Lecture 11: Sponsored search

Lecturer: Yishay Mansour *Scribe: Ben Pere , Jonathan Heimann , Alon Levin*

11.1 Sponsored Search

11.1.1 Introduction

Search engines like Google and Yahoo! monetize their services by auctioning off advertising space next to their search results. For example, Apple may bid to appear among the advertisements whenever users search for "ipod". These sponsored results look similar to the search results. Each position in the sponsored results is called **Slot**. Generally, advertisements that appear in a higher ranked slot (higher on the page) garner more attention and more clicks from users. Thus, all else being equal, merchents generally prefer higher ranked slots to lower ranked slots.

Advertisers bid for placement on the page in an auction-style format where the higher their-bid the more likely their adverstiment will appear above other advertisements on the page. By convension, sponsored search advertisers generally *pay per click*, namely, the merchant pays only when the user clicks on the adverstiment.

11.1.2 Formal Model

- n - Number of advertisers (bidders)
- $k \leq n$ - Number of slots
- v_i - Value of click for the ith advertiser

Goal
Find a maxmimum matching between advertisers and slots and set prices so that the mechanism is *strategy-proof* (*incentive-compatible*)

11.1.3 Solution: VCG

- i - Bidder (advertiser)
- j - Item (slot)

- α_{ij} - Probability that a user will click on jth slot when it is occupied by advertiser i, also known as the *Click Through Rate (CTR)*. We assume that $\forall j : \forall i, i' : i \neq i' \Rightarrow \alpha_{ij} = \alpha_{i'j}$. Meaning CTR is independent on which advertiser occupies it. We'll denote the jth slot's CTR as α_j.
- β_i - The click through rate for advertiser i (independant from slot)
- We denote $v_{ij} = v_i \beta_i \alpha_j$

Matching
In order to maximize the social welfare we need to match advertisers with spots:

- We denote $v'_i = v_i \beta_i$
- $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_k$
- We sort v'_i and match them with the corresponding α_i

VCG Prices
In general, **the price of the ith slot to the ith bidder** is the total price if the ith bidder didn't exist (hence didn't win any slot), minus the total price subtracting the ith bidder part ($v'_i \alpha_i$).

Formally:

$$(\sum_{j=1}^{i-1} v'_j \alpha_j + \sum_{j=i+1}^{k} v'_j \underline{\alpha_{j-1}}) - (\sum_{j=1}^{i-1} v'_j \alpha_j + \sum_{j=i+1}^{k} v'_j \underline{\alpha_j}) = \sum_{j=i+1}^{k} v'_j (\alpha_{j-1} - \alpha_j)$$

Example

We'll calculate the prices for advertisers x, y, z in figure 11.1. We'll first notice that $\alpha_a > \alpha_b > \alpha_c$ and $v'_x > v'_y > v'_z$ hence we match x, y, z with a, b, c respectively.

Prices Calculation:
We'll denote P_i the price of advertiser i

$$P_x = (0 + v'_y \alpha_a + v'_z \alpha_b) - (0 + v'_y \alpha_b + v'_z \alpha_c) = (20 + 5) - (10 + 2) = 25 - 12 = 13$$

$$P_y = (v'_x \alpha_a + v'_z \alpha_b) - (v'_x \alpha_a + v'_z \alpha_c) = (30 + 5) - (30 + 2) = 35 - 32 = 3$$

$$P_z = (v'_x \alpha_a + v'_y \alpha_b) - (v'_x \alpha_a + v'_y \alpha_b) = (30 + 10) - (30 + 10) = 40 - 40 = 0$$

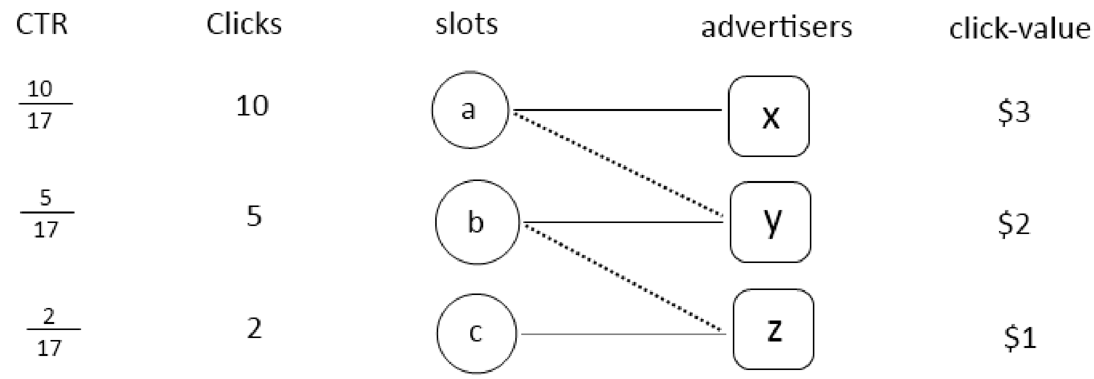

Figure 11.1: 3 advertisers x, y, z and 3 slots a, b, c

11.1.4 Generated Second Price Auction (GSP)

- The search engine assigns a weight β_j to each advertiser. The weight can be thought of as a relevance or quality metric. Through the lecture we've assume for simplicity $\forall j : \beta_j = 1$. Historically, the sponsored mechanism was introduced by Overture called *"rank by bid"*.
- Each bidder i bids b_i
- Score for the ith bidder is $b'_i = b_i \beta_i$
- We sort the bidders by their scores: $b'_1 \geq b'_2 \geq ... \geq b'_n$
- Now the bidders are sorted so that bidder i obtains slot i
- Payment for a click in slot i is

$$P_i = \frac{b_{i+1}\beta_{i+1}}{\beta_i} \leq b_i$$

GSP Strategy-Proofness

Claim 11.1 *GSP is not Strategy-Proof*

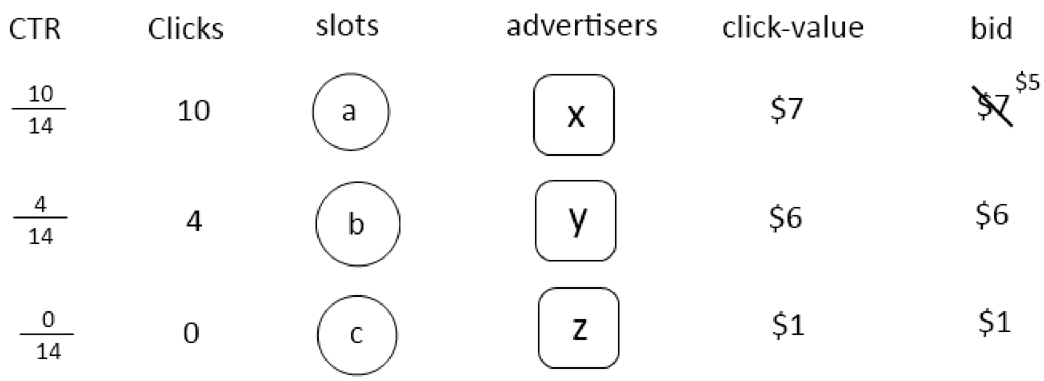

Figure 11.2: 3 advertisers x, y, z and 3 slots a, b, c where advertiser's x value is higher when he bids a number other than his real click-value

Definition Let g_i to denote the total gain of advertiser i, i.e.,

$$g_i = clicks \cdot (v_i' - b_i')$$

Proof:

Consider figure 11.2. We'll calculate the advertisers' gain in the case where $\forall i : b_i = v_i$, that is, advertisers bid their actual click value:

$$b_x' = b_y = 6 \Rightarrow g_x = 10 \cdot (7 - 6) = 10$$

$$b_y' = b_z = 1 \Rightarrow g_x = 4 \cdot (6 - 1) = 20$$

$$b_z' = 0 \Rightarrow g_z = 0$$

Now we'll change x's bid to be \$5:

$$b_x' = b_z = 1 \Rightarrow g_x = 4 \cdot (7 - 1) = 24$$

We've seen that if x lies about it's click-value (that is, bids differently than its value) his gain increases. Thus, x gains from lying, and therefore GSP mechanism is not *strategy-proof.* $\square$

Example - Existance of multiple equilibria under GSP

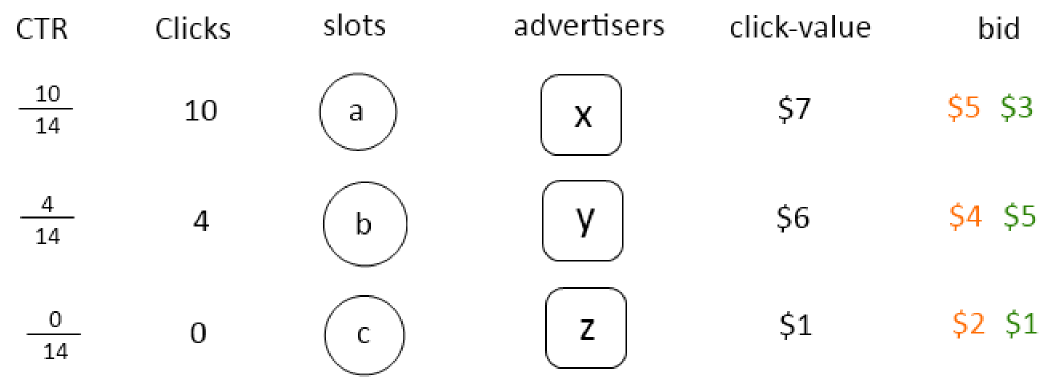

Figure 11.3: 3 advertisers x, y, z and 3 slots a, b, c, the orange bids and the green bids form 2 different equilbria under GSP

We'll now see that there are sets of bids $\{b_i\}$'s forming an equilbrium under GSP.

Consider the green and orange bid sets in figure 11.3. We'll see that both are an equilibrium under GSP.

Claim 11.2 *The orange bid set is an equilbrium under GSP*

Proof: We'll calculate the gains:

$$g_x = 10 \cdot (7 - 4) = 30$$

$$g_y = 4 \cdot (6 - 2) = 16$$

$$g_z = 0$$

Changing x's bid to \$3 isn't worthy:

$$g_x = 4 \cdot (7 - 2) = 20 < 30$$

Changing y's bid to so that $b_y > \$5$ isn't worthy:

$$g_y = 10 \cdot (6 - 5) = 10 < 16$$

Thus, the orange bid set is an equilbrium under GSP. □

Claim 11.3 *The green bid set is an equilbrium under GSP*

Proof: We'll calculate the gains:

$$g_x = 4 \cdot (7 - 1) = 24$$

$$g_y = 10 \cdot (6 - 3) = 30$$

$$g_z = 0$$

Changing x's bid so that $b_x > \$5$ isn't worthy:

$$g_x = 10 \cdot (7 - 5) = 20 < 24$$

Changing y's bid to \$2 isn't worthy

$$g_y = 4 \cdot (6 - 1) = 20 < 30$$

Thus, the green bid set is an equilbrium under GSP. □

Payment
Payment is the sum of prices for each advertiser: $\sum_i P_i$
GSP payment in the orange equilbrium:

$$10 \cdot 4 + 4 \cdot 2 + 0 = 48$$

GSP payment in the green equilbrium:

$$4 \cdot 1 + 10 \cdot 3 + 0 = 34$$

VCG Prices:

$$P_x = v'_y(\alpha_a - \alpha_b) + v'_z(\alpha_b - \alpha_c) = 6 \cdot (10 - 4) + 1 \cdot (4 - 0) = 36 + 4 = 40$$

$$P_y = v'_z(\alpha_b - \alpha_c) = 1 \cdot (4 - 0) = 4$$

$$P_z = 0$$

Total VCG Payment is $40 + 4 = 44$.

Summary

$$\frac{\text{GSP Payment in}}{\text{green equilbrium}} < \text{VCG Payment} < \frac{\text{GSP Payment in}}{\text{orange equilbrium}}$$

We found multi equilbria under GSP

11.1.5 Matching Market

Model

- n - buyers
- m - products
- Each buyer wants to buy exactly 1 product
- Social Optimum $\equiv$ Maximum Matching
- Each buyer i has a value v_j^i to item j
- Product j has price p_j
- Buyer i will choose a product out of $\arg\max_j\{v_j^i - p_j\}$

Goal
Set prices that clear the market, i.e. sell all the products to the buyers.

Example

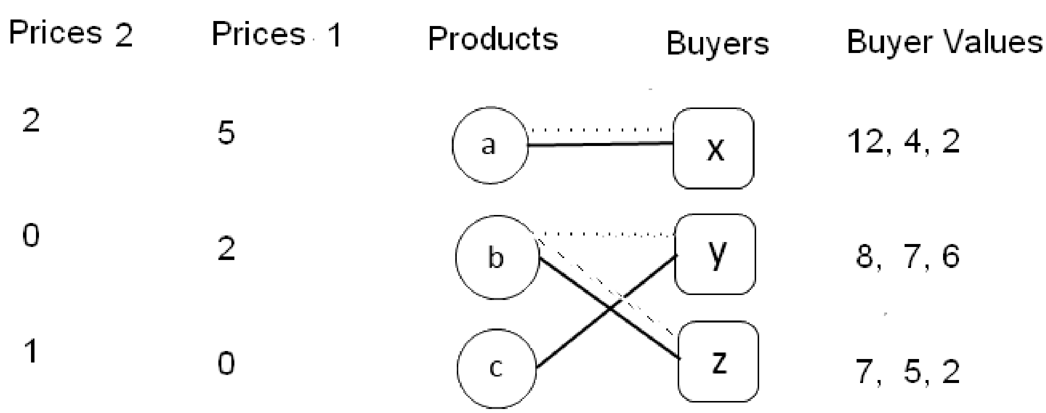

Figure 11.4: 3 buyers x, y, z and 3 products a, b, c, and values of each product for each buyer. Best response for prices 1 marked with solid lines, and for prices 2 with dashed lines. There's a perfect matching in BR graph for prices 1, i.e. those prices clear the market, while prices 2 don't clear the market.

For a set of prices for the products, we define *preferred-product graph*, or *best response graph* on buyers and products by constructing an edge between each buyer and his preferred product. There can be different edges from a buyer to the products, if he's indifferent for their prices. Now it's simple to see that the prices are *market-clearing* iff the resulting preferred-product graph has a perfect matching. We shall show an algorithm which constructs a set of market-clearing prices.

Algorithm for constructing a set of market-clearing prices

1. Set p_j=0
2. Start of the round: at this point we have a set of prices, with smallest price being 0
3. Build BR graph, given the prices
4. Find perfect matching in the graph
5. *If* there is, we're done: the current prices are market clearing
6. *Else* we find a set of buyers S with BR set $N(S)$ which satisfies $|S| > |N(S)|$
7. Increase prices in $N(S)$ by 1
8. *If* the minimal price p is non-zero, decrease each price p_j by p
9. *Goto* 2

Example run of the algorithm

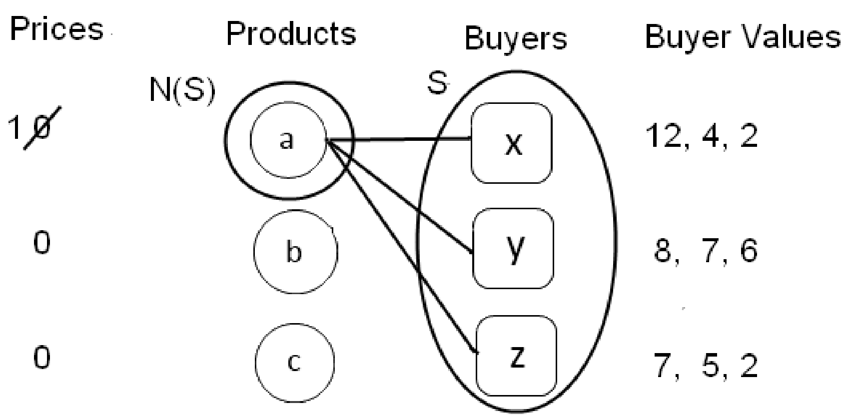

Figure 11.5: Round 1: 3 buyers x, y, z and 3 products a, b, c, and values of each product for each buyer. Starting prices are 0. Best response is graphed. There's no perfect matching, so S is chosen to be $\{x, y, z\}$ with $N(S) = \{a\}$

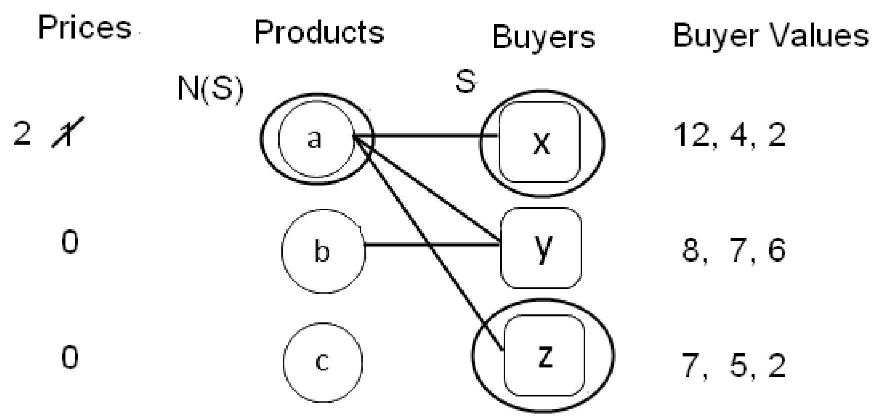

Figure 11.6: Round 2: There's no perfect matching, so S is chosen to be $\{x, z\}$ with $N(S) = \{a\}$

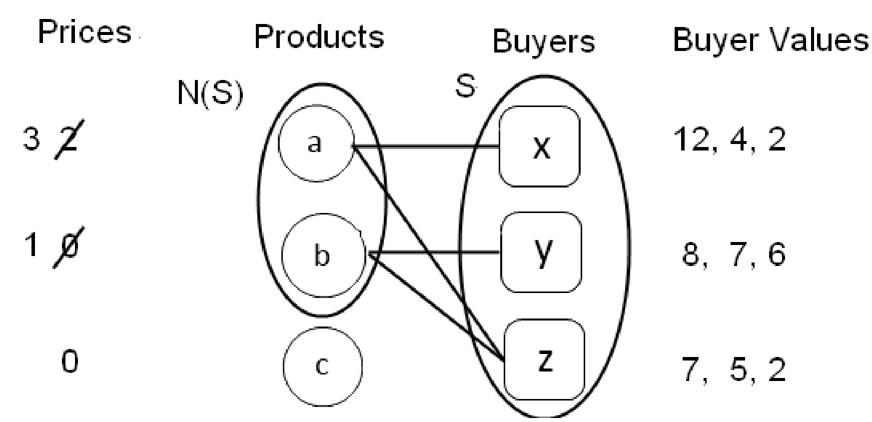

Figure 11.7: Round 3: There's no perfect matching, so S is chosen to be $\{x, y, z\}$ with $N(S) = \{a, b\}$

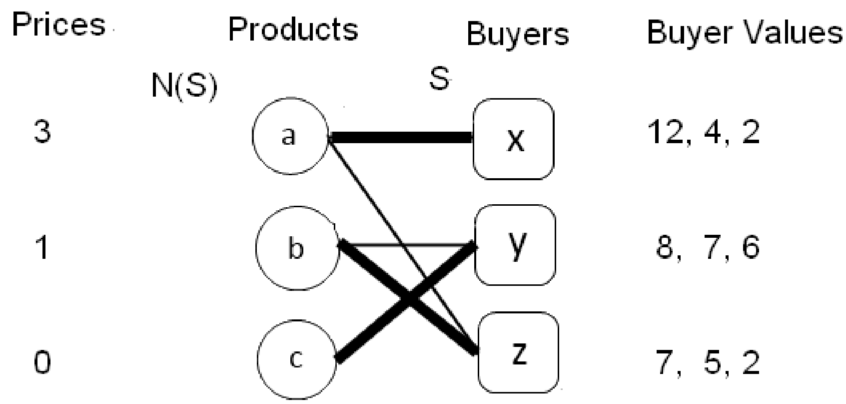

Figure 11.8: Round 4: There is perfect matching, marked by bold lines, so we have found market-cleaning prices.

Hall's theorem claims that if we do not have a perfect matching, then there is a set S, such that $|S| > N(S)$. This implies we will not fail in step 6.

Theorem 11.4 *Hall's theorem: Given a finite bipartite graph $G := (S \cup T, E)$ with two equally sized partitions S and T, a perfect matching exists if and only if for every subset X of S $|X| \leq |N(X)|$, where for a set X of vertices of G, $N(X)$ is the neighborhood of X in G.*

Claim 11.5 *Step 6 in the algorithm is viable*

Proof: Follows directly from Theorem 11.4, since there's such set S with $|S| > |N(S)|$ if and only if there's a perfect matching in G □

Claim 11.6 *The algorithm eventually stops, therefore achieving market-clearing prices.*

Proof: We prove the claim for integer prices and values for simplicity, the proof can be easily extended to the general case. Define potential function ϕ such that $\phi(Product_i) = p_i$, the product price, and $\phi(Buyer_i) = \max_k\{v_k^j - p_k\}$, the buyer's gain. The cummulative potential function Φ is the sum of ϕ's values over all buyers and all products. The starting prices are 0, so ϕ over all products is 0. Initially, $\phi(Buyer_i) = \max_k\{v_k^j\}$ since $p_j = 0$ in the beginning, therefore when the algorithm starts $\Phi = \sum_i \max_k\{v_k^j\}$, which is some finite non-negative integer value.

- The prices are always non-negative, since we decrease them only by the smallest price
- The gain of any buyer is non-negative in step 2, since there's a product with 0 price

- The potential Φ changes only when the prices change

Let Δ_p be the change in the potential of the prices, Δ_B the change in the potential of the products, and $\Delta_\Phi = \Delta_p + \Delta_B$ the change in the potential. We consider the two cases where prices change:
Decrease all prices by p:
$\Delta_p = \sum_{i=1}^{n} \phi(Product_i) = -np$, while $\Delta_B = \sum_{i=1}^{n} \phi(Buyer_i) = np$, so overall $\Delta\Phi = 0$.

Increase prices in $N(S)$ by 1:
$\Delta_p = \sum_{i=1}^{n} \phi(Product_i) = |N(S)|$, while $\Delta_B = \sum_{i=1}^{n} \phi(Buyer_i) = -|S|$, so overall $\Delta\Phi = |N(S)| - |S| < 0$. Since the initial value of Φ is bounded, $\Phi \geq 0$, $\Delta\Phi < 0$ and all the values are integers, the algorithm stops in finite time, reaching a perfect matching in preferred-product graph, and thus finds market clearing prices. $\square$

Theorem 11.7 *Market clearing prices maximize the social gain.*

Proof: Suppose there's a perfect matching in the preferred-product graph, i.e. prices p_i that clear the market. A perfect matching matches buyer i to some product j_i, thus having buyer i recieves payoff $v^i_{j_i} - p_{j_i}$. Given a perfect matching M, summing over all buyers payoff yields $\sum_{i=1}^{n} Payoff(M_i) = \sum_{i=1}^{n} Valuation(M_i) - \sum_{i=1}^{n} p_i$ so $Valuation(M) = Payoff(M) + \sum_{i=1}^{n} p_i$, so choosing M among the perfect matchings that maximizes buyer payoff will maximize the social gain, since the prices are constant. $\square$

11.1.6 Back to GSP

Now we show a way to convert a sponsored search instance into a matching market instance in such way that finding market clearing prices will be equivalent to finding a GSP equilibrium. The sponsored search slots are the matching market products, and the advertisers are the buyers. An advertiser's valuation for a slot is simply the product of it's own revenue per click and the clickthrough rate of the slot. Figure 11.9 shows a sponsored search instance translated into a matching market instance: advertiser x has click value 7, thus his item values are his click value multiplied by slot click rate: $7 \cdot 10$, $7 \cdot 4$, $7 \cdot 0$, and the item values for y and z are computed similarly. We chose market clearing prices 40, 40 and 0, which impose matching between advertisers and slots. The costs per click are the slot price divided by click rate - $\frac{40}{10} = 4$ for x, $\frac{4}{4} = 1$ for y and 0 for z. Player y bids x's cost per click $b_y = 4$, player z bids y's cost per click $b_z = 1$, and x bids some bid larger than y's, i.e., $b_x > b_y$. This is a GSP equilibrium.

General claim on GSP equilibrium construction:
Given advertiser's click values and slot click rates, we find market clearing prices $p_1, ..., p_n$. The matching maps j-th advertiser to $slot_j$ with price p_j. The price is for all the clicks

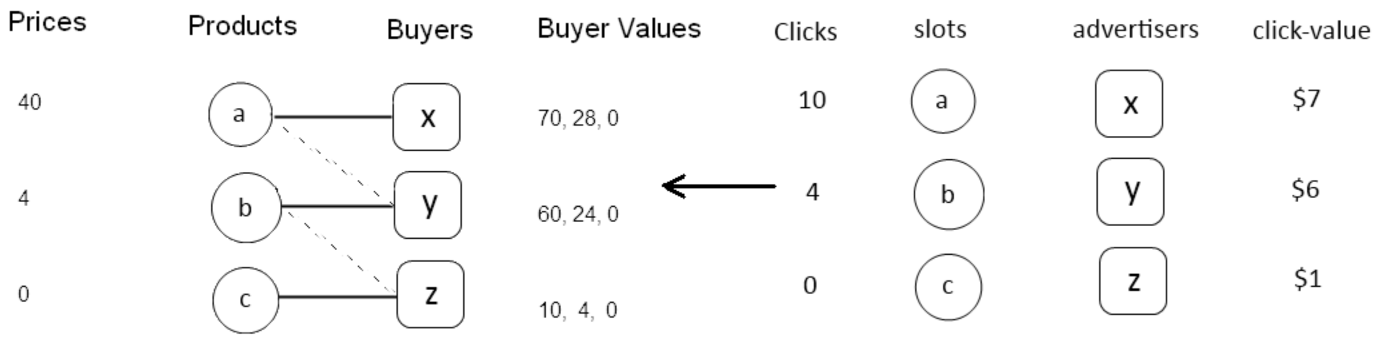

Figure 11.9: 3 advertisers x, y, z and 3 slots a, b, c, with click values for advertisers and click rates for slots. Constructing a matching market instance with market clearing prices imposes matching between advertisers and slots.

- Bids definition: for each $slot_j$ define cost per click cost $p_j^* = \frac{p_j}{\alpha_j}$

Claim 11.8 $p_1^* \geq p_2^* \geq ... \geq p_n^*$

Proof: Prices p_j clear the market. Consider buyer i and compare to place $l < i$. In i-th place the gain is $v_i - p_i^*$ per click, $\alpha_i(v_i - p_i^*)$ for all clicks. In l-th place - $\alpha_l \geq \alpha_j$, since l-th place has higher clickthrough rate. Since buyer i preferrs i-th place, his gain is larger than in l-th place - $v_i - p_l^* \leq v_i - p_i^* \Rightarrow p_i^* \leq p_l^*$ □

Now, given prices $p_1^*, ..., p_n^*$ we define the GSP bids. $i+1$-th buyer has $bid_{i+1} = p_i^*$, and 1-st buyer has $bid_1 = p_1^*$.

Claim 11.9 *The bids are a GSP equilibrium.*

Proof: Consider any change of bids:

Lowering a bid

If player i lowers his bid he will be between some player j and $j+1$ with $i < j$ and will pay j-th player price. It won't increase i-th player gain, since the prices clear the market.

Raising a bid

Player i raises his bid and get between some player j and $j-1$ with $i > j$ and pays $p_{j-1} \geq p_j$, which is higher than i-th player original price. □

11.1.7 GSP and VCG prices

- VCG price: $p_i = \sum_{j=i+1}^{k} v_j(\alpha_{j-1} - \alpha_j)$
- GSP bid: $bid_{i+1} = p_i^*$, $bid_1 = p_1^*$

The VCG prices can be rewritten as $p_{i-1} = p_i + v_i(\alpha_{i-1} - \alpha_i)$, so the prices are ascending and $v_i(\alpha_{i-1} - \alpha_i)$ is the marginal price of moving from price p_i to p_{i-1}. Every player can choose the highest price as long as the marginal cost is smaller than his value, and a player can lower his price until the gain is smaller than the marginal cost. This is exactly the VCG prices, therefore these bids are a GSP equilibrium.