E:1 **OR** F:1)

●
●
●

v_k: (A:1 **OR** D:1) **XOR** (C:1 **OR** F:1)
XOR (D:1 **OR** E:1 **OR** F:1)

- Every solution to Max-k-Cover implies an allocation with the same value.
- Every allocation implies a solution to Max-k-Cover with at least that value.
- ➔ Same approximation lower bound.
- A matching communication lower bound exists.

Open Questions – Narrowing the Gaps

Valuation Class	Value queries	Demand queries	General communication
General	$\leq m/(\log^{1/2} m)$ *(Holzman, Kfir-Dahav, Monderer, Tennenholz)* $\geq m/(\log m)$ *(Nisan-Segal, Dobzinki-Schapira)*	$\leq m^{1/2}$*(Blumrosen-Nisan)*	$\geq m^{1/2}$*(Nisan-Segal)*
CF	$\leq m^{1/2}$	**≤ log(m)**	**≥ 2**
XOS			**≤ 2** **≥ e/(e-1)**
SM	≤ 2(Lehmann,Lehmann,Nisan) ≥ e/(e-1)(**new**: Khot, Lipton,Markakis, Mehta)		≥ 1+1/(2m)(Nisan-Segal)
GS	1(Bertelsen, Lehmann)		