Bidder Selection Problem in Position Auctions:
A Fast and Simple Algorithm via Poisson Approximation

Nick Gravin — ITCS, SUFE
Yixuan Even Xu — IIIS, Tsinghua
Renfei Zhou — IIIS, Tsinghua

Bidder Selection in Online Ad Auction

Online ad auction:
Ad company sells ad slots to advertisers;
Real-time and automated.

Bidder selection:
Bidders' valuations are computed from a ML model;
Running the model for all bidders is costly and slow;
A prior distribution for each bidder is available.
Two-stage selection:
Filter out a fraction of bidders, then run the auction.

Bidder Selection Problem (Single-Item)

There are **n bidders** competing for an ad slot.
Bidder i has value $v_i \sim D_i$ from an independent distribution.
We need to **choose k bidders**, maximizing

$$\mathbb{E}_{v_1,\dots,v_n}[\max\{v_i \mid \text{bidder } i \text{ is chosen}\}].$$

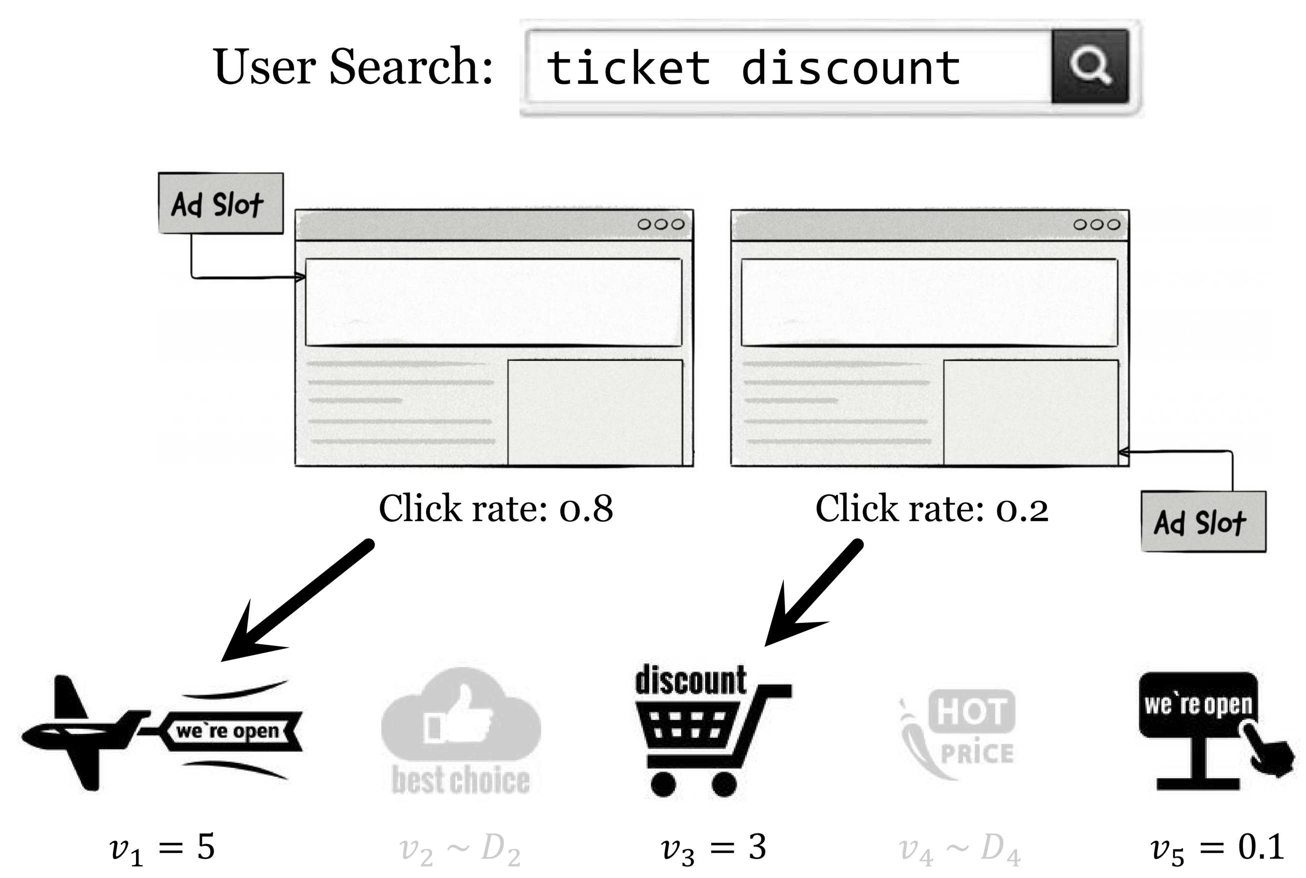
Exact optimum is NP-hard; aim for **$(1-\varepsilon)$-approximate.**

Bidder Selection Problem (Position Auction)

There are **n bidders** competing for **some ad slots.**
Bidder i has value $v_i \sim D_i$ from an independent distribution.
There is a non-negative weight sequence $w_1 \ge w_2 \ge \cdots \ge w_k$.
We need to **choose k bidders**, maximizing

$$\mathbb{E}_{v_1,\dots,v_n}\left[\sum_{i=1}^{k} v_{(i)} w_i\right],$$

where $v_{(i)}$ is the i-th largest value among k chosen bidders.



Previous Results on BSP

Previous $(1-\varepsilon)$-approximation (PTAS) algorithms on BSP:
[CHLLLL2016]: For **single-item auction**;
[MNPR2020]: For **single-item auction**;
[SS2021]: For ***L*-unit auctions** (i.e., position auctions with $w_i \in \{0, 1\}$)

All of them base on discretizing all possible distributions.
Bad dependency on ε:

$$2^{O(1/\varepsilon)^{O(1/\varepsilon)}}$$

Take years for small instances like $n = 3,\ k = 2,\ \varepsilon = 0.2$.
Not implementable in practice.

Our Results

There is a **polynomial-time** algorithm for BSP choosing k bidders out of n with approximation ratio

$$1 - O(k^{-1/4}).$$

This implies a PTAS for BSP for general position auctions.
The algorithm is **easily implemented**, runs **fast** and obtains **high-quality solutions** in experiments.

Main Technique: Poisson Approximation

Objective Function $\mathrm{SW}(\boldsymbol{x})$ → **Poisson Approximation** → Relaxed Objective $\widetilde{\mathrm{SW}}(\boldsymbol{x})$

Relaxed objective $\widetilde{\mathrm{SW}}(\boldsymbol{x})$ has 3 merits:

1. **Good approximation** ratio: $1 - O(k^{-1/4})$;
2. **Convex**, thus easy to optimize;
3. **Works** for general **position auctions** (not only single-item).

Algorithm Framework

1. Poisson approximation gives the **relaxed objective** $\widetilde{\mathrm{SW}}(\boldsymbol{x})$;
2. Run **convex optimization** to find (a fractional solution) $\boldsymbol{x}$ that maximizes $\widetilde{\mathrm{SW}}(\boldsymbol{x})$;
3. Use **rounding** techniques to transform $\boldsymbol{x}$ to an integer solution.

Experiments

We test homebrew implementations of 3 algorithms (using python + standard convex libraries):

1. Greedy for Submodular Welfare Maximization;
2. Local Search (a slow heuristic algorithm usually with good solution quality);
3. Our algorithm.

On large instances ($n = 1000,\ k = 200$):

	Local Search	Greedy	Our Algo
Running Time	> 1 week	1 day	45 sec
Relative Welfare	N/A	97.38%	100.00%

On all test cases, our algorithms achieves **> 99% approx.** compared to the benchmarks (Local Search & Greedy).

Future Directions

1. Bidder Selection Problem under different feasibility constraints, e.g., matroid, matching, and intersection of matroids;
2. Revenue maximization for other auction formats;
3. Improve the approximation ratio $1 - O(k^{-1/4})$.