Lecture 11:June 8

Lecturer:Yishay Mansour *Scribe:Nir Yosef, Itamar Nabriski, Nataly Sharkov*

11.1 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 cases here is when the valuation of a given set of resources is different form 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**.

11.2 Preliminaries

Throughout this lecture, we shall consider single-side combinatorial auctions, that is, auctions with single seller and multiple buyers.
Any such auction must specify three elements:

- The bidding rules (i.e., what one is allowed to bid for and when).
- The market clearing rules (i.e., when is it decided who bought what and who pays what)
- The information disclosure rules (i.e., what information about the bid state is disclosed to whom and when).

We consider only one-stage, sealed-bid *CAs*: each bidder submits zero or more bids, the auction clears, and the results are announced.
The third element of the specification is thus straightforward: no information is released about other bidders' bids prior to the close of the auction. The first element of the specification is almost as straightforward: each bidder may submit one or more bids, each of which mentions a subset of the goods and a price. One has to be precise, however, about the semantics of the collection of bids submitted by a single bidder, because, as was mentioned, the bid for a group doesn't necessarily equal to the sum of bids of its elements.
Only the second element of the above specification, the clearing policy, provides choices for the designer of the *CA*. There are two choices to be made here: which goods does every bidder receive, and how much does every bidder pay? We address these below.

11.2.1 The model

- $N = \{1..n\}$ set of players.
- $S = \{1..m\}$ set of resources (products).
- Θ - set of players private information, player i has information $\theta_i \in \Theta_i$ which is the inner state he is currently in.
- D - Mechanisms decision space - each vector specifies resources allocation amongst the players. $D = \{< s_1..s_n > \mid (\forall i \neq j \ s_i \cap s_j = \emptyset) \wedge (\bigcup_{1 \leq i \leq n} s_i \subseteq S) \ \}$.
- $V = \{V_1..V_n\}$ - set of preference functions $V_i : D \times \Theta_i \rightarrow R$ which is the value which player i attributes to every subset of S given its internal state θ_i.
- $\vec{t} = \{t_1..t_n\}$ - set of payments defined for each player by the mechanism $t : \Theta \rightarrow R^n$, $t_i(\Theta) \in R$.

Remark 11.1 ***Monotonicity*** *for every $s_1, s_2 \in S$ such that $s_1 \subseteq s_2$, the value attributed to s_2 will not be smaller to that of s_1. i.e. $s_1 \subseteq s_2 \Rightarrow V_i(s_1) \leq V_i(s_2)$ for any player i.*

11.2.2 Goals and assumptions

- Our goal will be guaranteeing **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)

- 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, this resource to be allocated to player X and not to player Y.

11.3 Mechanism Design for *CA*

In order to get an efficient allocation where for each player *telling the truth* is a dominant strategy we'll use the *VCG* mechanism.

11.3.1 *VCG* mechanism - definition

- Decision rule(resource allocation): $d =< s_1..s_n >\in D$ such that d = $ArgMax\ _{d\in D} \sum_i V_i(s_i, \theta_i)$. That is, the chosen allocation maximizes the sum of the declared valuations of the players.
- Payment scheme: $t_i(\theta) = \sum_{j\neq i} V_j(s_j, \theta_j) - VMax_{<s'_1..s'_n>|s'_i=\emptyset} \sum_{j\neq i} V_j(s_j, \theta_j)$. That is, each player receives a monetary amount that equals the sum of the declared valuations of all other players, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction.

Remark 11.2 *Note that A bidder knows his own inner state, but this information is private and neither the auctioneer nor the other players have access to it, thus both of the above are functions of the players' declarations rather than its inner state.*

In the previous lecture we've seen that this mechanism brings the social benefit (sum of all benefits calculated according to players' declarations) to a maximum while keeping truth-telling as a dominant strategy.

11.3.2 Problems with the implementation of *VCG* mechanism

The main problem we confront trying to implement *VCG* mechanism is a computational problem, as it turns out, finding such a maximum benefit allocation is a *NP-hard* optimization problem (moreover, in our case we need to calculate a maximum benefit allocation $n+1$ times) that, in the worst case, cannot be even approximated in a feasible way.
An additional problem is describing players' preferences: the domain of the preference function is the product of all subsets of S with player's internal state and as such, for a given state, its size is exponential in m.
Comment: the size of the domain mentioned above is under the assumption of no-externalities.

Without that assumption, the domain would have been much larger ($|D| \times |\Theta|$)

In the following sections we consider a simplified model of *CA* called *SMB* (single minded bidder) defined as:
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 c_i.

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

We have a compact description for the players' preferences $< s_i, c_i >$, thus overcoming the second problem, next we'll see that even for that simplified model, implementing *VCG* i.e. finding maximal allocations, is NP-hard.

11.3.3 Reduction from IS

Claim 11.3 *Finding an optimal allocation on* CA *with* SMB *model 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 on *SMB* model: Given an undirected graph $G = (V, E)$ let us build an instance of *CA* as follows:

- $S = E$: 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 $c_i = 1$.

For example, see following figure:

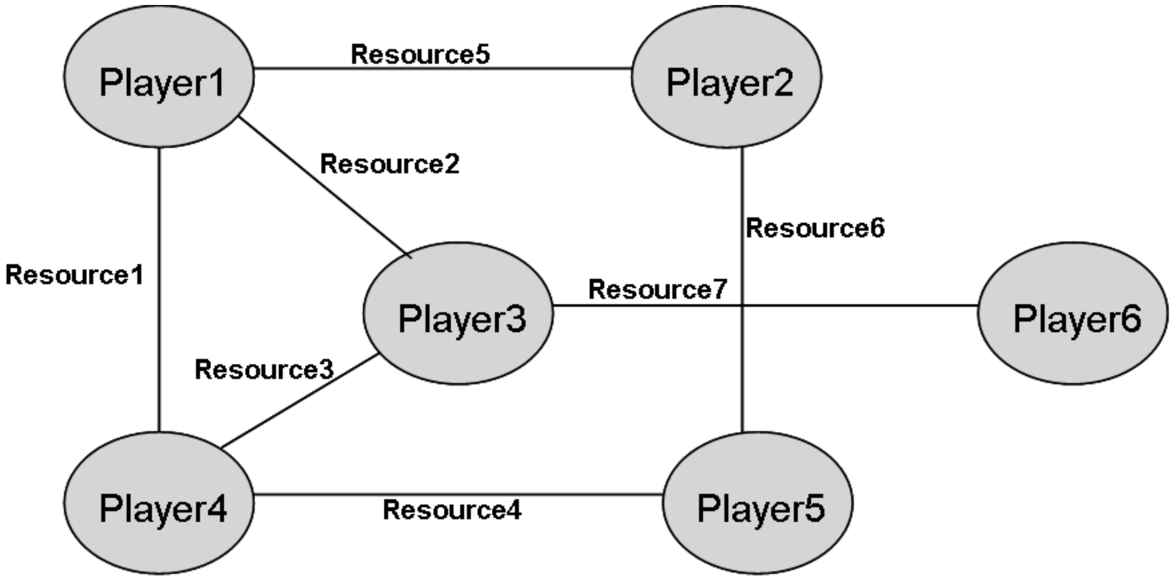

Fig.1 Reduction from IS on an undirected graph to finding optimal allocation on CA with SMB
For example: Player1 desired set of resources (s_1) is $\{2, 5, 1\}$

>From the definition of D above, it is easy to see that:

- any legal 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 legal 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 11.4 *Since we have $|E| \leq |V|^2$ resources and since no approximation scheme for IS has an approximation ratio of $|V|^{1-\epsilon}$ we get a bound of $\sqrt{m}$ on the approximation ratio for our problem where m is the number of resources.*

11.4 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 off from optimal by at most a constant, 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 truthful 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 i_1 of L ($< s_{i_1}, c_{i_1} >$); is granted, that is, the set s_{i_1} will be allocated to player i_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 bids previously granted. If it does, it denies (i.e., does not grant) the bid.

11.4.1 First sort criteria: c_i

Claim 11.5 *Using a greedy algorithm, G_1, with c_i as a sort criteria would yield an approximation factor of m*

Proof:
$\Rightarrow$ The ratio is at least m, proof by example:
Suppose we have a set $N = \{1..n\}$ of players (SMB's) and a set $S = \{1..m\}$ of resources where $m = n$. and suppose:

- Player 1 asks for all the resources and his value is $1 + \epsilon$, $[s_1 = S,\ c_1 = 1 + \epsilon]$

- $\forall 2 \leq i \leq n$ player i asks for resource i and his value is 1 , $[s_i = \{i\},\ c_i = 1]$

In this case it follows that $OPT = m$ but $G_1 = 1 + \epsilon$

$\Leftarrow$ 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) □

11.4.2 Second sort criteria: $\frac{c_i}{|s_i|}$

Claim 11.6 *Using a greedy algorithm, G_2, with $\frac{c_i}{|s_i|}$ as a sort criteria would yield an approximation factor of m.*

Proof:

$\Rightarrow$ The ratio is at least m, proof by 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 $[s_1 = 1, c_1 = 1]$
- Player 2 asks for all the resources and his value is $m - \epsilon$ $[s_2 = S, c_2 = m - \epsilon]$

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

$\Leftarrow$ The ratio can be at most m:
>From the greediness property of G_2, for any subset s_i(requested by player i) that was allocated by OPT and not allocated by Greedy there exists at least one other subset which was previously allocated by G_2 and because of which s_i was not allocated.
Let us consider the following function defined on the subsets allocated by OPT:

$$\forall_{i \in OPT} \quad J(i) = \begin{cases} j: \quad (j \in G_2) \wedge (s_i \cup s_j \neq \emptyset) & i \notin G_2 \\ i & \text{otherwise} \end{cases}$$

Explanation: for any subset s_i (requested by player i) that was allocated by OPT and not allocated by G_2, we take $s_{J(i)}$ as a subset because of which s_i was not allocated. And, for any subset s_i which was allocated both by OPT and G_2 we take $J(i)$ to be equal to i

Now, from the above definition of J and from the feasibility and greediness of G_2, we can conclude ($\forall_{i \in OPT}$):

1. $s_i \cap s_{J(i)} \neq \emptyset$
2. $\frac{c_i}{|s_i|} \leq \frac{c_{J(i)}}{|s_{J(i)}|}$

>From which follows: $c_i \leq \frac{|s_i|}{|s_{J(i)}|} c_{J(i)} \leq |s_i| c_{J(i)}$

And finally:

$$OPT = \sum_{i \in OPT} c_i \leq \sum_{i \in OPT} |s_i| c_{J(i)} \leq m \sum_{i \in OPT} c_{J(i)} \leq \sum_{j \in G_2} c_j = m \cdot G_2$$

- The third inequality is due to the fact that OPT is feasible i.e.,
$(s_1, s_2 \in OPT) \rightarrow (s_1 \cap s_2 = \emptyset)$ $\square$

Remark on notation: *for a player i and an algorithm ALG we say that $i \in ALG$ if the request of player i was granted by ALG*

11.4.3 Third sort criteria: $\frac{c_i}{\sqrt{|s_i|}}$

Claim 11.7 *Using a greedy algorithm, G_3, with $r_i = \frac{c_i}{\sqrt{|s_i|}}$ as a sort criteria would yield an approximation factor of $\sqrt{m}$*

Proof:
Consider the following two inequalities:

$G_3 = \sum_{j \in G_3} c_j \geq \sqrt{\sum_{j \in G_3} c_j^2} = \sqrt{\sum_{j \in G_3} r_j^2 |s_j|}$

- *Because* $\forall_{1<j<n}$, $c_j > 0$

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

- *The last inequality follows from:* $(\forall i_1, i_2 \in OPT i_1 \neq i_2) \rightarrow (s_{i_1} \cup s_{i_2} = \emptyset)$

Thus it is enough to compare $\sqrt{\sum_{j \in G_3} r_j^2 |s_j|}$ 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 G_3, there exists at most $|s_j|$ subsets which are allocated by OPT and rejected by G_3 because of s_j. Summing for all $i \in OPT$, we get:

$\sqrt{\sum_{i \in OPT} r_i^2} \leq \sqrt{\sum_{i \in OPT} r_{J(i)}^2} \leq \sqrt{\sum_{j \in G_3} r_j^2 |s_j|}$

And finally, we get:
$OPT \leq \sqrt{m} \sqrt{\sum_{i \in OPT} r_i^2} \leq \sqrt{\sum_{j \in G_3} r_j^2 |s_j|} \leq \sqrt{m} G_3$ □

11.5 Truthful Mechanism with Greedy Allocation in *SMB*

11.5.1 Greedy Allocation Scheme and *VCG* do not make a Truthful Mechanism in *SMB*

The following example illustrates a case where using G_2 and VCG doesn't yield a truthful mechanism (and simiraly for any G_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 truthful and bid below 9.5 (G_2's $\frac{v_i}{|s_i|}$), he would be better off gaining 0. Thus in this case being truthful is not a dominant strategy for R and thus this mechanism is not truthful.
We now explore the conditions necessary for a truthful greedy allocation mechanism in *SMB*.

11.5.2 Sufficient Conditions for a Thruthful Mechanism in SMB

Let $\{g_1, ..., g_n\}$ denote the set of allocations the mechanism grants to each player. For brevity all bids and valuations are not labeled by the player index and all pretain to player i
Definition Exactness: Either $g_i = s$ or $g_i = \emptyset$.

In other words player i is allocated all the goods he bid for or none at all. There are no partial allocations.

Definition Monotonicity: $s \subseteq g_i$, $s^{'} \subseteq s$, $v^{'} \geq v \Rightarrow s^{'} \subseteq g_i^{'}$.

This means that if player i's bid was granted for bidding $< s, v >$ then his bid would also be granted for bidding $< s^{'}, v^{'} >$ where $s^{'} \subseteq s$ and $v^{'} \geq v$. Thus if a bid for a set of goods is granted then a bid (with at least the same amount of money) for a subset of the goods will be granted as well.

Lemma 11.8 *In a mechanism that satisfies* Exactness *and* Monotonicity*, given a bidder i, a set s_i of goods and declarations for all other bidders in the game, there exists a* critical value v_c *such that:*

$$v_i < v_c \Rightarrow g_i = \emptyset$$

$$v_i > v_c \Rightarrow g_i = s_i$$

Note that we do not know if i's bid is granted when $v_i = v_c$ and that v_c can be infinite and thus for every v, $g_i = \emptyset$.

Proof: Assume by contradiction our mechanism supports *Exactness* and *Monotonicity*, but a v_c as described above does not exist then either:

1. For a bid v_i by player i, $g_i \neq s$ and $g_i \neq \emptyset$. But this contradicts *Exactness*. Contradiction.

2. For two different possible bids of player i, v_1, v_2: $v_1 < v_2$ and $g_{i_1} = s$, $g_{i_2} = \emptyset$. But this contradicts *Monotonicity*. Contradiction.

□

Definition Critical: $s \subseteq g_i \Rightarrow t_i = v_c$

This has two meanings:

1. The payment for a bid granted to player i does not depend on his bid but on the bids of the other players.

2. The payment equals exactly to the (critical) value below which the bid will not be granted.

Definition Participation: $s \not\subseteq g_i \Rightarrow t_i = 0$

This implies that if you are not granted the goods you bid for, you will not incur any payments.

Lemma 11.9 *In a mechanism that satisfies* Exactness *and* Participation*, a bidder whose bid is denied has a profit of zero.*

Proof:
By *Exactness*, the bidder gets nothing and thus his income is zero. By participation his payment (expenditure) is zero. Thus $profit = income - expenditure = 0 - 0 = 0$.

□

Lemma 11.10 *In a mechanism that satisfies* Exactness, Monotonicity, Participation *and* Critical *a truthful bidder's profit is nonnegative.*

Proof:
If player i's bid is denied, we conclude be lemma 11.9 that i's profit is zero. Assume i's bid is granted and his type is $< s, v >$. Being truthful, i declaration is $d_i =< s, v >$. Thus i is allocated s and his income is v. By lemma 11.8 , since i's bid is granted, $v \geq v_c$. By *Critical*, i's payment is v_c, thus his profit is $v - v_c \geq 0$.

□

Lemma 11.11 *In a mechanism that satisfies* Exactness, Monotonicity, Participation *and* Critical, *a bidder i of type $< s, v >$ is never better off declaring $< s, v' >$ for some $v' \neq v$ than being truthful.*

Proof:
For player i, compare the case i bids truthfully $< s, v >$ and the case he bids $< v', s >$. Let g_i be the goods he recieves for $< s, v >$ and g'_i be the goods he recieves for $< s, v' >$. There are three cases:

1. If i's bid is denied for $< s, v' >$ (thus $g' \neq s$), then by lemma 11.9, his profit is zero and by lemma 11.10 his profit for $< s, v >$ is nonnegative and the claim holds.
2. Assume i's bid is granted both for $< s, v' >$ and $< s, v >$ thus $g'_i = s$, $g_i = s$. If both bids are granted then in both cases the player gets goods that he values to be worth v. In both cases the player pays the same payment v_c (by *Critical*).Thus profit is identical in both cases and the player is not better off lying.
3. Assume i's bid is granted for $< s, v' >$ but denied for $< s, v >$ thus $g'_i = s$, $g_i = \emptyset$. It must be that $v \geq v_c \geq v'$. By lemma 11.9, being truthful gives i zero profit. Lying gives him profit $v - v_c \leq 0$.

□

Lemma 11.12 *In a mechanism that satisfies* Exactness, Monotonicity *and* Critical, *a bidder i declaring $< s, v >$ whose bid is granted ($g_i = s$), pays a price t_i where $t_i \geq t'_i$ and t'_i being the price payed for declaring $< s', v >$ where $s' \subseteq s$.*

Proof:
Since $< s, v >$ was granted, by *Monotonicity*, so would $< s', v >$. By *Critical*, the price t'_i payed for $< s', v >$ satisfies: for any $x < t'_i$ the bid $< s', x >$ would be denied. By *Critical*, for any $x > t_i$ the bid would be granted. Thus, it must that $t'_i \leq t_i$.

□

Using the above lemmas we will prove the following central Theorem:

Theorem 11.13 *If a mechanism satisfies* Exactness*,* Monotonicity*,* Participation *and* Critical*, then it is a truthful mechanism.*

Proof:

Suppose player i's type is $< s, v >$, we prove he is not better off declaring $< s', v' >$:
By lemma 11.10 the only case we must consider is when declaring $< s', v' >$ yields positive profit to i and by lemma 11.9 this means that this bid was granted. Assume, therfore that $g'_i = s'$.

1. Assume $s \not\subset s'$. By *SMB* defintion, player i's income is zero (he got the bundle he doesn't want...). Since, by *Critical*, his payment is non-negative, his profit cannot be positive.

2. Assume $s \subset s'$. Being an *SMB* , i's income from s' is the same as from s. By lemma 11.12 it is evident that instead of declaring $< s', v' >$, i would not be worse off declaring $< s', v >$. By lemma 11.11 it is evident that $< s', v >$ is not better off than being truthful, or in other words declaring $< s, v >$.

□

11.5.3 A Truthful Mechanism with Greedy Allocation

We shall now describe a *payment scheme* that used with greedy algorithms of type G_i creates a truthful mechanism for SMB.
The mechanism proposed is for G_2, i.e. sorting bids by $\frac{v_i}{|s_i|}$. This can easily be adapted to G_1,G_3 or any sort of G_3 with a different norm with no added complexity.
The payment computation is done in parallel with the execution of G_i. Each payment calculation takes $O(n)$ and thus computing all the payments is $O(n^2)$. Since G_i takes $O(n \log n)$ the total running time is $O(n^2)$.

Definitions

- $AverageCost_i = \frac{v_i}{|s_i|}$

- $NextBidder(i) : N \longrightarrow N$, returns the first bidder following i (in the the sorted descending list of bids, that is $AverageCost_i \geq AverageCost_{NextBidder(i)}$) whose bid was denied, but would be granted had we removed i from the game. Defined Formally:

  $$NextBidder(i) = min\{i|i < i, s(i) \cap s(i) \neq \emptyset, \forall l, l < i, l \neq i, l \; granted \Rightarrow s(l) \cap s(i) = \emptyset\}$$

- *Greedy Payment Scheme* (GPS). Let L be the sorted list created by G_i:

1. If $g_i = s_i$, i pays $AverageCost_{NextBidder(i)} \times |s_i|$ (if there is no next bidder payment is 0), else:
2. i pays 0.

Proposed Mechanism

Theorem 11.14 *G_i together with GPS comprise a truthful mechanism for the SMB.*

Proof:

We shall prove that G_i together with GPS satisfies *Exactness*, *Motonicity* , *Participation* and *Critical* and use Theorem 11.13 to conclude it is a truthful mechanism:

1. Exactness:
   By definiton of G_i.

2. Motonicity:
   For any G_i and a player i with bids of $< s, v >, < s', v' >$, if $g_i = s$, $s' \subseteq s$ and $v' \geq v$ then bidding $< s', v' >$ would put i in an equal or better place in L and thus $g'_i = s'$ as well.

3. Participation:
   By definiton of GPS.

4. Critical:
   For G_2 with GPS, but similarly for any type of G_i with a similar GPS, if player i bids $< s, v >$ and $g_i = s$ then i pays $AverageCost_{NextBidder(i)} \times |s|$. If i where to bid $< s, v' >$ such that $v' < AverageCost_{NextBidder(i)} \times |s|$ than he would lose the bid since $\frac{v'}{|s|} < AverageCost_{NextBidder(i)}$ and thus be rated below $NextBidder(i)$ in L. Thus the payment of i is equal to the critical value of i.

□

11.5.4 Examples

1. Let us return to the example we used in 11.5.1, but this time for For G_i with GPS:

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

We see the algorithm granted R his bid with a payment of 9.5 which is G's average value, G's bid is denied since some of his goods where allocated to R. B's bid is granted as well with no payment since there is no next player after him in L.

2. Another example of this algorithm at work:

Player	$< s_i, v_i >$	$\frac{v_i}{\lvert s_i \rvert}$	t_i
R	$(\{a\}, 20)$	20	0
G	$(\{b\}, 15)$	15	0
B	$(\{a,b\}, 10)$	10	0

R and G's bids are granted, B's bid is denied. Had R not participated G's bid would still be granted a and B's bid would still be denied, thus his payment is 0. Had G not participated, B's bid would still be denied, thus his payment is 0. In this case the allocation is also the efficient one.

11.6 Single-Unit Combinatorial Auctions

In a *Single-Unit Combinatorial Auction* bidders are interested in buying as many copies of the a single good as offered by the seller. In this case the term *auction* maybe a bit misleading, since the sellers acts more like a shopkeeper that chooses a price tag for the product he is selling without knowing the potential buyers' valuations.

11.6.1 Definitions

1. All buyer valuations of the good are within a given range, thus:

$$\forall i,\ v_i \in [1, w].$$

2. The highest valuation among buyers is denoted by

$$v^* = \max(v_i)$$

11.6.2 Single Copy Auction

In this type of auction only one copy of the good is sold. We construct an algorithm ALG to determine the price tag we will give the product as follows (we are interested, of course, in selling the product for the maximal bidder valuation):

We pick a price tag of 2^i $(0 \leq i \leq \log w)$ with probability of $\frac{1}{\log w}$. We define l such that:

$$2^{l-i} \leq v^* \leq 2^l$$

Effectively we cut the potential tag range into about $\log w$ segments, each segment being twice as wide as the segment preceding it. We randomly choose one of the segments with equal probability and fix the price to be in this segment. OPT, knowing all valuations, will, of course, select a price tag of v^*. Our ALG has $\frac{1}{\log w}$ chance of picking a price tag in the segment containing v^*, a price tag in this segment can be at the worst case equal to $v^*/2$. Thus the expected revenue generated by ALG is bounded from below by $2 \log w$. Thus we get a competitive ratio of:

$$\frac{v^*}{ALG} \leq 2 \log w.$$

11.6.3 Several Copies Auction

Assume several copies of the single product are for sale and they number $\log w$. OPT will always sell all the products for a total revenue of $v^* \log w$ (selling all the products to the buyer with the highest valuation).

Our algorithm, ALG, begins by selling the good with a price of 1 and after every sale we make, we double the price.

We consider the final price tag 2^l, that is the price tag where no willing buyers are left for the product or we run out of stock, and observe two cases:

1. If $2^l \leq v^*$, (actually it is exactly $2^l = v^*$, since this is the only possible way the seller can clear his stock), then the best price we got is no worse than v^*, yielding a competitive ratio of about $\log w$.

2. If $v^* < 2^l$, then exists player j that bought at 2^{l-1}, and so $2^{l-1} \leq v_j \leq v^*$. Thus, the last item sold guarantees the following:

   $$v(ALG) \geq v_j \geq \frac{1}{2}v^*$$

   and since

   $$v(OPT) \leq v^* \log w.$$

   In this case we get a competitive ratio of:

   $$\frac{v(OPT)}{v(ALG)} \leq 2 \log w.$$

11.7 Multi-Unit Combinatorial Auctions

In this part we study multi-unit combinatorial auctions. In a *Multi-Unit Combinatorial Auction* there are n types of goods, for each good i there are k_i copies for sale. We isolate our treatment to auctions where the number of copies of each good are relatively small.

11.7.1 Definitions

- Let U be the set of all possible bundles, thus every member of U is a bundle that may be requested by one of the bidders. Formally:

$$U = \{0, ..., k_1\} \times ... \times \{0, ..., k_n\}$$

- For each bidder j, there exists a valuation function:

$$v_j : U \longrightarrow \Re^+$$

- There exists a lower bound α and and upper bound β. Each bidder desires 0 or at least αk_i and at most βk_i units of good i .

- We simplify the problem by assuming 1 unit exists for each product but players can request fractional amounts of it (bids for each product are in the range $[0, ..1]]$).

- *Demand Oracle* . A demand oracle for valuation v accepts as input a vector of item prices $p_{(1)}, p_{(2)}...p_{(n)}$ and outputs the demand for the items at these prices, i.e. it outputs the vector $\lambda = (\lambda_{(1)}, \lambda_{(2)}, ..., \lambda_{(n)})$ that maximizes the surplus $v(\lambda) - \langle \vec{\lambda}, p \rangle = v(\lambda_{(1)}, \lambda_{(2)}, ..., \lambda_{(n)}) - \sum_i \lambda^{(i)} P^{(i)}$

- *Allocation.* An allocation is a collection of m non-negative vectors $\lambda_1, \lambda_2..., \lambda_m$, where $\lambda_j^{(i)}$ specifies the amount of good i that bidder j has received. An allocation is feasible if for all i, $\sum_j \lambda_j^{(i)} \leq 1$.

- *Value of an allocation.* The value of an allocation A is $V(A) = \sum_j v_j(\lambda_j)$. An allocation is optimal if it achieves the maximal value of any feasible allocation.

- *Direct Relevation mechanism.* A direct relevation mechanism recieves as input a vector of declared valuations $v_1, .., v_m$ and produces as output an allocation $\lambda_1, .., \lambda_m$ and a vector of payments $P_1, .., P_m$, where bidder j recieves λ_j and pays P_j.

- *Incentive Compatibility.* A direct relevation mechanism is inncentive compatible if for every bidder j, every valuation v_j, all declarations of the other bidders v_{-j}, and all possible "false declaratrions" v'_j we have that bidder j's utility with bidding v'_j is no more than his utility truthfully bidding v_j. I.e. Let λ_j and P_j be the mechanism's output with input (v_j, v_{-j}) and λ'_j and P'_j be the mechanism's output with input (v'_j, v_{-j}) then $v_j(\lambda) - P_j \geq v_j(\lambda') - P'_j$

11.7.2 The Online Algorithm

We present an *Online Algorithm* for the problem of *Multi-Unit Auction with Bounded Demand*. The idea of the algorithm is as follows :
At any point in time good i has a price of $P^{(i)}$. The bidders arrive one after the other, and when bidder j is considered he chooses which bundle he prefers according to the current prices. The prices $P^{(i)}$ are initialized to some parameter P_0 and are increased whenever a quantity of that good is allocated. The increase in price is exponential with a rate r per unit allocation.

Formally, the online Algorithm with Parameters P_0 and r is as follows,

- for each good i, $l_1^{(i)} = 0$
- for each bidder $j = 1$ to m
  - for each good i, $P_j^{(i)} = P_0 r^{l_j^{(i)}}$
  - Query j's demand oracle on the current prices and allocate:
    $Demand(P_j^{(1)}, ..., P_j^{(n)}) \longrightarrow (x_j^{(1)}, ..., x_j^{(n)})$
  - determine bidder j's payment as:
    $P_j^{total} = \sum_i x_j^{(i)} P_j^{(i)}$
  - update
    $l_{j+1}^{(i)} = l_j^{(i)} + x_j^{(i)}$

11.7.3 Analysis of Online Algorithm

The correctness of the algorithm involves three elements: incentive compatibility (as defined in previous lectures), validity and approximation ratio.

Lemma 11.15 *For any outer choice of parameters P_0 and r, the online algorithm is incentive compatible.*

The lemma follows from the theorem below:

Theorem 11.16 *A direct relevation mechanism is incentive compatible if and only if for every bidder j and every vector of bids of the other players v_{-j} it:*

1. *fixes a price $p_j(\lambda)$ for every possible allocation λ to bidder j, and whenever bidder j is allocated λ his payment is $p_j(\lambda)$. (Note that $p_j(\lambda)$ does not depend on v_j.)*

2. *allocates to j, λ that maximizes the value of $v_j(\lambda) - p_j(\lambda)$ over all λ that can be allocated to j (for any choice of v_j).*

Proof:

1. We show the two conditions are sufficient. Fix v_{-j} and v_j. Now consider an alternative "lie" v'_j for bidder j. Let λ and p be the mechanism's output for j with input (v_j, v_{-j}) and λ' and p' be the mechanism's output for for j with input (v'_j, v_{-j}). If $\lambda = \lambda'$ then the first condition ensures that $p = p' = p_j(\lambda)$, and thus both allocation and the payments with declaration v'_j are equivalent to those obtained with a truthful bid. If $\lambda \neq \lambda'$, then $p = p_j(\lambda)$, $p' = p_j(\lambda')$, and the second codition ensures that $v_j(\lambda) - p_j(\lambda) \geq v_j(\lambda') - p_j(\lambda')$, and thus the utility with declaration v'_j is less than that obtained with a truthful bid.

2. We show the two conditions are necessary.

   - Assume to the contrary that the first condition does not hold, i.e. that for some v_{-j}, and the valuations v_j and v'_j, the mechanism yields the same allocation λ to player j, but charges different payments $p > p'$, respectively, from him. Now it is clear that for the case where bidders' valuations are v_{-j} and v_j, for bidder j to declare v'_j instead of v_j will imorove his utility (since the allocation remains the same, while the payment decreases), contrary to the definition of incentive compatibility.
   - Now assume the first condition holds, but assume to the contrary that the second condtion doen't, i.e. that for some v_{-j} and valuation v_j, the mechanism allocates λ to j with the property that $v_j(\lambda) - p_j(\lambda) < v_j(\lambda') - p_j(\lambda')$, for some λ' that can be allocated to j, e.g. if he bids v'_j. But this exactly says that for the case where bidders' valuations are v_{-j} and v_j, the for bidder j to declare v'_j instead of v_j will improve his utility (since he is now allocated λ' and charged $p_j(\lambda')$), contrary to the definition of incentive compatibility.

□

Next we prove the validity of the algorithm. i.e. that it never allocates more than the available quantity of each good. This is true as long as the values of P_0 and r satisfy a certain condition. Let $l_j^{(i)} = \sum_{t=1}^{j-1} x_t^{(i)}$ the total allocation of good i to players in $[1..j-1]$. Let $l_*^{(i)} = l_{m+1}^{(i)}$ the total allocation to all players, $l_*^{(i)} \leq 1$. Let $v_{max} = \max_{j,\lambda} v_j(\lambda)$ be the highest valuation in the auction.

Lemma 11.17 *Let P_0, r be such that the condition $P_0 r^\gamma \geq \frac{v_{max}}{\alpha}$ holds, then $l_*^{(i)} \leq \gamma + \beta$ In particular for $\gamma = 1 - \beta$ the algorithm is valid.*

Proof: Assume to the contrary that $l^{(i)}_{j+1} > \gamma+\beta$, and let j be the first player that caused this to happen for some good i, i.e. $l^{(i)}_{j+1} > \gamma + \beta$ since no player is allocated more than β units of each good, we have that $(l^{(i)}_j > \gamma)$. It follows that $(P^{(i)}_j > P_0 r^\gamma \geq \frac{v_{max}}{\beta})$. Since player j is allocated at least α units of good i, his payment is at least $\alpha P^{(i)}_j > v_{max} \geq v_j(x_j)$. Thus player j's payment is more than his valuation for the bundle allocated, in contradiction to the definition of the demand oracle and the possibility of choosing the empty bundle and paying nothing. □

Our final step is to prove a bound on the approximation ratio. For an allocation algorithm A, let V(A) denote the total sum of bidders' valuations for the allocation produced, i.e. $V(A) = \sum_j v_j(x_j)$, where $(x_1, ..., x_m)$ is the allocation produced by A.

We now prove that:

$$V(ALG)(1 + \frac{r^\beta - 1}{\beta}) \geq V(OPT) - nP_0.$$

To get this conjecture we prove some additional lemmas.

Lemma 11.18 *For any j and $\overrightarrow{\lambda_j}$, $v_j(\overrightarrow{x_j}) \geq v_j - \langle\lambda_j, \overrightarrow{P_*}\rangle$, where P_* is the vector of the goods' prices at the end of allocation, $P_* = P^{(1)}_* ... P^{(n)}_*$, and where $P^{(i)}_* = P^{(0)} r^{l^{(i)}_*}$*

Proof: When bidder j is allocated than the inequality

$$v_j(\overrightarrow{x_j}) - \langle\overrightarrow{x_j}, \overrightarrow{P_j}\rangle \geq v_j(\overrightarrow{\lambda_j}) - \langle\overrightarrow{\lambda_j}, \overrightarrow{P_j}\rangle$$

takes place. It derives from definition of demand oracle.
Since $\overrightarrow{P_*} \geq \overrightarrow{P_j}$ for any j, then

$$v_j(\overrightarrow{x_j}) - \langle\overrightarrow{x_j}, \overrightarrow{P_j}\rangle \geq v_j(\overrightarrow{\lambda_j}) - \langle\overrightarrow{\lambda_j}, \overrightarrow{P_*}\rangle.$$

The last inequality holds true since $\overrightarrow{P_*} \geq \overrightarrow{P}$. Since $\langle\overrightarrow{x_j}, \overrightarrow{P_j}\rangle \geq 0$ the lemma holds. □

Corollary 11.19

$$V(ALG) \geq V(OPT) - \sum_i P^i_*$$

Since each bidder pays no more than the value of the bundle he gets, the total revenue is a lower bound for the total valuation. When the j is allocated we have

$$v_j(\overrightarrow{x_j})) \geq \langle\overrightarrow{x_j}, \overrightarrow{P_j}\rangle = \sum_i x^{(i)}_j P_{(0)} r^{l^{(i)}_j}$$

Summing for all bidders we have

$$V(ALG) = \sum_j v_j(x_j) \geq \sum_j \sum_i x_j^{(i)} P_0 r^{l_j^{(i)}} = \sum_i \sum_j x_j^{(i)} P_0 r^{l_j^{(i)}}$$

Let $R^{(i)} = \sum_j x_j^{(i)} P_0 r^{l_j^{(i)}}$ be the total revenue obtained for good i. Let $\Delta_j R^{(i)} = x_j^{(i)} P_0 r^{l_j^{(i)}}$, then $R^{(i)} = \sum_j \Delta_j R^{(i)}$. We denote $h = x_j^{(i)}, t = l_j^{(i)}$, so $\Delta_j R^{(i)} = \sum \Delta_j R^{(i)} = hP_0 r^t$.
Let $\Delta\overline{R^{(i)}}$ be the change when the price grows continuously. We compare this value to

$$\frac{\Delta\overline{R^{(i)}}}{P_0} = \int_t^{t+h} r^x dx = \frac{r^x}{\ln r}\Big|_t^{t+h} = \frac{r^{t+h} - r^t}{\ln r} = \frac{r^t}{\ln r}(r^h - 1).$$

Since the demand of any good is bounded by β we can bound the ratio between $\Delta\overline{R^{(i)}}$ and $\Delta R^{(i)}$, (in other words, bounding the ratio between the continuous and discrete evaluations).

$$\max_{h \leq \beta} \frac{\Delta\overline{R^{(i)}}}{\Delta R^{(i)}} = \max_{h \leq \beta} \frac{\frac{r^t}{\ln r}(r^h - 1)}{hr^t} = \max_{h \leq \beta} \frac{r^h - 1}{h \ln r} = \frac{1}{\ln r} \frac{r^\beta - 1}{\beta}$$

And so

$$R^{(i)} \geq \frac{\beta}{r^\beta - 1}(P_*^i - P_0),$$

where $P_* = r^{t+h}$ and $P_0 = r^t$.
Summing this result over all goods, we achieve the following bound:

Lemma 11.20

$$V(ALG) \geq \sum_j R^{(i)} \geq \frac{\beta}{r^\beta - 1}(\sum P_*^{(i)} - nP_0)$$

$$V(ALG) = \frac{r^\beta - 1}{\beta} + nP_0 \geq \sum_i P_*^{(i)}$$

$$V(ALG) \geq V(OPT) - \sum P_*^{(i)}$$

We obtain compatible, valid and approximation algorithm as long as following two conditions on the parameters P_0 and r hold:

1. $nP_0 \leq \frac{V(OPT)}{2}$
2. $r^{1-\beta} \geq \frac{v_{max}}{\alpha P_0}$.

And so

$$V(ALG)\frac{r^\beta - 1}{\beta} \geq \frac{1}{2}V(OPT).$$

Under these conditions no item is over allocated and the approximation ratio is $C = 2(1 + \frac{r^\beta - 1}{\beta})$

In order to obtain a complete online algorithm we need to choose parameters to our ALG. In our algorithm we choose them before any players arrive. This is possible, only if there exists an a priory known bounds v_{min} and v_{max} such that:

$$v_{min} \leq \max_j v_j(\beta, ..., \beta) \leq v_{max}$$

We will assume this condition holds.

Using the algorithm with $P_0 = \frac{v_{min}}{2n}$ and $r = \frac{v_{max}}{\alpha P_0}$ we achieve $2\frac{\frac{\rho}{2\alpha n}^{\frac{\beta}{1-\beta}} - 1}{\beta}$ - approximation to the optimal allocation, where $\rho = \frac{v_{max}}{v_{min}}$.

11.8 References

1. Mansour Y. ”Topics in Computational Game Theory” Course, Notes from lecture 11 presented at 11/6/2004, Tel Aviv University, Israel.

2. Lehmann D, O’Callaghan LI, Shoham Y. Truth Revelation in Approximately Efficient Combinatorial Auctions. Journal of the ACM, October 19, 2002.

3. Bartal Y, Gonen R, Nisan N. Incentive Compatible Multi Unit Combinatorial Auctions. ACM Press, 2003.