COMBINATORIAL AUCTION

A combinatorial auction is defined as:

- $N = \{1, \dots, n\}$ is the set of players;
- $I = \{1, \dots, m\}$ is the set of items
- $S = P(I) \setminus \emptyset$ is the set of possible bundles of items
- $\Theta_i = \{\theta_{i,s} : \theta_{i,s} \in \mathbb{R}^+, s \in S\}$ is the type of player i, composed of a parameter for every possible bundle
- $K = \{(s,i) : s \in S, i \in N$ and for every s there is at most one $i\}$ is an allocation specifying the set of allocated bundles and for each bundle the winning player
- $v_i(K, \Theta_i) = \sum'_{s:(s,i) \in K} \theta_{i,s}$ is the evaluation function of player i.

The problem of finding the optimal allocation is called COMB AUCTION.

There are $2^{|I|} - 1$ bundles on which each agent can potentially bid.
Is the complexity of this problem just related to the size of the input?
We restrict to SINGLE-MINDED bidders.

- A player is said single-minded when $\theta_{i,s} > 0$ for only one $s \in S$

<u>Theorem</u>: Even when players are single-minded the decision version of COMB-AUCTION, i.e., the problem of deciding whether there is an allocation K with a value $\sum'_{i \in N} \sum'_{s:(s,i) \in K} \theta_{i,s} \geq T$ is NP-hard.

What can we say about approximability? (6.1)

Theorem: There is no polynomial-time approximation algorithm of the COMB AUCTION problem with an approximation ratio better than $\Omega\left(\frac{1}{n}\right)$ and $\Omega\left(\frac{1}{\sqrt{m}}\right)$ unless $P = NP$.

Proof: Approximation preservation from Independent Set.
Any instance of IS can be formulated as an instance of CA
Therefore, any α-approximation of CA provides an α-approximation of IS.
But IS does not admit any poly-time approximation algorithm better than $\Omega\left(\frac{1}{n}\right)$, so the same holds for CA.
For CA, by construction of the proof, $m = O(n^2)$, being the graph undirected.
So, CA does not admit any poly-time approximation algorithm better than $\Omega\left(\frac{1}{\sqrt{m}}\right)$

APPROXIMATION ALGORITHM FOR CA

ApxCombAuction develops in the following steps:
1) Sort all the players on the basis of their unique non-zero $\theta_{i,S}$, in decreasing order in $\frac{\theta_{i,S}}{\sqrt{|S|}}$
2) For each player i allocate the bundle S such that $\theta_{i,S} > 0$ to i if S does not contain any item in some bundles previously allocated
3) Return the allocation found

Complexity: $O(n \log n)$

Theorem: Algorithm ApxCA provides a worst case approximation ratio of $\Omega\left(\frac{1}{\sqrt{m}}\right)$ of the optimal solution of COMB AUCTION.

Def: An allocation function $k(\theta)$ is weakly monotonic if: $\theta_i > \theta_i' \Rightarrow p_i(k(\theta_i, \theta_{-i})) \geq p_i(k(\theta_i', \theta_{-i}))$, $\forall i \in N$, $\forall \theta_i \in \Theta_i$, $\forall \theta_{-i} \in \Theta$

Theorem: The allocation function defined by ApxCombAuction is weakly monotone.

Proof: θ_i denotes the only $\theta_{i,S}$ s.t. $\theta_{i,S} > 0$
$p_i(k(\theta_i, \theta_{-i}))$ has binary values: it's equal to 1 if player i is selected by ApxCombAuction
If the bid of i is selected for $\theta_i = \overline{\theta_i}$, then is selected also for $\theta_i > \overline{\theta_i}$ (the position of i in the order reduces monotonically as θ_i increases)
If i is selected for $\theta_i = \overline{\theta_i}$, there is no player J preceding i s.t. the bundle of the bid of J contains some items contained in the bundle of i. So, if the position of i reduces, the bid of i is still selected.
Similarly it can be shown that if the bid of i is not selected for $\theta_i = \overline{\theta_i}$ is not selected neither for θ_i

Myerson payments for Combinatorial Auction.

Let $N' \subseteq N$ be the subset of players whose bids are in the allocation returned by ApxCombAuction. A simple algorithm to compute Myerson payments is the following:

1) if $i \notin N'$, $p_i(\theta) = 0$;
2) otherwise
   - extract the subproblem of COMB-AUCTION in which player i is removed;
   - let s_i be the bundle s.t. $\theta_{i,s} > 0$;
   - apply ApxCombAuction to the subproblem until a bid over s_j with $s_j \cap s_i \neq \emptyset$ is selected. Stop the algorithm;
   - if the algorithm terminated, $p_i(\theta) = 0$;
     otherwise, $p_i(\theta) = \sqrt{|s_i|} \cdot \frac{\theta_{j,s_j}}{\sqrt{|s_j|}}$

Complexity: $O(n)$

Example:

bundles

players

	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
1	3						
2		1					
3			2				
4				4			
5					6		
6						5	
7							7

$\frac{\theta_{1,s}}{\sqrt{s}} = \frac{3}{1} = 3$; $\frac{\theta_{2,s}}{\sqrt{|s|}} = \frac{1}{1} = 1$; $\frac{\theta_{3,s}}{\sqrt{|s|}} = \frac{2}{1} = 2$; $\frac{\theta_{4,s}}{\sqrt{|s|}} = \frac{4}{\sqrt{2}} \approx 2{,}82$;

$\frac{\theta_{5,s}}{\sqrt{|s|}} = \frac{6}{\sqrt{2}} \approx 4{,}24$; $\frac{\theta_{6,s}}{\sqrt{|s|}} = \frac{5}{\sqrt{2}} \approx 3{,}53$; $\frac{\theta_{7,s}}{\sqrt{|s|}} = \frac{7}{\sqrt{3}} \approx 4{,}04$

Players' order: $5 > 7 > 6 > 1 > 4 > 3 > 2$

$K = \{\underset{5}{\{1,3\}}, \underset{2}{\{2\}}\}$: accepts 5, discards 7, discards 6, discards 1, discards 4, discards 3, accepts 2.

$SW(K) = 6 + 1 = 7$

Payments:

$p_1 = p_3 = p_4 = p_6 = p_7 = 0,$

p_5: players' order: $7 > 6 > 1 > 4 > 3 > 2$: accepts 7: $\{1,2,3\} = S_7$: stop

$$p_5 = \sqrt{|S_5|} \cdot \frac{\theta_{7,S_7}}{\sqrt{|S_7|}} = \sqrt{2} \cdot \frac{7}{\sqrt{3}} \approx 5{,}72$$

p_2: players' order: $5 > 7 > 6 > 1 > 4 > 3 > 2$: accepts 5 : $p_2 = 0$

<u>Example:</u>

bundles

players	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
1	10						
2		12					
3			5				
4				15			
5					11		
6						7	
7							13

$\frac{\theta_{1,S_1}}{\sqrt{|S_1|}} = \frac{10}{1} = 10$; $\frac{\theta_{2,S_2}}{\sqrt{|S_2|}} = \frac{12}{1} = 12$, $\frac{\theta_{3,S_3}}{\sqrt{|S_3|}} = \frac{5}{1} = 5$, $\frac{\theta_{4,S_4}}{\sqrt{|S_4|}} = \frac{15}{\sqrt{2}} \approx 10{,}61$,

$\frac{\theta_{5,S_5}}{\sqrt{|S_5|}} = \frac{11}{\sqrt{2}} \approx 7{,}78$, $\frac{\theta_{6,S_6}}{\sqrt{|S_6|}} = \frac{7}{\sqrt{2}} \approx 4{,}95$, $\frac{\theta_{7,S_7}}{\sqrt{|S_7|}} = \frac{13}{\sqrt{3}} \approx 7{,}51$

players' order: $2 > 4 > 1 > 5 > 7 > 3 > 6$

$K = \{\underset{\substack{\downarrow \\ 2}}{\{2\}}, \underset{\substack{\downarrow \\ 1}}{\{1\}}, \underset{\substack{\downarrow \\ 3}}{\{3\}}\}$: accepts 2, discards 4, accepts 1, discards 5, discards 7, accepts 3

$sw(K) = 12 + 10 + 5 = 27$

Payments:

$p_4 = p_5 = p_6 = p_7 = 0$

p_2: players' order: $4 > 1 > 5 > 7 > 3 > 6$: accepts 4: $\{1,2\}$: stop

$$p_2 = \sqrt{|S_2|} \cdot \frac{\theta_{4,S_4}}{\sqrt{|S_4|}} = 1 \cdot \frac{15}{\sqrt{2}} \approx 10{,}61$$

p_1: players' order. $2 > 4 > 5 > 7 > 3 > 6$: accepts 2, discards 4, accepts 5: stop: $p_1 = \sqrt{|S_1|} \cdot \frac{\theta_{5,S_5}}{\sqrt{|S_5|}} = 1 \cdot \frac{11}{\sqrt{2}} \approx 7{,}78$

p_3: players' order: $2 > 4 > 1 > 5 > 7 > 6$: accepts 2, discards 4, accepts 1, discards 5, 7, 6 : $p_3 = 0$

KNAPSACK AUCTION - APPROXIMATION ⊛

APPROXIMATION ALGORITHM

The algorithm ApxKnapsack develops in the following steps:

1) sort all the items in decreasing order in $\frac{w_i}{s_i}$ and relabel items s.t. item 1 is the first in the order and item n is the last.
2) repeatedly add items to I' according to the above order while the capacity constraint is not violated; call $|I'| = n'$.

   $\sum_{i \leq n'} s_i \leq C$ and if $n' < n$, $C < \sum_{i \leq n'+1} s_i$

3) return $\max\left\{ \sum_{i \leq n'} w_i , w_{n'+1} \right\}$

Complexity: $O(n \log n)$

⊛ I = set of items
s_i = size of i-th item
w_i = value of i-th item
C = capacity

N = set of players
Θ_i = space of types of player i ; $\theta_i = w_i$
$K \subseteq I$: set of feasible allocations $\sum_{i \in K} s_i \leq C$

$$v_i(k, \theta_i) = \begin{cases} \theta_i , & \text{if } i \in K \\ 0 , & \text{otherwise} \end{cases}$$

Myerson payments in Knapsack auction: call $k(\theta) \subseteq I$ the allocation returned by ApxKnapsack. A simple algorithm to compute Myerson payments is the following:

- if $i \notin k(\theta)$, $p_i(\theta) = 0$
- • define $J = I \setminus \{i\}$ where I items are sorted in decreasing order in $\frac{w}{s}$
  • apply a logarithmic search to find the minimum $\bar{w}_i \in \left\{ s_i \frac{w_j}{s_j} : j \in J \right\}$ s.t. i is allocated

Complexity: $O(\log n)$

s_i	5	2	7	1	3
w_i	7	4	10	3	5

$C = 12$

$\frac{w_i}{s_i}$: $\frac{7}{5} = 1{,}4$, 2, $\frac{10}{7} \approx 1{,}42$, 3, $\frac{5}{3} = 1{,}6$ $\Rightarrow$ $i_4 > i_2 > i_5 > i_3 > i_1$

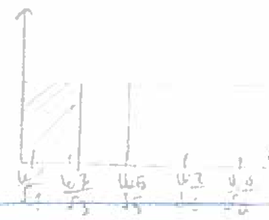
$K^* = \{i_4, i_2, i_5\}$ $\quad sw(K^*) = 12$; $\quad sw(K^{OPT}) = 19$

$p_3 = p_1 = 0$

p_4: $J = \{i_2, i_5, i_3, i_1\}$: $s_4 \cdot \frac{w_2}{s_2} = 2 = \overline{w}_1$: $K = \{i_4, i_2, i_5\}$, $sw(K) = 11$: $11 > 10$ po on;
$\overline{w}_1 = s_4 \cdot \frac{w_5}{s_5} = \frac{5}{3}$: $K = \{i_1, i_5, i_4\}$, $sw(K) = 9 + \frac{5}{3} > 10$; $\overline{w}_1 = s_1 \cdot \frac{w_3}{s_3} = \frac{10}{7}$: $K = \{i_2, i_5, i_3\}$, $sw(K) = 9 + \frac{10}{7} > \frac{10}{7}$: $p_1 = \frac{5}{3}$

p_2: $J = \{i_4, i_5, i_3, i_1\}$: $\overline{w}_2 = s_2 \cdot \frac{w_5}{s_5} = \frac{10}{3}$: $K = \{i_4, i_2, i_5\}$, $sw(K) = 3 + 5 + \frac{10}{3}$ . . . : ue po on;
$\overline{w}_2 = s_2 \cdot \frac{w_3}{s_3} = \frac{20}{7}$: $K = \{i_4, i_5, i_3\}$: $p_2 = \frac{20}{7}$

p_5: $J = \{i_4, i_2, i_3, i_1\}$: $s_5 \cdot \frac{w_3}{s_3} = \frac{30}{7} = \overline{w}_5$: $K = \{i_4, i_2, i_3\}$, $sw(K) = 3 + 5 + \frac{10}{7} > \frac{30}{7}$
$\therefore p_5 = \frac{w_3}{s_3} + \varepsilon$



s_i	1	5	3	4
w_i	15	10	9	5
$\frac{w_i}{s_i}$	15	2	3	5/4

$C = 8$
$sw(K^{OPT}) = 29$

$i_1 > i_3 > i_2 > i_4$: $K^* = \{i_1, i_3\}$: $sw(K^*) = 15 + 9 = 24$

$p_2 = p_4 = 0$

p_1: $J = \{i_3\}$ $\overline{w}_1 = s_1 \cdot \frac{w_3}{s_3} = 1 \cdot 3 = 3$: $K = \{i_1, i_3\}$ po on; $\overline{w}_1 = s_1 \cdot \frac{w_2}{s_2} = 2$: $i_3 > i_1 \geq i_2 > i_4$:
$K = \{i_3, i_2\}$: Kon, $p_1 = 3$

p_3: $J = \{i_1\}$ $s_3 \cdot \frac{w_2}{s_2} = 3 \cdot 2 = 6$: $i_1 > i_2 \geq i_3 > i_4$: $K = \{i_1, i_2\}$: $p_2 = \frac{w_2}{s_2} + \varepsilon$