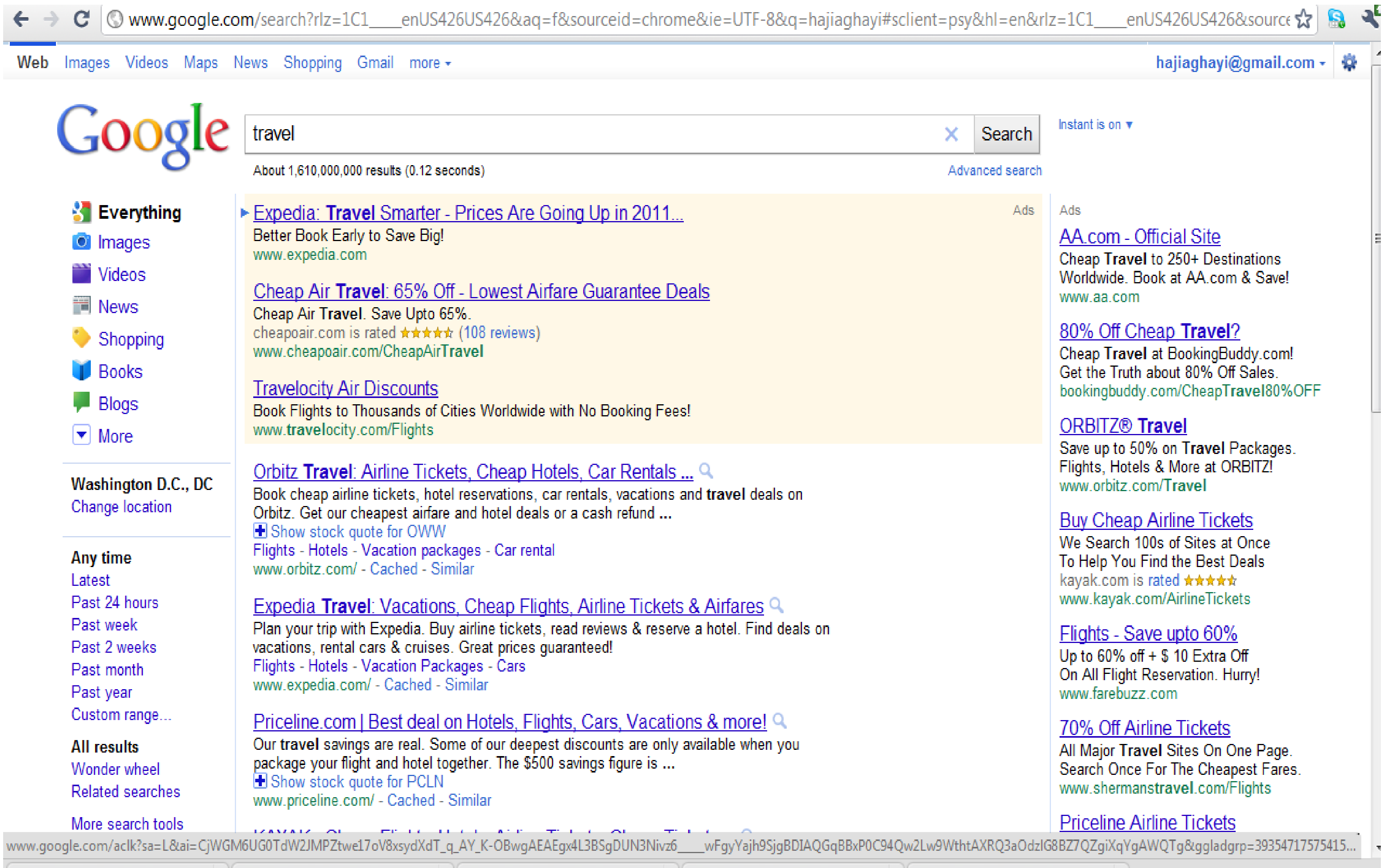
Advertisement (Online) Auctions and Prophet-Inequality Setting

Speaker:
Mohammad T. HajiAghayi,
University of Maryland,
College Park

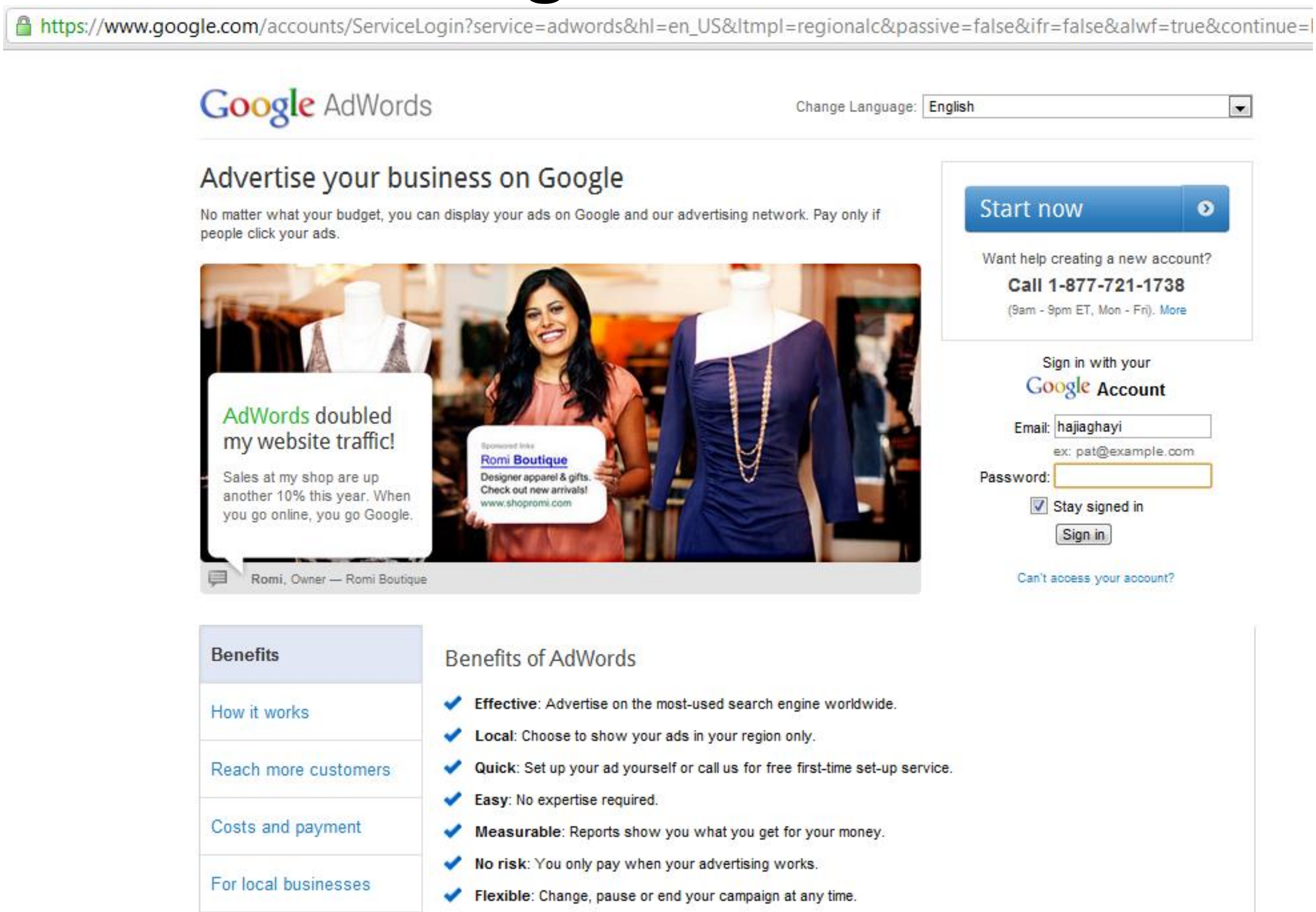
AdWord Auction



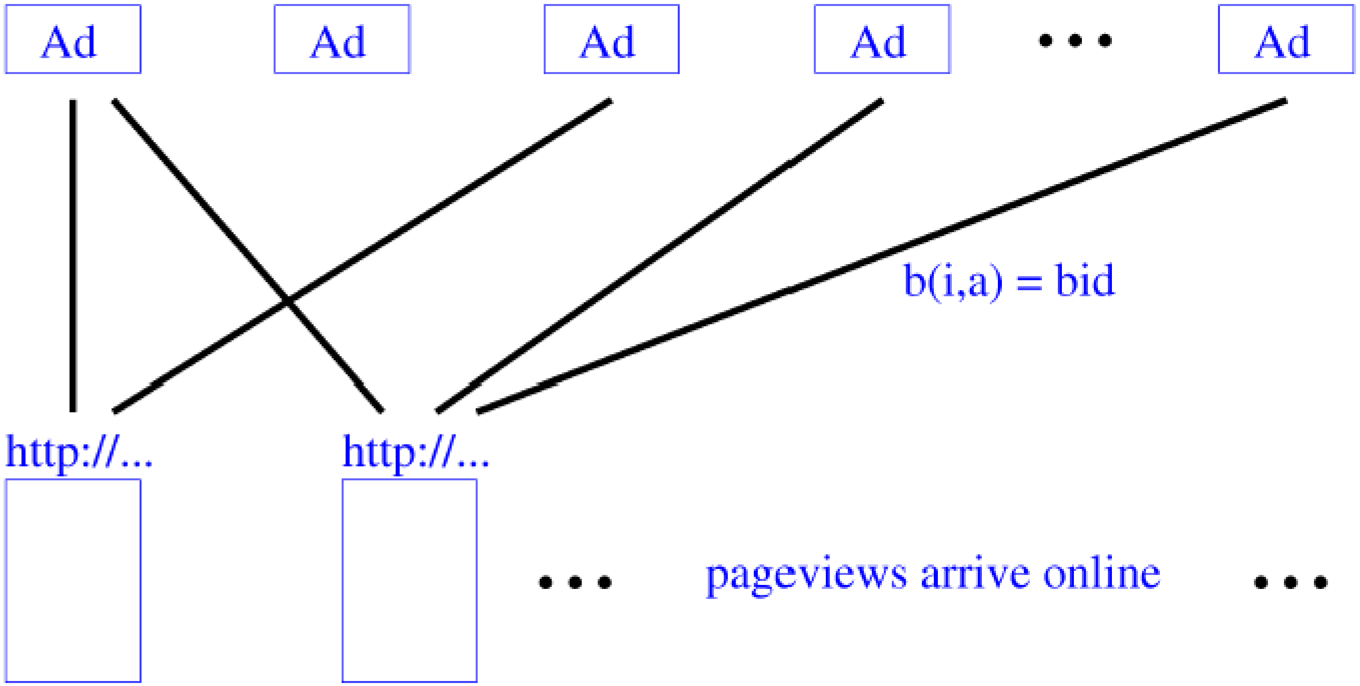
AdWord Auction

- Internet search companies such as Google, Yahoo, and MSN make billions of dollars out of it
- They decide what ads to display with each query to maximize revenue
- Users typing in query keywords, called **AdWords**
- Business place bids for individual AdWords together with a daily budget limit
- Search engines earn money from business when they show their ads in response to queries and charge them the second highest bid

Google AdWords

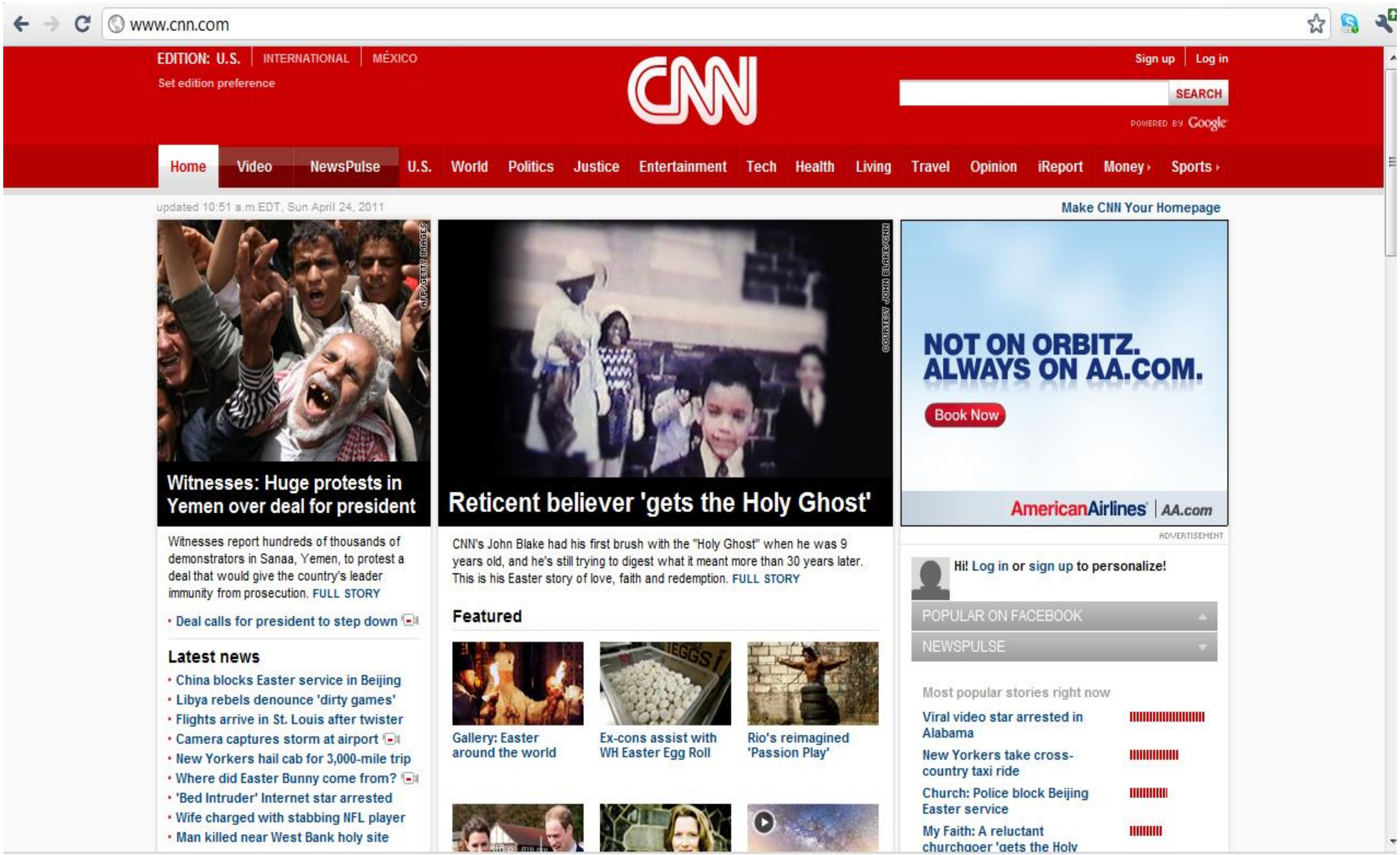


Online AdWord Auction



- When page arrives, assign an eligible ad.
  - revenue from assigning page i to ad a: b_{ia}
- "AdWords" (AW) problem:
  - Maximize revenue of ads served: $\max \sum_{i,a} b_{ia} x_{ia}$
  - Budget of ad a: $\sum_{i \in A(a)} b_{ia} x_{ia} \leq B_a$

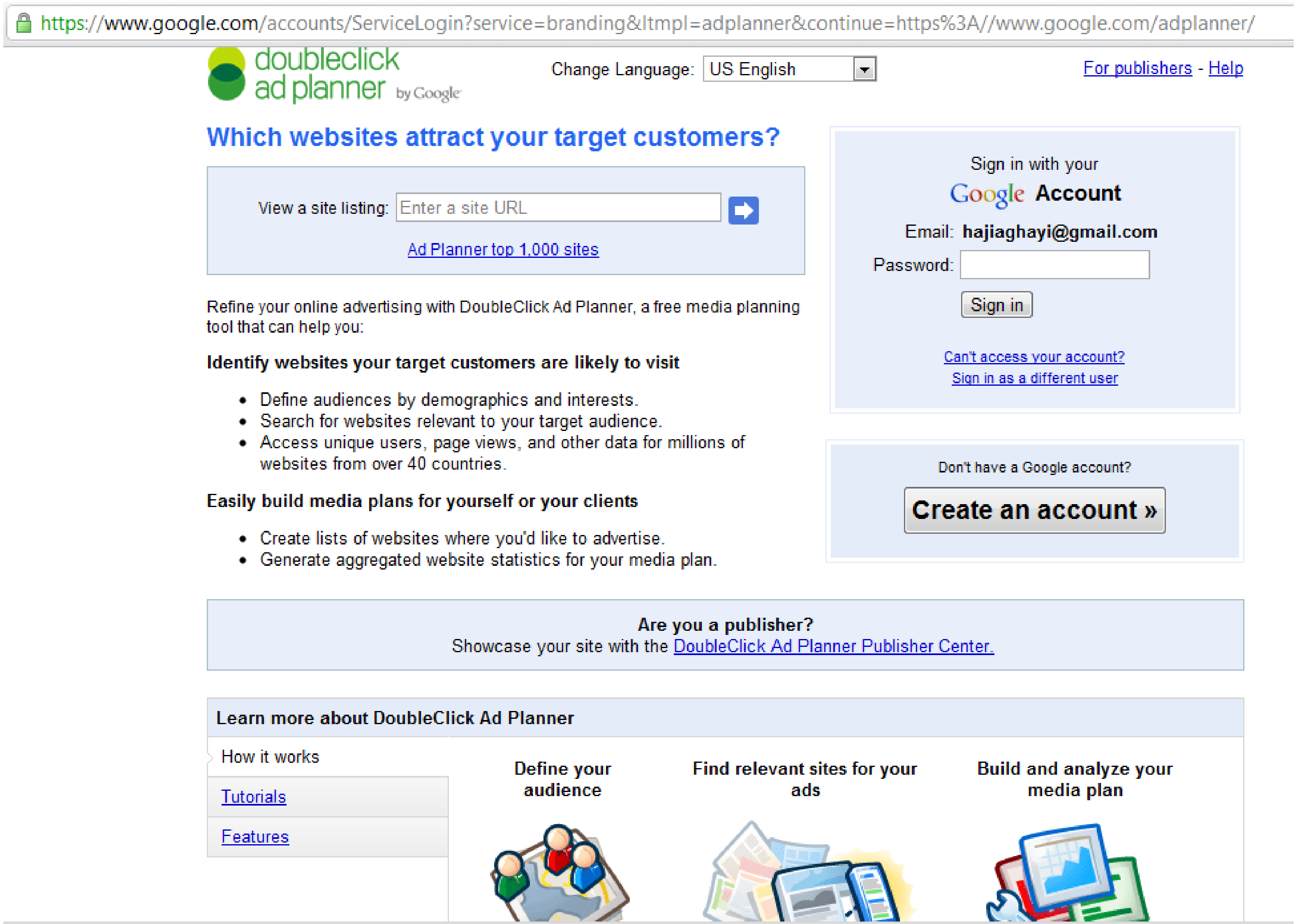
Display Ad Auction



Display Ad Auction

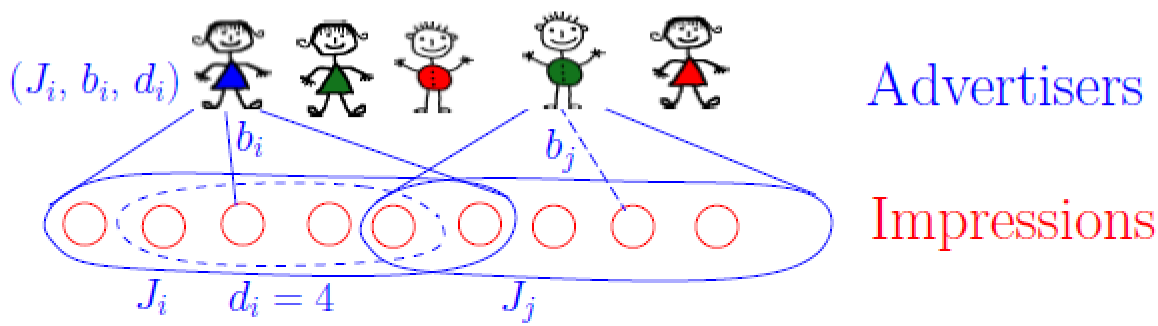
- **Impression**: Display/Banner Ads, Video Ads, Text Links
- Cost-Per-Impression (CMI/CPM)
- Done through offline negotiations+ Online allocations
- Q1, 2010: One Trillion Display Ads in US, $2.7 Billion
- Top publisher: Facebook, Yahoo and Microsoft sites
- Top Advertisers: AT&T, Verizon, Scottrade
- Ad Serving Systems e.g., Facebook, Google DoubleClick Ad Planner

DoubleClick Ad Planner



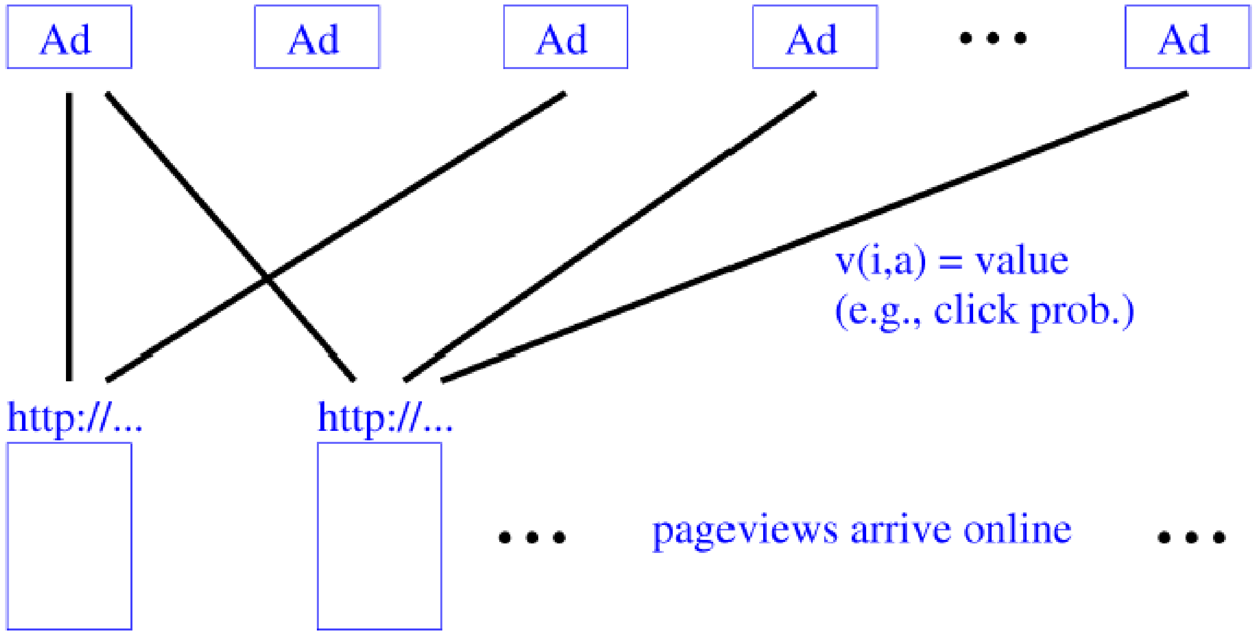
Google Ad Planning

- n advertisers, and set Y of impressions (items).
- Each advertiser i
  - Interested in a set J_i of impressions, (e.g, young women in Seattle),
  - Needs d_i impressions (Demand),
  - Value v_{it} (or Bid b_i) for each impression t,



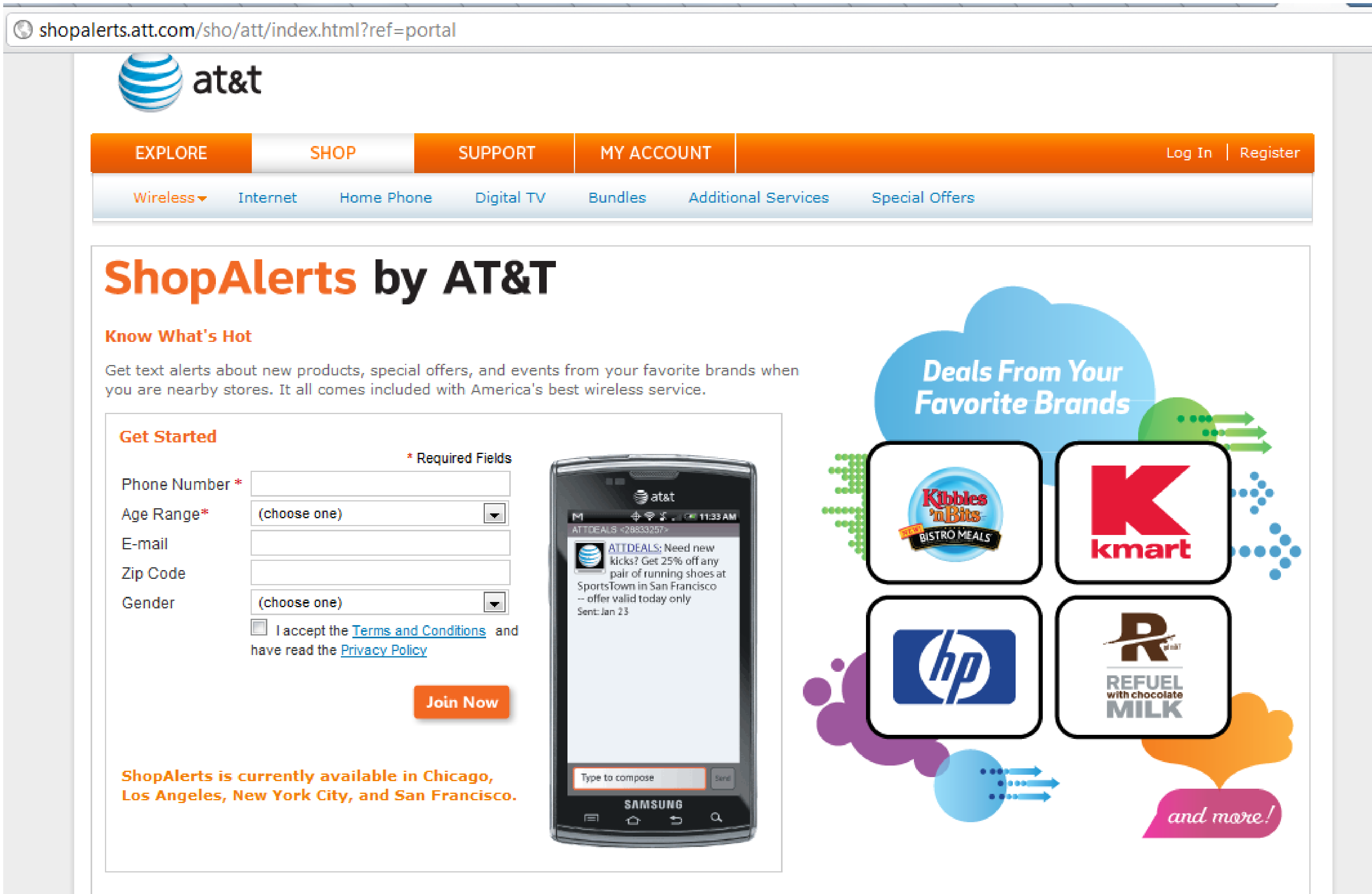
Efficiency (or Revenue) Maximization: Find an assignment with the maximum value.

Online Display Ad



- When page arrives, assign an eligible ad.
  - value of assigning page i to ad a: v_{ia}
- Display Ads (DA) problem:
  - Maximize value of ads served: $\max \sum_{i,a} v_{ia} x_{ia}$
  - Capacity of ad a: $\sum_{i \in A(a)} x_{ia} \leq C_a$

AdCell Auction

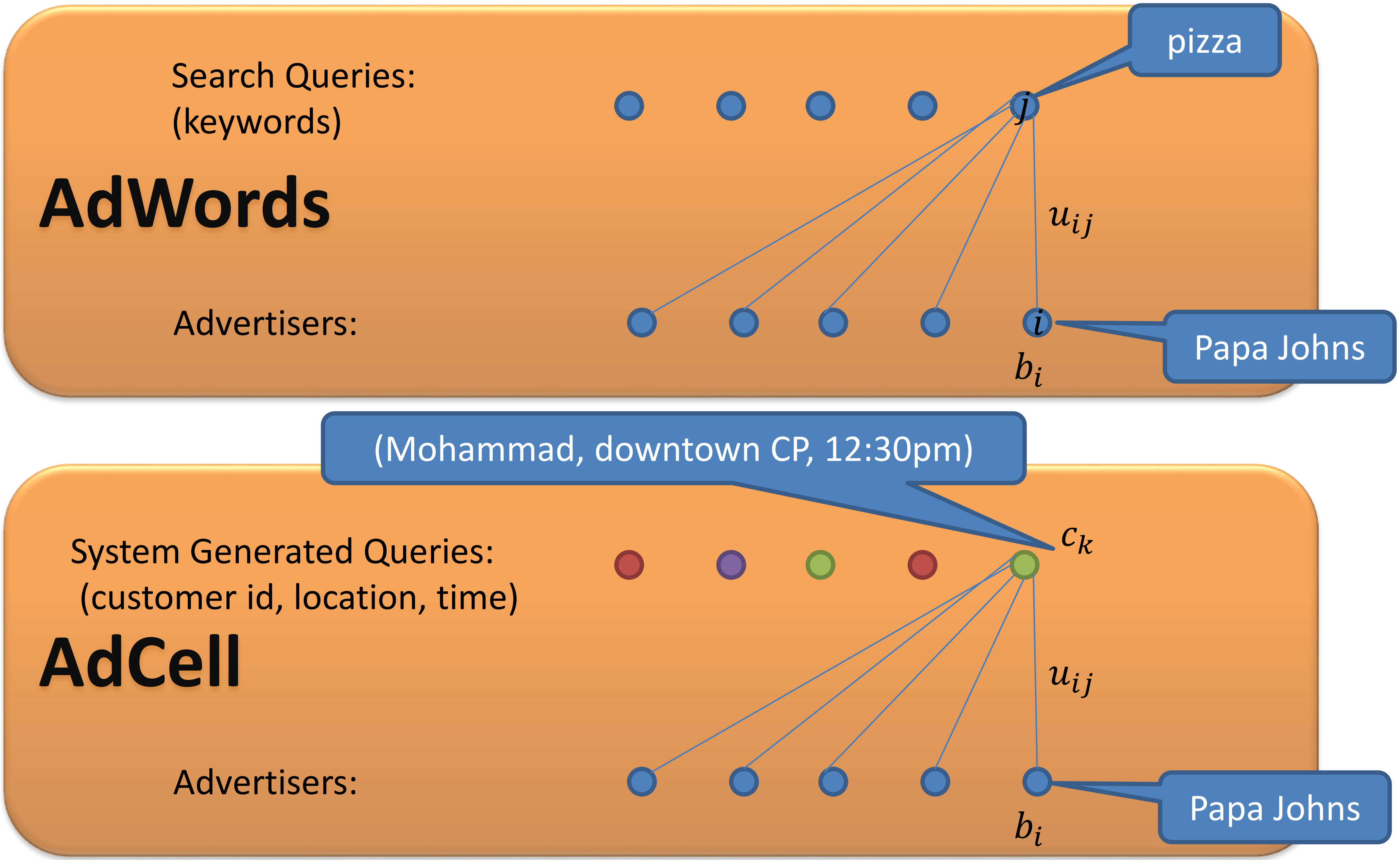


AdCell Auction

- Online Advertising
  - Major source of revenue

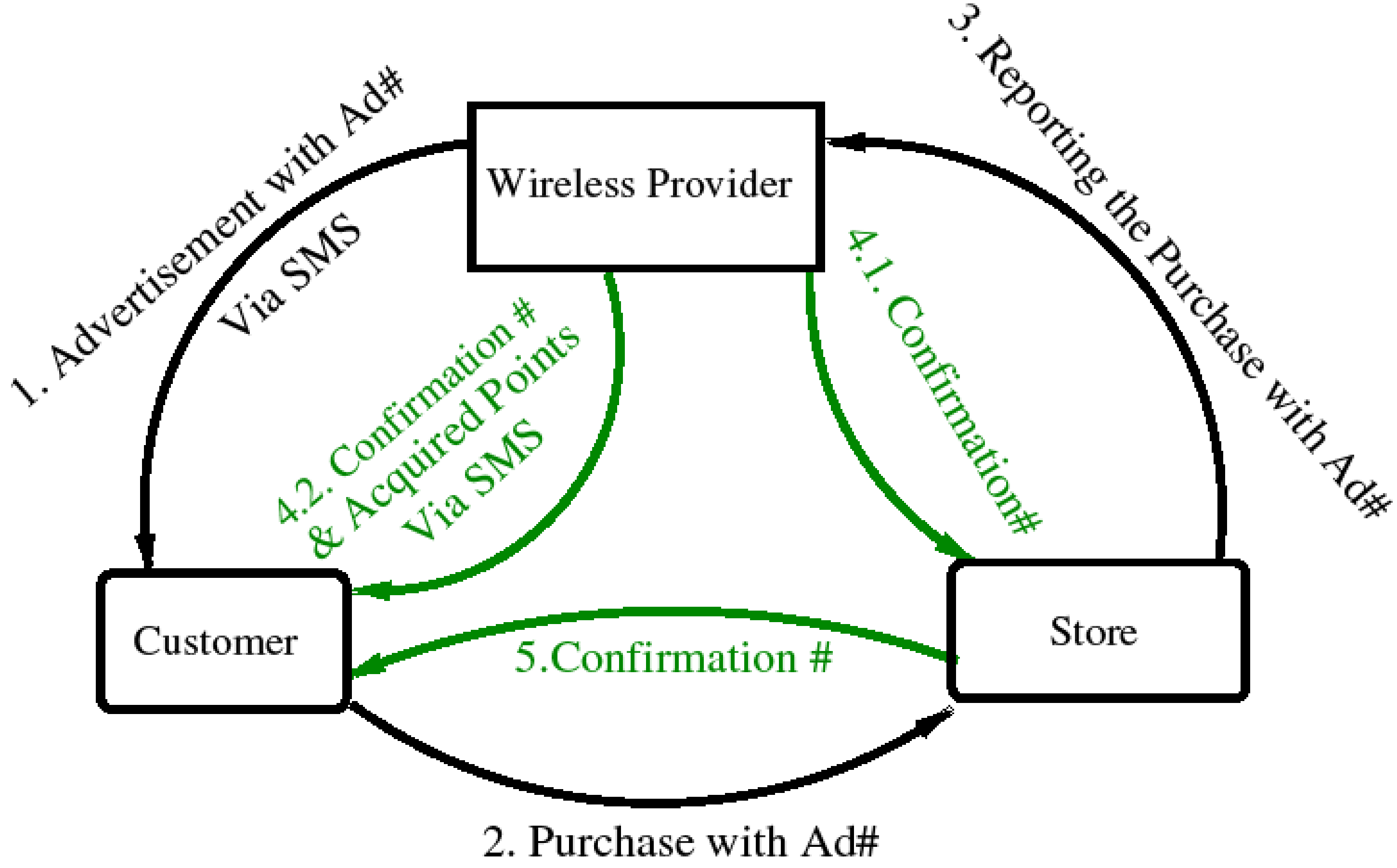
- AdCell vs AdWords
  - Intrusive delivery (TEXT,MMS, etc)
  - Limited number of Ads per customer
  - System generated queries
- ShopAlerts by AT&T
- Formulated in [AHLPS'11, ESA'11]

AdWords vs AdCell



Customer Policy

- AdCell is intrusive
- Incentivizing customers
  - Discount on service plan if they opt in
  - Limited number of ads per month



Wireless Provider
Customer
Store
1. Advertisement with Ad#
Via SMS
2. Purchase with Ad#
3. Reporting the Purchase with Ad#
4.1. Confirmation#
4.2. Confirmation #
& Acquired Points
Via SMS
5.Confirmation #

Online Bipartite Matching

- All these three problems are generalizations of Online Bipartite Matching:
- The input to the problem is:
  - bipartite graph G = (V_1 U V_2, E)
  - V_1 is the set of advertisers and V_2 is the set of keywords
  - the vertices in V_2 (keywords) arrive in an on-line fashion
  - the edges incident to each vertex u in V_2 are revealed when u arrives and determine the advertisers who want keyword u.
- When u arrives, the algorithm may match u to a previously unmatched adjacent vertex in V_1 , if there is one.
- Such a decision, once made, is irrevocable.
- The objective is to maximize the size of the resulting matching.

Online Bipartite Matching: Greedy

- Any **greedy** algorithm that always matches a vertex in V_2 if a match is possible constructs a maximal matching, and therefore such an algorithm has a ***competitive ratio*** of ½=**0.50** (by a double counting argument).
- ***Competitive ratio:*** The ratio of our algorithm to the *best* (*optimum*) offline algorithm.
- On the other hand, given any deterministic algorithm, it is easy to construct an instance that forces that algorithm to find a matching of size no greater than half of the optimum.

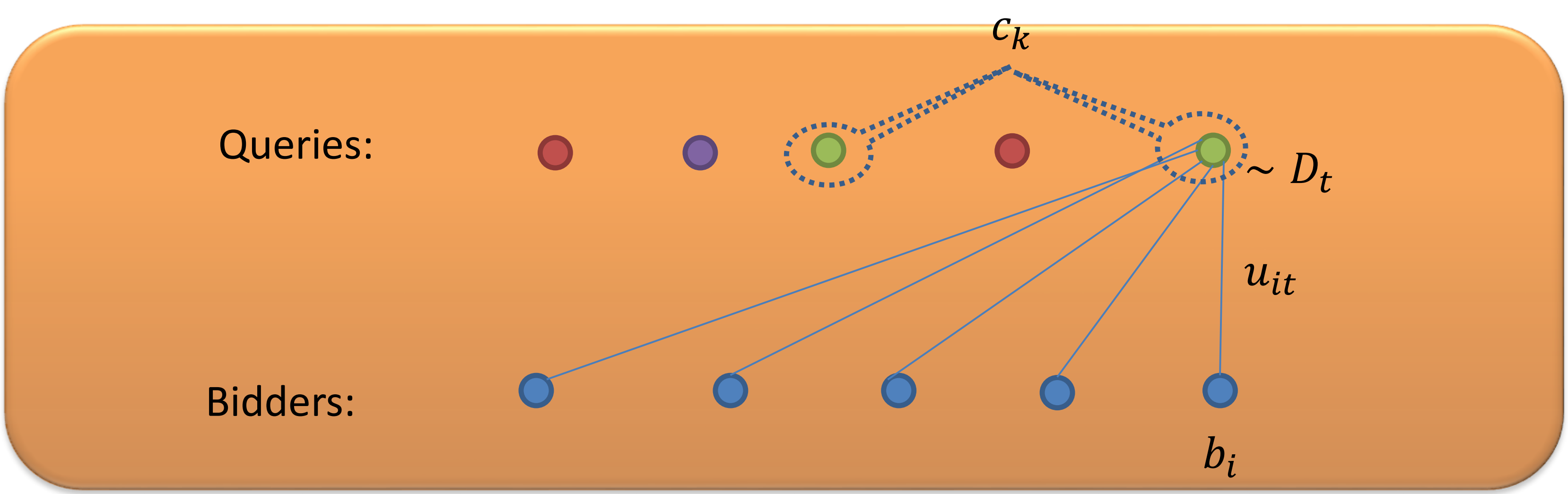
Online Bipartite Matching: Randomized

- Any **randomized** algorithm that chooses a single random ranking on the vertices in V_1
- When a vertex u in V_2 arrives among its unmatched neighbors assign u to the one than has the highest ranking
- This produces a competitive ratio 1-1/e≈**0.63**
- This is the best that we can do in the online world.
- However, if we know stochastic information like distributions of the keywords (the probability that a node u in V_2 arrives) and also the budget to the bid ratio is very large, we can obtain a competitive ratio very close to **1**.

Motivating Problems

- Online Advertising
  - AdWord Auction
    - Budget constraints (advertisers)
  - Display Ad Auction
    - Capacity constraints (advertisers)
  - AdCell Auction
    - Budget constraints (advertisers) & capacity constraints (customers)
- The goal is maximizing revenue
  - We charge the bids themselves (first price-auction)

Model



25

- D_t: t[th] query drawn from distribution D_t
- i: bidder, with budget b_i and bid u_{ij} for every query of type j in the support of D_t
- k: the set to which query x_t belongs.
- c_k: capacity of set k.

Use of History (model-based)

- Model
- 3 Variants of the problem
  - Only Budget Constraint
  - Only Capacity Constraint
  - Both Budget & Capacity Constraints
- Prophet Inq. Setting: Known but different distributions for different times, different customers, etc. (similarities to a Bayesian setting)
- As we have done emperical analysis using AT&T YellowPages data, IN PRACTICE we know probability distribution of 95% of queries (search keywords) [AHHKLZ’12].
- (Approximation) algorithms for real-world instead of the worst case

Secretary setting vs. Prophet Inq. setting

- Secretary problem is a classic optimal stopping theory problem studied since 1963 [Dynkin] and more recently in auction design [HKP04, ACM EC'04] (many follow-up work)
- Secretary setting: given a sequence of random variables $x_1,\dots,x_n$ drawn i.i.d. from the same distribution (or the random order model), an onlooker has to choose a certain number k of values and cannot choose a past value.
- Prophet Inq. is another well-studied optimal stopping theory problem since the 1970s [KS77, KS78, Ken87] and more recently in computer science [HKS07, AAAI'07]
- Prophet Inq. setting: given possibly different distributions of random variables $x_1,\dots,x_n$ an onlooker has to choose a certain number k of values and cannot choose a past value
- Called *Prophet Inq.* since a prophet with complete foresight has only a bounded advantage over an onlooker who observes the random variables one by one
- For $k=1$, the tight bound of ½ is known which we generalize.

IP (budgeted)

$$\text{maximize.} \quad \sum_i \min\left(\sum_t \sum_j \mathbf{x}_{ijt} u_{ij}, b_i\right) \qquad (B)$$

$$\forall j \in [n], t \in [T] \quad \sum_i \mathbf{x}_{ijt} \leq \mathcal{R}_{jt} \qquad (R)$$

$$\mathbf{x}_{ijt} \in \{0, 1\}$$

- b_i: budget of buyer i
- u_{ij}: bid of buyer i for a query of type j
- R_{jt}: Indicator random variable which is 1 iff the t^{th} query is of type j
- x_{ijt}: allocation variable

IP (capacitated)

$$\text{maximize.} \quad \sum_i \sum_t \sum_j \mathbf{x}_{ijt} u_{ij}$$

$$\forall j \in [n], t \in [T] \quad \sum_i \mathbf{x}_{ijt} \leq \mathcal{R}_{jt} \qquad (R)$$

$$\forall i \in [m] \quad \sum_t \sum_j \mathbf{x}_{ijt} \leq c_i \qquad (C)$$

$$\mathbf{x}_{ijt} \in \{0, 1\}$$

- c_i: capacity of advertiser i
- u_{ij}: bid of buyer i for a query of type j
- R_{jt}: Indicator random variable which is 1 iff the t^{th} query is of type j
- x_{ijt}: allocation variable

Expectation LPs

maximize. $\sum_i \sum_t \sum_j \mathbf{x}_{ijt} u_{ij}$

$\forall i \in [m]$ $\sum_t \sum_j \mathbf{x}_{ijt} u_{ij} \leq b_i$

$\forall j \in [n], t \in [T]$ $\sum_i \mathbf{x}_{ijt} \leq p_t(j);\ \mathbf{x}_{ijt} \geq 0$

$\mathbf{x}_{ijt} \geq 0$

maximize. $\sum_i \sum_t \sum_j \mathbf{x}_{ijt} u_{ij}$

$\forall j \in [n], t \in [T]$ $\sum_i \mathbf{x}_{ijt} \leq p_j(t)$

$\forall i \in [m]$ $\sum_t \sum_j \mathbf{x}_{ijt} \leq c_i$

$\mathbf{x}_{ijt} \geq 0$

Online Algorithm (budgeted)

$$\text{maximize.} \quad \sum_i \sum_t \sum_j \mathbf{x}_{ijt} u_{ij}$$

$$\forall i \in [m] \quad \sum_t \sum_j \mathbf{x}_{ijt} u_{ij} \leq b_i$$

$$\forall j \in [n], t \in [T] \quad \sum_i \mathbf{x}_{ijt} \leq p_t(j); \ \mathbf{x}_{ijt} \geq 0$$

$$\mathbf{x}_{ijt} \geq 0$$

- Algorithm Alg_B:
  - Solve this LP and let x^*_{ijt} denote the optimal assignment.
  - Upon arrival of query t of type j allocate it to buyer i w.p. $\frac{x^*_{ijt}}{p_{tj}}$

$\approx 1 - \frac{1}{\sqrt{2\pi k}}$

- Claim: Alg_B obtains at least $(1 - \frac{k^k}{e^k k!})$ of LP_B (in expectation)

  , where k is the minimum of the budget to the bid ratios
- $X_{ijt} = \begin{cases} u_{ij} & w.p. \quad x^*_{ijt} \\ 0 & w.p. \quad 1 - x^*_{ijt} \end{cases}$
- Expected revenue of Alg_B is: $\mathrm{E}[\sum_i \min(\sum_{j,t} X_{ijt}, b_i)]$
- Objective value of LP_B is: $\sum_i \sum_{j,t} E[X_{ijt}]$
- We show that for each i :

$$E[\min(\sum_{j,t} X_{ijt}, b_i)] \geq (1 - \frac{k^k}{e^k k!}) \min(\sum_{jt} E[X_{ijt}], b_i)$$

Expectation of Truncated Sum (ETS) vs Truncated Sum of Expectations (TSE)

- Theorem (ETS vs TSE):
  – If $X_1, \cdots, X_n$ are independent random variables, $k \in N$, $b \in R_+$ and all $X_j \in [0, \frac{b}{k}]$ then the following holds:
    - $E\left[\min\left(\sum_j X_j, b\right)\right] \geq \left(1 - \frac{k^k}{e^k k!}\right) \min\left(\sum_j E[X_j], b\right)$
      $\geq \left(1 - \frac{1}{\sqrt{2\pi k}}\right) \min\left(\sum_j E[X_j], b\right)$

Proof of ETS-vs-TSE

- Create $Y_1, \cdots, Y_n$ such th **Y_i = C w.p. $E[X_i]/C$ where $C=b/k$**
  - $\mathrm{E}[Y_i] = E[X_i]$
  - $Y_i \in \{0, \frac{b}{k}\}$
- Then
  - $E\left[\min\left(\sum_j Y_j, b\right)\right] \leq = \geq E\left[\min\left(\sum_j X_j, b\right)\right]$
- So it is enough to show that
  - $E\left[\min\left(\sum_j Y_j, b\right)\right] \geq \left(1 - \frac{k^k}{e^k k!}\right) \min\left(\sum_j E[Y_j], b\right)$

Proof of ETS-vs-TSE (cont'd)

- Take any Y_{j_1} and Y_{j_2}
- Replace them with RVs Y'_{j_1} and Y'_{j_2} such that
  - $Y'_{j_1}, Y'_{j_2} \in \{0, \frac{b}{k}\}$
  - $\mathrm{E}[Y'_{j_1}] = E[Y'_{j_2}] = E[Y_{j_1} + Y_{j_2}]/2$
- Then
  - $E[\min(\sum_j Y_j, b)] \geq E[\min(Y'_{j_1} + Y'_{j_2} + \sum_{j \neq j_1, j_2} Y_j, b)]$
- So it is enough to prove the theorem for the case where $\mathrm{E}[Y_{j_1}] = E[Y_{j_2}]$ for all j_1 and j_2.
- The final step involves simple algebraic manipulations which we omit.

A Less Simple Model: Capacity Only

- Consider one bidder with two query types q_1 and q_2
- Bid for q_1 is 1 and for q_2 is $(1-\varepsilon)/\varepsilon$
- The first query is of type q_1 with Prob. 1 and the second query is of type q_2 with Prob. ε
- The expected revenue of any online (randomized) algorithm is $\max\{1, \epsilon(\frac{1-\epsilon}{\epsilon})\} = 1$
- However optimum offline would allocate the second query if it arrives; otherwise allocate the first query. Its expected revenue is $(1-\epsilon) \times 1 + \epsilon(\frac{1-\epsilon}{\epsilon}) \approx 2$

A Less Simple Model: Capacity Only

- This is the lower bound ½ also for Prophet Inq. known from the 1970s for which we generalize the upper bound.

- This is in contrast to factors 1- 1/e and better for the same distribution case, e.g., [MSVV'05,FMMM'09]

- Note that without stochastic information we cannot get any approximation factor (generalizes the secretary problem in this setting).

The Capacitated Case

maximize. $$\sum_i \sum_t \sum_j \mathbf{x}_{ijt} u_{ij}$$

$\forall j \in [n], t \in [T]$ $$\sum_i \mathbf{x}_{ijt} \leq p_j(t)$$

$\forall i \in [m]$ $$\sum_t \sum_j \mathbf{x}_{ijt} \leq c_i$$

$$\mathbf{x}_{ijt} \geq 0$$

- Algorithm Alg_C:
  - Solve the LP and let x^*_{ijt} denote the optimal assignment.
  - Run a dynamic program for each advertiser: $E^r_{i,t}$
  - Upon arrival of query t of type j, select advertiser i w.p. $\frac{x^*_{ijt}}{p_{tj}}$ then allocate it to i iff $u_{ij} + E^{r-1}_{i,t+1} \geq E^r_{i,t+1}$

Dynamic Program of DP_i

- For each advertiser i we can compute $E^r_{i,t+1}$ using the following dynamic program:

$$\mathcal{E}^r_{i,t} = \sum_j \mathbf{x}^*_{ijt} \max\{u_{ij} + \mathcal{E}^{r-1}_{i,t+1}, \mathcal{E}^r_{i,t+1}\} + (1 - \sum_j \mathbf{x}^*_{ijt})\mathcal{E}^r_{i,t+1}$$

, where the quantity is the expected benefit we get from bidder i with only r remaining capacity at or after time t.

- Claim: DP_i obtains at least $\frac{1}{2}$ of LP_C for each advertiser$i$ (in expectation)

- We show that for each advertiser i :

$$E_{i,1}^{c_k} \geq \frac{1}{2} \sum_t \sum_t u_{ij} x_{ijt}^*$$

Stochastic Uniform Knapsack (SUK)

- A knapsack of capacity c
- Item $\boldsymbol{t}$ arrives with prob $p_{\boldsymbol{t}}$ and has value $v_{\boldsymbol{t}}$ *at time* $\boldsymbol{t}$
- Dynamic Program (DP):
  - At time t, if item $\boldsymbol{t}$ arrives: put it in the knapsack iff $v_j + E_{\boldsymbol{t}+1}^{r-1} \geq E_{t+1}^{r}$

$$E_t^r = \max(p_t v_t + (1 - p_t) E_{t+1}^{r-1}, E_{t+1}^{r})$$

Dynamic Program for SUK

- Dynamic Program (DP):
  - At time t, if item t arrives: put it in the knapsack iff $v_j + _t E^{r-1}_{k,t+1} \geq E^{r}_{k,t+1}$
  - $E^r_t = \max(p_t v_t + (1 - p_t) E^{r-1}_{t+1}, E^r_{t+1})$

- Theorem (SUT-DP):
  – Let $O_{DP} = E^c_1$ and $O^* = \sum_j p_j v_j$. If $\sum p_j \leq c$ then $\mathrm{O_{DP}} \geq \frac{1}{2} O^*$

Proof of Theorem (SUT-DP)

- $O_{DP} = E_1^c$
- $O^* = \sum_t p_t v_t$
- $E_t^r = \max(p_t v_t + (1 - p_t) E_{t+1}^{r-1}, E_{t+1}^r)$

- Lemma:
  - E_t^r has decreasing marginal value in r,
  i.e.: $E_t^r - E_t^{r-1} \leq E_t^{r-1} - E_t^{r-2}$,
  Therefore: $E_t^{r-1} \geq \frac{r-1}{r} E_t^r$
- Lemma:
  - $E_t^r \geq \max(p_t v_t + \left(1 - \frac{p_t}{r}\right) E_{t+1}^r, E_{t+1}^r)$

Proof of Theorem (SUT-DP)

- $O_{DP} = E_1^C$
- $O^* = \sum_t p_t v_t$
- $E_t^C \geq \max\left(p_t v_t + \left(1 - \frac{p_t}{r}\right) E_{t+1}^C, E_{t+1}^C\right)$
- Lemma:
  - WLOG, we may assume O^*=1
  - then O_{DP}/O^* is at least:

$$
\begin{aligned}
&\text{minimize.} && \boldsymbol{E}_1^C \\
&\forall t \in [u-1]: && \boldsymbol{E}_t^C - p_t \boldsymbol{v}_t - \left(1 - \frac{p_t}{r}\right) E_{t+1}^r \geq 0 && (\boldsymbol{\alpha}_t) \\
& && \boldsymbol{E}_u^C - p_u v_u \geq 0 && (\boldsymbol{\alpha}_{\boldsymbol{u}}) \\
&\forall t \in [u-1]: && \boldsymbol{E}_t^C - \boldsymbol{E}_{t+1}^C \geq 0 && (\boldsymbol{\beta}_t) \\
& && \sum_{t=1}^{u} p_t \boldsymbol{v}_t \geq 1 && (\gamma) \\
& && \boldsymbol{v}_t \geq 0, \quad \boldsymbol{E}_t^C \geq 0
\end{aligned}
$$

$u = t_{max}$

Proof of Theorem (SUT-DP)

$$
\begin{array}{lll}
\text{minimize.} & \boldsymbol{E}_1^C & \\
\forall t \in [u-1]: & \boldsymbol{E}_t^C - p_t \boldsymbol{v}_t - (1 - \frac{p_t}{r}) E_{t+1}^r \geq 0 & (\boldsymbol{\alpha}_t) \\
 & \boldsymbol{E}_u^C - p_u v_u \geq 0 & (\boldsymbol{\alpha_u}) \\
\forall t \in [u-1]: & \boldsymbol{E}_t^C - \boldsymbol{E}_{t+1}^C \geq 0 & (\boldsymbol{\beta}_t) \\
 & \sum_{t=1}^{u} p_t \boldsymbol{v}_t \geq 1 & (\boldsymbol{\gamma}) \\
 & \boldsymbol{v}_t \geq 0, \quad \boldsymbol{E}_t^C \geq 0 &
\end{array}
$$

$$
\begin{array}{lll}
\text{maximize.} & \boldsymbol{\gamma} & \\
\forall t \in [u]: & \boldsymbol{\gamma} p_t - \boldsymbol{\alpha}_t p_t \leq 0 & (\boldsymbol{v}_t) \\
 & \boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1 \leq 1 & (\boldsymbol{E}_1^C) \\
\forall t \in [2 \cdots u-1]: & \boldsymbol{\alpha}_t + \boldsymbol{\beta}_t - (1 - \frac{p_{t-1}}{C}) \boldsymbol{\alpha}_{t-1} - \boldsymbol{\beta}_{t-1} \leq 0 & (\boldsymbol{E}_t^C) \\
 & -(1 - \frac{p_{u-1}}{C}) \boldsymbol{\alpha}_{u-1} - \boldsymbol{\beta}_{u-1} \leq 0 & (\boldsymbol{E}_u^C) \\
 & \boldsymbol{\alpha}_t \geq 0, \quad \boldsymbol{\beta}_t \geq 0, \quad \boldsymbol{\gamma} \geq 0 &
\end{array}
$$

Proof of Theorem (SUT-DP)

$$
\begin{array}{lll}
\text{maximize.} & \gamma & \\
\forall t \in [u]: & \gamma \leq \boldsymbol{\alpha}_t & (\boldsymbol{v}_t) \\
& \boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1 \leq 1 & (\boldsymbol{E}_1^C) \\
\forall t \in [2 \cdots u-1]: & \boldsymbol{\alpha}_t + \boldsymbol{\beta}_t \leq (1 - \frac{p_{t-1}}{C})\boldsymbol{\alpha}_{t-1} + \boldsymbol{\beta}_{t-1} & (\boldsymbol{E}_t^C) \\
& 0 \leq (1 - \frac{p_{u-1}}{C})\boldsymbol{\alpha}_{u-1} + \boldsymbol{\beta}_{u-1} & (\boldsymbol{E}_u^C) \\
& \boldsymbol{\alpha}_t \geq 0, \quad \boldsymbol{\beta}_t \geq 0, \quad \gamma \geq 0 &
\end{array}
$$

- We construct a feasible dual that obtains ½:
  - Setting $\alpha_t = \gamma$ and $\beta_t = \beta_{t-1} - \frac{p_{t-1}}{C}\gamma$ and $\beta_1 = 1 - \gamma$
  - Then $\beta_t = 1 - \gamma - \gamma \sum_{k=1}^{t-1} \frac{p_k}{C} \geq 1 - 2\gamma$
  - So we can set $\gamma = 1/2$

Capacity at least k

- Theorem:

  If the capacity of each customer is at least k, then the same algorithm Alg_C obtains at least $1 - \frac{1}{\sqrt{k+3}}$ fraction of the optimum solution in expectation.

- The analysis is more complicated and needs a process called Sand/Barrier to analyze it [AHL'12, ACM EC'12].
- So if the capacities are large enough we obtain almost the optimum expectation.

General Model (Capacity and Budget)

- This is the case of AdCell Auction
- We write the combined LP of the capacitated case and the budgeted case by writing both set of constraints
- The algorithm Alg_{BC} is the same as that of the capacitated case (Alg_C) except we are using the new LP
- Theorem:
  - ✓ Alg_{BC} obtains $\left(1 - \frac{1}{\sqrt{k+3}}\right)\left(1 - \frac{1}{\sqrt{2\pi k}}\right) \approx 1$ fraction of the optimum in expectation (VERY PRACTICAL).
- The proof comes due to negative correlations.

Online Stochastic Generalized Assignment Problem

- Generalization of all problems considered so far, as well as stochastic knapsack
- Items arrive online, each with a value and a size maybe dependent to bins
- Upon arrival of an item, it can be placed in a bin or discarded
- Distribution is available about size and value but the items are coming in an adversarial order
- Objective is to maximize the value of the placement

Online Stochastic Generalized Assignment Problem

- Theorem: Under the assumption that no item takes more than 1/k capacity of any bin of non-zero value, we can obtain a $1-\frac{1}{\sqrt{k}}$ -competitive algorithm.

- The algorithm is competitive even when the adversary can do fractional assignment
  - ❑ application to banner advertisement (the problem is independent-set hard otherwise)
- The bound is tight for k=1

Online Stochastic Generalized Assignment Problem

Ideas of the proof:

- First we write an Expected LP as before
- Upon arrival of an item, we decide the bin that we want to try based on *x* variables as before
- To decide whether we really want to assign the item to the selected bin we are running a continues version of sand/barrier process (a.k.a. the magician problem)
- Intuitively a probabilistic algorithm which derives a new bound like *Chernoff* but with zero violation prob.
- The proof is involved; but uses induction as a basis

Questions?

تشكر

Obrigado