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

Nikolai Gravin*
nikolai@mail.shufe.edu.cn
Shanghai University of Finance and Economics
Shanghai, China

Yixuan Even Xu*
xuyx20@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Renfei Zhou*
zhourf20@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

## ABSTRACT

In the Bidder Selection Problem (BSP) there is a large pool of $n$ potential advertisers competing for ad slots on the user's web page. Due to strict computational restrictions, the advertising platform can run a proper auction only for a fraction $k < n$ of advertisers. We consider the basic optimization problem underlying BSP: given $n$ independent prior distributions, how to efficiently find a subset of $k$ with the objective of either maximizing expected social welfare or revenue of the platform. We study BSP in the classic multi-winner model of position auctions for welfare and revenue objectives using the optimal (respectively, VCG mechanism, or Myerson's auction) format for the selected set of bidders. This is a natural generalization of the fundamental problem of selecting $k$ out of $n$ random variables in a way that the expected highest value is maximized. Previous PTAS results ([Chen, Hu, Li, Li, Liu, Lu, NIPS 2016], [Mehta, Nadav, Psomas, Rubinstein, NIPS 2020], [Segev and Singla, EC 2021]) for BSP optimization were only known for single-item auctions and in case of [Segev and Singla 2021] for $l$-unit auctions. More importantly, all of these PTASes were computational complexity results with impractically large running times, which defeats the purpose of using these algorithms under severe computational constraints.

We propose a novel Poisson relaxation of BSP for position auctions that immediately implies that 1) BSP is polynomial-time solvable up to a vanishingly small error as the problem size $k$ grows; 2) there is a PTAS for position auctions after combining our relaxation with the trivial brute force algorithm. Unlike all previous PTASes, we implemented our algorithm and did extensive numerical experiments on practically relevant input sizes. First, our experiments corroborate the previous experimental findings of Mehta et al. that a few simple heuristics used in practice (e.g., Greedy for general submodular maximization) perform surprisingly well in terms of approximation factor. Furthermore, our algorithm outperforms Greedy both in running time and approximation on medium and large-sized instances, i.e., its running time scales better with the instance size.

## CCS CONCEPTS

• **Theory of computation → Approximation algorithms analysis**; **Algorithmic game theory and mechanism design**; • **Applied computing** → *Online auctions*.

## KEYWORDS

Bidder Selection, Submodular Maximization, Position Auctions

**ACM Reference Format:**
Nikolai Gravin, Yixuan Even Xu, and Renfei Zhou. 2024. Bidder Selection Problem in Position Auctions: A Fast and Simple Algorithm via Poisson Approximation. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3589334.3645418

## 1 INTRODUCTION

Online advertising is a big part of the modern e-commerce industry and a key to the monetization of many online businesses. The majority of ad slots on a user web page are sold in *real time* via an automated auction to a group of candidate advertisers. The whole process from the time when an auction is initiated based on the impression about advertising opportunity up to the time when ads are displayed on the user's page usually has to be completed in a few milliseconds. This makes it imperative for the platform (Ad exchange ADX, or Demand side DSP) to keep the auction processing and communication time under a strict limit. Meanwhile, a platform usually runs a complex ML model on each advertiser to get an accurate estimate of their auction score[1]. As some platforms already have or anticipate to have in the near future an excessive number of prospective advertisers, the comprehensive ML model can only be run on a fraction $k$ of $n$ advertisers due to strict time[2] limit. In practice, the platform handles this by a two-stage selection process: it filters out all but $k$ advertisers by running a much faster and less accurate ML model, and then it runs a proper auction for the remaining $k$ advertisers using the comprehensive and slow ML model. E.g., for the ADX platforms $n$ may be in the range $20 - 50$ and $k$ depends on the specific company, e.g., $k = 10$ or $k = 20$; in the case of DSPs, $n$ may vary a lot and can reach thousands, while $k$ cannot be too large, e.g., $k = 100$ or $k = 200$.

*The authors contributed equally to this research.
The full version of this paper is available at https://arxiv.org/abs/2306.10648.

Nikolai Gravin is supported by Science and Technology Innovation 2030 "New Generation of Artificial Intelligence" Major Project No.(2018AAA0100903), National Key R&D Program of China (2023YFA1009500), NSFC grants 62150610500, 61922052, and 61932002, Innovation Program of Shanghai Municipal Education Commission, and Fundamental Research Funds for Central Universities.

*WWW '24, May 13–17, 2024, Singapore, Singapore*

ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3645418

[1] An auction score is comprised of estimates for click-through-rate (CTR), ad relevance, and value-per-click bid.

[2] In some case, the platform may also estimate the bids to avoid communication lag.

This raises multiple practical challenges for the ADX and/or DSP platforms. A platform first needs to get $n$ rough score estimates, which can be viewed as $n$ distributions of the bidders' accurate auction scores. One problem is how to obtain these estimates online in a constantly changing environment. Another learning problem, recently considered by Goel et al. [9] is how to retrieve this information from strategic agents, who might affect the selection stage by adjusting their bids. Third, there is an underlying optimization question: for $n$ prospective bidders with known independent prior distributions $(D_i)_{i\in[n]}$ how to select $k < n$ of them with the objective of either maximizing expected score (social welfare) or platform's revenue. We focus on this basic optimization problem termed the Bidder Selection Problem (BSP).

The welfare maximization BSP for the VCG single-item (or more generally $\ell$-unit) auction is equivalent to the following fundamental algorithmic question: select $k$ out of $n$ independent random variables, with the objective of maximizing the expected maximum (expected sum of top $\ell$ values). This question (for single-item) has received significant attention under different names: model-driven optimization [8], $k$-MAX [6], team selection with test scores [13], subset selection for expected maximum [16], non-adaptive Probe-Max [21], which extends to the top-$\ell$-out-of-$k$ problem.

It is also natural to consider the selection problem with a more general set of objectives given by linear combinations of the top-$\ell$ objectives. The latter problem generalizes $\ell$-unit auctions and corresponds to the widely used position auction [7, 23] environment as described, e.g., in [11]. In position auction, there are $m$ sorted positions that appear alongside the search results and $n$ advertisers competing for these $m$ slots. Each slot $j \in [m]$ has a different click-through rate $w_j$ (additional multiplicative factor for the click probability on $j$-th position), which translates into the value $v_i \cdot w_j$ for advertiser $i$ if $i$'s ad is displayed at $j$-th position.

*Prior Results for the BSP.* The basic BSP of welfare maximization (second-price auction) was shown to be NP-hard by Goel et al. [8] and later by Mehta et al. [16]. On the positive side, there are a few known Polynomial Time Approximation Schemes (PTAS). First, Chen et al. [6] gave a dynamic programming-based polynomial time approximation scheme (PTAS). Later, Mehta et al. [16] and Segev and Singla [21] respectively proposed an Efficient PTAS (EPTAS) for the BSP. Both of their approaches are based on discretizing and enumerating all possible distributions for the maximum of a few random variables: [16] characterize all $n$ distributions into one of $C(\varepsilon)$ bins (where $C(\varepsilon) = O(1/\varepsilon)^{O(1/\varepsilon)}$ depends on the approximation guarantee $1 - \varepsilon$) and then run a brute force search with the complexity of $O\left(n \cdot \log(k)^{O(C(\varepsilon))}\right)$; [21] use a complex reduction to a multi-dimensional extension of *Santa Claus* problem of [2] with at least $O(1/\varepsilon)^{O(1/\varepsilon)^{O(1/\varepsilon^2)}}$ running time[3]. We would like to emphasize that the findings of Mehta et al. and Segev and Singla could only be considered as purely computational complexity results rather than real algorithms to be used in practice or even in testing/numerical experiments. Indeed, their approaches are rather involved and not very easy to implement, e.g., Segev and Singla only state an existential result without explicitly describing their algorithm. Furthermore, the running times of these EPTASes even for the basic $k$-MAX problem and small values of $n$ and $k$ are enormously large[4]. I.e., using any of the existing PTASes would completely defeat the purpose of a two-stage selection process.

For the revenue objective, Mehta et al. [16] also considered BSP for the second-price auction, which is equivalent to maximizing the expectation of the second largest value among selected $k$ random variables. They showed strong impossibility results: it is impossible to get any constant factor approximation in polynomial time under either of the exponential time, or the planted clique hypothesises.

*Auction Formats.* One of the most commonly used formats in advertising industry is the Generalized Second Price (GSP) auction. It was shown in [7, 23] that GSP of any position auction with any set of bidders has a Nash equilibrium equivalent to the welfare-maximizing outcome of the VCG. In some cases (e.g., Google's auction for selling contextual ads [24]) the platform's auction format is based directly on the VCG. Thus, in the BSP context with the welfare-maximization objective, it is most natural to analyze the VCG mechanism. For the revenue objective, it is natural to study the optimal Myerson's auction, as it gives an upper bound on the revenue of any other auction format. Then one can reduce revenue maximization to welfare maximization of the VCG format. I.e., it is w.l.o.g. to only study welfare-maximization BSP for the VCG format, which we do in the rest of our paper. It is also natural to study revenue maximization BSP for the GSP format (or by the revenue equivalence of the VCG format). Unfortunately, it does not admit polynomial time $O(1)$-approximation even for the basic single-item auction due to strong impossibility results of [16].

## 1.1 Our Results

We propose a novel relaxation of the BSP for a more general position auction environment, which we call Poisson (or Poisson-Chernoff) relaxation. It has the following theoretical guarantees.

(1) The relaxation is a continuous maximization problem with a concave objective that can be solved in time polynomial in $n$ and $k$. In fact, the objective of this relaxation is a nicely structured algebraic function that lends itself to efficient convex minimization solvers.
(2) With small adjustments (summarized in Algorithm 1), the relaxed objective *converges* at the rate $1 - O(k^{-1/4})$ to the actual social welfare of fractional BSP as the problem size $k$ grows (Theorem 4.4). The standard rounding of this fractional solution suffers only a small loss of $O(k^{-1/2})$, yielding $\left(1 - O(k^{-1/4})\right)$-approximation (Theorem 4.5) for the integral BSP.
(3) For the special case of the single-item auction, our algorithm achieves $1 - O\left(\sqrt{\ln k/k}\right)$ approximation (see the full version).

These results have immediate theoretical implications and can be implemented in practice unlike all previous PTAS algorithms, as our approach has far superior running time to the point where it outperforms some of the existing heuristics used in practice, such as the Greedy algorithm for general submodular maximization.

[3] We give a lower bound on the running time, as it is hard to say what is the exact time complexity of their approach, as they did not specify it explicitly and their result proceeds by a sequence of reductions with non-trivial running time dependencies.

[4] Note that $k$-MAX is a special case of general submodular optimization which already admits efficient $1 - 1/e$ approximation. Hence, $\varepsilon < 1/e$ and the time complexity of $2^{\Theta(1/\varepsilon)^{\Theta(1/\varepsilon)}} > 2^{9000}$ with conservative estimates of $\Theta(1/\varepsilon) = 2/\varepsilon$.

Furthermore, our results bring new theoretical insight for BSP: the BSP converges to a polynomial-time solvable optimization problem as the problem size $k$ grows.

*Theoretical Implications.* Our Algorithm 1 gives a $(1-\varepsilon)$ approximation to the BSP for any position auction with $\varepsilon = \Omega(k^{-1/4})$ and works in polynomial time (independent of $\varepsilon$). On the other hand, for small values of $\varepsilon$ ($\varepsilon = O(k^{-1/4})$), a straightforward exhaustive search algorithm gives a perfect solution in $O(n^k) = n^{\text{poly}(1/\varepsilon)}$ time. I.e., the combination of our relaxation with the brute force algorithm yields a PTAS:

Corollary 1.1. *BSP for any position auction admits a $(1-\varepsilon)$ PTAS that runs in $n^{\text{poly}(1/\varepsilon)}$ time.*

Moreover, the algorithm as in Corollary 1.1 is an EPTAS with a much better dependency on $\varepsilon$ than previous PTASes under a mild assumption that $k \geq \log n$:

Corollary 1.2. *BSP for position auctions admits a $(1-\varepsilon)$ EPTAS for any $k \geq \log n$ that runs in $O(\text{poly}(n,k)) + 2^{O(\varepsilon^{-8})}$ time.*

Proof. Algorithm 1 runs in $O(\text{poly}(n,k))$. We run brute force only when $\varepsilon = O(k^{-1/4})$, i.e., when $k^2 = O(\varepsilon^{-8})$. Then its running time is not more than $n^k \leq 2^{k^2} = 2^{O(\varepsilon^{-8})}$, since $k \geq \log n$. □

*Comparison with Previous Theoretical Results.* Our novel Poisson approximation approach yields more general theoretical results than previous work. E.g., as Segev and Singla [21] briefly mention how their scheme can be extended from $k$-MAX (i.e., single-item auction) to $\ell$-unit auctions, one may wonder if a similar approach also extends to the more general position auction environment. To the best of our knowledge, it does not. Indeed, their main idea is to consider two regimes for $\ell$: small (a constant) $\ell < 1/\varepsilon^3$, and large $\ell > 1/\varepsilon^3$. They claim that the former case can be handled with a similar approach (maybe with a significantly worse dependency of the running time on $\varepsilon$ than the case $\ell = 1$), and in the later case, one can use concentration bounds similar to the (3) case described in Section 3.1. This argument cannot be used for the objective that is a linear combination of welfare in 1-unit and $k/2$-unit auctions.

*Relevance in Practice.* Our relaxation is a white-box approach that can be easily adapted to different scenarios. E.g., if some bidders have to be included in the final solution (which is often the case in industry because of contract obligations), then our approach gives the same approximation with these additional constraints[5]. Also, depending on the type of problem instances, a few steps of our algorithm can be removed or simplified to fit the specific domain, which results in simpler and more efficient solutions.

Furthermore, unlike the case with all previous PTASes, we actually implemented our algorithm and did numerical experiments on several generated data sets of practically relevant sizes. Note that a company cares much more about the implementability and running time of their algorithms rather than its theoretical approximation guarantees[6], which has been the main focus of previous PTAS results. In contrast, our algorithm is not very complex and is based on standard continuous convex maximization methods, which means that it is much easier to understand and adopt by a platform's product team. The experiments are summarized in Section 5. We slightly simplified our theoretical Algorithm 1 to avoid the hard-coded efficiency loss of $O(k^{-1/4})$ in the approximation.

We observed that among all tested algorithms, our algorithm produced solutions that were always within 0.1% of the best-performing algorithm in terms of objective value. Furthermore, our algorithm's running time scales significantly slower than any other tested algorithm. For large instances where $n = 1000, k = 200$, our algorithm took less than 1 minute to complete, while even Greedy (which is a subroutine in all previous PTASes) took more than 1 day and Local Search could not finish within 1 week. These findings demonstrate the practical relevance of our approach.

## 1.2 Other Related Work

Mehta et al. [16] mention a few other scenarios besides bidder selection with similar mathematical formulations. The applications range from a two-tier solution for scoring documents in a search result [5], to filtering initial proposals in procurement auctions [20, 22], to voting theory [19]. Bei at al. [3] studied BSP with the revenue objective under multiple auction formats including Myerson's auction and gave constant factor approximation for the second-price auction with anonymous reserve. They also introduced another optimization framework for the BSP under costs, which is more challenging than the BSP under capacity constraint.

Poisson approximation is a well-developed technique from probability theory and statistics. A survey [18] mentions at least twenty different results on the basic question of approximating the sum of independent Bernoulli random variables by the Poisson distribution. In statistics, Poisson approximation is commonly used in Extreme Value Theory (EVT) with applications to structural and geological engineering, traffic prediction, and finance (see, e.g., a book [17]). It has also been used in theoretical computer science, e.g., [15] used Le Cam's Poisson approximation theorem for stochastic bin packing and knapsack problems and also for EUM introduced in [14].

In fact, the expected utility maximization (EUM) is closely related to our objective. EUM is formulated as choosing a feasible subset $S$ out of $n$ random variables $X_1, \ldots, X_n$ to maximize $\mathbf{E}[u(\sum_{i\in S} X_i)]$, where $u$ is a given utility function. The problem has been studied under capacity [4] or other combinatorial constraints [14, 15, 25] with a non-linear (typically concave) utility function. The BSP for single-item auction, i.e., the $k$-MAX problem, has a similar objective $\mathbf{E}[\max_{i\in S}\{X_i\}]$ to EUM but with max operator instead of the sum. When the distributions of $X_i$ are unknown, EUM becomes an online learning problem. Chen et al. [6] gave the first PTAS for $k$-MAX, but their main focus is on Combinatorial Multi-Armed Bandits.

## 2 PRELIMINARIES

A set of $n$ bidders wish to receive some service and each bidder $i \in [n]$ has a private non-negative value $v_i \in \mathbb{R}_{\geq 0}$ indicating how much they are willing to pay for it. We denote the vector of bidder values as $\mathbf{v} = (v_i)_{i\in[n]}$. By the revelation principle, we can restrict our attention to incentive compatible and individually rational single-round auctions $\mathcal{A}$, where each bidder $i$ submits a sealed bid $b_i$ to the auctioneer. The auctioneer then decides on a feasible allocation

[5] In fact, the approximation guarantee holds not only for the global objective, but also for the surplus objective, i.e., the additional gain to welfare from the non-fixed bidders.
[6] In the context of BSP any reasonable heuristic is preferable to a complex and slow algorithm with good approximation efficiency guarantees.

vector $\mathbf{a}(\mathbf{b}) = (a_i(\mathbf{b}))_{i\in[n]}$ and payments $\mathbf{p}(\mathbf{b}) = (p_i(\mathbf{b}))_{i\in[n]}$. The incentive compatibility and individual rationality mean that by bidding truthfully $b_i = v_i$, each bidder $i \in [n]$ (a) maximizes her utility $u_i(b_i, \mathbf{b}_{\text{-}i}) \overset{\text{def}}{=} v_i \cdot a_i(b_i, \mathbf{b}_{\text{-}i}) - p_i(b_i, \mathbf{b}_{\text{-}i})$ and (b) receives non-negative utility $u_i(v_i, \mathbf{b}_{\text{-}i}) \geq 0$. The seminal VCG mechanism (a second-price auction in the case of single-item auction) is an example of incentive compatible mechanism that also maximizes *social welfare* $\mathrm{SW}(\mathbf{a}, \mathbf{v}) = \sum_{i=1}^n a_i \cdot v_i$.

We study auctions in the Bayesian setting, where we assume that bidder values are drawn independently from known prior distributions $\mathbf{v} \sim \mathbf{D} = \prod_{i\in[n]} D_i$. We also use $D_i(\tau) = \mathbf{Pr}_{v_i\sim D_i}[v_i \leq \tau]$ to denote the cumulative distribution function. The auction designer is usually concerned about two objectives: the expected *social welfare* $\mathrm{SW} = \mathbf{E}_{\mathbf{v}\sim\mathbf{D}}[\mathrm{SW}(\mathbf{a}(\mathbf{v}), \mathbf{v})]$, and *revenue* $\textsc{Rev} = \mathbf{E}_{\mathbf{v}\sim\mathbf{D}}$ $[\sum_{i\in[n]} p_i(\mathbf{v})]$. The VCG mechanism maximizes the welfare on every valuation profile $\mathbf{v}$, and thus maximizes SW in expectation for any prior $\mathbf{D}$. The well-known Myerson's auction maximizes $\textsc{Rev}$. This auction reduces the problem of revenue maximization to *virtual welfare* maximization by transforming values $(v_i)_{i\in[n]}$ to virtual values $\varphi_i(v_i)$ for regular distribution $D_i$ and by doing ironing $\overline{\varphi}_i(v_i)$ for irregular distribution $D_i$. I.e., the expected revenue of Myerson's auction can be written as $\textsc{Rev} = \mathbf{E}_{\mathbf{v}}[\mathrm{SW}(\mathbf{a}(\mathbf{v}), \varphi(\mathbf{v}))]$ for regular distributions and $\textsc{Rev} = \mathbf{E}_{\mathbf{v}}[\mathrm{SW}(\mathbf{a}(\mathbf{v}), \overline{\varphi}(\mathbf{v}))]$ for general distributions.

*Auction Environments.* The single-item auction is an environment with the feasible allocations given by $\{\mathbf{a} : \sum_{i\in[n]} a_i \leq 1\}$. A more general $\ell$-unit auction environment for $\ell \in \mathbb{N}$ is given by the feasibility constraints: $\sum_{i\in[n]} a_i \leq \ell$ and $a_i \in [0,1]$ for all $i \in [n]$. A position auction environment further generalizes $\ell$-unit auctions. It is specified by a sorted weight vector $\mathbf{w} = (1 \geq w_1 \geq w_2 \geq \ldots \geq w_n \geq 0)$, which represents the click-through rate probabilities for $n$ advert positions[7]. Every bidder may get at most one position and each advert position can be assigned to at most one advertiser. Formally, the feasibility allocations can be specified with an assignment function $\pi : [n] \to [n]$ of $n$ advertisers to $n$ sorted slots as follows: $\{\mathbf{a} : \exists\pi,\ a_i \in [0, w_{\pi(i)}]$ for all $i \in [n]\}$.

*Bidder Selection.* In the Bidder Selection Problem (BSP) the seller first decides $x_i \in \{0,1\}$ which bidders $i \in [n]$ to invite to the auction. The selected set of bidders $S$ may not exceed a certain capacity $k \geq |S|$, i.e., $\sum_{i\in[n]} x_i \leq k$. Then the auctioneer runs an optimal auction $\mathcal{A}$ for the set $S$ of invited bidders: VCG mechanism for the welfare objective, and Myerson for the revenue objective. Since revenue of the Myerson's auction can be rewritten as the expected virtual welfare with independent (ironed) virtual values $(\varphi_i(v_i) : v_i \sim D_i)_{i\in S}$, the BSP for the revenue maximization is equivalent to the BSP for the welfare maximization. Thus it suffices to only consider the welfare maximization problem. Specifically, we denote by $\mathbf{v} \sim \mathbf{x} \cdot \mathbf{D}$ the independent draws of $v_i \sim x_i \cdot D_i$ (i.e., $v_i \sim D_i$ when $x_i = 1$, and $v_i = 0$ when $x_i = 0$) for all $i \in [n]$, then the BSP of the VCG mechanism for any $\ell$-unit/position auction with weights $\mathbf{w}$ can be written as follows:

$$\mathrm{OPT}(\ell) \overset{\text{def}}{=} \max_{\substack{\mathbf{x}\in\{0,1\}^n \\ |\mathbf{x}|_1 \leq k}} \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}\left[\sum_{i=1}^{\ell} v_{(i)}\right],$$

$$\mathrm{OPT}(\mathbf{w}) \overset{\text{def}}{=} \max_{\substack{\mathbf{x}\in\{0,1\}^n \\ |\mathbf{x}|_1 \leq k}} \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}\left[\sum_{i=1}^{n} w_i \cdot v_{(i)}\right],$$

where $v_{(i)}$ is the $i$-th largest value among $\{v_i\}_{i\in[n]}$. We want to obtain good approximation algorithms for these BSPs. I.e., we would like to find in polynomial time $\mathbf{x} \in \{0,1\}^n$ with $|\mathbf{x}|_1 \leq k$ such that $\mathbf{E}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}[\sum_{i=1}^{\ell} v_{(i)}] \geq (1-\varepsilon)\,\mathrm{OPT}(\ell)$ and $\mathbf{E}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}[\sum_{i=1}^{n} w_i \cdot v_{(i)}] \geq (1-\varepsilon)\,\mathrm{OPT}(\mathbf{w})$ for a small $\varepsilon > 0$. To simplify the presentation, we assume that all distributions $(D_i)_{i\in[n]}$ have finite supports and are given explicitly as the algorithm's input.

It has been observed before that the BSP's objective is a monotone submodular function of the set $S = \{i : x_i = 1\}$ of invited bidders, i.e., $\mathrm{SW}(S) + \mathrm{SW}(T) \geq \mathrm{SW}(S \cap T) + \mathrm{SW}(S \cup T)$ for any $S, T \subseteq [n]$. The same property holds for the $\ell$-unit and position auctions with an almost identical proof (see [6]).

We also consider the standard (in submodular optimization literature) multi-linear extension of the BSP objective. I.e., for a fractional $\mathbf{x} \in [0,1]^n$, we invite each bidder $i$ to the auction independently with probability $x_i$. We employ the same notation $\mathbf{v} \sim \mathbf{x} \cdot \mathbf{D}$ as in the integral problem, where $v_i \sim x_i \cdot D_i$ means that we first decide whether to invite $\xi_i \in \{0,1\}$ bidder $i$ according to a Bernoulli distribution $\xi_i \sim \mathrm{Ber}(x_i)$, then draw their value $v_i \sim \xi_i \cdot D_i$. The capacity constraint transforms into the bound on the expected number of invited bidders $\sum_{i=1}^n x_i \leq k$. We describe this new mathematical formulation in Section 3. In Section 4, we first consider this fractional relaxation of BSP and we discuss in Section 4.2 how to obtain a good solution to the integral BSP from the fractional problem.

## 3 NEW MATHEMATICAL FORMULATION

In this section, we give a fractional relaxation of BSP. As the welfare of any position auction can be written as a linear combination of $\ell$-unit auctions, we begin with a fractional relaxation of BSP for the $\ell$-unit auction. Specifically, the expected social welfare with a fractional set of bidders $\mathbf{x}$ is $\mathrm{SW}(\mathbf{x}, \ell) = \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}[\sum_{i=1}^{\ell} v_{(i)}]$, where $v_{(i)}$ denotes the $i$-th largest value among $\{v_i\}_{i\in\{1,2,\ldots,n\}}$. Hence, the following mathematical program represents BSP for $\ell$-unit auction:

$$\begin{array}{ll} \text{Maximize} & \mathrm{SW}(\mathbf{x}, \ell) \\ \text{Subject To} & \sum_{i=1}^n x_i \leq k, \quad x_i \in [0,1], \forall i \in \{1,2,\ldots,n\}. \end{array} \tag{1}$$

As $\ell$ is fixed, whenever it is clear from the context, we will simply write $\mathrm{SW}(\mathbf{x})$.

*Bernoulli Representation.* We now derive an explicit formula for the social welfare $\mathrm{SW}(\mathbf{x})$. Fixing a threshold $\tau$, the expected number of bidders with values exceeding $\tau$ among the highest $\ell$ bidders is $\mathbf{E}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}[\min(\sum_{i=1}^n \mathbb{I}[v_i \geq \tau], \ell)]$. Therefore, by integrating[8] over $\tau \in [0, +\infty)$, we get

$$\mathrm{SW}(\mathbf{x}) = \int_0^{+\infty} \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}\left[\min\left(\sum_{i=1}^n \mathbb{I}\Big[v_i \geq \tau\Big],\ \ell\right)\right] \mathrm{d}\tau.$$

[7] The number of available positions $m$ is usually smaller than the number of bidders $n$, in which case we simply let $w_{m+1} = \cdots = w_n = 0$.

[8] The function inside the integral is piece-wise constant, i.e., it is constant between consecutive values of the threshold $\tau$ in the supports of $\{D_i\}_{i\in[n]}$.

Note that $\mathbb{I}[v_i \geq \tau]$ for $v_i \sim x_i \cdot D_i$ is a Bernoulli random variable, and $\mathbf{E}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}[\min(\sum_{i=1}^n \mathbb{I}[v_i \geq \tau], \ell)]$ is the minimum of a sum of independent Bernoulli random variables and $\ell$. To simplify notations, we explicitly define the probabilities of Bernoulli random variables $\mathbb{I}[v_i \geq \tau]$:

$$q_i(x_i, \tau) \overset{\text{def}}{=\!=} \Pr_{v_i \sim x_i \cdot D_i}[v_i \geq \tau] = x_i \cdot (1 - D_i(\tau))$$

$$\text{and let} \quad \mathbf{q}(\mathbf{x}, \tau) \overset{\text{def}}{=\!=} (q_i)_{i\in[n]}. \tag{2}$$

*Definition 3.1 (Bernoulli Objective).* For a vector $\mathbf{q} \in [0,1]^n$ and $\ell$, the **Bernoulli objective term** of $\mathbf{q}$ and $\ell$ is a function $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \ell)$:

$$\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \ell) \overset{\text{def}}{=\!=} \mathop{\mathbf{E}}_{\mathbf{z}\sim\mathrm{Ber}(\mathbf{q})}\left[\min\left(\sum_{i=1}^n z_i,\ \ell\right)\right],$$

$$\text{then} \quad \mathrm{SW}(\mathbf{x}) = \int_0^{+\infty} \mathrm{H}_{\mathrm{ber}}(\mathbf{q}(\mathbf{x}, \tau), \ell)\, \mathrm{d}\tau. \tag{3}$$

*Position Auctions.* A position auction is given by a vector of non-negative weights[9] $\mathbf{w} : (w_1 \geq w_2 \geq \cdots \geq w_n \geq 0)$. The highest social welfare we get from the set $S$ of invited bidders is $\sum_{i=1}^{|S|} v_{(i)} \cdot w_i$, where $v_{(1)} \geq \cdots \geq v_{(|S|)}$ are ordered values of bidders in $S$. Thus the expected social welfare for a fractional set $\mathbf{x}$ is

$$\mathrm{SW}(\mathbf{x}, \mathbf{w}) = \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathbf{x}\cdot\mathbf{D}}\left[\sum_{i=1}^n v_{(i)} \cdot w_i\right] = \sum_{\ell=1}^n (w_\ell - w_{\ell+1}) \cdot \mathrm{SW}(\mathbf{x}, \ell),$$

where $w_{n+1} \overset{\text{def}}{=\!=} 0$. Then the respective fractional BSP program for position auctions is as follows.

$$\begin{array}{ll} \text{Maximize} & \mathrm{SW}(\mathbf{x}, \mathbf{w}) \\ \text{Subject To} & \sum_{i=1}^n x_i \leq k, \quad x_i \in [0,1], \forall i \in \{1, 2, \ldots, n\}. \end{array} \tag{4}$$

We will often omit dependency on $\mathbf{w}$ in SW whenever it is clear from the context. We further consider the *Bernoulli representation* for position auctions:

$$\mathrm{SW}(\mathbf{x}, \mathbf{w}) = \int_0^{+\infty} \mathrm{H}_{\mathrm{ber}}(\mathbf{q}(\mathbf{x}, \tau), \mathbf{w})\, \mathrm{d}\tau,$$

$$\text{where} \quad \mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \mathbf{w}) \overset{\text{def}}{=\!=} \sum_{\ell=1}^n (w_\ell - w_{\ell+1})\, \mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \ell). \tag{5}$$

As in (2), $\mathbf{q}(\mathbf{x}, \tau)$ represents the probabilities of each bidder's value exceeding $\tau$. $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \mathbf{w})$ is called the **Bernoulli objective term** for position auctions.

## 3.1 Overview of Our Approach

The fractional relaxation (4) is still neither convex nor concave and thus is too unwieldy. The central idea of our paper is to use instead *Poisson approximation* to the Bernoulli objective terms $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}(\mathbf{x}, \tau), \mathbf{w})$. Specifically, we substitute each Bernoulli random variable $z \sim \mathrm{Ber}(p)$ with $p = q_i(x_i, \tau)$ by the Poisson random variable $y \sim \mathrm{Pois}(p)$ with the same expectation as $z$. We use the following Poisson objective term $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ to approximate $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \ell)$.

$$\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell) \overset{\text{def}}{=\!=} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathrm{Pois}(\mathbf{q})}\left[\min\left(\sum_{i=1}^n y_i,\ \ell\right)\right] = \mathop{\mathbf{E}}_{Y\sim\mathrm{Pois}(\sum_{i=1}^n q_i)}[\min(Y,\ \ell)].$$

The advantage of the Poisson approximation $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ is that the resulting functions $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ and $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \mathbf{w})$ are *concave* in $\mathbf{q}$. This in turn allows us to efficiently solve the optimization problem for the Poisson approximation analogous to (4). This Poisson approximation works well[10] in the following situations.

1. When the probabilities $p_i = q_i(x_i, \tau)$ of $z_i \sim \mathrm{Ber}(p_i)$ are small (i.e., $p_i \leq \delta, \forall i \in [n]$). This allows us to handle the crucial contribution to the welfare comprised of small probability tail events for large thresholds $\tau$.
2. When $\mathbf{E}[\sum_{i=1}^n z_i]$ is large (when thresholds $\tau$ are small). Indeed, by Chernoff bounds the sums of independent random variables (both Bernoulli and Poisson) $\sum_{i=1}^n z_i$ and $\sum_{i=1}^n y_i$ are close to their expectations. In fact, we simply use Chernoff objective term $\mathrm{H}_{\mathrm{cher}}(\mathbf{q}, \ell) \overset{\text{def}}{=\!=} \min\left(\sum_{i=1}^n q_i,\ \ell\right)$ instead of Poisson approximation in this case.
3. When $\ell$ is large. In this case, either the concentration inequality gives a good approximation when $\mathbf{E}[\sum_{i=1}^n z_i] = \mathbf{E}[\sum_{i=1}^n y_i]$ is large, or when this expectation is much smaller than $\ell$ then the probability that either of $\sum_{i=1}^n z_i$ or $\sum_{i=1}^n y_i$ exceeds the threshold $\ell$ is small.

Then the algorithmic framework for the BSP is rather straightforward. We need to solve the following *concave* program of $\mathbf{x}$:

$$\begin{array}{ll} \text{Maximize} & \sum_{\ell=1}^n (w_\ell - w_{\ell+1}) \int_0^{+\infty} \mathrm{H}_{\mathrm{pois}}(\mathbf{q}(\mathbf{x}, \tau), \ell)\, \mathrm{d}\tau \\ \text{Subject To} & \sum_{i=1}^n x_i \leq k, \quad x_i \in [0,1], \forall i \in \{1, 2, \ldots, n\}. \end{array} \tag{6}$$

We define the approximate social welfare $\widetilde{\mathrm{SW}}(\mathbf{x}, \mathbf{w})$ to be the objective of (6). Here, $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}(\mathbf{x}, \tau), \ell) = \mathbf{E}_{Y\sim\mathrm{Pois}(\sum q_i)}[\min(Y,\ \ell)]$ and all functions under the integral are piece-wise constant with the number of pieces bounded by the size of the union of the supports of $D_i$. The optimal fractional solution $\mathbf{x}$ can be computed rather efficiently using continuous convex optimization. Finally, we use a rounding procedure (similar to the multi-linear extension of submodular function) to the fractional solution of (6) to obtain an integral solution with a small loss to the approximation guarantee.

# 4 OUR ALGORITHM

*Poisson approximation* is the central idea of this paper. We use the Poisson Objective to approximate Bernoulli objective terms.

*Definition 4.1 (Poisson Objective).* For $\mathbf{q} \in [0,1]^n$ and $\ell \in \mathbb{N}$, the **Poisson objective term** is a function $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ given by

$$\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell) \overset{\text{def}}{=\!=} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathrm{Pois}(\mathbf{q})}\left[\min\left(\sum_{i=1}^n y_i,\ \ell\right)\right] = \mathop{\mathbf{E}}_{Y\sim\mathrm{Pois}(\sum_{i=1}^n q_i)}[\min(Y,\ \ell)].$$

Note that the latter equality is an important property of Poisson distribution: the sum of independent Poisson random variables $y_i \sim \mathrm{Pois}(q_i)$ follows the Poisson distribution $\mathrm{Pois}(\sum_{i=1}^n q_i)$. Another crucial property is the concavity of Poisson approximation.

Claim 4.2 (Concavity of Poisson). *$\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ is a concave function in $\mathbf{q} \in [0,1]^n$ for $\forall \ell \in \mathbb{N}$.*

[9] Usually weights are only for the first $k$ slots, as we only select a set of $k$ bidders. In this case, we simply assume that $w_{k+1} = \ldots = w_n = 0$.

[10] The sum of Poisson random variables approaches the sum of Bernoulli random variables in TV and other statistical distances.

Proof. Let $\lambda = \sum_{i=1}^{n} q_i$. Notice that $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ only depends on $\lambda$, which is linear in $\mathbf{q}$. We use $\mathrm{H}_{\mathrm{pois}}(\lambda)$ to represent this function. Then, we only need to prove that $\mathrm{H}_{\mathrm{pois}}(\lambda)$ is concave in $\lambda$. Rewrite

$$\mathrm{H}_{\mathrm{pois}}(\lambda) = \ell - \sum_{j=0}^{\ell-1} \Pr_{Y\sim\mathrm{Pois}(\lambda)}[Y = j] \cdot (\ell - j) = \ell - \sum_{j=0}^{\ell-1} \frac{\lambda^j}{j!} e^{-\lambda} \cdot (\ell - j).$$

By a straightforward differentiation of the partial series we get

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\mathrm{H}_{\mathrm{pois}} = \sum_{j=0}^{\ell-1} \frac{\lambda^j}{j!} e^{-\lambda}, \qquad \frac{\mathrm{d}^2}{\mathrm{d}\lambda^2}\mathrm{H}_{\mathrm{pois}} = -e^{-\lambda}\frac{\lambda^{\ell-1}}{(\ell-1)!} < 0.$$

Therefore, $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ is concave in $\lambda$, and thus concave in $\mathbf{q}$. □

*Chernoff Approximation.* As we mentioned earlier Poisson approximation is useful for small probabilities $q_i \le \delta$. For another extreme case where $\lambda = \sum_{i=1}^{n} q_i$ is large, due to concentration bounds, Poisson approximation also works well with $O(\lambda^{-1/2})$ relative error, because both $\sum_{i=1}^{n} \mathrm{Ber}(q_i)$ and $\mathrm{Pois}(\lambda)$ concentrate around $\lambda$. To optimize the presentation for an easier understanding, we will use **Chernoff objective term** $\mathrm{H}_{\mathrm{cher}}(\mathbf{q}, \ell) \overset{\mathrm{def}}{=\!=} \min(\sum_{i=1}^{n} q_i, \ell)$ as an alternative of Poisson in this case. It is also concave in $\mathbf{q}$.

In a number of cases $\mathrm{H}_{\mathrm{pois}}(\mathbf{q}, \ell)$ and $\mathrm{H}_{\mathrm{cher}}(\mathbf{q}, \ell)$ are good approximations to $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \ell)$. In the full version of this paper, we present these approximation guarantees along with a simple algorithm that illustrates our approach in an important special case.

## 4.1 Algorithm for Position Auctions

Below, we consider Bidder Section Problem for position auctions. We first analyze the fractional BSP, and then explain in Section 4.2 how to do integral rounding of the fractional solution with only a small loss to the approximation guarantee. For the fractional BSP, we give an efficient polynomial time $(1 - O(k^{-4}))$-approximation algorithm (see Theorem 4.4 for the exact statement).

Recall that the Bernoulli objective term for position auctions $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \mathbf{w})$ is defined in Section 3. The expected welfare $\mathrm{SW}(\mathbf{x}) = \mathrm{SW}(\mathbf{x}, \mathbf{w})$ is written as an integral of $\mathrm{H}_{\mathrm{ber}}$. Below, we will also need the Chernoff objective term

$$\mathrm{H}_{\mathrm{cher}}(\mathbf{q}, \mathbf{w}) \overset{\mathrm{def}}{=\!=} \sum_{\ell=1}^{n} (w_\ell - w_{\ell+1})\, \mathrm{H}_{\mathrm{cher}}(\mathbf{q}, \ell).$$

We sometimes slightly abuse notations and write $\mathrm{H}_{\mathrm{ber}}(\mathbf{x}, \tau)$ and $\mathrm{H}_{\mathrm{cher}}(\mathbf{x}, \tau)$ instead of $\mathrm{H}_{\mathrm{ber}}(\mathbf{q}, \mathbf{w})$ and $\mathrm{H}_{\mathrm{cher}}(\mathbf{q}, \mathbf{w})$.

In order to effectively use the Poisson approximation we would like to have the small probability assumption $q_i \le \delta$, which is achieved by fixing a small bidder set $S_{\mathrm{fix}}$ properly (see the full version for formal approximation guarantees).

*Fixing Small Bidder Set.* We will fix a small set of bidders $S_{\mathrm{fix}}$ (set $x_i = 1$ for $i \in S_{\mathrm{fix}}$) with $|S_{\mathrm{fix}}| = \varepsilon \cdot k$ and make sure that all other bidders $i \notin S_{\mathrm{fix}}$ have only a small probability $\Pr[v_i \ge \tau] \le \delta$ to exceed any of the thresholds $\tau > \eta$ for certain $\eta > 0$. This allows us to use Poisson approximation for the high range thresholds $\tau > \eta$ and bidders $i \in M \overset{\mathrm{def}}{=\!=} [n] \setminus S_{\mathrm{fix}}$. On the other hand, for the low range thresholds $\tau \le \eta$, we would like to see a certain number $\ell^*$ of bidders $i \in S_{\mathrm{fix}}$ to exceed the threshold $v_i \ge \tau$. To this end, we choose $S_{\mathrm{fix}}$ so that the expected number of bidders $i \in S_{\mathrm{fix}}$ with $v_i \ge \tau$ is at least $\ell^*$. We can achieve the following guarantees for $\varepsilon, \delta$, and $\ell^*$.

Claim 4.3 (Small Bidder Set). *Let $\varepsilon \in [\frac{\ell^*}{\delta \cdot k}, 1)$ be a multiple of $1/k$ for $\ell^* \in \mathbb{R}_{\ge 0} : \ell^* < k$ and $\delta \in (0, 1)$. We can find in polynomial time a threshold $\eta \ge 0$ and a set $S_{\mathrm{fix}} \subseteq [n]$ of size $|S_{\mathrm{fix}}| = \varepsilon \cdot k$:*

(a) $\forall\, 0 \le \tau \le \eta, \quad \sum_{i \in S_{\mathrm{fix}}} \Pr_{v_i \sim D_i}[v_i \ge \tau] \ge \ell^*;$

(b) $\forall\, i \notin S_{\mathrm{fix}}, \quad \Pr_{v_i \sim D_i}[v_i > \eta] < \delta.$

Proof. We search through all thresholds $\tau$ in the supports of $\{D_i\}_{i \ge 1}$ and find two consecutive threshold values $\eta$ and $\eta_+ > \eta$ such that $|\{i : \Pr[v_i \ge \eta] \ge \delta\}| \ge \varepsilon \cdot k$, but a similar number of bidders $|\{i : \Pr[v_i > \eta] = \Pr[v_i \ge \eta_+] \ge \delta\}| < \varepsilon \cdot k$ for the next value $\eta_+$. We place each bidder $i$ with $\Pr[v_i > \eta] \ge \delta$ into $S_{\mathrm{fix}}$ and fill the remaining positions in $S_{\mathrm{fix}}$ up to $\varepsilon \cdot k$ with bidders from $\{i : \Pr[v_i \ge \eta] \ge \delta > \Pr[v_i \ge \eta_+]\}$.

Thus, every bidder $i \notin S_{\mathrm{fix}}$ has $\Pr[v_i > \eta] = \Pr[v_i \ge \eta_+] < \delta$ as required by condition (b). On the other hand, $|S_{\mathrm{fix}}| = \varepsilon \cdot k$ and $\Pr[v_i \ge \eta] \ge \delta$ for every $i \in S_{\mathrm{fix}}$, which implies (a), since $\forall \tau \le \eta$,

$$\sum_{i \in S_{\mathrm{fix}}} \Pr_{v_i \sim D_i}[v_i \ge \tau] \ge \sum_{i \in S_{\mathrm{fix}}} \Pr_{v_i \sim D_i}[v_i \ge \eta] \ge \delta \cdot |S_{\mathrm{fix}}| = \delta \cdot \varepsilon \cdot k \ge \ell^*.$$

Thus we constructed in polynomial time the desired threshold $\eta$ and set $S_{\mathrm{fix}}$. □

We need to balance three parameters $\ell^*$, $\delta$, and $\varepsilon$, which must satisfy the conditions of Claim 4.3. Specifically, we choose $\ell^* = k^{1/2}$, $\varepsilon = k^{-1/4}$ rounded up to a multiple of $1/k$, and $\delta = \varepsilon \ge k^{-1/4}$. Claim 4.3 leads to Algorithm 1 (which we will present shortly).

For thresholds $\tau > \eta$, any bidder outside $S_{\mathrm{fix}}$ has only $\le \delta$ probability to exceed the threshold, which is ideal for applying the Poisson approximation. Therefore, to achieve the approximation guarantees when the probability of exceeding the threshold is low, we recalculate the **adjusted Poisson objective term** by applying Poisson approximation only on these bidders. Specifically, we let $M \overset{\mathrm{def}}{=\!=} [n] \setminus S_{\mathrm{fix}}$ and define $Z_{\mathrm{fix}}(\tau) \overset{\mathrm{def}}{=\!=} \sum_{i \in S_{\mathrm{fix}}} \mathbb{I}[v_i \ge \tau]$ (as $\mathbf{x}_{S_{\mathrm{fix}}} = \mathbf{1}_{S_{\mathrm{fix}}}$, random variable $Z_{\mathrm{fix}}(\tau)$ has Poisson binomial distribution). Indeed, we may calculate all probabilities $\Pr[Z_{\mathrm{fix}}(\tau) = j]$ for each $j \in [0, \varepsilon \cdot k]$ in polynomial time, and define the adjusted Poisson objective term as a conditional expectation depending on $Z_{\mathrm{fix}}(\tau)$:

$$\mathrm{G}_{\mathrm{pois}}(\mathbf{x}_M, \tau) \overset{\mathrm{def}}{=\!=} \sum_{j=0}^{|S_{\mathrm{fix}}|} \Pr[Z_{\mathrm{fix}}(\tau) = j] \cdot \Bigg( \sum_{\ell=1}^{j} w_\ell + \sum_{\ell=j+1}^{n} (w_\ell - w_{\ell+1}) \cdot \mathrm{H}_{\mathrm{pois}}(\mathbf{x}_M, \ell - j, \tau) \Bigg), \qquad (7)$$

where $\mathrm{H}_{\mathrm{pois}}(\mathbf{x}_M, \ell - j, \tau) = \mathop{\mathbf{E}}_{Y\sim\mathrm{Pois}(\lambda_M)}[\min\{Y, \ell - j\}]$, and $\lambda_M = \sum_{i \in M} q_i(x_i, \tau)$. For the low-range thresholds $\tau \le \eta$, we use the Chernoff objective $\mathrm{H}_{\mathrm{cher}}(\mathbf{x}, \tau) = \mathrm{H}_{\mathrm{cher}}(\mathbf{q}(\mathbf{x}, \tau), \mathbf{w})$ to approximate $\mathrm{H}_{\mathrm{ber}}(\mathbf{x}, \tau) = \mathrm{H}_{\mathrm{ber}}(\mathbf{q}(\mathbf{x}, \tau), \mathbf{w})$ for $\mathbf{x} = (\mathbf{x}_M, \mathbf{1}_{S_{\mathrm{fix}}})$. Importantly, unlike the high-range thresholds, we do not recalculate $\mathrm{H}_{\mathrm{cher}}$ as a function of $\mathbf{x}_M$, but use Chernoff approximation for the entire $n$-dimensional vector $\mathbf{x} = (\mathbf{x}_M, \mathbf{x}_{S_{\mathrm{fix}}})$ with $\mathbf{x}_{S_{\mathrm{fix}}} = \mathbf{1}_{S_{\mathrm{fix}}}$. Thus, as $\ell^* = k^{1/2}$ grows with $k$, $\mathrm{H}_{\mathrm{cher}}(\mathbf{x}, \tau) \to \mathrm{H}_{\mathrm{ber}}(\mathbf{x}, \tau)$.

```
Algorithm 1: Fractional BSP for Position Auctions
Let $\ell^* = k^{1/2}$; $\varepsilon$ be $k^{-1/4}$ rounded up to a multiple of $1/k$ $\left(\varepsilon = \frac{\lceil k \cdot k^{-1/4} \rceil}{k}\right)$; $\delta = \varepsilon$.
(1) Find $\eta$ and $S_{\text{fix}}$ according to Claim 4.3. Set $x_i = 1$ for $\forall i \in S_{\text{fix}}$. Let $M \overset{\text{def}}{=} [n] \setminus S_{\text{fix}}$.
(2) Define the approximate welfare using adjusted Poisson objective (7):
        $\widetilde{\text{SW}}(\mathbf{x}_M) \overset{\text{def}}{=} \int_0^{\eta} \text{H}_{\text{cher}}((\mathbf{x}_M, \mathbf{1}_{S_{\text{fix}}}), \tau)\, \mathrm{d}\tau$
            $+ \int_{\eta}^{+\infty} \text{G}_{\text{pois}}(\mathbf{x}_M, \tau)\, \mathrm{d}\tau,$    (8)
(3) Return $\widetilde{\mathbf{x}}^* = (\widetilde{\mathbf{x}}^*_M, \mathbf{1}_{S_{\text{fix}}})$, where $\widetilde{\mathbf{x}}^*_M$ is the solution to the concave program in $\mathbf{x}_M$:
        Maximize      $\widetilde{\text{SW}}(\mathbf{x}_M)$
        Subject To    $\sum_{i \in M} x_i \le k - \varepsilon \cdot k, \quad x_i \in [0,1]\ \forall i \in M.$    (9)
```

*Algorithm.* Our main algorithm is summarized as Algorithm 1.

Theorem 4.4. *Algorithm 1 works in polynomial time and is a* $(1 - O(\varepsilon))$*-approximation, i.e.,* $(1 - 43\, k^{-1/4})$*-approximation to the fractional BSP for any position auction.*

The proof of this theorem is deferred to the full version.

## 4.2 Rounding

We conclude Section 4 by presenting the rounding algorithm, which takes our solution $\widetilde{\mathbf{x}}^*$ to the fractional BSP produced by Algorithm 1 and returns a solution to the integral BSP.

Our fractional relaxation works as the standard multi-linear extension of submodular functions, which corresponds to sampling a random set of bidders $S \sim \prod_{i=1}^{n} \text{Ber}(x_i)$ in the integral BSP. To align the notations for fractional and integral BSPs, we shall use vectors $\mathbf{y}, \mathbf{z} \in \{0,1\}^n$ for the respective sets of selected bidders. Specifically, we use $\mathbf{y} \sim \text{Ber}(\mathbf{x})$ to represent the random set $S$ in the multi-linear extension. Our rounding procedure is in Algorithm 2.

```
Algorithm 2: Rounding: algorithm for Integral BSP
(1) Run Algorithm 1 to obtain a fractional solution $\mathbf{x}$.
(2) Sample an integral solution $\mathbf{y} \sim \text{Ber}(\mathbf{x})$, with $\mathbf{y} \in \{0,1\}^n$.
(3) • If $|\mathbf{y}|_1 \le k$, return $\mathbf{z} = \mathbf{y}$,
    • Else ($|\mathbf{y}|_1 > k$), return $k$ bidders $\mathbf{z} \sim \binom{\mathbf{y}}{k}$ chosen uniformly at random from $\mathbf{y}$.
```

Theorem 4.5. *Algorithm 2 works in polynomial time and in expectation is a* $\left(1 - 43\, k^{-1/4} - O(k^{-1/2})\right)$*-approximation to the integral BSP for any position auction.*

The proof of Theorem 4.5 is deferred to the full version. Since $\text{SW}(\mathbf{z}) \le \text{OPT}$, by running the rounding algorithm a few times and taking the best produced solution, we get a slightly worse approximation guarantee of $(1 - O(k^{-1/4}))\text{OPT}$ with high probability.

# 5 NUMERICAL EXPERIMENTS

We focus on testing welfare maximization Bidder Selection Problem for position auctions. In our experiments, we used synthetically generated prior distributions, as (1) the BSP is a pure optimization problem, which ignores the issue of data retrieval (2) due to companies' strict nondisclosure rules, it is infeasible to experiment on real historical data. We generally followed the AuctionGym [12] setup, a popular online simulation environment for advertising auctions run by Amazon, in the design of our synthetic data.

*Implementation.* We implemented Algorithms 1 and 2 as well as its modified version for BSP. In this simpler modification, we used a slightly different objective $\widetilde{\text{SW}}(\mathbf{x}) \overset{\text{def}}{=} \int_0^{+\infty} \text{H}_{\text{pois}}(\mathbf{x}, \tau)\, \mathrm{d}\tau$ than (8): we did not fix any small bidder set, only applied Poisson approximation, and used the same rounding step. The original Algorithm 1 was designed with the worst-case theoretical guarantees in mind, while the modified one is more practically oriented and retains only the most important Poisson approximation. The modification did not affect the run-time much, but allowed us to avoid hard-coded approximation loss of $O(k^{-1/4})$ due to the potentially suboptimal decision of fixing a small bidder set $S_{\text{fix}}$. We implemented the practical variant in Python with the help of Gurobi [10], a well-known commercial convex optimization solver, and present its comparison[11] with benchmark algorithms in Table 1.

*Benchmarks.* Ideally, we would like to compare our solutions to the optimum, which is usually not possible, as BSP is an NP-hard problem even for the case of single-item auction [8]. It is also infeasible to use any of the existing PTAS algorithms, as only [6] implemented their PTAS but could only run experiments on tiny input sizes of ($n = 9, k = 3$), while EPTASes of [16, 21] are pure computational complexity results with unrealistically large estimates on running times for inputs of any size. Instead, we used two well-known heuristic algorithms as our benchmarks: **Greedy** for submodular maximization following numerical experiments in [6, 16], and **Local Search** mentioned in [3]. They are easy to implement and run in feasible times on most of our datasets. Note that if Local Search starts with the solution produced by Greedy, it can only improve upon it, i.e., it seems reasonable to use Local Search as a main reference point for approximation efficiency guarantees. We also tested Local Search against the exact optimum computed by Brute Force on small instances ($n = 50, k = 5$) and found that they always produced the same results. We implemented Greedy, Local Search, and Brute Force in Python to ensure a fair comparison with our algorithm. We also limited the number of threads used by our algorithm to 1, as the benchmark algorithms are not parallelizable.

*Datasets.* We generated each of the $n$ prior distributions as a log-normal distribution $\text{Lognormal}(\mu, \sigma^2)$ as in AuctionGym [12]. We selected parameters $\mu$ and $\sigma$ of each individual distribution by

[11] We also compared the performances of the modified version and the theoretical version of our algorithm in Appendix A.1. The approximation efficiency of the theoretical version was worse than the modified version as expected.

**Table 1: Experimental results of Local Search, Greedy, and our algorithm. The "solution" column for each algorithm denotes the average relative quality of the produced solution to that of the best-performing algorithm that terminated in 1 week. Error bars denote the standard deviation. The "time" column denotes the average running time of each algorithm.**

| Setting | | Local Search | | Greedy | | Our Algorithm | |
|---|---|---|---|---|---|---|---|
| $n$ | $k$ | Solution | Time | Solution | Time | Solution | Time |
| 50 | 5 | 100.00% ± 0.00% | 1.01 seconds | 98.93% ± 0.50% | **0.21 seconds** | 99.99% ± 0.03% | 1.29 seconds |
| | 10 | 100.00% ± 0.00% | 6.05 seconds | 98.71% ± 0.38% | **1.11 seconds** | 99.99% ± 0.04% | 1.59 seconds |
| | 20 | 100.00% ± 0.00% | 65.23 seconds | 99.17% ± 0.28% | 9.02 seconds | 99.99% ± 0.01% | **2.58 seconds** |
| 200 | 10 | 100.00% ± 0.00% | 1.02 minutes | 98.06% ± 0.35% | 8.10 seconds | 99.99% ± 0.01% | **2.83 seconds** |
| | 20 | 100.00% ± 0.00% | 10.83 minutes | 97.92% ± 0.24% | 44.72 seconds | 100.00% ± 0.00% | **3.88 seconds** |
| | 40 | 100.00% ± 0.00% | 60.66 minutes | 97.97% ± 0.26% | 170.29 seconds | 99.99% ± 0.00% | **3.97 seconds** |
| 1000 | 50 | 100.00% ± 0.00% | 33.71 hours | 97.18% ± 0.14% | 0.90 hours | 99.99% ± 0.00% | **21.54 seconds** |
| | 100 | N/A | > 1 week | 97.15% ± 0.15% | 4.26 hours | 100.00% ± 0.00% | **41.21 seconds** |
| | 200 | N/A | > 1 week | 97.38% ± 0.11% | 27.65 hours | 100.00% ± 0.00% | **45.16 seconds** |

drawing them independently from continuous uniform distributions $\mathcal{U}[0, 0.2]$ and $\mathcal{U}[0, 0.5]$, respectively.[12] We then discretized each distribution to a common, finite support $\{0\} \cup \{1 + \frac{i}{50} \mid i = 0, 1, \ldots, 49\}$ by moving probability mass on each discretized interval inside $[0, 2]$ to its left point and by redistributing the mass on $(2, +\infty)$ to the discrete points, proportional to their respective probabilities. The weights $\mathbf{w}$ of the position auction on each instance were set as: $w_i = 1$ for $i \in [1, 0.2k]$, $w_i = 0.2$ for $i \in (0.2k, 0.6k]$, and $w_i = 0$ for $i \in (0.6k, k]$. We constructed datasets with 3 different $n \in \{50, 200, 1000\}$. For each $n$, we used 3 different values of $k$: for $n = 50$, we set $k \in \{5, 10, 20\}$; for $n = 200$, we set $k \in \{10, 20, 40\}$; and for $n = 1000$, we set $k \in \{50, 100, 200\}$. The general idea was to capture practically relevant scenarios of different scales, and also have our benchmarks solve them in a reasonable time. Moreover, we picked $k$ so that it is always significantly smaller than $n$. Note that this puts our algorithm at a disadvantage, as our Poisson relaxation gets more accurate as $k$ grows.

*Results.* The numerical experiments are given in Table 1. We ran Local Search, Greedy, and our algorithm on all 9 combinations of $n$ and $k$. We recorded the approximate efficiency ("solution" column) and the running time of each algorithm (if an algorithm did not terminate in 1 week, we would stop it and write "N/A" for the respective dataset). We measure efficiency as the relative quality of the produced solution with respect to the solution of the best-performing algorithm that terminated in 1 week on that test case.

(1) As shown in Table 1, Greedy performs surprisingly well: on each generated dataset, it produced solutions within 5% of the best-performing algorithm. It is much better than the $(1 - 1/e)$-approximation guarantee for general submodular maximization. Similar observations have been made in [16].
(2) Our modified algorithm produced solutions that were always within 0.1% of the solution of the best-performing algorithm and also had very small variance. I.e., our algorithm is effective and consistently produces good results. Moreover, its effectiveness improves as the problem size $k$ grows, which concurs with our theoretical analysis.
(3) The running time is the most crucial parameter in the context of BSP, as the optimization algorithm must stop within strict time limits and approximate efficiency is only a secondary objective. According to Table 1, the running time of our algorithm scales much slower than that of Greedy and Local Search with $n$ and $k$. The time complexities of Greedy and one iteration of Local Search are both $O(nk^3 \cdot |\text{Support}|)$. When $k$ is constant ($k = 5$), this time complexity is linear in $n$, but even then our algorithm has comparable running time and when $k = \Theta(n)$, it is much faster than the benchmarks.

## 6 CONCLUDING REMARKS

In this paper, we studied a more general setting of position auction than all previous work on the Bidder Selection Problem. We proposed a new relaxation that can be solved in time polynomial in $n$ and $k$, and the polynomial is rather small. The proposed Poisson approximation approach is much simpler than previous PTAS *complexity results* for $k$-MAX or non-adaptive probing, and it can be implemented in practice. We also did extensive numerical experiments on inputs with practically relevant sizes and observed that our algorithm outperforms some commonly used heuristics, such as Greedy for general submodular maximization. Furthermore, we showed that the Poisson approximation approach also yields good theoretical guarantees. Namely, that BSP becomes solvable in polynomial time for any fixed $\varepsilon$, when the problem size $k$ grows. Our algorithm is the first one with a nearly perfect efficiency guarantee of PTAS, that is relevant in the application domain of the BSP and can be used by a company. Indeed, all previous PTASes had enormous running times and high implementation complexities that made them completely irrelevant to the problem, which was motivated by getting a speedup of $n/k$ magnitude.

A natural next step would be to consider BSP of various auction environments under richer sets of feasibility constraints such as matroid, matching, and intersection of matroids. Another interesting direction is to identify conditions where it is possible to efficiently optimize the revenue of BSP for VCG/GSP auction formats.

[12] AuctionGym uses comparable $\mu = 0.1$ and $\sigma = 0.2$.

# REFERENCES

[1] MOSEK ApS. 2022. *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.* MOSEK. http://docs.mosek.com/10.0/toolbox/index.html

[2] Nikhil Bansal and Maxim Sviridenko. 2006. The Santa Claus Problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006* (Seattle, WA, USA) *(STOC '06)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/1132516.1132522

[3] Xiaohui Bei, Nick Gravin, Pinyan Lu, and Zhihao Gavin Tang. 2023. Bidder Subset Selection Problem in Auction Design. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, Nikhil Bansal and Viswanath Nagarajan (Eds.). SIAM, 3788–3801. https://doi.org/10.1137/1.9781611977554.ch147

[4] Anand Bhalgat and Sanjeev Khanna. 2014. A Utility Equivalence Theorem for Concave Functions. In *Integer Programming and Combinatorial Optimization*, Jon Lee and Jens Vygen (Eds.). Springer International Publishing, Cham, 126–137.

[5] Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien. 2003. Efficient Query Evaluation Using a Two-Level Retrieval Process. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (New Orleans, LA, USA) *(CIKM '03)*. Association for Computing Machinery, New York, NY, USA, 426–434. https://doi.org/10.1145/956863.956944

[6] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. 2016. Combinatorial Multi-Armed Bandit with General Reward Functions. In *Advances in Neural Information Processing Systems*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.), Vol. 29. Curran Associates Inc., Red Hook, NY, USA, 1651–1659. https://proceedings.neurips.cc/paper/2016/hash/aa169b49b583a2b5af89203c2b78c67c-Abstract.html

[7] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review* 97, 1 (March 2007), 242–259. https://doi.org/10.1257/aer.97.1.242

[8] Ashish Goel, Sudipto Guha, and Kamesh Munagala. 2010. How to probe for an extreme value. *ACM Trans. Algorithms* 7, 1 (2010), 12:1–12:20. https://doi.org/10.1145/1868237.1868250

[9] Gagan Goel, Renato Paes Leme, Jon Schneider, David Thompson, and Hanrui Zhang. 2023. Eligibility Mechanisms: Auctions Meet Information Retrieval. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 3541–3549. https://doi.org/10.1145/3543507.3583478

[10] Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual.

[11] Jason Hartline. [n. d.]. Mechanism Design and Approximation. http://jasonhartline.com/MDnA/. Accessed: 2022-01-10.

[12] Olivier Jeunen, Sean Murphy, and Ben Allison. 2023. Off-Policy Learning-to-Bid with AuctionGym. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4219–4228.

[13] Jon Kleinberg and Maithra Raghu. 2018. Team Performance with Test Scores. *ACM Trans. Econ. Comput.* 6, 3–4, Article 17 (oct 2018), 26 pages. https://doi.org/10.1145/3274644

[14] Jian Li and Amol Deshpande. 2019. Maximizing Expected Utility for Stochastic Combinatorial Optimization Problems. *Mathematics of Operations Research* 44, 1 (February 2019), 354–375. https://doi.org/10.1287/moor.2017.0927

[15] Jian Li and Wen Yuan. 2013. Stochastic Combinatorial Optimization via Poisson Approximation. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing* (Palo Alto, California, USA) *(STOC '13)*. Association for Computing Machinery, New York, NY, USA, 971–980. https://doi.org/10.1145/2488608.2488731

[16] Aranyak Mehta, Uri Nadav, Alexandros Psomas, and Aviad Rubinstein. 2020. Hitting the high notes: Subset selection for maximizing expected order statistics. *Advances in Neural Information Processing Systems* 33 (2020), 15800–15810.

[17] S.Y. Novak. 2011. *Extreme Value Methods with Applications to Finance.* CRC Press, Boca Raton.

[18] S. Y. Novak. 2019. Poisson approximation. *Probability Surveys* 16 (2019), 228 – 276. https://doi.org/10.1214/18-PS318

[19] Ariel D. Procaccia, Sashank Jakkam Reddi, and Nisarg Shah. 2012. A Maximum Likelihood Approach For Selecting Sets of Alternatives. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, Nando de Freitas and Kevin P. Murphy (Eds.). AUAI Press, Corvallis, Oregon, 695–704. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2333&proceeding_id=28

[20] Zhou Rong and Zhixue Lu. 2007. Two-stage procurement auction with bidders of asymmetric capacity. *Systems Engineering-Theory & Practice* 27, 12 (2007), 36–41.

[21] Danny Segev and Sahil Singla. 2021. Efficient Approximation Schemes for Stochastic Probing and Prophet Problems. In *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, Péter Biró, Shuchi Chawla, and Federico Echenique (Eds.). ACM, New York, NY, USA, 793–794. https://doi.org/10.1145/3465456.3467614

[22] Guofu Tan. 1992. Entry and R & D in procurement contracting. *Journal of Economic Theory* 58, 1 (1992), 41–60.

[23] Hal Varian. 2007. Position auctions. *International Journal of Industrial Organization* 25, 6 (2007), 1163–1178. https://EconPapers.repec.org/RePEc:eee:indorg:v:25:y:2007:i:6:p:1163-1178

[24] Hal R. Varian and Christopher Harris. 2014. The VCG Auction in Theory and Practice. *American Economic Review* 104, 5 (May 2014), 442–45. https://doi.org/10.1257/aer.104.5.442

[25] Jiajin Yu and Shabbir Ahmed. 2016. Maximizing expected utility over a knapsack constraint. *Operations Research Letters* 44, 2 (2016), 180–185.

# A ADDITIONAL ASPECTS OF THE NUMERICAL EXPERIMENTS

## A.1 Comparison between Our Theoretical and Modified Algorithms

We compare the performance of the theoretical version and the modified version of our algorithm. Due to certain limitations on the convex objectives in Gurobi, we implemented the theoretical version of our algorithm in MATLAB with the help of another convex optimization library, Mosek [1], and ran it on the same set of test inputs as in Section 5. As the implementations are in different programming languages, we do not compare their running time and only compare their approximation efficiency.

As shown in Table 2, we can see that the theoretical version of our algorithm performs slightly worse than the modified version. This is due to the potentially suboptimal decision of fixing a small bidder set $S_{\text{fix}}$. This step is helpful when we analyze our algorithm theoretically, but it may not be optimal in practice. Therefore, we choose to use the modified version of our algorithm in Section 5.

## A.2 Notes on the Discretization

Recall that in Section 5, we discretized each generated distribution to a common, finite support $\{0\} \cup \{1 + \frac{i}{50} \mid i = 0, 1, \ldots, 49\}$ by moving probability mass on each discretized interval inside $[0, 2]$ to its left point and by redistributing the mass on $(2, +\infty)$ to the discrete points, proportional to their respective probabilities.

We picked this unusual discretization to make instances more challenging for Greedy, as without it Greedy and other heuristics like our algorithm produce solutions with nearly optimal approximation efficiency (over 99% for both algorithms). Indeed, two distributions in a well-structured family of log-normal distributions are likely to have strong dominance relation: a distribution $D_1 = \text{Lognormal}(\mu_1, \sigma_1^2)$ is always preferable to $D_2 = \text{Lognormal}(\mu_2, \sigma_2^2)$, whenever $\mu_1 \geq \mu_2$ and $\sigma_1 \geq \sigma_2$. Our choice of discretization was solely based on the comparison between Greedy and Local Search, i.e., after a few empirical trials of different discretizations we simply stopped once the average efficiency of Greedy was less than 99% that of the Local Search.

**Table 2: Experimental results of Local Search, Greedy, and both the modified version and the theoretical version of our algorithm. For each algorithm, we show the average relative quality of the produced solution to that of the best-performing algorithm that terminated in 1 week. Error bars denote the standard deviation.**

| Setting | | Benchmarks | | Our Algorithm | |
|---|---|---|---|---|---|
| $n$ | $k$ | Local Search | Greedy | Modified | Theoretical |
| 50 | 5 | 100.00% ± 0.00% | 98.93% ± 0.50% | 99.99% ± 0.03% | 95.28% ± 1.10% |
| | 10 | 100.00% ± 0.00% | 98.71% ± 0.38% | 99.99% ± 0.04% | 97.43% ± 0.84% |
| | 20 | 100.00% ± 0.00% | 99.17% ± 0.28% | 99.99% ± 0.01% | 99.53% ± 0.23% |
| 200 | 10 | 100.00% ± 0.00% | 98.06% ± 0.35% | 99.99% ± 0.01% | 95.36% ± 0.79% |
| | 20 | 100.00% ± 0.00% | 97.92% ± 0.24% | 100.00% ± 0.00% | 98.50% ± 0.45% |
| | 40 | 100.00% ± 0.00% | 97.97% ± 0.26% | 99.99% ± 0.00% | 99.85% ± 0.10% |
| 1000 | 50 | 100.00% ± 0.00% | 97.18% ± 0.14% | 99.99% ± 0.00% | 99.80% ± 0.09% |
| | 100 | N/A | 97.15% ± 0.15% | 100.00% ± 0.00% | 99.96% ± 0.02% |
| | 200 | N/A | 97.38% ± 0.11% | 100.00% ± 0.00% | 99.96% ± 0.02% |