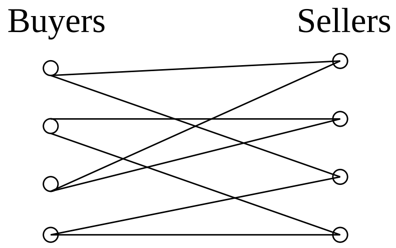
Lecture 17
Multi Item auction

October 09, 2002 transcribed by Dhan Mahesh

In this class we would be discussing Multi Item Auction. There are n buyers and m sellers each buyer wants to buy only one item out of the many available. Each seller has a single item to sell.



One of the scenarios is a market of second-hand cars where each seller brings a car for sale. Buyers are interested in at most one car each. Buyers have preferences over cars. Also depending on the price of the car. So each buyer will rank car and assigns some value to it.
For example:

Car Model	Value (in Lakhs)	Price (in Lakhs)
Maruti 800	1.0	0.5
Lancer	5.0	1.0

Which car would a buyer prefer? How would his/her preference be represented?

$v_{i,j}$ stands for the maximum a buyer i is willing to pay for item j.
p_j stands for the price at which item j can be bought.

Buyers may want to maximize their surplus : $v_{i,j} - p_j$ or the profit ratio $v_{i,j}/p_j$

Every seller has a reserve price r_j on item j. The reserve price represents the minimum price that a seller expects for the item and he/she would like to maximize his/her surplus that is given by $p_j - r_j$

If the utilities are quasi linear in money then the social optimal for the economy is to maximize the total surplus. Define variable $x_{i,j}$ which is 1 when buyer i gets item j and 0 otherwise since each buyer can buy only one item.

$$\sum_j x_{i,j} \leq 1$$

Also an item can be bought by atmost one buyer. Therefore

$$\sum_i x_{i,j} \leq 1$$

Surplus for buyer i can be given by $(v_{i,j} - p_j)x_{i,j}$ .The total buyer surplus is given by $\sum_j x_{i,j}(v_{i,j} - p_j)$
Similarly , total seller surplus is given by $\sum_i x_{i,j}(p_j - r_j)$
Therefore the total surplus given by

$$\begin{gathered}\sum_i X_{i,j}(v_{i,j} - p_j) + \sum_j \sum_i X_{i,j}(p_j - r_j) \\ = \sum_i \sum_j X_{i,j}(v_{i,j} - r_j)\end{gathered}$$

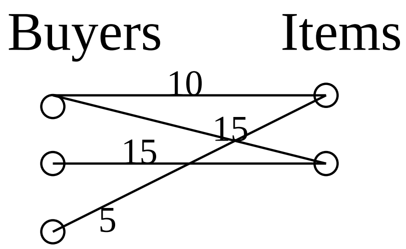
Note that the total surplus is independent of the prices. Therefore the problem of effecient allocation is same as finding maximum weight matching in a bi partite graph matching with edge weights equal to $v_{i,j} - r_j$. This is formally stated as follows:

$$\max_{x_{i,j}} (v_{i,j} - r_j)x_{i,j}$$

$$\textbf{such that } x_{i,j} \in \{0,1\} \ \ \& \sum_i x_{i,j} \leq 1 \ \ \& \sum_j x_{i,j} \leq 1$$

Maximum weighted bipartite matching gives allocation but the problem of how much buyer has to pay to buy that item remains unsolved. Even to get allocation also, one has to get teh maximum price a buyer is willing to pay for an item and its reserve price from seller. Would they give their current prices?

VCG procedure will give solution but let us see an alternative solution.
In the special case when $m = 1$ (there is a single item) ascending english auction leads to efficient allocation.
Final price = Second highest value.
If two items and many buyers: For example, two items and three buyers as below:



Let us assume reserve prices are zero.
Assume open cry ascending auction.
buyer 1 and buyer 3 competes with eachother for item 1 till Rs 5 and then buyer 1 can outbid buyer 3 by quoting some price u such that $5 < u \leq 10$. Similarly in case of item 2, buyer 1 and buyer 2 would be competing for item 2 till Rs 15 and then if buyer 1 gets chance he/she has to decide which item to choose. It is obvious that he/she would choose item 1 and gets it at some u ($5 < u \leq 10$). So buyer 2 would get item 2 at some price y such that $10 < y \leq 15$.
In general what procedure will give us stable and efficient allocation? That procedure should lead to stable matching (Self enforcing)

Bidding Procedure/strategy

General Case: At every step buyer i places a bid on item j such that $v_{i,j} - p_j$ is maximized and then rests until somebody outbids him/her.This procedure goes on like this. What will happen if we go like this (sooner or later, bidding stops as price will come to $v_{i,j}$ So, we are going to get with a matching and price. This procedure converges to (M,P)– (Matching, Price).

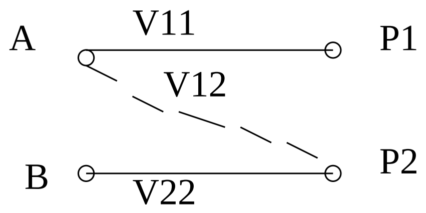
Theorem 0.1 *Ascending Price simultaneous auctions lead to nearly efficient allocations.[Bertsekas]*

Stability:
A situation where when an auction terminates no buyer is interested in switching.

Definition An allocation X of a price $\vec{p}$ to be stable if
$\forall i, j \quad Sb_i + Ss_j \geq v_{i,j} - r_j$ where the Sb_i stand for the surplus of buyer i in the allocation X and Ss_j is the surplus of seller of item j that is

$$v_{i,k} - p_k + p_j - r_j \geq v_{i,j} - r_j$$

If this condition is not satisfied then i & j can deviate from current allocation & be both better of , destabilizing X.
Claim: (M,P) is stable matching.



seller y tries to break up only if he gets more surplus. Buyer A tries to break up only if he gets more surplus.
(A,y) would destabilize the matching iff

$$v_{1,2} - (p_2 + \varepsilon) \geq v_{1,1} - p_1$$

$$\implies v_{1,2} - p_2 > v_{1,1} - p_1$$

Let us see under what conditions we would get stable matching?
After A places bid on x, price of y can increase (or stay unchanged). So his/her surplus on y can't be more than on x at a future time. He/she placed a bid on y of p_i which is less that p_2

$$v_{1,1} - p_1 = v_{1,2} - p_i$$

$$\implies v_{1,2} - p_2 > v_{1,2} - p_i \quad \text{as } p_2 > p_i$$

$$\implies v_{1,1} - p_1 > v_{1,2} - p_2$$

Lets see whether this leads to maximum surplus / efficient allocation ?
Assume number of buyers = number of items. We would be getting a unique stable matching but lots of prices possible.
Lets look at efficient allocation

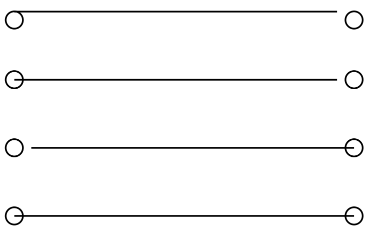


Fig 5(i): Our algo

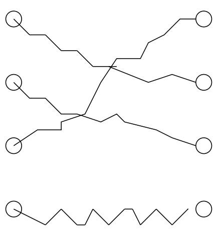
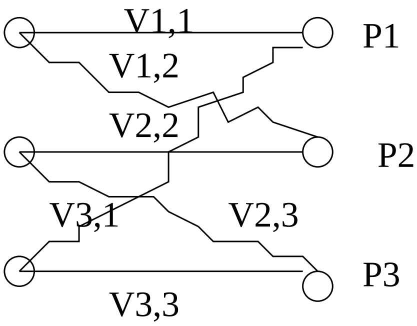


Fig 5(ii): Optimal (given by some other algo)

So, each buyer will have 2 edges and each seller has 2 edges.
So, we would get some disjoint cycles. As there are equal number of buyers and sellers, if we start from any buyer we will come back to that buyer.
Now we have to prove that the sum of the weights picked by our algo is atleast equal to that of Optimal.



$$v_{1,1} - p_1 \geq v_{1,2} - p_2$$

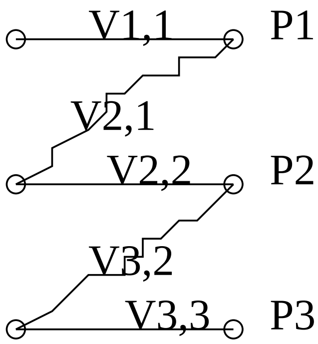
$$v_{2,2} - p_2 \geq v_{2,3} - p_3$$

$$v_{3,3} - p_3 \geq v_{3,1} - p_1$$

add them up.

$$v_{1,1} + v_{2,2} + v_{3,3} \geq v_{1,2} + v_{2,3} + v_{3,1}$$

This is true for all the cycles. Therefore our algo will give the correct maximum weighted bipartite matching.
If not all items are matched.
(assume for 3 buyers and 3 items)



$$v_{1,1} - p_1 \geq v_{1,3} - p_3$$

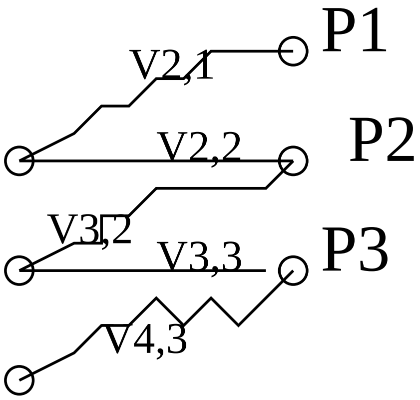
$$v_{2,2} - p_2 \geq v_{2,1} - p_1$$

$$v_{3,3} - p_3 \geq v_{1,3} - p_3$$

add them up

$$v_{1,1} + v_{2,2} + v_{3,3} \geq v_{3,2} + v_{2,1} + v_{1,3}$$

but $v_{1,3} = 0$;
another case:



$$v_{2,2} - p_2 \geq v_{2,1} - p_1$$

$$v_{3,3} - p_3 \geq v_{3,2} - p_2$$

$$0 \geq v_{4,3} - p_3$$

$$v_{2,2} + v_{3,3} \geq v_{2,1} + v_{3,2} + v_{4,3} - p_1$$

$$(v_{2,2} - r_2) + (v_{3,3} - r_3) \geq (v_{2,1} - r_1) + (v_{3,2} - r_2) + (v_{4,3} - r_3) \text{ since } p_1 = r_1$$

Therefore our allocation gives better results.
So we can say that this procedure gives better results which are competitive with respect to buyers and are stable

By VCG:
Take for every item each buyer is willing to pay and all the sellers will sit and give to one who bids the maximum. We exclude that bidder and find the price.
There can be different prices but same stable matching.

This method would be applicable for recruitment (employers and employees).