Advertising on the Web

Mining of Massive Datasets
Jure Leskovec, Anand Rajaraman, Jeff Ullman
Stanford University
http://www.mmds.org

History of Web Advertising

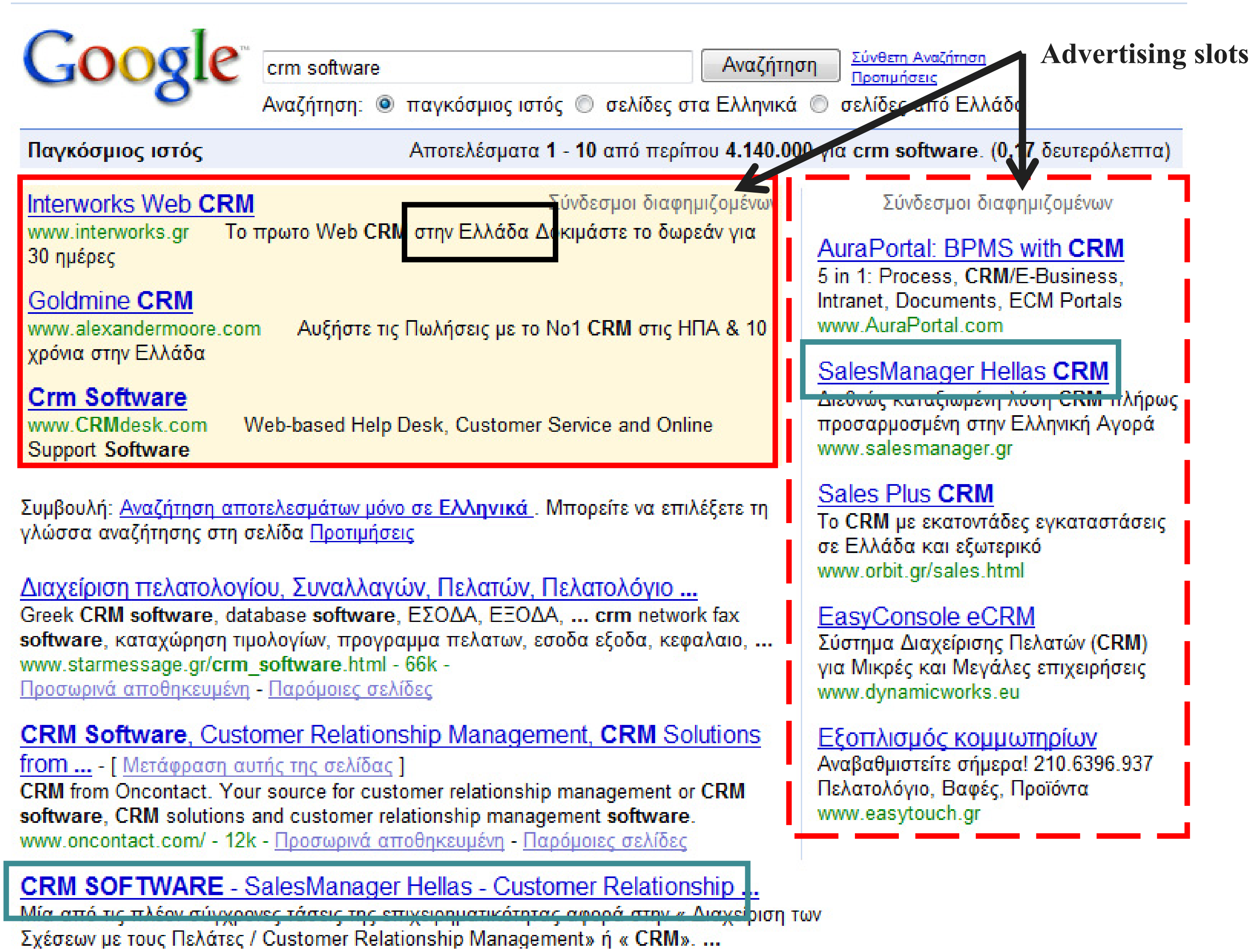
- **Banner ads (1995-2001)**
  - Initial form of web advertising
  - Popular websites charged *X*\$ for every 1,000 "impressions" of the ad
    - Called "**CPM**" rate (Cost per thousand impressions)
    - Modeled similar to TV, magazine ads
  - From **untargeted** to **demographically targeted**
  - **Low click-through rates**
    - Low ROI for advertisers

CPM…cost per *mille*
Mille…thousand in Latin

Performance-based Advertising

- Introduced by Overture around 2000
  - Advertisers **bid** on **search keywords**
  - When someone searches for that keyword, the **highest bidder's ad is shown**
  - Advertiser is charged only if the ad is clicked on
- Similar model adopted by Google with some changes around 2002
  - Called **Adwords**

Sponsored Search - AdWords



Ads vs. Search Results

Web Results **1** - **10** of about **2,230,000** for **geico**. (**0.04** seco

GEICO Car Insurance. Get an auto insurance quote and save today **...**
GEICO auto insurance, online car insurance quote, motorcycle insurance quote, online insurance sales and service from a leading insurance company.
www.**geico**.com/ - 21k - Sep 22, 2005 - Cached - Similar pages
Auto Insurance - Buy Auto Insurance
Contact Us - Make a Payment
More results from www.geico.com »

Geico, Google Settle Trademark Dispute
The case was resolved out of court, so advertisers are still left without legal guidance on use of trademarks within ads or as keywords.
www.clickz.com/news/article.php/3547356 - 44k - Cached - Similar pages

Google and **GEICO** settle AdWords dispute | The Register
Google and car insurance firm **GEICO** have settled a trade mark dispute over **...** Car insurance firm **GEICO** sued both Google and Yahoo! subsidiary Overture in **...**
www.theregister.co.uk/2005/09/09/google_**geico**_settlement/ - 21k - Cached - Similar pages

GEICO v. Google
... involving a lawsuit filed by Government Employees Insurance Company (**GEICO**). **GEICO** has filed suit against two major Internet search engine operators, **...**
www.consumeraffairs.com/news04/**geico**_google.html - 19k - Cached - Similar pages

Sponsored Links

Great Car Insurance Rates
Simplify Buying Insurance at Safeco
See Your Rate with an Instant Quote
www.Safeco.com

Free Insurance Quotes
Fill out one simple form to get
multiple quotes from local agents.
www.HometownQuotes.com

5 Free Quotes. 1 Form.
Get 5 Free Quotes In Minutes!
You Have Nothing To Lose. It's Free
sayyessoftware.com/Insurance
Missouri

Web 2.0

- **Performance-based advertising works!**
  - Multi-billion-dollar industry
- **Interesting problem:**
  What ads to show for a given query?
  - (Today's lecture)
- **If I am an advertiser, which search terms should I bid on and how much should I bid?**
  - (Not focus of today's lecture)

Adwords Problem

- **Given:**
  - **1.** A set of bids by advertisers for search queries
  - **2.** A click-through rate for each **user-advertiser-query** triple
  - **3.** A budget for each advertiser (say for 1 month)
  - **4.** A limit on the number of ads to be displayed with each query
- **Respond to each search query with a set of advertisers such that:**
  - **1.** The size of the set is **no larger than the limit** on the number of ads per query
  - **2.** Each advertiser has bid on the search query
  - **3.** Each advertiser has **enough budget** left to pay for the ad if it is clicked upon

Adwords Problem

- A stream of queries arrives at the search engine: $q_1, q_2, \ldots$
- Several advertisers bid on each query
- When query q_i arrives, search engine must pick a subset of advertisers whose ads are shown

- **Goal: Maximize search engine's revenues**
  - **Simple solution:** Instead of raw bids, use the "**expected revenue per click**" (i.e., **Bid*CTR**)
- **Clearly we need an online algorithm!**

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
A	$1.00	1%	1 cent
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents

Click through rate

Expected revenue

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents
A	$1.00	1%	1 cent

Complications: Budget

- Two complications:
  - Budget Exhaustion
  - CTR of an ad is unknown

- Each advertiser has a limited budget
  - Search engine guarantees that the advertiser will not be charged more than their daily budget

Complications: CTR

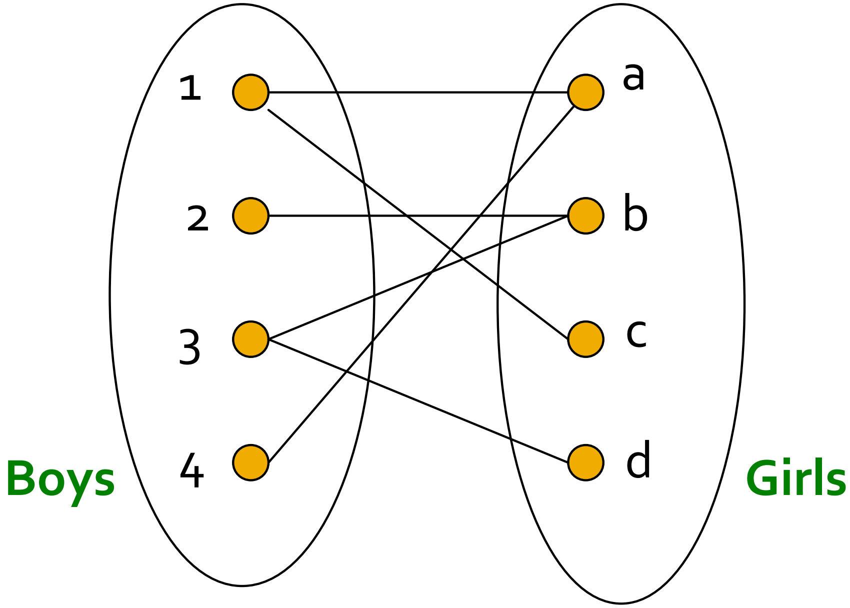
- **CTR: Each ad has a different likelihood of being clicked**
  - **Advertiser 1** bids $2, click probability = 0.1
  - **Advertiser 2** bids $1, click probability = 0.5
  - **Clickthrough rate (CTR)** is measured **historically**
    - **Very hard problem: Exploration vs. exploitation**
      Exploit: Should we keep showing an ad for which we have good estimates of click-through rate
      or
      Explore: Shall we show a brand new ad to get a better sense of its click-through rate

Online Bipartite Matching

Online Algorithms

- **Classic model of algorithms**
  - You get to see the entire input, then compute some function of it
  - In this context, "offline algorithm"
- **Online Algorithms**
  - You get to see the input one piece at a time, and need to make irrevocable decisions along the way
  - **Similar to the data stream model**

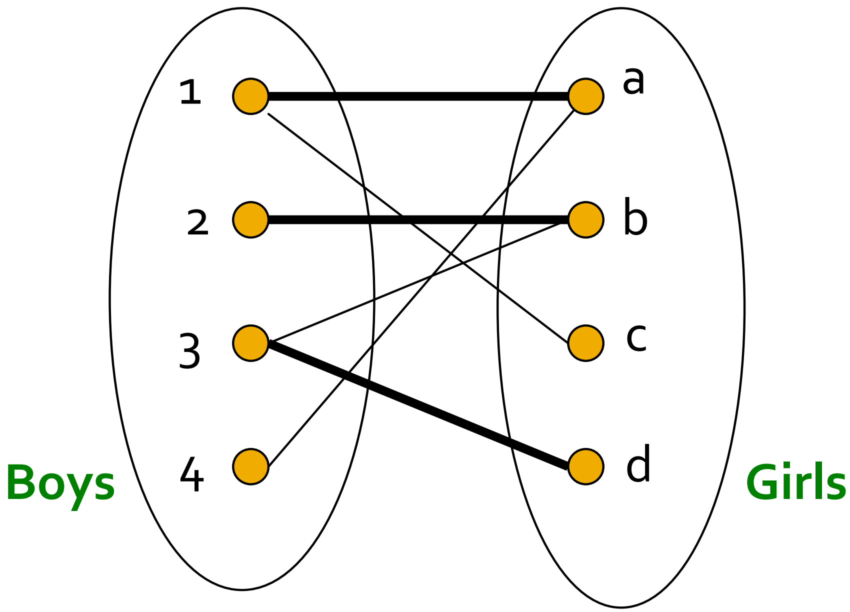
Example: Bipartite Matching



Nodes: Boys and Girls; Edges: Preferences

Goal: Match boys to girls so that maximum number of preferences is satisfied

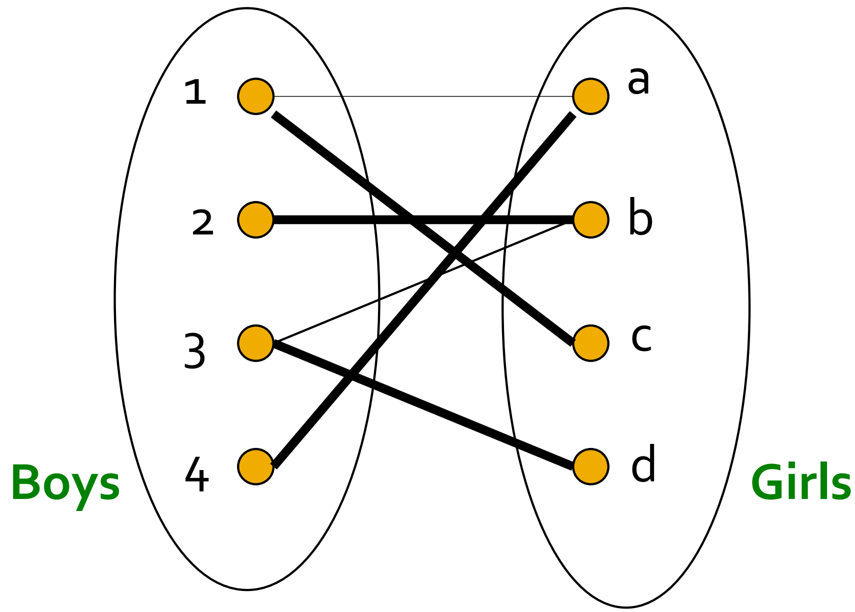
Example: Bipartite Matching



M = {(1,a),(2,b),(3,d)} is a **matching**

Cardinality of matching = |M| = 3

Example: Bipartite Matching



M = {(1,c),(2,b),(3,d),(4,a)} is a
perfect matching

Perfect matching … all vertices of the graph are matched
Maximum matching … a matching that contains the largest possible number of matches

Matching Algorithm

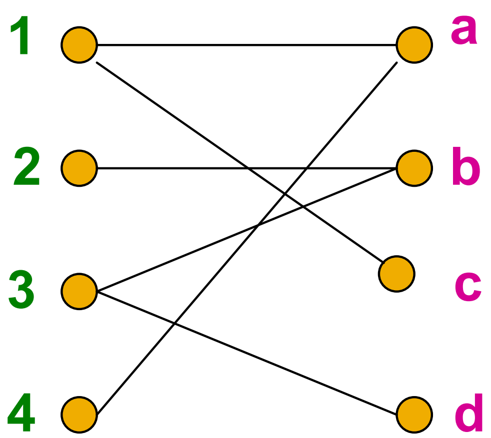
- **Problem: Find a maximum matching for a given bipartite graph**
  - A perfect one if it exists
- There is a polynomial-time offline algorithm based on augmenting paths (Hopcroft & Karp 1973, see http://en.wikipedia.org/wiki/Hopcroft-Karp_algorithm)
- **But what if we do not know the entire graph upfront?**

Online Graph Matching Problem

- Initially, we are given the set **boys**
- In each **round**, **one girl's choices are revealed**
  - That is, girl's **edges** are revealed
- **At that time, we have to decide to either:**
  - Pair the **girl** with a **boy**
  - Do not pair the **girl** with any **boy**
- **Example of application:**

  Assigning tasks to servers

Online Graph Matching: Example



(1,a)
(2,b)
(3,d)

Greedy Algorithm

- **Greedy algorithm for the online graph matching problem:**
  - Pair the new girl with **any** eligible boy
    - If there is none, do not pair girl
- **How good is the algorithm?**

Competitive Ratio

- For input ***I***, suppose greedy produces matching M_{greedy} while an optimal matching is M_{opt}

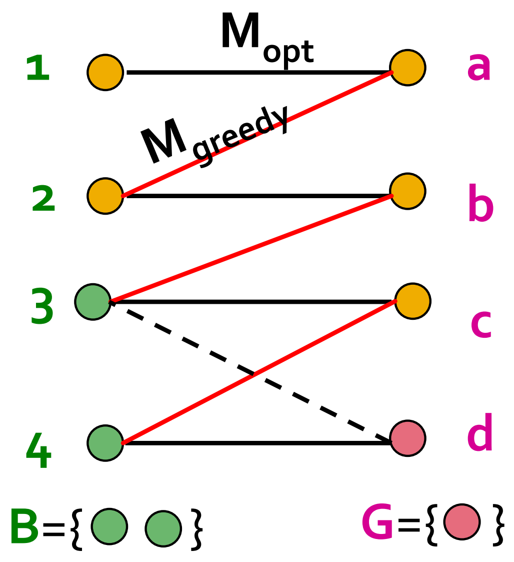
Competitive ratio =

$$\min_{\text{all possible inputs } I} \left(|M_{greedy}| / |M_{opt}|\right)$$

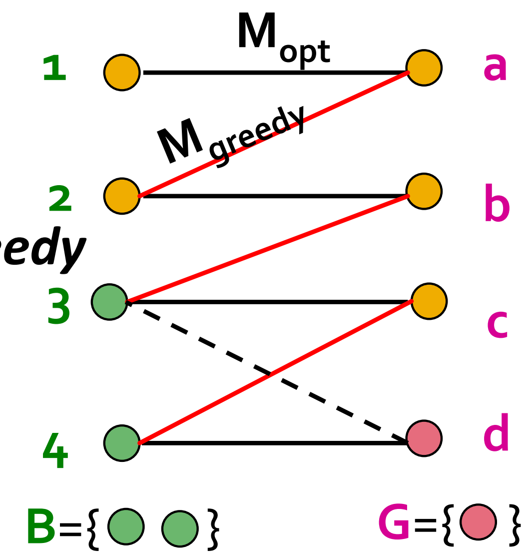
(what is greedy's <u>worst</u> performance <u>over all possible</u> inputs *I*)

Analyzing the Greedy Algorithm

- Consider a case: $M_{greedy} \neq M_{opt}$
- Consider the set G of girls matched in M_{opt} but not in M_{greedy}
- Then every boy B adjacent to girls in G is already matched in M_{greedy}:
  - If there would exist such non-matched (by M_{greedy}) boy adjacent to a non-matched girl then greedy would have matched them
- Since boys B are already matched in M_{greedy} then **(1)** $|M_{greedy}| \geq |B|$

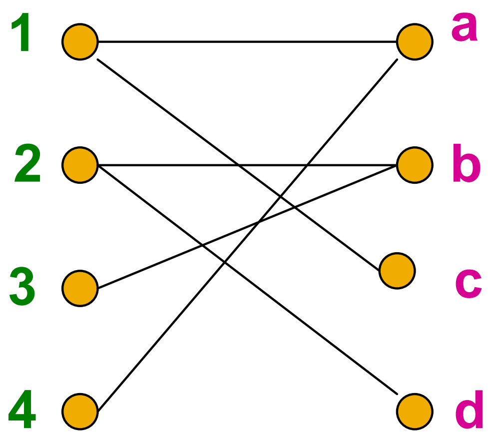


Analyzing the Greedy Algorithm



- **Summary so far:**
  - Girls ***G*** matched in M_{opt} but not in M_{greedy}
  - **(1)** $|M_{greedy}| \geq |B|$
- There are at least $|G|$ such boys ($|G| \leq |B|$) otherwise the optimal algorithm couldn't have matched all girls in ***G***
  - **So:** $|G| \leq |B| \leq |M_{greedy}|$
- By definition of ***G*** also: $|M_{opt}| \leq |M_{greedy}| + |G|$
  - Worst case is when $|G| = |B| = |M_{greedy}|$
- $|M_{opt}| \leq 2|M_{greedy}|$ **then** $|M_{greedy}|/|M_{opt}| \geq 1/2$

Worst-case Scenario



(1,a)
(2,b)

Back to Adwords: Budget Exhaustion

Adwords Problem

- A stream of queries arrives at the search engine: $q_1, q_2, \ldots$
- Several advertisers bid on each query
- When query q_i arrives, search engine must pick a subset of advertisers whose ads are shown

- **Goal: Maximize search engine's revenues**
  - **Simple solution:** Instead of raw bids, use the **"expected revenue per click"** (i.e., **Bid*CTR**)
- **Clearly we need an online algorithm!**

Greedy Algorithm

- **Our setting: Simplified environment**
  - There is **1** ad shown for each query
  - All advertisers have the same budget ***B***
  - All ads are equally likely to be clicked
  - Value of each ad is the same (=**1**)
- **Simplest algorithm is greedy:**
  - For a query pick any advertiser who has bid **1** for that query
  - **Competitive ratio of greedy is 1/2**

Bad Scenario for Greedy

- **Two advertisers A and B**
  - ***A*** bids on query ***x***, ***B*** bids on ***x*** and ***y***
  - Both have budgets of **$4**
- **Query stream: *x x x x y y y y***
  - Worst case greedy choice: ***B B B B*** **_ _ _ _**
  - Optimal: **A A A A B B B B**
  - **Competitive ratio = ½**
- **This is the worst case!**
  - **Note:** Greedy algorithm is deterministic – it always resolves draws in the same way

BALANCE Algorithm [MSVV]

- **BALANCE** Algorithm by [Mehta, Saberi, Vazirani and Vazirani]
  - **For each query, pick the advertiser with the largest unspent budget**
    - Break ties arbitrarily (**but in a deterministic way**)

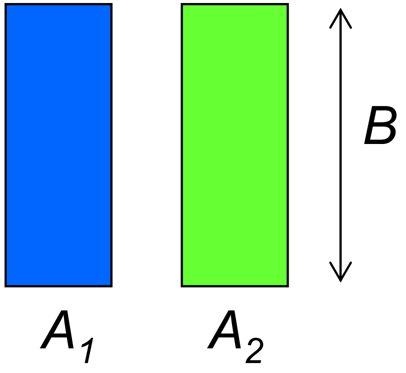
Example: BALANCE

- **Two advertisers A and B**
  - **A** bids on query ***x***, **B** bids on ***x*** and ***y***
  - Both have budgets of **$4**
- **Query stream:** ***x x x x y y y y***
- **BALANCE choice: A B A B B B _ _**
  - Optimal: **A A A A B B B B**
- **In general:** For **BALANCE** on **2** advertisers
  Competitive ratio = ¾

Analyzing BALANCE

- **Consider simple case (w.l.o.g.):**
  - **2** advertisers, A_1 and A_2, each with budget **B** (≥1)
  - Optimal solution exhausts both advertisers' budgets
- **BALANCE must exhaust at least one advertiser's budget:**
  - **If not, we can allocate more queries**
    - Whenever BALANCE makes a mistake (both advertisers bid on the query), advertiser's unspent budget only decreases
    - Since optimal exhausts both budgets, one will for sure get exhausted
  - Assume BALANCE exhausts A_2's budget, but allocates x queries fewer than the optimal
  - **Revenue:** $BAL = 2B - x$

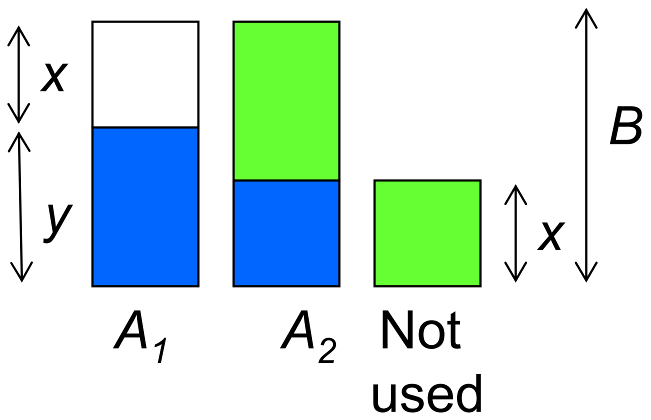
Analyzing Balance



Queries allocated to A_1 in the optimal solution

Queries allocated to A_2 in the optimal solution

Optimal revenue = **2B**
Assume Balance gives revenue **= 2B-x = B+y**



Unassigned queries should be assigned to A_2
(if we could assign to A_1 we would since we still have the budget)

Goal: Show we have $y \geq x$

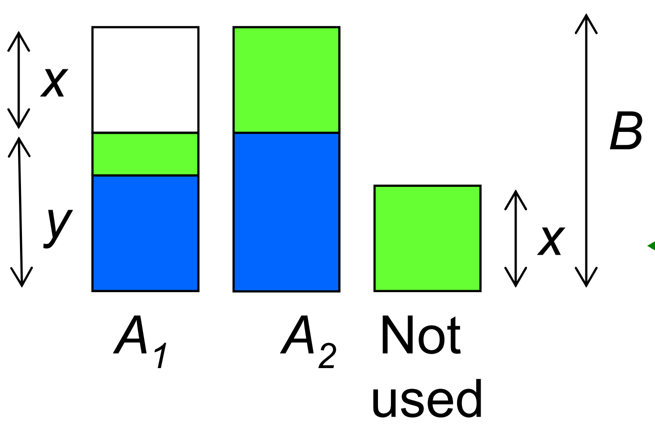
Case 1) $\leq$ ½ of A_1's queries got assigned to A_2
then $y \geq B/2$

Case 2) > ½ of A_1's queries got assigned to A_2
then $x \leq B/2$ and $x + y = B$

Balance revenue is minimum for $x = y = B/2$

Minimum Balance revenue = $3B/2$

Competitive Ratio = 3/4



BALANCE exhausts A_2's budget

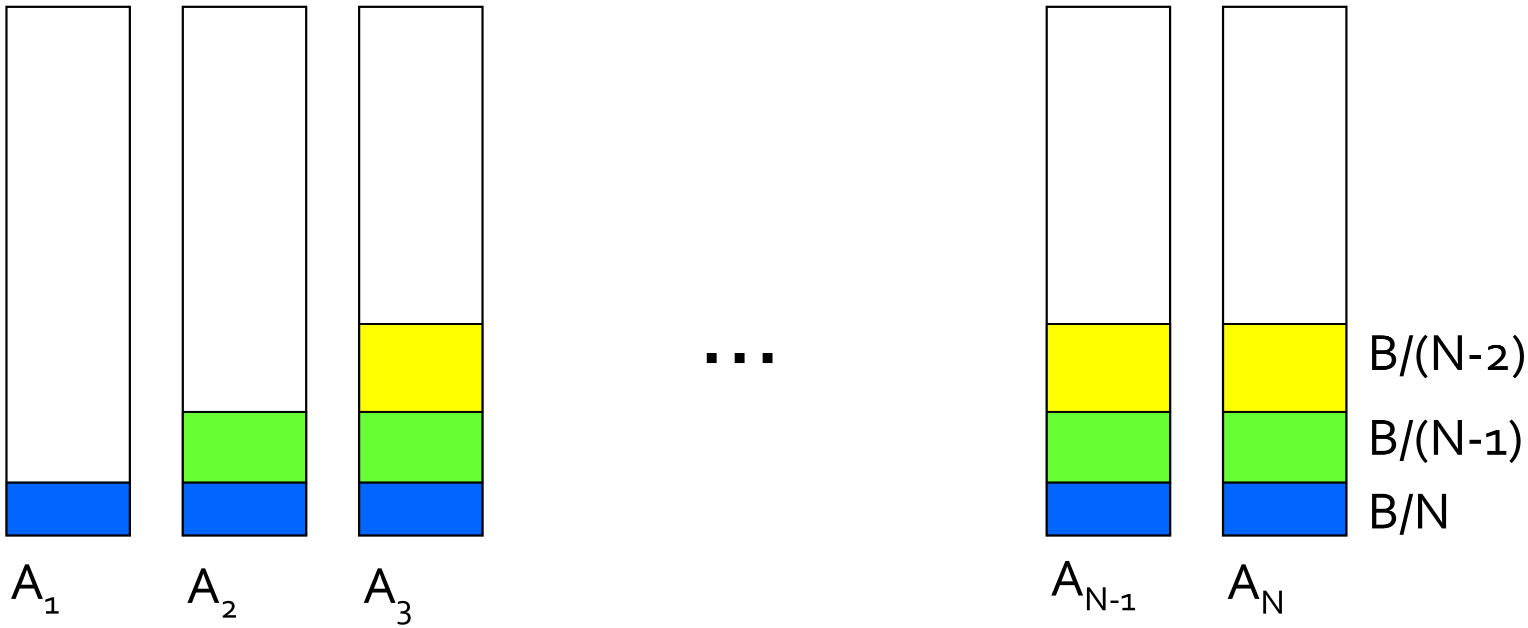
BALANCE: General Result

- **In the general case, worst competitive ratio of BALANCE is 1–1/e = approx. 0.63**
  - Interestingly, no online algorithm has a better competitive ratio!
- **Let's see the worst case example that gives this ratio**

Worst case for BALANCE

- ***N* advertisers:** $A_1, A_2, \ldots A_N$
  - Each with budget $B > N$
- **Queries:**
  - $N \cdot B$ queries appear in N rounds of B queries each
- **Bidding:**
  - Round **1** queries: bidders $A_1, A_2, \ldots, A_N$
  - Round **2** queries: bidders $A_2, A_3, \ldots, A_N$
  - Round ***i*** queries: bidders $A_i, \ldots, A_N$
- **Optimum allocation:**
  Allocate round ***i*** queries to A_i
  - Optimum revenue $N \cdot B$

BALANCE Allocation



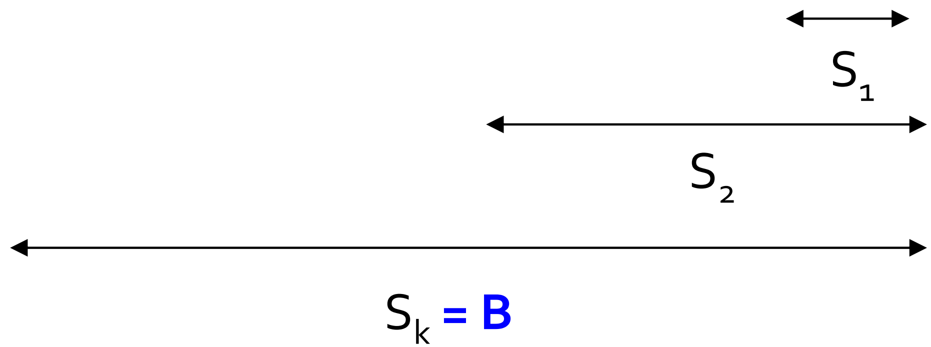
BALANCE assigns each of the queries in round 1 to **N** advertisers.
After *k* rounds, sum of allocations to each of advertisers $A_k,\dots,A_N$ is

$$S_k = S_{k+1} = \cdots = S_N = \sum_{i=1}^{k-1} \frac{B}{N-(i-1)}$$

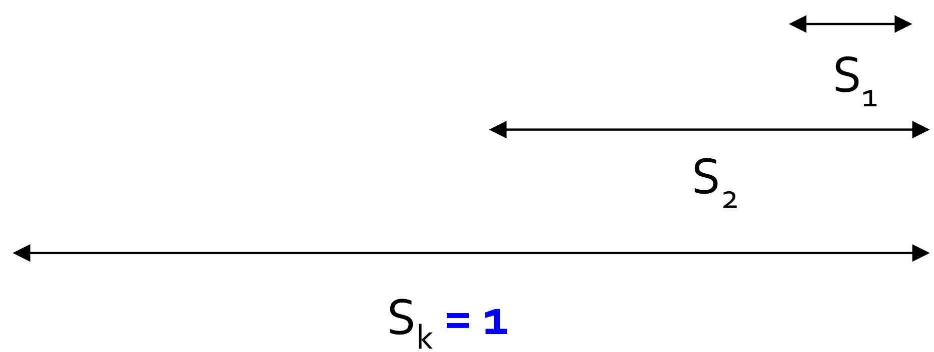
If we find the smallest *k* such that $S_k \geq B$, then after *k* rounds we cannot allocate any queries to any advertiser

BALANCE: Analysis

$B/1 \quad B/2 \quad B/3 \quad \ldots \quad B/(N-(k-1)) \ldots B/(N-1) \quad B/N$



$1/1 \quad 1/2 \quad 1/3 \quad \ldots \quad 1/(N-(k-1)) \ldots 1/(N-1) \quad 1/N$



BALANCE: Analysis

- **Fact:** $H_n = \sum_{i=1}^{n} 1/i \approx \ln(n)$ for large n
  - Result due to Euler

1/1   1/2   1/3   …   1/(N-(k-1)) … 1/(N-1)   1/N

ln(N)

ln(N)-1 $S_k = 1$

- $S_k = 1$ implies: $H_{N-k} = ln(N) - 1 = ln(\frac{N}{e})$
- We also know: $H_{N-k} = ln(N-k)$
- So: $N - k = \frac{N}{e}$
- Then: $k = N(1 - \frac{1}{e})$

N terms sum to ln(*N*).
Last *k* terms sum to 1.
First *N-k* terms sum
to ln(*N-k*) but also to ln(*N*)-1

BALANCE: Analysis

- So after the first $k=N(1-1/e)$ rounds, we cannot allocate a query to any advertiser
- **Revenue = $B \cdot N\,(1-1/e)$**
- **Competitive ratio = $1-1/e$**

General Version of the Problem

- **Arbitrary bids and arbitrary budgets!**
- Consider we have 1 query $\boldsymbol{q}$, advertiser $\boldsymbol{i}$
  - Bid = $\boldsymbol{x_i}$
  - Budget = $\boldsymbol{b_i}$
- **In a general setting BALANCE can be terrible**
  - Consider two advertisers $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$
  - $\boldsymbol{A_1}$: $\boldsymbol{x_1 = 1}$, $\boldsymbol{b_1 = 110}$
  - $\boldsymbol{A_2}$: $\boldsymbol{x_2 = 10}$, $\boldsymbol{b_2 = 100}$
  - Consider we see **10** instances of **q**
  - BALANCE always selects $\boldsymbol{A_1}$ and earns **10**
  - Optimal earns **100**

Generalized BALANCE

- **Arbitrary bids:** consider query ***q***, bidder ***i***
  - Bid = x_i
  - Budget = b_i
  - Amount spent so far = m_i
  - Fraction of budget left over $f_i = 1-m_i/b_i$
  - Define $\psi_i(q) = x_i(1-e^{-f_i})$
- Allocate query ***q*** to bidder ***i*** with largest value of $\psi_i(q)$
- **Same competitive ratio (1-1/e)**

A Primer on Auctions: The Second Price Mechanism and Myerson's Lemma

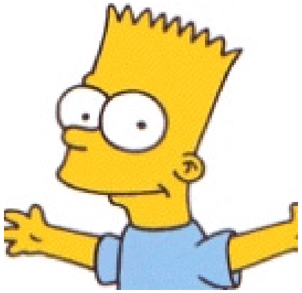
Slides: Vangelis Markakis

(modified for this lecture)

Auctions

- An **auctioneer** with some items (e.g., advertising space) on sale
- A set of **bidders** (e.g., advertisers) express preferences over items
- **Preferences** are submitted through **bids**

Single-Item Auctions



1 indivisible good

Set of players
N = {1, 2, …, n}

Sealed Bid Auctions

- We assume every player has a **valuation v_i** for obtaining the good (click on her ad)
- **Available strategies**: each bidder is asked to submit a bid b_i
  - $b_i \in [0, \infty)$
- The submitted bid b_i may differ from the real value v_i of bidder i

First Price Auction

Auction rules

- Let **b** = $(b_1, b_2, \ldots, b_n)$ the vector of all the offers
- Winner: The bidder with the highest offer
    - Ignore ties (they are broken in arbitrary fixed way).
- Winner's payment: the bid declared by the winner
- Utility function of bidder i,

$$u_i(\mathbf{b}) = \begin{cases} v_i - b_i, & \text{if i is the winner} \\ 0, & \text{otherwise} \end{cases}$$

- Bidding true value in 1PA is not incentive compatible!

Auction Mechanisms

Definition: For the single-item setting, an auction mechanism receives as input the bidding vector **b** = $(b_1, b_2, \ldots, b_n)$ and consists of

- an allocation algorithm (who wins the item)
- a payment algorithm (how much does the winner pay)

Natural mechanisms should satisfy individual rationality:

- Non-winners do not pay anything
- If the winner is bidder i, her payment will not exceed b_i (it is guaranteed that no-one will pay more than what she declared)

Auction mechanisms

Aligning Incentives: Ideally, we would like mechanisms that do not provide incentives for strategic behavior

Definition: A mechanism is called truthful (or strategyproof, or incentive compatible) if for every bidder i, and for every profile $\mathbf{b}_{-i}$ of the other bidders, it is a dominant strategy for i to declare her real value v_i, i.e., it holds that

$$u_i(v_i, \mathbf{b}_{-i}) \geq u_i(b', \mathbf{b}_{-i}) \text{ for every } b' \neq v_i$$

Vickrey (2nd Price) Auction

[Vickrey '61]

- Allocation algorithm: the highest bidder
- Payment algorithm: the winner pays the 2^{nd} highest bid
- Hence, the auctioneer offers a discount to the winner

Observation: the payment does not depend on the winner's bid!

  - The bid of each player determines if he wins or not, but not what he will pay

Vickrey (2nd Price) Auction

[Vickrey '61] (Nobel prize in economics, 1996)

Theorem: The 2nd price auction is a truthful mechanism

Proof sketch:

Fix a bidder i, and let $\mathbf{b}_{-i}$ be an arbitrary bidding profile for the rest of the players

Let $b^* = \max_{j \neq i} b_j$

Consider now all possible cases for the final utility of bidder i, if he plays v_i

- $v_i < b^*$
- $v_i > b^*$
- $v_i = b^*$
- In all these different cases, we can prove that bidder i does not become better off by deviating to another strategy

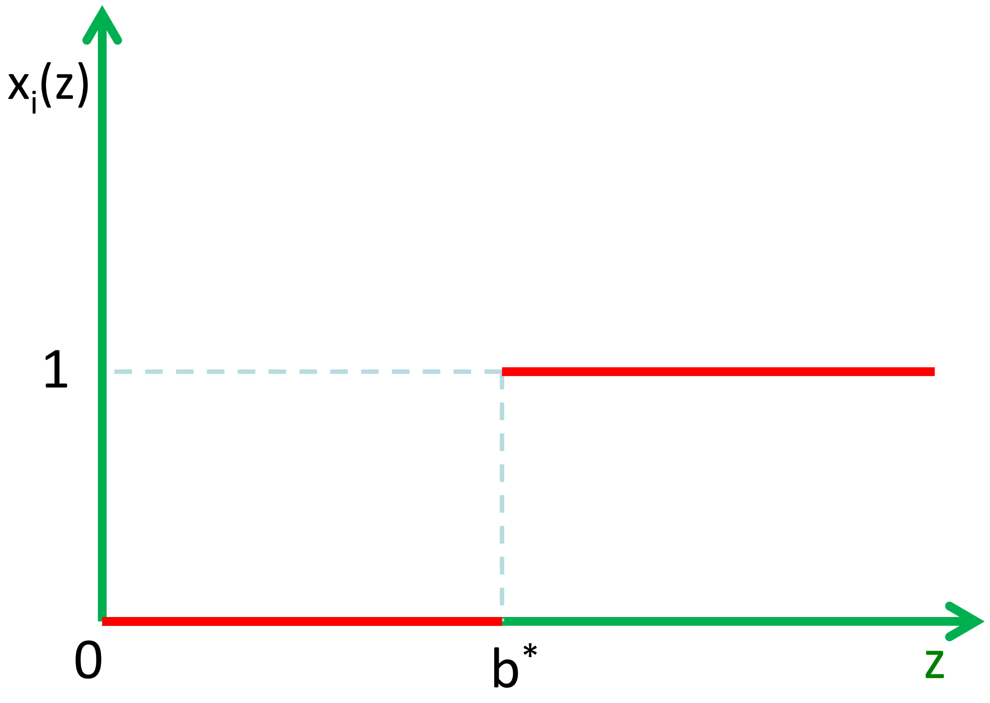
Myerson's Lemma

Definition: An allocation rule is **x** **monotone** if for every bidder i, and bids profile $\mathbf{b}_{-i}$, the allocation $x_i(z, \mathbf{b}_{-i})$ is non-decreasing in z

- Theorem [Myerson '81]:For every single-parameter environment,
  - An allocation rule **x** can be turned into a truthful mechanism if and only if it is monotone
  - If **x** is monotone, then there is a unique payment rule **p**, so that (**x**, **p**) is a truthful mechanism
    - Subject to the constraint that if $b_i = 0$, then $p_i = 0$

Myerson's Lemma and Payment Formula

- For the payment rule, we need to look for each bidder at the allocation function $x_i(z, \mathbf{b}_{-i})$
- For the single-item truthful auction:
  - Fix $\mathbf{b}_{-i}$ and let $b^* = \max_{j \neq i} b_j$



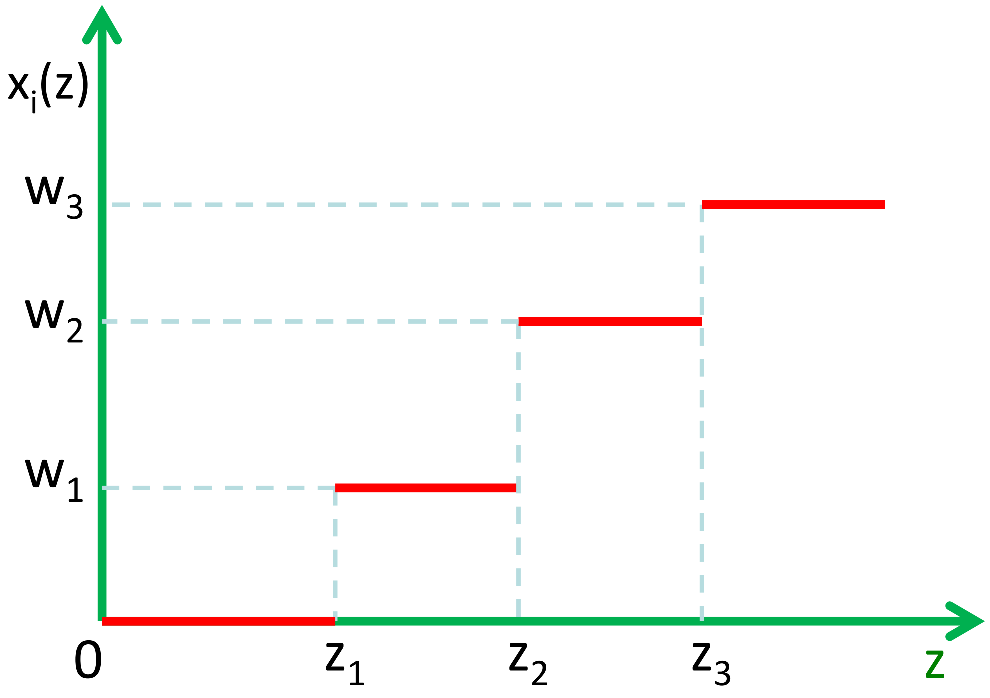
Facts:

- For any fixed $\mathbf{b}_{-i}$, the allocation function is piecewise linear with 1 jump
- The Vickrey payment is precisely the value at which the jump happens
- The jump changes the allocation from 0 to 1 unit

Myerson's Lemma and Payment Formula

For most scenarios of interest

- The allocation is piecewise linear with multiple jumps
- The jump determines how many extra units the bidder wins

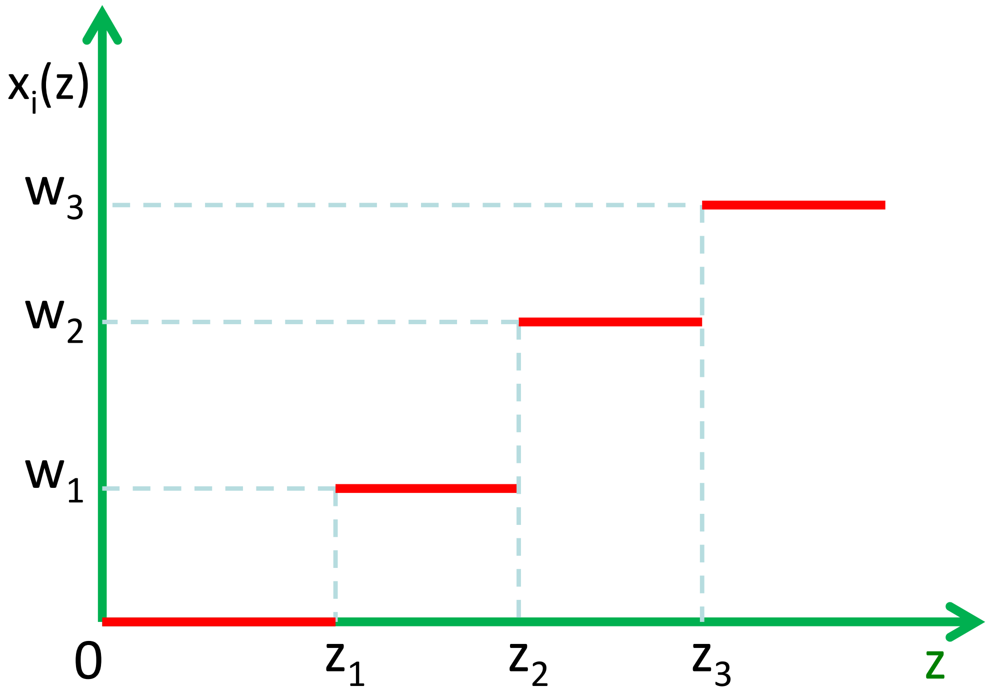


- Suppose bidder i bids b_i
- Look at the jumps of $x_i(z, b_{-i})$ in the interval $[0, b_i]$
- Suppose we have k jumps
- Jump at z_1: w_1
- Jump at z_2: $w_2 - w_1$
- Jump at z_3: $w_3 - w_2$
- ...
- Jump at z_k: $w_k - w_{k-1}$

Myerson's lemma and payment formula

For most scenarios of interest

- The allocation is piecewise linear with multiple jumps
- The jump determines how many extra units the bidder wins



Payment formula

- For each bidder i at a profile b, find all the jump points within $[0, b_i]$
- $p_i(b) = \Sigma_j z_j \cdot [\text{jump at } z_j]$
  $= \Sigma_j z_j \cdot [w_j - w_{j-1}]$
- The formula can also be generalized for monotone but not piecewise linear functions

Sponsored Search Auctions: Myerson’s Lemma and Generalized Second Price

Sponsored Search Auctions

- For a fixed search term (e.g. *ipod*)
  - n advertisers
  - k slots (typically k << n)
  - An auction is run for **every single search**
  - Each advertiser (bidder) is interested in getting himself displayed in **one of the slots**
    - And usually in a slot as **high** as possible

Sponsored Search Auctions

- Bidders submit an **initial budget** which they can refresh weekly or **monthly**
- Bidders also submit an **initial bid** which they can adjust as often as they wish
- The auction selects the winners to be displayed
- Different charging models exist: Pay Per Click, Pay Per Impression, Pay Per Transaction
- Currently, most popular is **Pay Per Click**
- A bidder is charged only if someone clicks on the bidder's ad

Sponsored Search Auctions

- We will focus on the bidders' side
- Model parameters for each bidder i
  - Private information: v_i = maximum amount willing to pay per click = value/happiness derived from a click (private information)
  - Each bidder i submits a bid b_i for willingness to pay per click (b_i may differ from v_i)
  - We will ignore the budget parameter for now.
    - In many cases, it is large enough and cannot affect the game
  - Hence, we have a single-parameter problem

Sponsored Search Auctions

- Model parameters for each slot j
  - α_j = Click-through rate (CTR) of slot j = probability that a user will click on slot j
  - Assume it is **independent** of who occupies slot j
    - We can generalize to the case **where the rates are weighted by a quality score** of the advertiser who takes each slot
  - The search engines update regularly the click-through rates and statistics show that

$$\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \ldots \geq \alpha_k$$

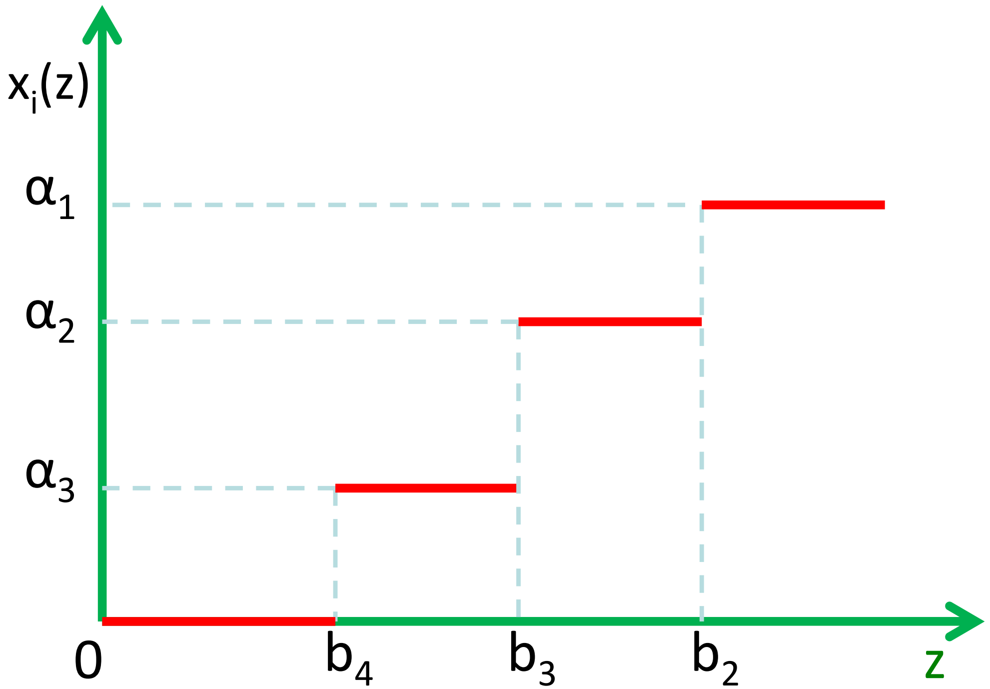
  - Users tend to click on higher slots

Sponsored Search Auctions

- Most natural allocation rule: for i=1 to k, **give to the i-th highest bidder the i-th best slot** in terms of CTR
  - Remaining n-k bidders do not win anything
  - For convenience, assume that $b_1 \geq b_2 \geq b_3 \geq \ldots \geq b_n$
- **Expected value** of a winning bidder i: $\alpha_i v_i$
- Is this rule **monotone**?
  - Yes, bidding higher can only get you a better slot
- Hence we can apply Myerson's formula to find the payment rule
  - For each bidder i, $x_i(b_i, b_{-i}) \in \{0, \alpha_k, \alpha_{k-1}, \ldots, \alpha_1\}$

Myerson's Lemma for Slot Auctions

- Let's analyze the highest bidder with bid b_1
- Suppose we have 3 slots and n>3 bidders



- Look at the jumps of x_i in the interval $[0, b_1]$
- Jump at $b_4 = \alpha_3$
- Jump at $b_3 = \alpha_2 - \alpha_3$
- Jump at $b_2 = \alpha_1 - \alpha_2$

Total payment:

$b_4\alpha_3 + b_3 (\alpha_2 - \alpha_3) + b_2 (\alpha_1 - \alpha_2)$

Myerson's Lemma for Slot Auctions

- More generally, for the i-th highest bidder, there will be k-i+1 jumps

$$p_i(\mathbf{b}) = \sum_{j=i}^{k} b_{j+1}[\alpha_j - \alpha_{j+1}]$$

- This would have been the payment if bidders cared for impressions and not for clicks
- Under **pay-per-click**, no actual payment takes place at the end of every auction, unless there is a click by a user
- Need to scale so that expected per-click payment is $p_i(\mathbf{b})$
- Proposed per-click payment to bidder in i-th slot: $p_i(\mathbf{b})/\alpha_i$
- By Myerson, no other payment can achieve truthfulness with the same allocation rule

Sponsored Search Auctions in Practice

- In practice most engines do not use the payment of Myerson's lemma
- But they use the same allocation rule
- The **Generalized Second Price Mechanism** (GSP) - initial version:
  - The search engine ranks the bids in decreasing order: $b_1 \geq b_2 \geq \ldots \geq b_n$
  - The i-th highest bidder takes the i-th best slot
  - Every time there is a click on slot *i*, bidder *i* pays b_{i+1}

Generalized Second Price (GSP)

- A better version:
  - The search engine keeps a quality score q_i for each bidder *i*
    - Yahoo, Bing (till a few years ago): q_i is the click-through rate of *i* (probability of a user clicking on an ad of bidder *i* on first slot)
    - Google: q_i depends on click-through rate, relevance of text and other factors
  - The search engine ranking is in decreasing order of $q_i \times b_i$
    $q_1 \times b_1 \geq q_2 \times b_2 \geq \ldots \geq q_n \times b_n$
  - The first k bidders of the ranking are displayed in the k slots
  - Every time there is a click on slot *i*, bidder *i* pays minimum bid required to keep his position, i.e. $(q_{i+1} \times b_{i+1}) / q_i$

Generalized Second Price (GSP)

- Myerson's lemma implies GSP cannot be truthful
  - Otherwise, its payment rule would coincide with the Myerson formula
- GSP was employed probably by accident!
  - As an attempt to use something simple that looked close to truthful
- Nevertheless...
  - For a long period, revenue from GSP was 95% of Google's revenue
  - Still nowadays an important percentage of search engines' revenue
  - Theoretical analysis: the Nash equilibria of GSP have revenue at least as high as the revenue of truthful bidding
  - Further connections also exist between GSP outcomes and the outcome of the truthful mechanism