Computational Game Theory **Spring Semester, 2009/10**

Lecture 12: May 31, 2010

Lecturer: Yishay Mansour *Scribe: Eran Cohen, Bar Avidan, Michael Gendelman* [1]

12.1 Introduction

In this lecture we consider **Combinatorial Auctions** (abbreviated *CA*), that is, auctions where instead of competing for a single resource we have multiple resources. The resources assignments and bids are defined on subsets of resources and each player has a valuation defined on subsets of the resource set he was assigned. The interesting case here is when the valuation of a given set of resources is different from the sum of valuations of each resource separately (the whole is *different* from the sum of its parts). That could happen when we have a set of complementary products, that is, each product alone is useless but the group has a significantly larger value (for example - left and right shoes). On the other hand we might have a set of substitutional products where the opposite takes place (for example - tickets for a movie - no use of having two tickets if you are going alone).
In these cases there is an importance for pricing groups of resources rather than single resources separately, i.e. in the absence of complementarity and substitutability (if every participant values a set of goods at the sum of the values of its elements), one should organize the multiple auction as a set of independent simple auctions, but, in the presence of these two attributes, organizing the multiple auction as a set or even a sequence of simple auctions will lead to less than optimal results, in such a case we use **Combinatorial Auctions**.

12.2 Definitions

12.2.1 The CA model

- $N = \{1, \ldots, n\}$ set of players.
- $X = \{1, \ldots, m\}$ set of resources (products).
- $V_i : 2^X \to \mathbb{R}, i \in N$
  Each player has a value function, mapping a subset of products to their value.

[1] Based on a previous scribe done by Nir Yosef, Itamar Nabriski, Nataly Sharkov

- $V = V_1 \times \cdots \times V_n$
- $A = \{a_1, \ldots, a_n\}$ - The set of allocations. A feasible allocation A holds that $\forall i, j \in A :$ $a_i \cap a_j = \emptyset$
- $\vec{p} = \{p_1, \ldots, p_n\}$ - A set of payments defined for each player by the mechanism $p :\in \mathbb{R}^n$
- $u_i = V_i(S) - p_i$ - Each player's utility function, which is quasi-linear in the payment.

12.2.2 Goals and assumptions

- Our goal is to achieve **Efficiency** - find a *pareto-optimal* allocation, that is, no further trade among the buyers can improve the situation of some trader without hurting any of them. This is typically achieved by using an assignment which brings the sum of benefits to a maximum.
- An alternative goal - maximizing Seller's revenue (will not be discussed on this lecture, but in the next lecture).
- Assumption - **no-externalities** : Players' preferences are over subsets of S and do not include full specification of preferences about the outcomes of the auction (the resulting allocation). Thus, a player cannot express externalities, for example, that he would prefer, if he does not get a specific resource, that this resource to be allocated to player X and not to player Y.

12.2.3 Examples

- *Substitutional products*: $S, T : S \cap T = \emptyset, V(S \cup T) \leq V(S) + V(T)$
- *Complementary Products*: $S, T : S \cap T = \emptyset, V(S \cup T) \geq V(S) + V(T)$
- *Additive values*: $\forall i \in N : V_i(S) = \sum_{j \in S} V_i(\{j\})$

  Additive values are both substitutional and complementary. In order to optimally solve (in terms of seller's revenue and sum of benefits) such an auction, one can simply use a seperate auction for this item.
- *Unit demand*: $\forall i \in N : V_i(S) = max_j\{V_i(\{j\})\}$

We will use two simple assumptions:

- **Motonicity**(*free-disposal*): for every $S, T \subseteq X$ such that $S \subseteq T$, the value attributed to T will not be smaller to that of S, i.e., $S \subseteq T \Rightarrow V_i(S) \leq V_i(T)$ for any player i.

- **Normalizing** $V(\emptyset) = 0$.

Following from the two above assumptions:

- $\forall S \subseteq X : V(S) \geq 0$

12.3 Mechanism Design for *CA*

In order to get an efficient allocation where for each player *telling the truth* is a dominant strategy we might use the *VCG* mechanism. However, using VCG with the general model described above has a clear disadvantage: VCG requires each player's value function, which is $O(2^m)$ bits. We will overcome this problem by inspecting a simpler mechanism called *Single Minded Bidders* (SMB)

12.3.1 *Single Minded Bidders* mechanism - definition

Definition Single Minded Bidder: For every player i there exists a single set $s_i \subseteq S$ which he wants and for which he is willing to pay the (non-negative) price w_i.

$$V_i(s) = \begin{cases} w_i & s_i \subseteq S \\ 0 & \text{otherwise} \end{cases}$$

Clearly, we have a compact description for the players' preferences $< s_i, w_i >$, thus overcoming our initial problem, next we'll see that even for that simplified model, implementing *VCG*, i.e., finding maximal allocations, is *NPC*.

12.3.2 Reduction from Independent Set (IS)

Claim 12.1 *Finding an optimal allocation in* CA *with* SMB *bidders is NP-hard*

Proof: We prove the claim by showing a reduction from the graph-theory problem of maximum independent set to a maximum allocation problem for *SMB* bidders: Given an undirected graph $G = (V, E)$ let us build an instance of *CA* as follows:

- $X = E$: every edge is considered as a resource
- $N = V$: every vertex is considered as a player
- for each player (vertex) i, define s_i as the set of all edges (resources) coming out of that vertex and $w_i = 1$.

For example, see following figure:

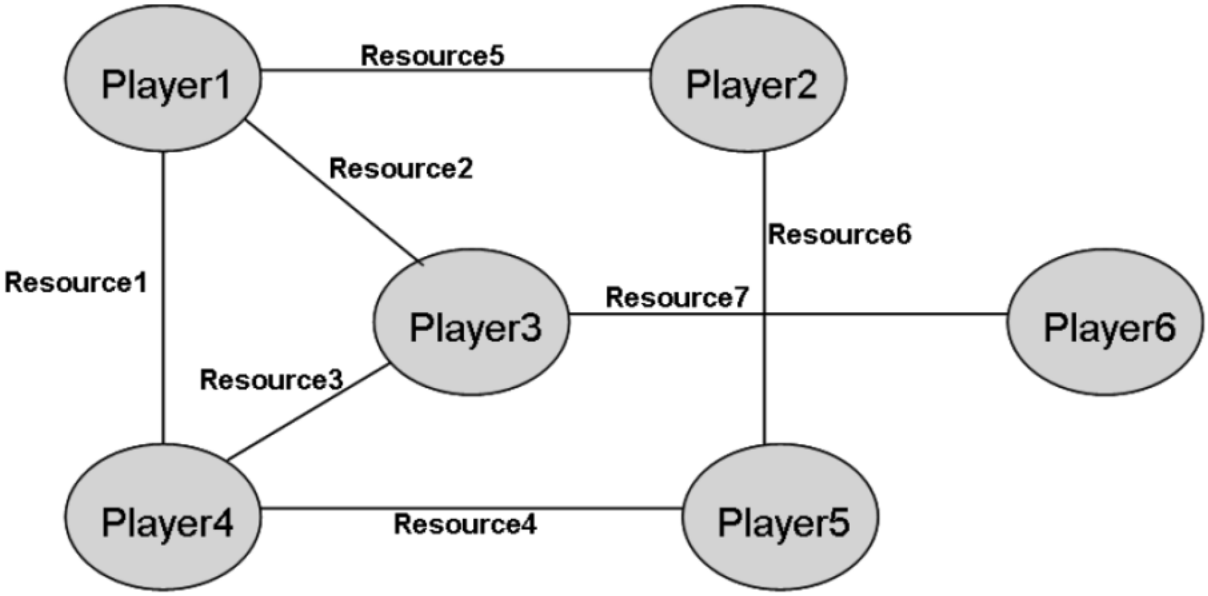

Fig.1 Reduction from IS on an undirected graph to finding optimal allocation on CA with SMB. For example: Player1 desired set of resources (s_1) is $\{2, 5, 1\}$

- any feasible allocation defines an independent set (the set of all players(vertices) with a non-zero benefit) with the same value
- on the other hand, any independent set Δ defines a feasible allocation (Allocate s_i for every player(vertex) i such that $i \in \Delta$) with the same value as well.

Thus, finding a maximal social benefit is equivalent to finding a maximum independent set. From the above reduction and since *IS* is in *NPC*, we conclude the same on the problem of finding an optimal allocation. □

Corollary 12.2 *Since no approximation scheme for IS has an approximation ratio less than* $|V|^{1-\epsilon}$*, and for CA we have* $m \leq n^2$ *resources, we get a bound of* $m^{\frac{1}{2}-\epsilon}$ *on the approximation ratio for our problem where m is the number of resources.*

12.4 The greedy allocation

As we have seen, for all practical purposes, there does not exist a polynomial-time algorithm for computing an optimal allocation, or even for computing an allocation that is guaranteed to be at least the optimal times a constant, for any given constant. One approach to meeting this difficulty is to replace the exact optimization by an approximated one. Next,

we shall propose a family of algorithms that provide such an approximation. Each of those algorithms runs in a polynomial time in n, the number of single-minded bidders. Finally, we (unfortunately) see that the properties guaranteed by the mechanism (such as strategy-proof bidding, to be defined later), disappear when using these approximated allocations.
(**comment** - *traditional analysis of established CA mechanisms relies strongly on the fact that the goods are allocated in an optimal manner*).

General description of the algorithms:

- First phase: the players are sorted by some criteria. The algorithms of the family are distinguished by the different criteria they use.

- Second phase: a greedy algorithm generates an allocation. Let L be the list of sorted players obtained in the first phase. The bid of the first player of L ($< s_1, w_1 >$) is granted, that is, the set s_1 will be allocated to player 1. Then, the algorithm examines all other player of L, in order, and grants its bid if it does not conflict with any of the previously granted sets. If it conflicts, the bid is denied (i.e., does not grant).

Payment:
For each player i, the payment p_i will be the minimal w_i which he has to bid in order to win (losers pay none).

Sort criterias: We will talk about 3 greedy algorithms, with 3 sort criterias:

- First sort criteria: $f_1 = w_i$

- Second sort criteria: $f_2 = \frac{w_i}{|s_i|}$

- Third sort criteria: $f_3 = \frac{w_i}{\sqrt{|s_i|}}$

12.5 Strategy-Proof Mechanism with Greedy Allocation in *SMB*

12.5.1 Greedy Allocation Scheme and *VCG* do not make a Strategy-Proof Mechanism in *SMB*

The following example illustrates a case where using f_2 and VCG doesn't yield a strategy-proof mechanism (and simiraly for any f_i):

Player	$< s_i, v_i >$	$\frac{v_i}{\|s_i\|}$	t_i
R	$(\{a\}, 10)$	10	$8 - 19 = -11$
G	$(\{a, b\}, 19)$	9.5	0
B	$(\{b\}, 8)$	8	$10 - 10 = 0$

Since the t_i's represent the value gained by the other players in the auction minus the value gained by the other players had i not participated in the auction, R ends up with a lose of 11. Had R not been strategy-proof and bid below 9.5 (f_2's $\frac{v_i}{|s_i|}$), he would be better off gaining 0. Thus in this case being strategy-proof is not a dominant strategy for R and thus this mechanism is not strategy-proof.
We now explore the conditions necessary for a strategy-proof greedy allocation mechanism in *SMB*.

12.5.2 Sufficient Conditions for a Strategy-Proof Mechanism in SMB

Theorem 12.3 *The mechanism for Single Minded Bidders is strategy proof if, and only if, it holds both of the following:*

1. *Monotonicity: Given a winning bid $< s_i^*, w_i^* >$, any $< s_i^*, w_i' >, < s_i', w_i^* >$ is a winning bid, if $w_i^* \leq w_i'$, or $s_i' \subseteq s_i^*$.*
2. *Critical-Fee: Winning player pays the minimal payment for him to win.*

Proof: Given a truthful bid $< s_i^*, w_i^* >$, then $u_i^* \geq 0$ (the player has a non-negative gain). We will show that any other bid $< s_i', w_i' >$ has a lower gain. There are two possibilities we need to consider:

- $s_i' \neq s_i^*$:
  - $s_i' \subset s_i^*$, in this case $v_i' = 0$ (the player doesn't get all the items he wanted), meaning $u_i' \leq u_i^*$
  - $s_i^* \subset s_i'$, in this case, $v_i' = v_i$, yet, by the critical-fee, we know the $p^* \leq p'$, meaning $u_i' \leq u_i^*$
- $w_i' \neq w_i^*$:
  - if he wins in both cases, we know that by the critical fee $p^* = p'$.

– if giving a different w'_i made him win the auction, we know that $w^*_i \leq p' \leq w'_i$, thus, $u'_i \leq 0$.

$\square$

Corollary 12.4 *Greedy Single Minded Bidders are strategy proof.*

12.5.3 First sort criteria: $f_1(w_i) = w_i$

Claim 12.5 *Using a greedy algorithm, with $f_1(w_i) = w_i$ as a sort criteria has an approximation ratio of m*

Proof:
$\Rightarrow$ The ratio is at least m, as shown by the following example:
Suppose we have a set $N = \{1, \ldots, n\}$ of players of players (SMBs) and a set $S = \{1, \ldots, m\}$ of resources where $m = n$, and suppose:

- Player 1 asks for all the resources and his value is $1 + \epsilon$, $[s_1 = X,\, w_1 = 1 + \epsilon]$
- $\forall i : 2 \leq i \leq n$ player i asks for resource i and his value is 1, $[s_i = \{i\},\, w_i = 1]$

In this case it follows that $OPT = m$ but $f_1 = 1 + \epsilon$

$\Leftarrow$ The ratio can be at most m because the value of the first player in a greedy allocation is higher than that of any player in OPT (follows immediately from the feasibility of OPT), so its value is at least $\frac{1}{m}$ from OPT. $\square$

12.5.4 Second sort criteria: $f_2(w_i, s_i) = \frac{w_i}{|s_i|}$

Claim 12.6 *Using a greedy algorithm, with $f_2(w_i, s_i) = \frac{w_i}{|s_i|}$ as a sort criteria has an approximation ratio of m.*

Proof:
$\Rightarrow$ The ratio is at least m, as shown by the following example:
Assuming we have a set of two players and a set of resources similar to the above, suppose:

- Player 1 asks for resource 1 and his value is 1 $[s_1 = 1,\, w_1 = 1]$
- Player 2 asks for all the resources and his value is $m - \epsilon$ $[s_2 = X,\, w_2 = m - \epsilon]$

In this case it follows that $OPT = m - \epsilon$ but $f_2 = 1$

$\Leftarrow$ The ratio can be at most m:
Let G_2 be, any player i which his requests s_i were allocated by OPT and not allocated by f_2. $\forall i \in G_2$ there exists a player j $s.t.$:

- $s_i \cap s_j \neq \emptyset$
- $f_2(s_j, w_j) \geq f_2(s_i, w_i)$

For each player $i \in G_2$ we match such a player j, and denote: $J(i) = j$, and let $G_J = \{j \mid \exists i \; J(i) = j\}$.
Now, from the above definition of J and from the feasibility and greediness of f_2, we can conclude:
$\frac{w_i}{|s_i|} \leq \frac{w_{J(i)}}{|s_{J(i)}|}$

From which follows: $w_i \leq \frac{|s_i|}{|s_{J(i)}|} w_{J(i)}$

And finally:

$\sum_{i \in G_2} w_i \leq \sum_{i \in G_2} \frac{|s_i|}{|s_{J(i)}|} w_{J(i)} \leq \sum_{j \in G_J} |s_i| w_j \leq m \sum_{j \in G_J} w_j$
The second inequality follows since each $j \in G_J$ can have at most $|s_j|$ players i s.t. $J(i) = j$.
Since $\sum_{i \in OPT - G_2} w_i = \sum_{j \in f_2 - G_J} w_j$ we have that $OPT \leq m f_2$. □

12.5.5 Third sort criteria: $f_3(w_i, s_i) = \frac{w_i}{\sqrt{|s_i|}}$

Claim 12.7 *Using a greedy algorithm, with $f_3(w_i, s_i) = \frac{w_i}{\sqrt{|s_i|}}$ as a sort criteria has an approximation ratio of $\sqrt{m}$.*

Proof:
$\Rightarrow$ The ratio is at least $\sqrt{m}$, as shown by the following example:
Suppose we have a set $N = \{1, \ldots, n\}$ of players (*SMB*s) and a set $X = \{1, \ldots, m\}$ of resources where $m = n$, and suppose:

- Player 1 asks for all the resources and his value is $m + \epsilon$, $[s_1 = X, \; w_1 = m + \epsilon]$
- $\forall i : 2 \leq i \leq n$ player i asks for resource i and his value is $\sqrt{m}$, $[s_i = \{i\}, \; w_i = \sqrt{m}]$

In this case it follows that $OPT = m\sqrt{m}$ but $f_3 = m + \epsilon$

$\Leftarrow$ The ratio is at most $\sqrt{m}$:
Consider the following two inequalities, let $r_j = \frac{w_j}{\sqrt{|s_j|}}$.

$f_3 = \sum_{j \in f_3} w_j \geq \sqrt{\sum_{j \in f_3} w_j^2} = \sqrt{\sum_{j \in f_3} r_j^2 |s_j|}$,

- *Because* $\forall_{1<j<n} : w_j > 0$

$OPT = \sum_{i \in OPT} r_j \sqrt{|s_j|} \leq \sqrt{\sum_{i \in OPT} r_i^2} \sqrt{\sum_{i \in OPT} |s_i|} \leq \sqrt{\sum_{i \in OPT} r_i^2} \sqrt{m}.$

- *The first inequality follows from: Cauchy-Schwarz inequality*
- *The last inequality follows from:* $(\forall i, j \in OPT : i \neq j) \rightarrow (s_i \cap s_j = \emptyset)$

Thus it is enough to compare $\sqrt{\sum_{j \in f_3} r_j^2 |s_j|}$ and $\sqrt{\sum_{i \in OPT} r_i^2}$

Let us consider the function $J(i)$ as in the last proof. In the same manner we can conclude $\forall i \in OPT$:

1. $s_i \cap s_{J(i)} \neq \emptyset$
2. $r_i \leq r_{J(i)}$

From the feasibility of OPT it follows that for every subset s_j allocated by f_3, there exists at most $|s_j|$ subsets which are allocated by OPT and rejected by f_3 because of s_j.
Summing for all $i \in OPT$, we get:

$\sqrt{\sum_{i \in OPT} r_i^2} \leq \sqrt{\sum_{i \in OPT} r_{J(i)}^2} \leq \sqrt{\sum_{j \in f_3} r_j^2 |s_j|}$

Where the second inequality follows since at most $|s_j|$ values of i have $J(i) = j$.

And finally, we get:
$OPT \leq \sqrt{m} \sqrt{\sum_{i \in OPT} r_i^2} \leq \sqrt{\sum_{j \in f_3} r_j^2 |s_j|} \leq \sqrt{m} f_3$ □

12.6 Gross Substitute

Definition Gross Substitute function A value function is GS if for all prices $\vec{p} \leq \vec{q}$ the demand for products in $T = \{j | p_i = p_j\}$ did not fall when we changed from $\vec{p}$ to $\vec{q}$:
For each

$$S' \in \arg\max_s (V_i(s) - \textstyle\sum_{j \in S} p_j)$$

There exists

$$S'' \in \arg\max_s (v_i(s) - q(s))$$

such that $S' \cap T \subseteq S''$

A simple case of a GS value function: Unit Demand.
SMB is not a GS, for example:

$$S = \{a, b\}$$
$$w = 10$$

With prices (3,3) the player will require S
With prices (8,3) the player will require $\emptyset$
Thus, $T = \{b\}$

Therefore, SMB is not a GS

Definition An allocation $S_1, \ldots, S_n$ with prices $p_1, \ldots, p_n$ is a Walrasian ϵ-equilibrium if:

1. $j \notin \bigcup_{i \in N} S_i$ then $p_j = 0$ or equivalently $\{j | p_j > 0\} \subseteq \bigcup_{i \in N} S_i$
2. For each $i \in N$ the set S_i is a Best Response w.r.t the following prices:
   - p_j for $j \in S_i$
   - $p_j + \epsilon$ for $j \notin S_i$

A Walrasian ϵ-equilibrium does not always exist (with $\epsilon = 0$)
For example: Two players, two products.

Player 1:

$$v_1(\{a, b\}) = 3$$
$$v_1(\{a\}) = v_1(\{b\}) = 0$$

Player 2:

$$v_2(\{a\}) = v_2(\{b\}) = v_2(\{a, b\}) = 2$$

If player 2 gets a then the price of b has to be 0, and the price of $a \leq 2$. In that case player 1 would like to get $\{a, b\}$. Therefore the price for both a and b is at least 2. At these prices, player 1 would not like to get any product (and hence the prices should be 0).

The following algorithm computes an ϵ-Walrasian equilibrium for gross substitute bidders.

Algorithm:

for each $j \in X$
  do
  $p_j := 0$
  end for
for each $i \in N$
  do
  $S_i := \emptyset$
  end for
loop
  for each player $i \in N$ compute the demand D_i with the following prices:
    do
    p_j when $j \in S_i$
    $p_j + \epsilon$ when $j \notin S_i$
    end for
  if $\forall i \; D_i = S_i$
   then
     return (an equilibrium was found)
   else
     Find i such that $S_i \neq D_i$
     Update:
     for $j \in D_i - S_i$
       do
       $p_j = p_j + \epsilon$
       end for
     $S_i := D_i$
     for $i \neq k$
       do
       $S_k := S_k - D_i$
       end for
   end if
  end loop
returns
Allocation $S_1, \ldots, S_n$
Prices $p_1, \ldots, p_m$

Claim 12.8 *In each stage, for every player i: $S_i \subseteq D_i$*

Proof:

- At start $S_i = \emptyset$
- Looking at an update step of player i:
  - For player i itself: $S_i := D_i$ and D_i is the new demand of i.
  - For player $k \neq i$: The prices in S_k did not rise after the change. Because of the GS properties, S_k is part of his Best Response.

$\square$

Theorem 12.9 *For players with GS value function the algorithm finds:*

1. *A Walrasian ϵ-equilibrium*
2. *An allocation that is at most ϵm of the maximum social welfare.*

Proof:

1. From the previous claim we can see that every item that was previously allocated will stay allocated. Thus:

$$\{j|p_j > 0\} \subseteq \bigcup_{i \in N} S_i$$

At the end of the algorithm: $S_i = D_i$ therefore it is an ϵ-BR for every player, and a Walrasian ϵ-equilibrium.

2. Let $S_1^*, \ldots, S_n^*$ be an allocation of the algorithm and $S_1, \ldots, S_n$ some other allocation (for $\epsilon = 0$).
$\forall i \in N$:

$$V_i(S_i^*) - \sum_{j \in S_i^*} p_j \geq V_i(S_i) - \sum_{j \in S_j} p_j$$

Now we sum both sides, over the players:

$$\sum_{i \in N} V_i(S_i^*) - \sum_{j \in \bigcup_{i \in N} S_i^*} p_j \geq \sum_{i \in N} V_i(S_i) - \sum_{j \in \bigcup_{i \in N} S_i} p_j$$

Since for any $j \notin \bigcup_{i \in N} S_i^* : p_j = 0$. So we get:

$$\sum_{j \in \bigcup_{i \in N} S_i^*} p_j \geq \sum_{j \in \bigcup_{i \in N} S_i} p_j$$

And thus,

$$\sum_{i \in N} V_i(S_i^*) \geq \sum_{i \in N} V_i(S_i)$$

For $\epsilon > 0$:

$$V_i(S_i^*) - \sum_{j \in S_i^*} p_j \geq V_i(S_i) - \sum_{j \in S_i} p_j - \epsilon |S_i - S_i^*|$$

Hence, the difference from the maximum social welfare $\leq \epsilon \sum_{i \in N} |S_i - S_i^*| \leq \epsilon m$.

□

The algorithm is not strategy-proof, For example:

	a	b	$\{a, b\}$
Alice	4	4	4
Bob	5	5	10

In this case, the algorithm sets the prices $p_a = p_b = 4$ and allocates both to Bob. Bob pays 8, and his payoff is $10 - 8 = 2$.

$$p_a = p_b = 4$$
$$S_{Bob} = \{a, b\}$$
$$S_{Alice} = \emptyset$$

Bob's strategic behaviour:
If Bob bids only on a during the auction (claims value zero for b), then the auction would stop at zero prices, allocating a to Bob and b to Alice. With this demand reduction, Bob improves his payoff to 5.
That is: $D_{Bob} = \{a\}$
The outcome is:

$$p_a \approx p_b \approx 0$$
$$S_{Bob} = \{a\}$$
$$S_{Alice} = \{b\}$$

Bob's payoff = 5, Alices's payoff = 4