Algorithmic Game Theory December 12, 2013

Lecture 8

Lecturer: Michal Feldman Scribe: Ophir Friedler, Yogev Vaknin, Jonathan Kaletchtein

1 Combinatorial auction

In the previous lecture we talked about the general case of combinatorial auctions where we have n players and m different unsplittable products, and an allocation space, which is a set of vectors $s_1, ..., s_n$ where s_i is the set of products that player i gets, which implies $s_i \cap s_j = \phi$ for the allocation to be valid.

Throughout our analysis we are going to make two standard assumptions about the valuation functions of players:

$$S \subseteq T \Rightarrow v_i(S) \le v_i(T) \qquad \text{monotonicity} \qquad (1)$$

$$v_i(\phi) = 0 \qquad \text{normalization} \qquad (2)$$

Examples for combinatorial auctions are financial auctions like the stock market, or electromagnetic frequencies, where M is expected to be very large, therefore any algorithm we would suggest would have to be at least $poly(M)$.

Throughout the rest of the lecture we are going to discuss three main issues: representation, complexity of computation, and incentives:

Representation: Every player has a valuation function $v_i : 2^M \to \mathbb{R}_+$ where M might be very large, how can we represent each player's valuation?

Complexity: Can we efficiently find an allocation that maximizes the social welfare $\sum_{i=1}^{n} v_i(s)$?

Incentives: How do we build a truthful mechanism?

Notice that another assumption we have made here is that each player knows its valuation function without any computation. Let us consider a few examples:

Example 1 *Additive valuations: Each player i values a set S by the sum of the values of its elements:* $\forall S, v_i(S) = \sum_{j \in S} v_i(j)$.

Representation: Each player is able to pass her valuation by passing m numbers, therefore representation is not an issue.

Complexity: In this case social welfare is maximized when each product is be given to the player that values it the most. This allocation is simple and can be computed efficiently.

Incentives: Perform second price auctions on each of the products separately, we have seen that this implies a truthful mechanism.

Definition 1 *A sub additive function v is a function that satisfies:*

$$v(S+T) \leq v(S) + v(T)$$

Example 2 *An extreme case of substitutional products (the valuation functions are sub-additive function):*

Unit demand bidders: For every agent i and product j, set a value for $v_i(\{j\})$. For any set that is not a singleton:

$$v_i(S) = max_{j \in S} v_i(\{j\}) \tag{3}$$

Representation: Again each player is able to pass her valuation by passing m numbers, therefore representation is not an issue.

Complexity: The problem is equivalent to finding a maximal weighted matching between people and products, which has a polynomial time solution, therefore finding an allocation that maximizes social welfare can be done efficiently.

Incentives:In previous lectures we have seen that calculating payments for the VCG mechanism is done by finding an allocation that maximizes social welfare twice, and this can be done efficiently. Therefore VCG can be run efficiently, and VCG is a truthful mechanism.

Definition 2 *Given a valuation function v, complementary sets products S and T are such that:*

$$v(S+T) > v(S) + v(T)$$

Example 3 *An extreme case of complementary valuation (Single minded bidders)*
Each agent i values a package S_i^* to v_i^*, hence:
$v_i(S) = v_i^*$ *if* $S_i^* \subseteq S$ *else* 0

This is very similar to the case where players have single parameter as private information, however in this case the private information of a player is (v_i, S_i^*), which is more private information than one parameter (therefore, this is NOT a single parameter case).

Representation: Each player is able to pass her valuation by passing $2m$ numbers, therefore representation is not an issue.

We are now going to discuss the issues of **complexity** and **incentives**.

Complexity:

Claim 3 *Maximizing social welfare for single minded bidders is* $NP-Hard$

Proof: Let us see a reduction from $Max-IS$ to our problem:
Given a graph $G=(V,E)$, define the auction where V is the set of players, and E is the set of products. The valuation of each player i is $v_i^*=1$, and the package $S_i^*=\{e \in E : i \in e\}$ then the set of "winners" satisfy $S_i \cap S_j = \emptyset$ if and only if the set of corresponding nodes is an independent set in the graph.
Therefore the induced social welfare is exactly the number of nodes in the independent set.
$\Rightarrow |IS| \geq k \Leftrightarrow SW \geq k$ ■

It is known that this problem is hard to approximate to a factor of $\forall \epsilon : n^{1-\epsilon}$ when n is the number of nodes. Since in our case $|IS| = SW$ the reduction preserves approximations, therefore our problem is hard to approximate up to a factor of $m^{\frac{1}{2}-\epsilon}$.

Claim 4 *There exist a polynomial mechanism that provides an approximation of* $\sqrt{m}$*.*

This is the best approximation the original IS problem, therefore this is the best approximation we can hope for.

Case 1: private case that can be solved in polynomial time: Single minded bidders
The m products are m consecutive months each player wants to rent an apartment in some interval.

Claim 5 *It's possible to find an optimal Social Welfare in polynomial time.*

Proof: Via dynamic programming:
$k=$ The number of months from the beginning

$$v(k) = max\{v(k-1), max_i(v_i[j \dots k] + v(j-1))\} \tag{4}$$

on each step we can take player that it's interval ends in k, we will choose the player that maximizes $v_i[j \dots k] + v(j-1)$. ■

Case 2: seconde private case that can be solved in polynomial time: for each player $|S_i^*| = 2$
For each i, $|S_i^*| = 2$. In this case we can represent each agent is an edge, and each product is a node, and v_i is the weight of the edge. This case is equivalent to *max-weighted-matching*, and therefore it can be solved in polynomial time.

A general mechanism:

Sort the agents by:

$$\frac{V_1^*}{\sqrt{|S_1^*|}} \geq \cdots \geq \frac{V_n^*}{\sqrt{|S_n^*|}} \tag{5}$$

The algorithm returns the set of winners W:

Algorithm 1 Allocation algorithm

```
w ← φ
for i = 1...n do
  if S_i^* ∩ (∪_{j∈w} S_j^*) = φ then
    w ← w ∪ {i}
  end if
end for
```

Payments: Each player that wins pays the smallest value v_i^* he could say and still win. Formally, to win, a player needs:

$$\frac{v_i^*}{\sqrt{|S_i^*|}} \geq \frac{v_j^*}{\sqrt{|S_j^*|}} \tag{6}$$

When j is the first player s.t

$$S_j \cap S_i \neq \emptyset$$

But:

$$\forall k \in W, k < j : \quad S_j \cap S_k = \emptyset$$

that is:

$$p_i = v_j^* \frac{\sqrt{|S_i^*|}}{\sqrt{|S_j^*|}}$$

Two important attributes we will use:

- **Monotonicity:**

$$(v_i, S_i) \ \ wins \Rightarrow \forall v_i^{'} \geq v_i, S_i^{'} \subseteq S_i : \left(v_i^{'}, S_i^{'}\right) \ \ wins \tag{7}$$

- The payment is the critical value

Both attributes are true in our mechanism.

Claim 3 The mechanism is truthful

Proof:
Let (v_i, S_i) be the real values, we will prove bidding $\left(v_i^{'}, S_i^{'}\right)$ isn't profitable.
For lying to be profitable, it must be that $S_i \subseteq S_i^{'}$.

$$\left(v_i^{'}, S_i^{'}\right) <_i^{monotonocity} \left(v_i^{'}, S_i\right) <_i^{(*)} (v_i, S_i) \tag{8}$$

$(*)$:
If $\left(v_i^{'}, S_i\right)$ doesn't win then the claim is trivial because if you lose the profit is 0, and in that case you can only improve.
Else, there are two possibilities:

1. If (v_i, S_i) wins as well, then the payment is the same.
2. If (v_i, S_i) doesn't win, then the new payment is greater than v_i, but then the utility from winning is negative ■

Claim 4 The mechanism gives a $\sqrt{m}$ approximation

Proof:
We need to prove that:

$$\frac{1}{\sqrt{m}} \sum_{i \in OPT} v_i \leq \sum_{i \in W} v_i \tag{9}$$

When:

$$W = \{i | player\ i\ wins\ in\ our\ algorithm\}$$

$$OPT = \{i | player\ i\ wins\ when\ social\ wellfare\ is\ maxemized\}$$

Denote OPT_i to be the set of players from OPT who didn't win because of player i .
Formally:

$$OPT_i = \left\{j \in OPT, j \geq i | S_i^* \cap S_j^* \neq \emptyset\right\} \tag{10}$$

Then:

$$OPT \subseteq \bigcup_{i \in W} OPT_i$$

It is enough to prove that

$$\forall i \in W : \sum_{j \in OPT_i} v_j^* \leq \sqrt{m} * v_i^* \tag{11}$$

Proof: $\forall j \in OPT_i$:

$$v_j^* \leq \frac{v_i \sqrt{|S_j^*|}}{\sqrt{|S_i^*|}} \tag{12}$$

This is true because OPT_i are the set of players who didn't win because of i and therefore j was after i in the sorting.
If we sum $\forall j \in OPT_i$ we will get:

$$\sum_{j \in OPT_i} v_j^* \leq \sum_{j \in OPT_i} \frac{v_i \sqrt{|S_j^*|}}{\sqrt{|S_i^*|}} \Leftrightarrow \tag{13}$$

$$\sum_{j \in OPT_i} v_j^* \leq \frac{v_i}{\sqrt{|S_i^*|}} \sum_{j \in OPT_i} \sqrt{|S_j^*|}$$

Using the Cauchy-Schwartz inequality, and the fact that the size of the vector $(1, 1..., 1)$ is at most $|S_i^*|$:

$$\sum_{j \in OPT_i} \sqrt{|S_j^*|} = (1, 1..., 1) \cdot (\sqrt{|S_1^*|}, ..., \sqrt{|S_t^*|}) \leq \sqrt{|S_i^*|} \sqrt{\sum_{i=1}^{t} |S_i^*|} \tag{14}$$

$$\leq^{(*)} \sqrt{|S_i^*|} \sqrt{m}$$

$^{(*)}$: If $i \in OPT_i, a$ factor of $\sqrt{2}$ is added.
To remove it, we need to redefine OPT_i s.t $i \notin OPT_i$ and then

$$OPT \subseteq \bigcup_{i \in W} OPT_i \cup \{i : OPT_i = \emptyset\} \tag{15}$$

Altogether:

$$\sum_{j \in OPT_i} v_j^* \leq \frac{v_i^*}{|\sqrt{|S_i^*|}} * \sqrt{|S_i^*|} * \sqrt{m} = v_i^* \sqrt{m} \tag{16}$$

■

Conclusion: The mechanism presented gives a $\sqrt{m}$ approximation and is truthful, and therefore the demand to truthfulness doesn't weaken the approximation ability.