Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0000-0000 | EISSN 0000-0000 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Concise Bid Optimization Strategies with Multiple Budget Constraints

Arash Asadpour
NYU Stern School of Business, aasadpou@stern.nyu.edu

MohammadHossein Bateni, Kshipra Bhawalkar, and Vahab Mirrokni
Google Inc., {bateni, kshipra, mirrokni}@google.com

A major challenge faced by marketers attempting to optimize their advertising campaigns is to deal with budget constraints. The problem is even harder in the face of multidimensional budget constraints, particularly in the presence of many decision variables involved and the interplay among the decision variables through such constraints. Concise bidding strategies help advertisers deal with this challenge by introducing fewer variables to act on.

In this paper, we study the problem of finding optimal concise bidding strategies for advertising campaigns with multiple budget constraints. Given bid landscapes—i.e., the predicted value (e.g., number of clicks) and the cost per click for any bid—that are typically provided by ad-serving systems, we optimize the value of an advertising campaign given its budget constraints. In particular, we consider bidding strategies that consist of no more than k different bids for all keywords. For constant k, we provide a PTAS to optimize the profit, whereas for arbitrary k we show how a constant-factor approximation algorithm can be obtained via a combination of solution enumeration and dependent LP rounding techniques which can be of independent interest. In addition to being able to deal with multi-dimensional budget constraints, our results do not assume any specific payment scheme and can be applied on pay-per-click, pay-per-impression, or pay-per-conversion models. Also, no assumption about the concavity of value or cost functions is made.

Finally, we evaluate the performance of our algorithms on real datasets in regimes with up to 6-dimensional budget constraints. In the case of a single budget constraint, where uniform bidding (introduced by Feldman et al. 2007 and currently used in practice) has provable performance guarantee, our algorithm beats the state of the art by an increase of 1% to 6% in the expected number of clicks. This is achieved by only two or three clusters—in contrast with the single cluster permitted in uniform bidding. With multiple budget constraint, the gap between the performance of our algorithm and an enhanced version of uniform bidding grows to an average of 5% to 6% (and as high as 35% in higher dimensions).

Key words: Internet Advertising, Revenue Management, Budget Constraints, Bidding Strategies, Uniform Bidding, Concise Bidding.

Subject classifications: Integer Programming: Relaxation and Applications; Analysis of Algorithms: Suboptimal Algorithms.

Area of review: Optimization.

1. Introduction

The Internet has become a major advertising medium, with billions of dollars at stake; according to the recent report of the Interactive Advertising Bureau (see the latest available in IAB 2017), Internet advertising revenues in the United States totaled $72.5 billion in 2016 with sponsored search accounting for a total of 48% of this revenue (with 24% coming from mobile search and 24% from nonmobile search). Search engines provide simple ways to quickly set up an advertising campaign, track expenses, monitor effectiveness of the campaigns, and tinker with campaign parameters. This has made it relatively easy even for small advertisers to enter online advertising market. Even with all the available tools to monitor and optimize ad campaigns, proper allocation of the *marketing budget* is far from trivial. A major challenge faced by the marketers attempting to optimize their campaigns is in the sheer number of variables they can possibly change. The problem is even more challenging in the presence of *multiple budget constraints*; i.e., in setting up a campaign that aims to target various categories of users or queries, or target a diverse set of demographics, the goal of an advertiser may be to allocate at least a fraction of its budget to each category, and therefore, it may be facing several budget constraints at the same time. Even within a single advertising channel on a particular search engine the advertiser can optimize by reallocating the budget across different keywords, choosing a particular bidding strategy to use within a single ad auction, and deciding on the daily advertising budget or what demographics of users to target. This is in particular challenging in the presence of many decision variables involved and an interplay among these variables.

In order to deal with the challenge, we propose *concise bidding strategies* to help advertisers by introducing fewer variables to act on. The idea is to consider the set of keywords that an advertiser may be interested in bidding on, and group them into a small number of clusters such that the advertiser is going to have the same bid on each cluster. (Some of the keywords may end up not being in any of the clusters, which means that the advertiser will not bid on them.) Such concise bidding strategies are inspired by *uniform bidding strategies*. Uniform bidding is a popular and practical way of bidding and has been widely used in industry and is shown to achieve relatively good results (Feldman et al. 2007).

In this paper, we develop near-optimal concise bidding strategies for allocating advertising budgets across different keywords in a general setting in the presence of multiple budget constraints. In the following, we first motivate the problem and give an overview of our contributions, before elaborating on our model and our results in the following sections.

1.1. Motivations and Challenges

Any online advertising market–such as sponsored search–consists of three main players: *Users*, *Advertisers*, and *Publishers (or search engines)*. In sponsored search, users pose queries on a search

engine like Bing or Google, declaring their intention and interests. Advertisers seek to place ads and target them to users' intentions as expressed by their queries, and finally publishers (or search engines in the case of sponsored search) provide a suitable mechanism for showing ads to users, through an ad-serving system. A common mechanism for allocating ads to users works as follows. First, the advertisers bid on the search query issued by the user. Then, the search engine runs an *auction* at the time the user poses the query to determine the advertisements that will be shown to the user. A lot of research has focused on the algorithmic and game-theoretic problems behind such advertising markets, both from the publisher/search engine's point of view (see Aggarwal et al. 2006, Varian 2007, Edelman et al. 2005, Mehta et al. 2007, Devanur and Hayes 2009, Dobzinski et al. 2012, and Goel et al. 2012), and advertiser's point of view (see Borgs et al. 2007, Feldman et al. 2007, Rusmevichientong and Williamson 2006, Zhou et al. 2008, Muthukrishnan et al. 2010, Even-Dar et al. 2009, and Archak et al. 2012). In this paper, we focus on optimization problems faced by advertisers.

More specifically, when a user submits a search query to a search engine, she receives next to the search results a number of ads. If the user finds any of the ads interesting and relevant, she may click on the ad. The advertisers interested in a search query submit their bids and the auction determines (1) which ads "win" to be displayed to the user, and (2) how much each user is charged. Charging can be based on "impressions" (each time the ad is displayed to the user), "clicks" (only if the the user clicks on the ad), or "conversions" (only if the user purchases the product or installs the software). In sponsored search, the advertisers mainly pay if the user clicks on their ad (the "pay-per-click" model), and the amount they pay is determined by the auction mechanism, but will be no larger than their bid.

While the impact of a bidding strategy is a complicated phenomenon based on complex dynamics among other advertisers' bidding strategies and the arrival pattern of user queries, search engines help advertisers optimize their campaigns by providing general statistics about the final predicted cost and value (e.g., number of clicks) of a bidding strategy. In particular, they provide for each advertiser a set of *bid landscapes*[1] for keywords (see Feldman et al. 2007); i.e., for each keyword w, advertisers get bid landscape functions value_w and $\overrightarrow{\mathsf{cost}}_w$ of bids on keyword w, which we will discuss in detail later.

We emphasize that our results can directly apply to *cost-per-click* (CPC), *pay-per-impression* (PPI), or *pay-per-conversion* (PPC) models, as we do not assume how the bid landscapes are produced. Also, our results apply to more general value or cost functions than that considered in the current literature. In particular, unlike the works of Feldman et al. (2007) or Muthukrishnan et al. (2010), no concavity assumption regarding these functions are made.

[1] Also referred to as "bid simulator" or "traffic estimator" by Google Support (2013b,c).

To set up an advertising campaign, advertisers specify a set of user queries (or keywords), determine a bid for each type of query/keyword, and declare an upper bound on their *advertising budget* for the campaign. Next, we discuss these constraints.

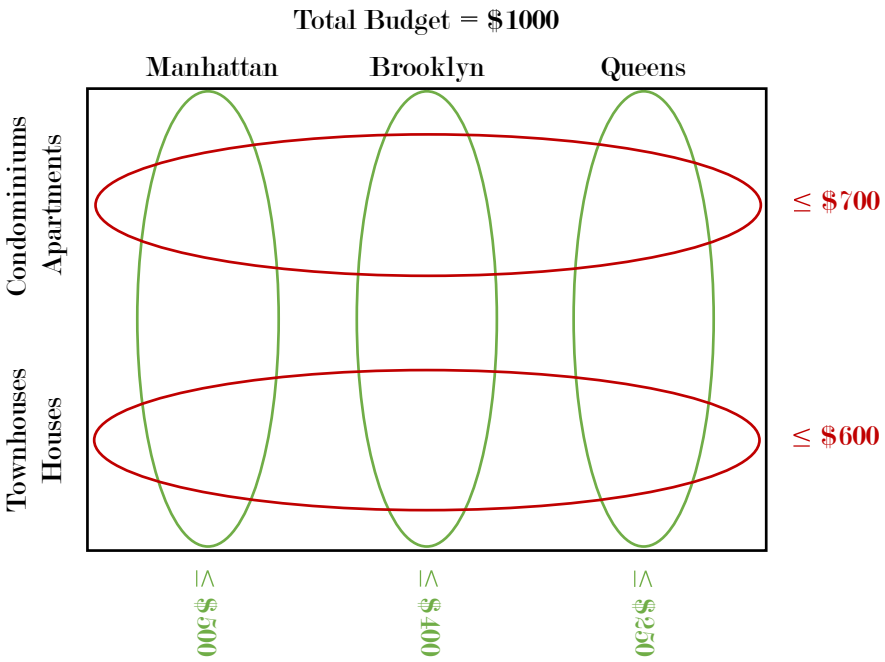
1.1.1. Further Challenges: Multidimensional Budget Constraints. Budget constraints play a major role in setting up an online advertising campaign, both from the auctioneer's point of view and in the advertiser's marketing strategy. It gives advertisers a robust knob to hedge against the uncertainty in the cost and benefits of the advertising campaign. In fact, some automated tools provided by search engines ask for a budget as part of the input, as can be seen for instance in Google Support (2011, 2013a). While setting up an advertising campaign, marketers often aim to target a diverse set of demographics, and therefore need to spread their budget spent on various keywords. One way to enforce a diversified spending is to set an upper bound on the budget spent on a subset of keywords corresponding to a subcategory of users targeted in the campaign or a particular category of keywords.

EXAMPLE 1. Consider an advertising campaign by a real-estate agency website to generate customers (or leads) for rentals in three of the boroughs in New York City, namely, Manhattan, Brooklyn, and Queens. Given a $1000 daily budget for the whole campaign, the advertiser might want to diversify the campaign throughout different boroughs, and therefore, spend at most $500 of the budget for the keywords related to Manhattan, at most $400 for those related to Brooklyn, and at most $250 for those related to Queens. Moreover, the advertiser might want to diversify among the different rental types as well, and to spend at most $700 on the keywords relevant to condominiums and/or apartments and at most $600 on those searches relevant to townhouses and/or houses (see Figure 1).

By this example, we would like to emphasize that even very natural preferences such as the above (and consequently, their relevant budget constraints) could have very complicated structures. In particular, the budget constraints are not limited to the special multidimensional case in which the constraints are only defined over disjoint subsets of keywords. As a result, different budget constraints interact with each other and their corresponding decisions would affect one another. We want our model to be able to capture such general multidimensional budget constraints. We describe the details of these budget constraints in Section 3.

1.1.2. Concise Bidding Strategies. Advertisers usually need to submit their bidding strategy to the search engine ahead of time, so that whenever a relevant query enters the system, the auction can run in real time. Advertisers can optimize their campaigns by reallocating their budget across different keywords, choosing a particular bidding strategy to use, or deciding on what demographics of users to target. This optimization is particularly challenging when facing many

Figure 1 **An example for multi-dimensional budget constraints: NYC housing market.**



decision variables or the interplay among them through the multidimensional budget constraints. To deal with this challenge, we propose *concise bidding* strategies by introducing fewer variables to act on. The idea is to represent the bidding strategy by a small number of bids with each bid acting on a cluster of keywords, i.e., dividing the target set of keywords into a small number of clusters so that the advertiser is going to have the same bid on each cluster.

Concise bidding strategies can be considered as a generalization of *uniform bidding strategies*, introduced by the seminal paper of Feldman et al. (2007). In uniform bidding strategies, the advertiser bids uniformly on all keywords. Uniform bidding, although naïve at the first glance, has been shown to achieve relatively good results, both in theory and practice (see Feldman et al. 2007 and Muthukrishnan et al. 2010). In fact, the simplicity of uniform bidding along with its reasonable performance makes it a desirable solution in practice which is robust and less reliant on the uncertain information provided by the advertising tool. Using such strategies, advertisers understand what their campaign is doing and where it is spending the budget.

On the other hand, search engines like to provide the advertisers with the ability to bid differently on each keyword, as employing more complicated bidding strategies—in particular, using this ability to bid differently on different keywords—may benefit the advertiser, search engine users, and the search engine company. However, finding a different bid value for each keyword will result in information overload for the advertiser. It may make the campaign management overwhelming

and impossible. Therefore, we take a middle-ground approach, and instead of declaring only one bid for all keywords, we cluster the keywords into a small number of subsets and apply a uniform bidding strategy on each subset, i.e., we use k distinct bids and let each bid act on an appropriate subset of keywords.

The Strengths. From a methodological point of view, we emphasize that uniform bidding, although effective in many current challenges, mainly applies to a specific setting with a *single* budget constraint and *concave* cost and value functions. Our approach on the contrary, can work with multiple budget constraints. Moreover, it does not assume that the cost or the value functions are concave. Moreover, in presence of weights on the clicks (which is crucial in order to model the pay-per-conversion systems), uniform bidding can perform arbitrarily bad (see Feldman et al. 2007). Our approach can on the other hand can model *cost-per-click* (CPC), *pay-per-impression* (PPI), or *pay-per-conversion* (PPC) systems as it does not assume how the bid landscapes are produced.

From a practical perspective, using small number of bids will likely lead to having large size clusters, which makes the solution more robust to the possible errors in individual points of bid landscapes provided by the advertising tool.

From the managerial point of view, clusters can help the managers differentiate between different levels of keyword profitability by providing a measure of *importance*. For instance, consider a scenario where for $k = 2$ in Example 1, the bids in the optimum solution end up to be 1\$ and 2\$ respectively for the two clusters. Also, the keyword "dishwasher" falls into the first cluster while the keyword "near subway" falls into the second cluster in the optimum solution. This can be a hint that the users searching for apartments near subway can be potentially more important for the advertising campaign than those who look for apartments with dishwashers.

Goal. Given all the above, our goal is to help advertisers find optimal concise bidding strategies respecting multiple budget constraints. Given a number k, a set of keywords relevant to an ad campaign, a value bid landscape and multiple budget landscapes for each keyword, the advertiser's goal is to find k clusters of keywords, and a bid for each cluster so as to maximize the value the advertiser receives from this bidding strategy (e.g., the expected number of clicks) subject to its budget constraints.

1.2. Our Results and Techniques

In this paper, we propose concise bidding strategies, and develop an algorithm to find optimal concise bidding strategies for allocating advertising budgets across different keywords in a general setting in the presence of multiple budget constraints. We formalize the concise bid optimization problem with multiple budget constraints as motivated and sketched above, and formally defined

in Section 3, and present approximation algorithms for this problem. With super-constant number of budget constraints r, the problem does not admit a reasonable approximation algorithm as it is harder than *set packing*. Coupled with the known hardness for set packing (Hastad 1999), the bid clustering problem does not admit any $\omega(r^{1-\epsilon})$ approximation unless $\mathsf{NP} \subseteq \mathsf{ZPP}$. In this paper, we focus on the problem with a constant number of budget constraints r. We emphasize that this is a reasonable assumption, as usually the number of budget constraints is considerably smaller than the number of keywords. For instance, in Example 1 there are three budget constraints related to the location and two more related to the type of the real estate, compared to all the possible keywords related to the search of real estate in New York City.

Our main theoretical contribution in this paper is a constant-factor approximation algorithm for arbitrary number of clusters k. This result is obtained using a dependent LP rounding technique (performed in four stages) combined with solution enumeration. The linear programming (LP) formulation of this problem and the dependent rounding approach used to obtain an integral solution are of independent interest. The rounding algorithm is very simple and fast to implement, as it runs in *linear*-time. However, its analysis uses new techniques to bound the loss incurred.

For the case of constant number of clusters k, we provide a polynomial-time approximation scheme (PTAS) to optimize the value. This PTAS is based on a careful dynamic program that enumerates various ways to satisfy the budget constraints. We emphasize that if a small violation of budget constraints (with a factor of $1+\epsilon$) is permitted, it is relatively easy to extend the standard PTAS for the knapsack problem to solve our problem with multiple budget constraints. However, to eliminate the budget violations completely is involved and requires careful enumeration and modification of a concept used in the previous PTAS called the *residual* instance.

Finally, we evaluate our algorithms on real datasets from sponsored search in Google search engine, and compare their performance with the uniform bidding strategy. We show that even in a simple setting of maximizing the expected number of clicks subject to one budget constraint (for which uniform bidding is provably good), using a small number of clusters can improve the expected number of clicks by 1% to 6%. We see more improvement by increasing the number of clusters when the budget constraint is tighter. We also evaluate our algorithm on datasets with more budget constraints, and notice significant improvement (as high as 35%) compared to uniform bidding. Moreover, we observe that we lose less than 1% when we round the solution from fractional LP solution to a feasible integral solution.

2. Related Work

As a central issue in online advertising, optimizing under budget constraints has been studied extensively both from publishers' (or search engines') point of view (see, for example, Mehta et al.

2007, Devanur and Hayes 2009, Dobzinski et al. 2012, and Goel et al. 2012), and from advertisers' point of view (see Borgs et al. 2007, Feldman et al. 2007, Rusmevichientong and Williamson 2006, Zhou et al. 2008, Muthukrishnan et al. 2010, Even-Dar et al. 2009, and Archak et al. 2012). One well-studied problem from publisher's perspective is to deal with online allocation of ads in the presence of budget constraints, as studied by Mehta et al. (2007), Devanur and Hayes (2009), and Feldman et al. (2009, 2010). Another line of research is dedicated to designing efficient mechanisms addressing incentive issues, and respecting budget constraints by Dobzinski et al. (2012) and Goel et al. (2012). More relevant to this paper, the bid optimization with budget constraints has also been studied from advertisers' perspective: either in a repeated auction setting by Borgs et al. (2007), or in the context of broad-match ad auctions by Even-Dar et al. (2009), or the case of long-term carryover effects by Archak et al. (2012).

This work is most related to the seminal paper of Feldman et al. (2007) in which the authors propose uniform bidding as a means for bid optimization in the presence of budget constraints in sponsored-search ad auctions. Our results differ from those of Feldman et al. (2007) and Muthukrishnan et al. (2010) in several aspects: the uniform bidding result and its guaranteed approximation ratio of $1 - 1/e$ applies to CPC settings where the goal is to maximize the expected number of clicks, and when the cost and value bid landscapes follow a specific structure and form concave functions. Besides, those results apply only in the case of one budget constraint and neither the methods nor the proofs can be easily generalized to settings with multiple budget constraints. Our results however does not assume any specific payment model (cost-per-click, pay-per-impression, or pay-per-conversion) and apply to any set of monotone[2] cost and value bid landscape functions (e.g., for the case of maximizing conversions). Thus, no concavity assumption is made on the landscape functions. More importantly, our results can handle multiple budget constraints.

We emphasize that in our theoretical bounds, we compare our solution to the best solution with the same number of clusters, whereas Feldman et al. (2007) compare their solution to the optimum of the knapsack problem with arbitrary number of clusters and also in a more general query language model. As a result, we can get a PTAS for the case of constant number of clusters, but Feldman et al. (2007) can only get a $1 - 1/e$ approximation ratio. In order to generalize the results of Feldman et al. (2007) to multiple budget constraints we developed new solution concept and a set of tools and techniques for this problem, including a dependent LP rounding that may be of independent interest. In order to fairly compare our algorithm with that of Feldman et al. (2007), in a series of experimental studies we compare the performance of our algorithm and that

[2] As we will see later, we only use the monotonicity assumption in the preliminary stage in order to to discretize the set of relevant bids (see Lemma 1). The assumption is not necessary if the set of possible bids were already small.

of Feldman et al. (2007) with the optimal (fractional) solution of the LP relaxation with arbitrary number of clusters. (See Section 5 for the results of the experiments.)

Finally, we note that there have been other efforts by the search engines to simplify bidding strategies for the advertisers. Bateni et al. (2014) study *multiplicative bidding* which is a bid adjustment language adopted by AdWords and Bing Ads (see for instance Google Support 2014). In multiplicative bidding, the advertiser can set bid adjustments for different search queries. For instance the advertiser can set an adjustment of 1.5, 1.2, and 0.9 for the real-estate queries in Manhattan, Brooklyn, and Queens, respectively. Also, the adjustment for apartments and townhouses can be set at 1 and 1.1, respectively. In this example, the bid of the advertiser for a query regarding "townhouses in Brooklyn" will be $\$1.2 \times 1.1 = \$1.32$. Bateni et al. (2014) provide an $O(\log n)$-approximation for the advertiser's optimization problem. In addition to assuming monotonicity of values, dealing with a single budget constraint, and a super-constant approximation ratio, multiplicative bidding does not achieve the robustness (with respect to the bid landscapes) that uniform or concise bidding provide due to their eventual large size clusters.

3. Preliminaries

Let $[k]$ for an integer k denote the set $\{1, 2, \ldots, k\}$. We denote a vector v by $\vec{v}$ to emphasize its multidimensionality. The length of the vector is often omitted and understood from the context; it is denoted by r (the total number of budget constraints), unless otherwise specified, since our vectors are mostly used for capturing the multidimensional resource constraints. To denote different components of a vector $\vec{v}$ of length r, we use the notation $v^{(q)}$ for $q \in [r]$. For any real number z, we denote by $\vec{z}$ the vector all whose components are z. The length of these vectors is understood from the context. We say $\vec{v} \leq \vec{u}$ if they have the same length and every component of $\vec{v}$ is at most the corresponding component of $\vec{u}$. Otherwise, we can write $\vec{v} \not\leq \vec{u}$.

3.1. Formal problem definition

In order to optimize their campaigns, we assume that advertisers get "bid landscapes", similar to the setting of Feldman et al. (2007), as an input. That is, for each keyword w, they get

(i) a nonnegative function $\mathsf{value}_w : \mathbb{R} \mapsto \mathbb{R}$ that maps any given bid to the expected value it generates (e.g., number of clicks), and

(ii) a nonnegative function $\overrightarrow{\mathsf{cost}}_w : \mathbb{R} \mapsto \mathbb{R}^r$ mapping any given bid to an r-dimensional cost vector incurred by the advertiser.

We assume that these functions are left-continuous, but they do not necessarily satisfy Lipschitz smoothness conditions. We refer the reader to Feldman et al. (2007) for a discussion on how these landscapes are derived.

In addition, we have an r-dimensional budget limit vector (or resource usage vector), and a number k indicating the number of clusters we can produce in our suggested bidding strategy. The bid clustering problem is formally defined as follows.

PROBLEM 1. Given are an integer k, a number r of budget constraints (resources), a real vector $\vec{L} \in \mathbb{R}^r$ denoting the resource limits, a set $\mathcal{K}$ of keywords, a value landscape function $\mathsf{value}_w : \text{bids} \mapsto \mathbb{R}$ for each keyword $w \in \mathcal{K}$, and a cost landscape function $\overrightarrow{\mathsf{cost}}_w : \text{bids} \mapsto \mathbb{R}^r$ for each keyword $w \in \mathcal{K}$. Find a partition of a subset of keywords into k clusters $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_k$ and a set of associated bids b_j for $j \in [k]$ such that the expected resource consumption of the advertiser is no more than his budget vector $\vec{L}$, i.e., $\sum_{j\in[k]} \sum_{w\in\mathcal{K}_j} \overrightarrow{\mathsf{cost}}_w(b_j) \le \vec{L}$, and the advertiser's value (e.g., expected number of clicks), i.e., $\sum_{j\in[k]} \sum_{w\in\mathcal{K}_j} \mathsf{value}_w(b_j)$ is (approximately) maximized.

We say an algorithm is α-approximation if it is guaranteed to find a solution that satisfies the budget constraint, while achieving a total value of at least α times that of the optimum solution.

We refer to each of the r budget constraints as a *resource* (or *budget*) constraint. The q-th resource constraint for any given $1 \le q \le r$ is in the form of $\sum_{j\in[k]} \sum_{w\in\mathcal{K}_j} \mathsf{cost}_w^{(q)}(b_j) \le L^{(q)}$. For ease of exposition, for any given $w \in \mathcal{K}$, $b \in$ bids, and $1 \le q \le r$, we use the shorthands $c^q_{w,b} = \mathsf{cost}_w^{(q)}(b)$ to refer to the cost incurred by the q-th resource constraint when bidding b on keyword w. Similarly, we use $p_{w,b} = \mathsf{value}_w(b)$ for the expected value achieved by bidding b on keyword w. Finally, throughout this paper, we use $n = |\mathcal{K}|$ to denote the number of keywords, k for the number of clusters, and r for the number of budget (or resource) constraints.

Note that different resource constraints do not necessarily correspond to disjoint sets of keywords. We emphasize that similar to Example 1, different resource constraints can correspond to different markets, each drawing from all keywords, albeit with different coefficients. However, these markets may have some products or characteristics in common.

3.2. Approach

As discussed earlier, we know that if r, the number of different budget constraints (or resource constraints), is not a constant, the bid-clustering problem even with no restriction on the number of clusters is inapproximable. This is due to the fact that this problem is harder than the set packing (see Section A) or the independent set problem which are known to be inapproximable within a factor better than $\omega(n^{1-\epsilon})$ unless $\mathsf{NP} \subseteq \mathsf{ZPP}$, due to Hastad (1999). As a result, henceforth we assume that r is a small constant. In fact, the running times of our algorithms depend exponentially on this parameter. We note that all the previous work in advertising bid optimization, including Feldman et al. (2007), Muthukrishnan et al. (2010), Even-Dar et al. (2009), and Archak et al. (2012), only study the case of $r = 1$.

Our approach relies on the three following initial observations.

Observation 1: Uniform Resource Limits. The first observation is that it is enough to consider problems with *uniform resource limits*. In other words, we can assume without loss of generality that the resource usage limit vector $\vec{L} = \vec{1}$. To do this, we consider each resource separately. Therefore, if a resource limit q in $\vec{L}$ is positive, we can scale it to 1 while modifying $\mathsf{cost}_w^{(q)}(b)$ appropriately for all $w \in \mathcal{K}$, i.e., $\mathsf{cost}_w^{(q)}(b) \leftarrow \mathsf{cost}_w^{(q)}(b)/L^{(q)}$. On the other hand, a limit of zero in L for some resource q implies that we cannot place a bid b on a keyword w if $\mathsf{cost}_w^{(q)}(b) > 0$. Hence, by setting such values of $\mathsf{cost}_w^{(q)}(b)$ to ∞ we can change the limit of q in L to 1.

Observation 2: Few Potential Bids. The second observation is that we can focus only on a *small set of potential bids* without incurring much loss. Although the cost and value landscapes have a continuous nature (provided to us, perhaps, by oracle access), we can settle with a polynomial-size description thereof while incurring a small loss in the guarantees. In particular, we show that there are only a polynomial number of different bid values that matter. This observation is formalized in the following lemma whose proof can be found in Section C in the appendix.

LEMMA 1. *Given any $\zeta > 0$, we can find (in polynomial time) a set B of size $\mathrm{poly}(n, \frac{1}{\zeta})$ such that there exists a $(1-\zeta)$-approximate solution to Problem 1 that only uses bids from B.*

For the rest of the paper we assume that we deal with a polynomial size subset of bids B, derived from the above Lemma 1.

Observation 3: PTAS For Constant k. The third and the final observation is that if k, the number of permissible clusters, is constant, then we can design a polynomial-time approximation scheme (PTAS) for the problem. This means that for any constant $\epsilon > 0$, we can design an algorithm that runs in polynomial time with respect to the size of the problem, and provides a $(1-\epsilon)$-approximation to the optimum. The PTAS and its analysis are discussed in Section B in the appendix. There, we first discuss designing a simpler PTAS that violates the budget constraints by a factor of $1+\epsilon$. The idea is based on a dynamic programming formulation of the problem. Later, we refine this algorithm to provide a more involved PTAS that does not violate the budget constraints at all. The results of Section B in the appendix is summarized in the following theorem.

THEOREM 1. *There exists a polynomial-time approximation scheme for the bid-clustering problem if k is a constant.*

In what follows, we use the above observation to present a constant factor approximation for the general formulation of Problem 1.

4. The Approximation Algorithm

In this section, we approach to the bid-clustering problem by solving a linear-programming relaxation and then applying a four-stage dependent randomized rounding scheme. Even though the analysis is not straightforward, the algorithm is very simple to implement and efficient in practice.

4.1. Linear Programming Relaxation

Here we introduce a linear programming relaxation for the problem. We will argue that we only need to round this LP for the case that each bid has a small contribution to the solution, otherwise, a PTAS from Theorem 1 suffices to find a good solution for keywords with big contributions in the solution. The IP formulation is the same as below where Eq. (6) is replaced with $x_{i,b}, y_b \in \{0,1\}$.

$$\max \sum_{i\in\mathcal{K}, b\in B} p_{i,b}x_{i,b} \tag{1}$$

$$\text{s.t.} \sum_{i\in\mathcal{K}, b\in B} c^q_{i,b}x_{i,b} \le 1 \qquad \forall q\in[r] \tag{2}$$

$$\sum_{b\in B} y_b \le k \tag{3}$$

$$x_{i,b} \le y_b \qquad \forall i\in\mathcal{K}, b\in B \tag{4}$$

$$y_b \le 1 \qquad \forall b\in B$$

$$\sum_{b\in B} x_{i,b} \le 1 \qquad \forall i\in\mathcal{K} \tag{5}$$

$$x_{i,b}, y_b \ge 0 \qquad \forall i\in\mathcal{K}, b\in B. \tag{6}$$

$$(\text{or, } x_{i,b}, y_b \in \{0,1\} \quad \forall i\in\mathcal{K}, b\in B \text{ in the IP formulation})$$

Consider a variable $x_{i,b}$ for each keyword $i\in\mathcal{K}$ and each bid $b\in B$. In the IP formulation, $x_{i,b}$ is a binary variable that denotes whether the advertiser should place a bid b on keyword i. In the LP relaxation, we relax it to a positive real variable. In addition, there is a variable y_b for each relevant bid b, denoting whether there exists a cluster with bid b. We remind the reader that we use the shorthands $p_{i,b} = \mathsf{value}_i(b)$ and $c^q_{i,b} = \overrightarrow{\mathsf{cost}}^{(q)}_i(b)$ in order to make the LP more concise and readable. Constraint (2) represents the r budget limit constraints, where the limits (i.e., $L^{(q)}$'s) are all normalized to 1 as a result of Observation 1 in Section 3.2. Constraint (3) implies that there are in total no more than k bids (and hence, no more than k corresponding clusters). Constraint (4) guarantees that the advertiser may bid b on keyword i only if there exists a cluster corresponding to that bid. Finally, Constraint (5) implies that the advertiser should only place one bid on a specific keyword (or in other words, the clusters do not have any intersection, as desired by the problem).

Let OPT denote the value of the optimum solution of Problem 1, or equivalently, the optimum solution of the IP formulation of the problem. For any *integral* solution $\mathsf{sol} = (x, y)$, let $c^q_b(\mathsf{sol})$ be defined as follows for every $b\in B$ and $q\in[r]$.

$$c^q_b(\mathsf{sol}) := \sum_{i\in\mathcal{K}} c^q_{i,b}x_{i,b}.$$

Let $0 < \epsilon < (2+\log r)^{-1}$ be given.

DEFINITION 1 (CONDITION SCP). For any given $\epsilon > 0$, an *integral* solution sol of Problem 1 is called to satisfy "Condition SCP" (small costs and profit) if and only if we have $c_b^q(\mathsf{sol}) \le \epsilon$ for all $b \in B$ and $q \in [r]$ and also, we have $p_{i,b}x_{i,b} \le \epsilon\mathsf{OPT}$ for all $i \in \mathcal{K}$ and $b \in B$.

The crucial observation is that if there exists an LP solution that satisfies Condition SCP, then the solution can be rounded to obtain an approximation ratio of $0.539 - f(\epsilon)$ where $f = O(\sqrt{\epsilon})$. This is shown in Section 4.2. Based on this observation, we consider two cases.

(I) The first case happens when there exists an *integral* solution $\mathsf{sol}_1 = (x, y)$ for the problem with value at least $\beta\mathsf{OPT}$ (for some value β to be determined later) which satisfies Condition SCP, i.e., for every q and b we have $\sum_i c_{i,b}^q x_{i,b} \le \epsilon$, and for every i and b with $x_{i,b} = 1$ we have $p_{i,b} \le \epsilon\mathsf{OPT}$. The first constraint can be written as $\sum_i c_{i,b}^q x_{i,b} \le \epsilon y_b$ due to Constraint (4) and the fact that $x_{i,b}$'s and y_b's are binary. Moreover, $\sum_i c_{i,b}^q x_{i,b} \le \epsilon$ means that $x_{i,b} = 0$ if $c_{i,b}^q > \epsilon$ for some $q \in [r]$. Thus, in this case the optimum LP value will remain at least $\beta\mathsf{OPT}$ if we add the following constraints to the LP relaxation (1-6). (See Remark 1 for Eq. (8).)

$$x_{i,b} = 0, \qquad \text{if } \exists q : c_{i,b}^q > \epsilon, \forall i, b, \tag{7}$$

$$x_{i,b} = 0, \qquad \text{if } p_{i,b} > \epsilon\mathsf{OPT}, \forall i, b, \tag{8}$$

$$\sum_i c_{i,b}^q x_{i,b} \le \epsilon y_b, \qquad \forall b, q. \tag{9}$$

In this case, we can round the solution of the new LP relaxation (1-6) with the additional constraints (7-9) using the rounding of Section 4.2. Theorem 4 guarantees that we find an integral solution with value at least $(0.539 - \delta)\beta\mathsf{OPT}$ for this problem where $\delta = 6\sqrt{\epsilon(2 + \log r)} + \epsilon r$.

(II) The second case happens when there is an integral solution $\mathsf{sol}_2 = (x, y)$ for the problem with value at least $(1 - \beta)\mathsf{OPT}$ that uses nothing but large-cost or large-profit bids, i.e., for every chosen bid b (where $y_b = 1$) there exists a $q \in [r]$ so that $c_b^q(\mathsf{sol}_2) > \epsilon$ or there exists an $i \in \mathcal{K}$ such that $p_{i,b}x_{i,b} > \epsilon\mathsf{OPT}$. Therefore, for every chosen bid b we have $\sum_q c_b^q(\mathsf{sol}_2) + \sum_i p_{i,b}x_{i,b}/\mathsf{OPT} \ge \epsilon$. Let κ be the number of chosen bids. If we sum up the inequality over all chosen bids b, we get $\sum_{q,b} c_b^q(\mathsf{sol}_2) + \sum_{i,b} p_{i,b}x_{i,b}/\mathsf{OPT} \ge \kappa\epsilon$. However, for all bids b that are not chosen, we have $x_{i,b} = 0$. Therefore, $\sum_q \sum_{i,b} c_{i,b}x_{i,b} + \sum_{i,b} p_{i,b}x_{i,b}/\mathsf{OPT} \ge \kappa\epsilon$. On the other hand, Constraint (3) of the LP and the optimality of OPT results in the left hand side being upper bounded by $r + 1$. As a result, $r + 1 \ge \kappa\epsilon$, or equivalently, there exists a solution with value at least $(1 - \beta)\mathsf{OPT}$ that uses at most $(r + 1)\epsilon^{-1}$ clusters. However, since r and ϵ are given constants, the total number of possible clusters is bounded by a constant. Therefore, we can invoke Theorem 1 to find an approximate solution with value at least $(1 - \beta - \epsilon)\mathsf{OPT}$ in this case for any constant $\epsilon > 0$.

Note that the above cases are defined in such a way that at least one of them holds in any instance. Hence, we can always get a solution whose value is at least the following for $\delta = 6\sqrt{\epsilon(2+\log r)} + \epsilon r$.

$$\min\{(0.539-\delta)\beta, 1-\beta-\epsilon\} \cdot \mathsf{OPT}.$$

Therefore, choosing $\beta = \frac{1}{1.539}$ yields an approximation ratio of $1-\beta-\delta \approx 0.3506$. This is summarized in the following theorem.

THEOREM 2. *The solution of LP (1-6) can be rounded to a* 0.35*-approximate integral solution in polynomial time.*

Finally, we emphasize that if a better analysis or some different rounding approach is found for rounding LP (1-6) with the additional constraints (7-9), then Theorem 2 can be improved by considering the same two cases above. This is presented in the following corollary.

COROLLARY 1. *If LP (1-6) with additional constraints (7-9) can be rounded to an integral* α*-approximate solution, then LP (1-6) in general can be rounded to an* $\frac{\alpha}{1+\alpha}$*-approximate integral solution in polynomial time.*

We make a conjecture on this corollary in Section 4.4.

REMARK 1. We note that the value of OPT is not known in advance. However, one can run a binary search in $O(\log(\mathsf{OPT}))$ to find the highest value for which either the LP (1-6) with the additional constraints (7-9) can be rounded to a solution with value at least $(0.539-\delta)\beta\mathsf{OPT}$ or the PTAS for $(r+1)\epsilon^{-1}$ clusters returns a solution with value at least $(1-\beta-\epsilon)\mathsf{OPT}$.

4.2. Rounding the LP

In this section we show how to round LP (1-6) with the additional constraints (7-9). The rounding procedure is done in four stages. In the first stage, we modify the fractional solution so that only k nonzero y_b variables remain. In the second stage, $x_{i,b}$'s are scaled proportional to the rounded values of y_b's. These two stages are done carefully without losing more than $1 - O(\sqrt{\epsilon(1+\log r)})$ factor in the objective value for an arbitrary small value of $\epsilon > 0$. At this point, we have an LP solution that is *almost* feasible—some constraints (5) may be violated. The third stage addresses this issue by modifying $x_{i,b}$ variables so that, at the end, all LP constraints are satisfied. The fourth stage, which we may even skip depending on the type of solution we need, is a standard randomized assignment of each keyword to one bid, and may lose up to a factor $1 - O(\sqrt{\epsilon(1+\log r)} + \epsilon r)$. Algorithm 1 provides a summary.

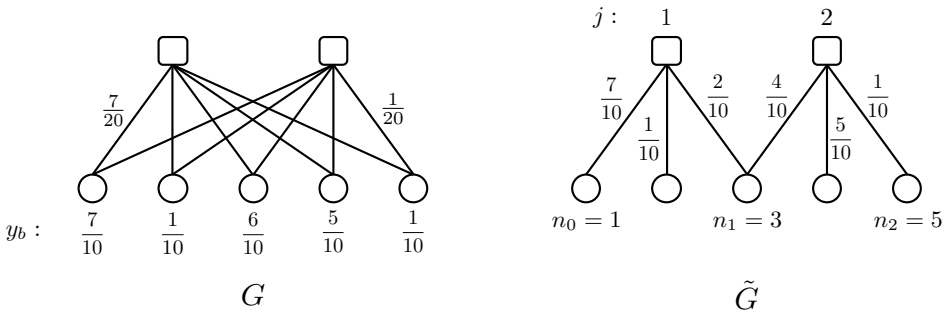
Stage i: rounding y_b's. We first note that by adding dummy bids we can assume that the constraint (3) is binding, i.e., $\sum_b y_b = k$. To see this, let us assume that $\sum_b y_b = k' < k$. We add $m = \lceil k - k' \rceil$ dummy bids, namely $y_{|B|+1}, y_{|B|+2}, \cdots, y_{|B|+m}$ where $y_b = 1$ for $|B|+1 \le b \le |B|+m-1$ and $y_{|B|+m} = 1 - \{k-k'\}$ where $\{x\} = x - \lfloor x \rfloor$ denotes the fractional part of x. Also, $x_{i,b}$ for all i and $|B|+1 \le b \le |B|+m$ is set to 0. It is easy to check that the current solution is feasible and also, its objective value is the same as before. Also, note that the dimension of the vector y is increased by at most k. Hence, from now on we assume that $\sum_{b \in B} y_b = k$.

In order to provide an intuition for the way we round y_b variables, we start by a graph representation of the problem. Let $G = (V, E)$ be a bipartite graph where $V = [k] \cup B$, and $E = [k] \times B$. Also, for every edge (j, b) where $j \in [k]$ and $b \in B$, we define $f_{j,b} := y_b / k$. (See the left graph in Figure 2.) Note that

$$\sum_{b \in B} f_{j,b} = \frac{1}{k} \sum_{b \in B} y_b = 1, \qquad \forall j \in [k], \text{ and} \tag{10}$$

$$\sum_{j=1}^{k} f_{j,b} = \frac{k y_b}{k} = y_b, \qquad \forall b \in B. \tag{11}$$

Figure 2 **An example of the bipartite graph G and its transformation to the acyclic graph $\tilde{G}$.**



Note that any matching in this graph that saturates all the vertices in $[k]$ can be associated to a 0/1 solution $(Y)_B$ that selects exactly k bids. More specifically, Y_b will be 1 if and only if b is saturated in the matching. Therefore, we focus on sampling matchings from G that saturate all the vertices in $[k]$. For the sake of brevity, we call such matchings *semi-perfect*.

In order to sample a semi-perfect matching from G, we first eliminate cycles from G in a way that the marginals are preserved, i.e., Eq. (10–11) still hold. In order to do so, let $n_1 \le n_2 \le \cdots \le n_{k-1}$ be the indices of the *border* bids, such that

$$\sum_{b=1}^{n_j - 1} y_b < j \le \sum_{b=1}^{n_j} y_b. \tag{12}$$

For any border bid $b = n_j$, define

$$\tilde{f}_{j,b} := j - \sum_{b=1}^{n_j-1} y_b \qquad \text{and} \qquad \tilde{f}_{j+1,b} := \sum_{b=1}^{n_j} y_b - j. \tag{13}$$

Also, for any non-border bid $n_{j-1} < b < n_j$, define

$$\tilde{f}_{j,b} = y_b. \tag{14}$$

Finally, $\tilde{f}_{j,b}$ for all other edges will be zero. For the sake of consistency, we define $n_0 := 1$ and $n_k := |B|$. See the right graph of Figure 2 for an example.

By our construction, it is straightforward to check that we have the equivalent of Eq (10–11) with $\tilde{f}$, i.e.,

$$\sum_{b\in B} \tilde{f}_{j,b} = 1, \qquad \forall j \in [k], \text{ and} \tag{15}$$

$$\sum_{j=1}^{k} \tilde{f}_{j,b} = y_b, \qquad \forall b \in B. \tag{16}$$

Moreover, our construction ensures that the edges with positive values of $\tilde{f}$ do not form any cycle.

We denote this graph by $\tilde{G}$. We employ the idea of the Bayesian dependent rounding of Asadpour and Saberi (2010) in order to sample a random matching from a forest (an undirected graph with no cycle) with given marginal probabilities. The main idea of the rounding method of Asadpour and Saberi (2010) is to scan the vertices of the forest one by one in some arbitrary order and fix the matching for the edges adjacent to the vertex currently being scanned according to the current marginal probabilities (by choosing at most one of the edges connected to the vertex). After that, the algorithm updates the marginal probabilities over all the other edges in the graph, according to a Bayesian update rule, and proceeds. However, the algorithm of Asadpour and Saberi (2010) runs in $\Theta(|V| \cdot |E|)$. For the purpose of our problem, we provide a simpler implementation of this algorithm that runs in linear time in the size of the problem, i.e., in $O(|B|)$ time.

Note that our algorithm needs one scan over the values of y_b in order to find the values of $n_1 \le n_2 \le \cdots \le n_{k-1}$ and also $\tilde{f}_{j,b}$'s in Lines 1 and 2. After that, in the "for loop" of Line 4 all the bids will be scanned once more (once for each non-border bid and twice for each border bid). Therefore, Stage i runs in $O(k + |B|)$ time. Note that in our problem we always have $|B| > k$ (otherwise, one could simply keep all the bids). Consequently, the running time of Stage i is effectively $O(|B|)$. An example of the first stage of rounding after the first iteration of the "for loop" in Line 4 is given in Figure 3. In the left picture, $n_1 = 3$ is *not* matched to $j = 1$ in the first iteration, i.e., $Y_3 = 0$ in the "if condition" of Line 6 at the beginning of the second iteration of the for loop. In the right picture, n_1 is matched to $j = 1$.

ALGORITHM 1: `A 0.539-approximation to the LP (1)-(6) with additional constraints (7-9)`

Input: Vectors $y \in [0,1]^{|B|}$, $x \in [0,1]^{k\times|B|}$ as a feasible solution of LP (1–6, 7-9). An $0 < \epsilon < (2+\log r)^{-1}$.

Output: Binary vectors $Y^\star \in [0,1]^{|B|}$, $X^\star \in [0,1]^{k\times|B|}$ as a feasible solution of LP (1–6).

Find the values of $n_1 \le n_2 \le \cdots \le n_{k-1}$ according to Eq. (12). Let $n_0 := 1$ and $n_k := |B|$.

Find the values $\tilde{f}_{j,b}$ for $j \in [k]$ and $b \in B$ according to Eq. (13–14). Let $\tilde{f}_{0,b} := 0$.

`*** stage i: rounding` y_b`'s ***`

$Y \leftarrow (0)_{|B|}$.

for $j = 1$ to k **do**

  Let $\lambda := \tilde{f}_{j-1,n_{j-1}}$ and $\mu := \tilde{f}_{j,n_{j-1}}$.

  if $Y_{n_{j-1}} = 0$ **then**

    Let $\pi_{n_{j-1}} := \frac{1}{1-\lambda}\tilde{f}_{j,n_{j-1}}$.

    Let $\pi_b := \frac{1-\lambda-\mu}{(1-\lambda)(1-\mu)}\tilde{f}_{j,b}$, for all $n_{j-1}+1 < b \le n_j$.

  else

    Let $\pi_{n_{j-1}} := 0$.

    Let $\pi_b := \frac{1}{1-\mu}\tilde{f}_{j,b}$, for all $n_{j-1}+1 < b \le n_j$.

  Pick exactly one $b_j \in \{n_{j-1}, n_{j-1}+1, \cdots, n_j\}$ according to the probabilities π_b.

  $Y_{b_j} \leftarrow 1$.

`*** stage ii: scaling` $x_{i,b}$`'s ***`

for $i \in \mathcal{K}$ and $b \in B$ **do**

  if $Y_b = 0$ **then**

    $X^{(\mathrm{ii})}_{i,b} \leftarrow 0$.

  else

    $$X^{(\mathrm{ii})}_{i,b} \leftarrow \frac{x_{i,b}}{y_b} \cdot \frac{1}{1+2\sqrt{\epsilon(2+\log r)}}.$$

`*** stage iii: trimming` $X^{(\mathrm{ii})}_{i,b}$`'s ***`

for $i \in \mathcal{K}$ **do**

  if $\sum_b X^{(\mathrm{ii})}_{i,b} > 1$ **then**

    $$X^{(\mathrm{iii})}_{i,b} \leftarrow \frac{X^{(\mathrm{ii})}_{i,b}}{\sum_{b'} X^{(\mathrm{ii})}_{i,b'}} \text{ for all } b \in B.$$

`*** stage iv: rounding` $X^{(\mathrm{iii})}_{i,b}$`'s ***`

for $i \in \mathcal{K}$ **do**

  pick (at most) one of the bids b_i according to the probabilities induced by $X^{(\mathrm{iii})}_{i,b}$'s.

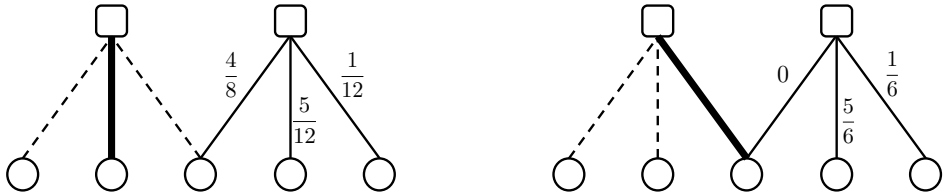
  $\tilde{X}^{(\mathrm{iv})}_{i,b_i} \leftarrow 1$. $\tilde{X}^{(\mathrm{iv})}_{i,b} \leftarrow 0$ for all $b \neq b_i$.

Remove some $\tilde{X}^{(\mathrm{iv})}_{i,b}$'s if necessary to satisfy the small violations in costs, as explained in the text. Denote the rest by $X^{(\mathrm{iv})}_{i,b}$.

$Y^\star \leftarrow Y$ and $X^\star \leftarrow X^{(\mathrm{iv})}$.

return $Y^\star$ and $X^\star$.

Figure 3 **An example of the $\tilde{G}$ after the first iteration of the "for loop". The thick edge means that the edge is fixed to be in the matching. The dashed edges mean that the edges are not going to be in the matching. The numbers are the updated marginals (the values of π_b) in the second iteration of the "for loop".**



We first prove that the algorithm is well-defined, in other words, the probabilities of the random pick in Line 12 always add up to 1.

CLAIM 1. *In the j-th iteration of the "for loop" in Line 4 the probability distribution defined by π_b for $b \in \{n_{j-1}, n_{j-1}+1, \cdots, n_j\}$ is proper, i.e., all π_b's are non-negative and also, $\sum_{b=n_{j-1}}^{n_j} \pi_b = 1$.*

The proof can be found in the appendix. Next, we prove two properties of the output of our rounding approach $(Y)_{|B|}$ which we are going to use later in the analysis, namely, being margin-preserving and imposing negative correlation.

LEMMA 2 **(Margin Preservation)**. *For every $b \in B$ we have $\mathbf{Pr}[Y_b = 1] = y_b$.*

LEMMA 3 **(Negative Correlation)**. *All Y_b's are negatively correlated, i.e., for every $S \subseteq B$, we have $\mathbf{Pr}[\wedge_{b\in S}(Y_b = 1)] \leq \prod_{b\in S} y_b$, and $\mathbf{Pr}[\wedge_{b\in S}(Y_b = 0)] \leq \prod_{b\in S}(1-y_b)$.*

Both proofs can be found in the appendix. We use these properties in order to prove concentration results on the cost-related budget constraints. In Section 4.3 some relevant works in the literature of sampling combinatorial objects are discussed and compared to our method.

Stage ii: scaling $x_{i,b}$'s. In order to satisfy constraint (4) in the LP which requires $x_{i,b} \leq y_b$ for every $i \in \mathcal{K}$ and $b \in B$, we scale the $x_{i,b}$ variables proportional to the rounding result of y_b's. More formally, the outcome of Stage ii of Algorithm 1, denoted by $X^{(\text{ii})}$, is calculated as follows:

$$X_{i,b}^{(\text{ii})} := \begin{cases} \dfrac{x_{i,b}}{y_b} \cdot \dfrac{1}{1+2\sqrt{\epsilon(2+\log r)}}, & \text{if } Y_b = 1, \\ 0, & \text{if } Y_b = 0. \end{cases} \tag{17}$$

The scaling guarantees that Y and $X^{(\text{ii})}$ satisfy constraint (4) of the LP formulation. We analyze the cost constraints of the LP, i.e., Eq. (2) that maintains $\sum_{i,b} c_{i,b}^q x_{i,b} \leq 1$ for every $q \in [r]$. We prove the following claim.

CLAIM 2. *After Stage ii, with probability at least $1 - 0.1r^{-0.2}$ we have $\sum_{i,b} c_{i,b}^q X_{i,b}^{(ii)} \leq 1$ for all $q \in [r]$.*

Proof of Claim 2. Note that for every fixed b, if $y_b > 0$, we have

$$\sum_i c^q_{i,b} X^{(\mathrm{ii})}_{i,b} = \frac{1}{1+2\sqrt{\epsilon(2+\log r)}} \sum_i c^q_{i,b} \frac{x_{i,b}}{y_b} Y_b. \tag{18}$$

However, as a result of Lemma 2 we have $\mathbf{E}[Y_b] = y_b$. By using Eq. (2) of the LP, we have

$$\mathbf{E}\left[\sum_{i,b} c^q_{i,b} X^{(\mathrm{ii})}_{i,b}\right] = \frac{1}{1+2\sqrt{\epsilon(2+\log r)}} \sum_{i,b} c^q_{i,b} x_{i,b} \le \frac{1}{1+2\sqrt{\epsilon(2+\log r)}}. \tag{19}$$

On the other hand, note that the coefficient of Y_b in Eq. (18), i.e., $\sum_i c^q_{i,b} x_{i,b}/y_b$ is bounded by ϵ due to Constraint (9) when $y_b > 0$. Also, when $y_b = 0$, due to the construction of the algorithm $\tilde{x}_{i,b} = 0$ for all $i \in \mathcal{K}$. Therefore, Eq. (18) can be written in the following form,

$$\sum_{i,b} c^q_{i,b} X^{(\mathrm{ii})}_{i,b} = \frac{1}{1+2\sqrt{\epsilon(2+\log r)}} \sum_b \gamma^q_b Y_b, \tag{20}$$

where $0 \le \gamma^q_b \le \epsilon$ for all bids b and $q \in [r]$, and also, $\mathbf{E}[\sum_b \gamma^q_b Y_b] \le 1$ due to Eq. (19). Due to the negative correlation between Y_b's proved in Lemma 3, we can use Chernoff bounds for *independent random variables* to study the deviation from the expected value (see Panconesi and Srinivasan 1997). We use the following bound from Chung and Lu (2004). We emphasize that some symbols (for the random variables and their indices) used in Chung and Lu (2004) are modified to be consistent with our notation.

THEOREM 3 **(Theorem 6 in Chung and Lu 2004).** *Let Z_b (for $b \in B$) be independent random variables satisfying $Z_b \le \mathbf{E}[Z_b] + M$, for every b. We consider the sum $Z = \sum_b Z_b$ with expectation $\mathbf{E}[Z] = \sum_b \mathbf{E}[Z_b]$ and variance $\mathbf{Var}[Z] = \sum_b \mathbf{Var}[Z_b]$. Then we have*

$$\mathbf{Pr}[Z \ge \mathbf{E}[Z] + \lambda] \le \exp\left(-\frac{\lambda^2}{2(\mathbf{Var}[Z] + M\lambda/3)}\right).$$

To complete our proof, for any given q define $Z_b := \gamma^q_b Y_b$. Since γ^q_b's are fixed and Y_b's are negatively correlated according to Lemma 3, the bound of Theorem 3 is valid (see Panconesi and Srinivasan 1997). Note that $0 \le Z_b \le \gamma^q_b$. Also, as a result of Eq. (20), $\mathbf{E}[Z] = \mathbf{E}[\sum_b \gamma^q_b Y_b] \le 1$. Moreover, $\mathbf{Var}[Z_b] \le (\gamma^q_b)^2 y_b$. By using $\gamma^q_b \le \epsilon$, we have, $\sum_b \mathbf{Var}[Z_b] \le \sum_b (\gamma^q_b)^2 y_b \le \epsilon \sum_b \gamma^q_b y_b \le \epsilon$. (We emphasize that since Y_b's are not independent, $\sum_b \mathbf{Var}[Z_b]$ is not necessarily the same as $\mathbf{Var}[Z]$. However, due to the established negative correlation, the work of Panconesi and Srinivasan 1997 allows us to use Theorem 3 *as if* Z_b's were independent.) Finally, define $M := \epsilon$ and let $\lambda := 2\sqrt{\epsilon(2+\log r)}$. Since $\epsilon \le (2+\log r)^{-1}$, we have

$$\begin{aligned}
\mathbf{Pr}\left[Z \ge \mathbf{E}[Z] + 2\sqrt{\epsilon(2+\log r)}\right] &\le \exp\left(-\frac{4\epsilon(2+\log r)}{2(\epsilon + 2\epsilon\sqrt{\epsilon(2+\log r)}/3)}\right) \\
&\le \exp\left(-\frac{4\epsilon(2+\log r)}{2(\epsilon + 2\epsilon/3)}\right) \\
&\le \exp\left(-6(2+\log r)/5\right) \le (e^2 r)^{-1.2}.
\end{aligned}$$

Hence, for any fixed q, we have $\sum_b \gamma_b^q Y_b \le 1 + 2\sqrt{\epsilon(2+\log r)}$ (or, equivalently, by using (20), $\sum_b c_{i,b}^q X_{i,b}^{(\text{ii})} \le 1$) with probability at least $1-(e^2 r)^{-1.2}$. Applying union bound over all $q \in [r]$ guarantees that with probability at least $1 - e^{-2.4} r^{-0.2}$ we have $\sum_b c_{i,b}^q X_{i,b}^{(\text{ii})} \le 1$ for all $q \in [r]$. Therefore, with probability at least $1 - 0.1 r^{-0.2}$ all the budget constraints are preserved with $X^{(\text{ii})}$. This completes the proof of Claim 2. □

Let LP be the objective value of the linear program. Further, denote by $\mathsf{LP}^{(l)}$ the objective values for the LP solutions after Stage $l \in \{\text{ii}, \text{iii}, \text{iv}\}$. We emphasize that, although LP itself is a certain value, each $\mathsf{LP}^{(l)}$ is a random variable. We have $\mathbf{E}\left[X_{i,b}^{(\text{ii})}\right] = x_{i,b}/(1+2\sqrt{\epsilon(2+\log r)})$ for all i, b. Therefore, the objective of the LP after Stage ii can be bounded as follows.

$$\mathbf{E}\left[\mathsf{LP}^{(\text{ii})}\right] \ge \frac{1}{1+2\sqrt{\epsilon(2+\log r)}}\mathsf{LP}. \tag{21}$$

Note that $X^{(\text{ii})}$ and Y may still be infeasible, due to constraint (5). Stage iii deals with this issue.

Stage iii: trimming $X_{i,b}^{(\text{ii})}$'s to get a feasible solution. Note that after rounding the y_b variables and scaling the $x_{i,b}$ variables appropriately, some constraints (5) may be violated. In particular, for certain keywords i, we may have $\sum_b X_{i,b}^{(\text{ii})} > 1$. For each such keyword, we trim down all $X_{i,b}^{(\text{ii})}$ variables at the same rate to obtain $\sum_b X_{i,b}^{(\text{iii})} = 1$. Clearly, these operations do not violate any new constraints, and fix all the violated ones, hence, the result is a feasible solution. It only remains to show the loss in the objective value due to these operations is not too much. More specifically, we prove the following.

LEMMA 4. *We have* $\mathbf{E}\left[\mathsf{LP}^{(iii)}\right] \ge \lambda \mathbf{E}\left[\mathsf{LP}^{(ii)}\right]$ *for a positive constant value of* $\lambda = 0.539968$.

In order to provide an insight into our approach, we first provide a weaker proof of the above lemma for $\lambda = \frac{1}{4}$. At the end of the proof we will strengthen the analysis to $\lambda = 0.539968$.

Let random variable X_i denote $\sum_b X_{i,b}^{(\text{ii})}$. Furthermore, define random variable $P_i = \sum_b p_{i,b} X_{i,b}^{(\text{ii})}$, which is the contribution of keyword i to the objective value after Stage ii. For the simplicity of notation, we define $\xi = (1+2\sqrt{\epsilon(2+\log r)})^{-1}$. We remind that $\mathbf{E}\left[X_{i,b}^{(\text{ii})}\right] = \xi x_{i,b}$, $\mathbf{E}[P_i] = \xi \sum_b p_{i,b} x_{i,b}$, and $\mathbf{E}\left[\mathsf{LP}^{(\text{ii})}\right] = \sum_i \mathbf{E}[P_i]$.

Consider the contribution of all keywords i with $X_i > 2$ to the value of $\mathsf{LP}^{(\text{ii})}$. Let random variable $\hat{P}_i$ denote this contribution. By the following inequalities, we show that $\mathbf{E}[\hat{P}_i] \le \mathbf{E}[P_i]/2$, which means that in expectation at least half of the value of $\mathsf{LP}^{(\text{ii})}$ comes from keywords i with $X_i \le 2$.

$$\begin{aligned}\mathbf{E}\left[\hat{P}_i\right] &= \mathbf{E}\left[P_i \middle| X_i > 2\right] \cdot \mathbf{Pr}\left[X_i > 2\right] \\ &= \mathbf{E}\left[\sum_b p_{i,b} X_{i,b}^{(\text{ii})} \middle| X_i > 2\right] \cdot \mathbf{Pr}\left[X_i > 2\right]\end{aligned}$$

$$= \sum_b \mathbf{E}\left[p_{i,b}X_{i,b}^{(\mathrm{ii})} \middle| X_i > 2\right] \cdot \mathbf{Pr}\left[X_i > 2\right]$$
$$= \sum_b \mathbf{E}\left[p_{i,b}X_{i,b}^{(\mathrm{ii})} \middle| X_i > 2 \wedge X_{i,b}^{(\mathrm{ii})} > 0\right] \cdot \mathbf{Pr}\left[X_{i,b}^{(\mathrm{ii})} > 0\right] \cdot \mathbf{Pr}\left[X_i > 2 \middle| X_{i,b}^{(\mathrm{ii})} > 0\right]$$

Note that the condition "$X_{i,b}^{(\mathrm{ii})} > 0$" implies that y_b has been rounded up to 1, i.e., $Y_b = 1$. Thus, the first expectation is equal to $p_{i,b}\xi x_{i,b}/y_b$. Moreover, $\mathbf{Pr}\left[X_{i,b}^{(\mathrm{ii})} > 0\right] = \mathbf{Pr}\left[Y_b = 1\right] = y_b$. Therefore,

$$\mathbf{E}\left[\hat{P}_i\right] = \sum_b p_{i,b}\xi\frac{x_{i,b}}{y_b} \cdot y_b \cdot \mathbf{Pr}\left[X_i > 2 \middle| X_{i,b}^{(\mathrm{ii})} > 0\right]. \tag{22}$$

We now define $X_i^{-b} := X_i - X_{i,b}^{(\mathrm{ii})} = \sum_{b' \neq b} X_{i,b'}^{(\mathrm{ii})}$ and rewrite Eq. (22) as the following.

$$\mathbf{E}\left[\hat{P}_i\right] = \xi \sum_b p_{i,b}x_{i,b} \cdot \mathbf{Pr}\left[X_i^{-b} > 2 - X_{i,b}^{(\mathrm{ii})} \middle| X_{i,b}^{(\mathrm{ii})} > 0\right]. \tag{23}$$

Note that conditioned on $X_{i,b}^{(\mathrm{ii})} > 0$ we have $X_{i,b}^{(\mathrm{ii})} = \xi x_{i,b}/y_b$. Using Markov's inequality, we get

$$\mathbf{E}\left[\hat{P}_i\right] \leq \xi \sum_b p_{i,b}x_{i,b} \cdot \frac{\mathbf{E}\left[X_i^{-b} \middle| X_{i,b}^{(\mathrm{ii})} = \xi\frac{x_{i,b}}{y_b}\right]}{2 - \xi\frac{x_{i,b}}{y_b}} = \xi \sum_b p_{i,b}x_{i,b} \cdot \frac{\sum_{b' \neq b} \mathbf{E}\left[X_{i,b'}^{(\mathrm{ii})} \middle| X_{i,b}^{(\mathrm{ii})} = \xi\frac{x_{i,b}}{y_b}\right]}{2 - \xi\frac{x_{i,b}}{y_b}}. \tag{24}$$

Note that by Eq. (17) for all b where $y_b > 0$, we have $\mathbf{E}\left[X_{i,b'}^{(\mathrm{ii})} \mid X_{i,b}^{(\mathrm{ii})} = \xi x_{i,b}/y_b\right] = \mathbf{E}\left[X_{i,b'}^{(\mathrm{ii})} \mid y_b = 1\right] = \xi x_{i,b'}/y_{b'}\mathbf{Pr}\left[y_{b'} = 1 \mid y_b = 1\right]$ which, due to the negative correlation result of Lemma 3 is bounded from above by $\xi x_{i,b'}/y_{b'}\mathbf{Pr}\left[y_{b'} = 1\right] = \xi x_{i,b'}$. Through linearity of expectation for the numerator of Eq. (24) as well as $\mathbf{E}\left[\sum_{b'} X_{i,b'}^{(\mathrm{ii})}\right] = \xi \sum x_{i,b'} \leq 1$, we obtain the following.

$$\mathbf{E}\left[\hat{P}_i\right] \leq \xi \sum_b p_{i,b}x_{i,b} \cdot \frac{1 - \xi\frac{x_{i,b}}{y_b}}{2 - \xi\frac{x_{i,b}}{y_b}} \leq \xi \sum_b p_{i,b}x_{i,b} \cdot \frac{1}{2} = \frac{1}{2}\mathbf{E}\left[P_i\right]. \tag{25}$$

Now, note that in Stage iii, for each keyword i with $X_i > 1$ we scale down all $X_{i,b}$ variables by X_i. Hence, for each keyword i with $X_i \leq 2$, we lose at most a factor of 2 in the scaling process of Stage iii. However, Inequality (25) shows that at least half of the value of $\mathbf{E}[\mathsf{LP}^{(\mathrm{ii})}]$ is coming from such keywords. Hence, a quarter of $\mathbf{E}[\mathsf{LP}^{(\mathrm{ii})}]$ is preserved after Stage iii, or, $\lambda \geq \frac{1}{4}$.

The above analysis is suboptimal for two reasons. First, it ignores the contribution of $p_{i,b}x_{i,b}$ to the objective if X_i happens to be greater than two. Second, it treats all keywords i for which $X_i \leq 2$ similarly, and divides all of them by two, although some may only require a small scaling factor (or none at all). We take advantage of these observations to achieve $\lambda = 0.539968$.

Let $f(t)$ for $t \geq 0$ be a density function denoting what portion of the objective is due to keywords with $X_i = t$. In particular, $\int_0^\infty f(t)dt = 1$. Clearly, the rounding algorithm for Stage iii obtains a ratio of $\lambda = \int_{t=0}^1 f(t)dt + \int_{t=1}^\infty \frac{1}{t}f(t)dt$. The following lemma formalizes how we can take advantage of this idea. The proof can be found in the appendix.

LEMMA 5. *Let $G(t)$ be a nonincreasing function such that $G(1)=1$ and $G(t)\geq 0$ for $t\geq 0$. If $G(t)\geq \int_{x=t}^{\infty} f(x)dx$ for every $t\geq 1$, then we have $\lambda \geq -\int_{t=1}^{\infty} \frac{dG(t)}{tdt}dt$.*

The first choice for $G(t)$ comes from a modification of the derivation (22)-(25). If we replace the violation factor 2 for X_i in the derivation by $t\geq 1$, we get that $\mathbf{E}[P_i|X_i>t]\leq \frac{1}{t}\mathbf{E}[P_i]$. Summing over all keywords i, we have $G(t)=\frac{1}{t}$ for $t\geq 1$. One can verify that $G(t)$ satisfies all conditions of Lemma 5. Plugging $G(t)=1/t$ into the lemma gives $\lambda\geq \int_1^{\infty}\frac{1}{t^3}dt=-\frac{1}{2t^2}\big|_1^{\infty}=\frac{1}{2}$.

This is already an improvement over the weaker analysis, but we improve it even further. We observed that a modification of (22)-(25) shows $\mathbf{E}[P_i|X_i>t]\leq \frac{1}{t}\mathbf{E}[P_i]$. In fact, this is $\mathbf{E}[P_i|X_i>t]\leq \max_b \mathbf{Pr}\left[X_i^{-b}>t-\xi\frac{x_{i,b}}{y_b}\middle| X_{i,b}^{(\mathrm{ii})}=\xi\frac{x_{i,b}}{y_b}\right]\cdot\mathbf{E}[P_i]$.

From Lemma 3 we know that X_i^{-b} is the sum of negatively correlated random variables each of which takes values in the range $[0,1]$. Therefore, by a similar generalization of Chernoff bound to negatively correlated random variables that we used in the analysis of Stage ii (see Panconesi and Srinivasan 1997), for $s\geq 1$ and $\mu=\mathbf{E}\left[X_i^{-b}\right]\leq 1$, we have

$$\mathbf{Pr}\left[X_i^{-b}>s\right]\leq \left(\frac{e^{\frac{s}{\mu}-1}}{\frac{s}{\mu}^{\frac{s}{\mu}}}\right)^{\mu}=\exp\left(-\mu\left[1+\frac{s}{\mu}\left(\ln\frac{s}{\mu}-1\right)\right]\right) \tag{26}$$

$$\leq \exp\left(-\left[1+s\left(\ln s-1\right)\right]\right), \tag{27}$$

where the last derivation comes from the fact that the derivative of (26) is always positive for $\mu<1$. Let us denote by $\delta(s)$ the right-hand side of (27).

We already know $\mathbf{E}\left[X_i^{-b}\middle|X_{i,b}^{(\mathrm{ii})}=\xi\frac{x_{i,b}}{y_b}\right]\leq 1-\xi\frac{x_{i,b}}{y_b}$ due to negative correlation in rounding y_b variables. Also, $(t-\xi\frac{x_{i,b}}{y_b})/(1-\xi\frac{x_{i,b}}{y_b})\geq t$ for any $t\geq 1$, since $0\leq \xi\frac{x_{i,b}}{y_b}\leq 1$. The fact that $\delta(s)$ is decreasing in s yields $\mathbf{Pr}\left[X_i^{-b}>t-\xi\frac{x_{i,b}}{y_b}\middle|X_{i,b}^{(\mathrm{ii})}=\xi\frac{x_{i,b}}{y_b}\right]\leq\delta(t)$. Hence,

$$\mathbf{E}[P_i|X_i>t]\leq \delta(t)\cdot\mathbf{E}[P_i].$$

This bound is generally better than the one derived previously from applying Markov's inequality. We would plug it into Lemma 5, however, doing so only gives a bound $\lambda\geq 0.4443$. This is inferior to the 0.5 bound we derived above. We note that, although $\delta(t)$ is a better bound than $1/t$ for large values of t, it is not so if $t\leq e$. To obtain the stronger result, then, we use $G(t)=\min\left(\frac{1}{t},\delta(t)\right)$. Plugging this in Lemma 5 and integrating numerically gives $\lambda\geq 0.539968$.

Stage iv: rounding $X_{i,b}^{(\mathrm{iii})}$ variables. We can simply pick at most one bid b for each keyword i with probability equal to $X_{i,b}^{(\mathrm{iii})}$. This rounding is independent across keywords and has negative correlation within keyword (since each keyword i chooses at most one bid). Therefore, it enjoys concentration properties (via Chernoff bounds) for the cost vector due to Panconesi and Srinivasan (1997). Before presenting the result, we remind the reader that for any i and b where $X_{i,b}^{(\mathrm{iii})}>0$ we have $x_{i,b}>0$ due to the construction of our algorithm, and thus, $c_{i,b}^q\leq\epsilon$ due to Constraint (7).

CLAIM 3. *For $\tilde{X}^{(iv)}_{i,b}$ calculated in Line 24 of the approximation algorithm, with probability $1-0.2r^{-0.2}$ we have $\sum_{i,b} c^q_{i,b}\tilde{X}^{(iv)}_{i,b} \le 1+2\sqrt{\epsilon(2+\log r)}$ for all $q\in[r]$ and also, $\sum_{i,b} p_{i,b}\tilde{X}^{(iv)}_{i,b} \ge (1-2\sqrt{\epsilon(2+\log r)})\sum_{i,b} p_{i,b}X^{(iii)}_{i,b}$.*

Proof of Claim 3. Similar to the proof of Claim 2, we invoke the result of Theorem 3. For any given budget constraint $q\in[r]$, we define $Z_{i,b} := c^q_{i,b}\tilde{X}^{(\text{iv})}_{i,b}$ and let $\lambda := 2\sqrt{\epsilon(2+\log r)}$ and $M := \epsilon$. We have $\mathbf{E}\left[\sum_{i,b} Z_{i,b}\right] = \mathbf{E}\left[\sum_{i,b} c^q_{i,b}\tilde{X}^{(\text{iv})}_{i,b}\right] = \mathbf{E}\left[\sum_{i,b} c^q_{i,b}X^{(\text{iii})}_{i,b}\right] \le 1$. Also, due to the fact that $c^q_{i,b}\le\epsilon$ and similar to the argument in the proof of Claim 2, we have $\sum_{i,b}\mathbf{Var}[Z_{i,b}]\le\epsilon$. Therefore, using Theorem 3, for any given q we have $\mathbf{Pr}\left[\sum_{i,b} c^q_{i,b}\tilde{X}^{(\text{iv})}_{i,b} > 1+\lambda\right] \le 0.1r^{-1.2}$. Similarly, for the objective, and by using the Chernoff for the lower tail (see Theorem 7 of Chung and Lu 2004), by defining $Z_{i,b} := p_{i,b}\tilde{X}^{(\text{iv})}_{i,b}$, $\lambda := 2\sqrt{\epsilon(2+\log r)}(\sum_{i,b} p_{i,b}X^{(\text{iii})}_{i,b})$, and $M:=0$ we have $\mathbf{Pr}\left[\sum_{i,b} p_{i,b}\tilde{X}^{(\text{iv})}_{i,b} < (1-\lambda)\sum_{i,b} p_{i,b}X^{(\text{iii})}_{i,b}\right] \le \exp(-2(2+\log r)) \le 0.1r^{-2}$. Hence, by applying union bound, we establish that the probability of violation with more than a factor of $1+2\sqrt{\epsilon(2+\log r)}$ for any of the r budget constraints, or losing more than a factor of $1-2\sqrt{\epsilon(2+\log r)}$ in the objective is less than $r\times 0.1r^{-1.2}+0.1r^{-2}\le 0.2r^{-0.2}$. This completes the proof. □

As the final step, we need to fix the budget constraints that are (slightly) violated according to Claim 3. We show that since individual contributions to cost are small, we can remove, and throw away a small portion of the cost with a loss in value that is no more than $2\sqrt{\epsilon(2+\log r)}\mathsf{LP}^{(\text{iii})} + \epsilon r\mathsf{OPT}$. In order to do so, for each dimension where the budget constraint is violated, throw away a minimal subset of keywords (i.e., set their bids to zero) by picking them in increasing order of profit over cost ratio until the cost is sufficiently reduced to satisfy the constraint. This greedy procedure ensures that we do not throw away too much cost. In particular, the loss factor in the profit is no more than the violation factor in the budgets (i.e., at most $2\sqrt{\epsilon(2+\log r)}$). Finally, for each dimension, we may have to throw away one last keyword in order to make the solution integer again. This will cost us the profit of one keyword and bid, bounded by $\epsilon\mathsf{OPT}$ in each dimension, hence, at most $\epsilon r\mathsf{OPT}$ in total.

If we denote by $X^{(\text{iv})}_{i,b}$ the $\tilde{X}^{(\text{iv})}_{i,b}$'s that are not thrown away, the total loss of the Stage iv of the rounding is bounded by $2\sqrt{\epsilon(2+\log r)}\mathsf{LP}^{(\text{iii})}$ from randomized rounding according to Claim 3 plus $2\sqrt{\epsilon(2+\log r)}\mathsf{LP}^{(\text{iii})} + \epsilon r\mathsf{OPT}$ according to the final removal of bids explained above. Therefore, we have the following claim.

CLAIM 4. *With probability at least $1-0.2r^{-0.2}$ we have $\sum_{i,b} c^q_{i,b}X^{(iv)}_{i,b}\le 1$ for every $q\in[r]$, and $\mathbf{E}\left[\mathsf{LP}^{(iv)}\right] \ge (1-4\sqrt{\epsilon(2+\log r)})\mathsf{LP}^{(iii)} - \epsilon r\mathsf{OPT}$.*

Combining the Analysis of Stages i to iv. Claim 4, combined by the fact that $\mathbf{E}\left[\mathsf{LP}^{(\mathrm{ii})}\right] \geq (1-2\sqrt{\epsilon(2+\log r)})\mathsf{LP}$ due to Eq. (21), the fact that $\mathbf{E}\left[\mathsf{LP}^{(\mathrm{iii})}\right] \geq 0.539\mathbf{E}\left[\mathsf{LP}^{(\mathrm{ii})}\right]$ in Stage iii according to Lemma 4, and the fact that $\mathsf{OPT} \leq \mathsf{LP}$ achieves the following bound on the performance of the rounding algorithm.

$$\begin{aligned}
\mathbf{E}\left[\mathsf{LP}^{(\mathrm{iv})}\right] &\geq (1-4\sqrt{\epsilon(2+\log r)})\mathbf{E}\left[\mathsf{LP}^{(\mathrm{iii})}\right] - \epsilon r\mathsf{OPT} \\
&\geq 0.539(1-4\sqrt{\epsilon(2+\log r)})\mathbf{E}\left[\mathsf{LP}^{(\mathrm{ii})}\right] - 0.539\epsilon r\mathsf{OPT} \\
&\geq 0.539(1-6\sqrt{\epsilon(2+\log r)})\mathsf{LP} - 0.539\epsilon r\mathsf{OPT} \\
&\geq (0.539-6\sqrt{\epsilon(2+\log r)} - \epsilon r)\mathsf{LP}.
\end{aligned}$$

Also, the solution and the corresponding inequality above always remains feasible for the Stages i and iii, whereas it remains feasible in each of the Stages ii and iv with probability $1-0.1r^{-0.2}$ and $1-0.2r^{-0.2}$ due to Claims 2 and 4, respectively. Therefore, we have our main theoretical result.

THEOREM 4. *Algorithm 1 provides a $(0.539-\delta)$-approximate integral solution to the LP (1-6) with additional constraints (7-9) for $\delta = 6\sqrt{\epsilon(2+\log r)} + \epsilon r$ with probability at least $1-0.3r^{-0.2}$.*

In Section 4.4 we provide an instance of the aforementioned LP with an integrality gap of 0.63.

REMARK 2. We emphasize that our rounding procedure is very fast. In fact, it can be implemented in linear running time in terms of the size of its input. This can be done by rounding at first only the y_b variables in $O(|B|)$ running time, and then rounding $x_{i,b}$ variables accordingly in $O(|\mathcal{K}|\cdot|B|)$ running time.

4.3. Some Notes on the Rounding of Stage i

In this section we discuss some relevant literature regarding the sampling problem in Stage i. The problem of interest is to sample k balls among $m = |B|$ given balls at random with known marginal probabilities y_b. (We note that in our problem, each ball represents a bid.) In this paper, we showed how the problem can be seen as a problem of sampling a random matching in a given bipartite graph $G = (V, E)$ where in our problem $V = [k] \cup B$. Note that any set of marginal probabilities $(F)_{[k]\cup B}$ on the edges of the bipartite graph that satisfy Eq. (10-11) can be seen as a doubly substochastic matrix. However, every doubly substochastic matrix F can be decomposed as the convex combination of 0/1 doubly substochastic matrices $M_1, M_2, \cdots$, i.e., $F = \sum_l \alpha_l M_l$, such that α_l's are non-negative and $\sum_l \alpha_l = 1$. (To see this, one can first make F a square matrix by adding $|B|-k$ rows of zero, then use the result of von Neumann 1953 to turn it into a doubly stochastic matrix, and finally, use Birkhoff 1946's decomposition of doubly stochastic matrices into permutation matrices. Note that since the summation of each of the first k rows are 1, the von Neumann method will not modify any of the entries in the original $k \times |B|$ matrix.)

Note that any such decomposition is equivalent to sampling a random matching which preserves the marginals on edges (and consequently on the vertices, i.e., the bids). This is done by sampling the matching corresponding to each 0/1 matrix M_l in the decomposition with probability α_l.

However, such decompositions are not necessarily unique, and we are particularly looking for something that in addition to preserving the marginals, impose negative correlation on the vertices. (We need this in the later stages of our algorithm in order to get the concentration results.) This is done by adapting the maximum-entropy sampling algorithm of Asadpour and Saberi (2010), when the underlying graph is a forest. We designed a linear-time implementation of this algorithm for the purpose of our problem.

Srinivasan (2001) also provides an alternative way of sampling k balls among m given balls with known marginals. The method of Srinivasan (2001) is based on choosing two arbitrary balls at each step, and picking or throwing out one of them (according to certain probabilities), then continuing with the remaining balls. Srinivasan (2001) shows that this method also produces negative correlation among the balls. Therefore, it can also be used for the purpose of our algorithm as the rounding of Stage i. The two main differences between our algorithm and that of Srinivasan (2001) are as follows.

- Our algorithm can potentially be used if the order of k chosen balls (in the context of this paper, the bids) matters. This is due to the fact that our algorithm works with the marginal probabilities on the edges and has the ability to match a specific ball to any number $1, 2, \cdots, k$ with the desired property.

- Our algorithm uses k random draws to choose the k random balls. However, Srinivasan (2001) uses $|B|$ such draws.

Finally, there is another direct way of choosing k balls among m given balls with known marginals, which is to directly sample from the maximum entropy distribution over the cardinality matroid. The underlying distribution will sample each subset S where $|S| = k$ with probability $\alpha^\star_S$, the optimum solution of the following convex program.

$$\begin{aligned} \text{maximize} \quad & \sum_S -\alpha_S \log \alpha_S \\ \text{s.t.} \quad & \sum_{S \ni b} \alpha_S = y_b \qquad \forall S \in B : |S| = k, \text{ and } b \in B \\ & \alpha_S \geq 0, \qquad \forall b \in B. \end{aligned}$$

This convex program, even though exponential in size, can be solved in polynomial time due to the fact that cardinality matroids are balanced (see Singh and Vishnoi 2014). The potential advantage of this distribution to both ours and Srinivasan (2001) is that it does not introduce add any artificial structure to the problem as a result of maximizing the entropy (see the discussion

in Asadpour et al. 2017). In particular, it allows for any two bid to be in the final solution together whereas, in our approach if the bids are connected to a single vertex in $[k]$, then they will not be chosen at the same time in the final solution. Similarly, it happens in Srinivasan (2001) that if the bids are chosen together at some point in the rounding they will not be present in the final solution together. However, the maximum entropy rounding in this problem requires using the Ellipsoid method and therefore, may not be practical for the large scale problems similar to what we face in our model.

4.4. Upper Bound For the Integrality Gap

We show that the integrality gap of the LP we wrote earlier is bounded from above by $1-\frac{1}{e}\approx 0.63$. The instance has $m=|B|$ bids, $n=\binom{m}{\frac{m}{k}}$ keywords, and $r=1$ resource. We assume m is divisible by k and that $m\geq k^2$. We do not specify the bids themselves, rather we only think of them abstractly and merely specify their landscapes. Each keyword has a distinct subset of $\frac{m}{k}$ "feasible" bids; i.e., $c_{i,b}=0$ for feasible (keyword, bid) pairs and 1 otherwise. The resource limit is also $L=0$, which implies that only feasible bids can be used for keywords. The values are $p_{i,b}=1$ for all i,b.

There is a fractional solution of value n as follows. For all b, we have $y_b=k/m$, hence in total $\sum_b y_b=k$. For eack keyword i, $x_{i,b}=k/m$ if b is a feasible bid for it. Therefore, the cost is zero. The value obtained from each keyword is $\sum_b x_{i,b}=\frac{m}{k}\times\frac{k}{m}=1$, hence, a total of n overall.

However, we argue that no integral solution can be better than $n(1-\frac{1}{e})$. The optimal integral solution picks a subset B' of k bids. By the construction of our example, there exists exactly $\binom{m-k}{\frac{m}{k}}$ keywords i where $c_{i,b}=1$ for all $b\in B'$, i.e., all bids in B' are infeasible for i. If the integral solution collects the profit of all the other keywords, its value will be at most $n-\binom{m-k}{\frac{m}{k}}$. Therefore, the approximation ratio is bounded from above by,

$$\frac{\binom{m}{\frac{m}{k}}-\binom{m-k}{\frac{m}{k}}}{\binom{m}{\frac{m}{k}}}=1-\frac{(m-\frac{m}{k})(m-\frac{m}{k}-1)\cdots(m-\frac{m}{k}-k+1)}{m(m-1)\cdots(m-k+1)}\leq 1-\left(\frac{m-\frac{m}{k}-k+1}{m-k+1}\right)^k\leq 1-\left(1-\frac{1}{k-1}\right)^k,$$

where the last inequality is a result of $m\geq k^2$. Taking k to ∞ yields the claimed result.

We conclude this section by a conjecture stating that this bound is tight.

Conjecture 1. *The integrality gap of the LP (1-6) with additional constraints (7-9) is indeed* $1-\frac{1}{e}\approx 0.63$. *In particular, there exists a polynomial time algorithm to achieve this bound. Consequently, the integralty gap of the LP (1-6) will be* at least $1-\frac{1}{1.63}\approx 0.386$.

5. Experimental Study

In order to evaluate the practical performance of our algorithms, we apply our algorithms to real datasets collected from a sponsored search advertising system, and compare our results with a baseline method, i.e., uniform bidding. For a set of (randomly selected, anonymous) advertisers,

we consider a set of queries on which they might wish to advertise. We use a traffic estimator tool to estimate the number of clicks the advertiser will get (as a measure for value) and the cost he will have to pay (as a measure for $\overrightarrow{\text{cost}}$) when he places a bid b. Such tools are provided for major sponsored search advertising systems (Bing Ads 2013, Google Support 2013b).

There are three datasets related to three different companies. Each dataset contains varying number of queries (from tens to tens of thousands). These companies have market caps with different orders of magnitude. They also belong to different sectors of the market. In order to preserve confidentiality, we denote these data sets by S, M, and L standing for a small size, a medium size, and a large size data set. The small size data set contains less than one hundred queries, while the medium size and the large size data sets contain a few thousands and more than ten thousand queries. Especially, the large data sets corresponds to advertisers that sell a wide variety of products so their queries exhibit high semantic diversity. For each query we obtain estimates of clicks and cost for bids in the range $[0.10, 2]$. We run our algorithm (based on the linear programming relaxation of the problem and the subsequent dependent randomized rounding), an appropriate version of the uniform bidding algorithm, and an algorithm that computes the optimal bid for each query against each of the datasets (based on solving the integer programming formulation of the problem using open source softwares available from COIN-OR[3] and GLPK[4]). We apply these algorithms at different budget values to see the impact of changing the budget in the relative performance of different algorithms.

Finally, all values are normalized and expressed as a percentage of the optimal. This, in addition to ensuring anonymity, makes it easier to compare their relative performance, too.

5.1. A one-dimensional budget constraint

Our initial experiments involve only one budget constraint. Note that this is the setting in which the uniform-bidding algorithm was proposed and analyzed (Feldman et al. 2007).

We first see how the total amount of clicks that can be obtained grows as the number of permitted bids increases. Figure 4 plots the performance of the bid allocation we find for different numbers of bids. The ratio of "the optimal fractional LP solution for a given number of bids k" to "the optimal fractional LP solution for arbitrary number of bids (i.e., after removing constraint (2) from the LP or equivalently, setting $k = \infty$)" is plotted in the graph. It shows how the objective value of the *fractional* LP (before the rounding step) for each dataset grows with the number of clusters. This follows immediately from the fact the set of feasible solution for the linear programming formulation only expands with larger values of k. From a managerial perspective, it confirms the

[3] Available at http://www.coin-or.org/projects/.

[4] Available at https://www.gnu.org/software/glpk/.

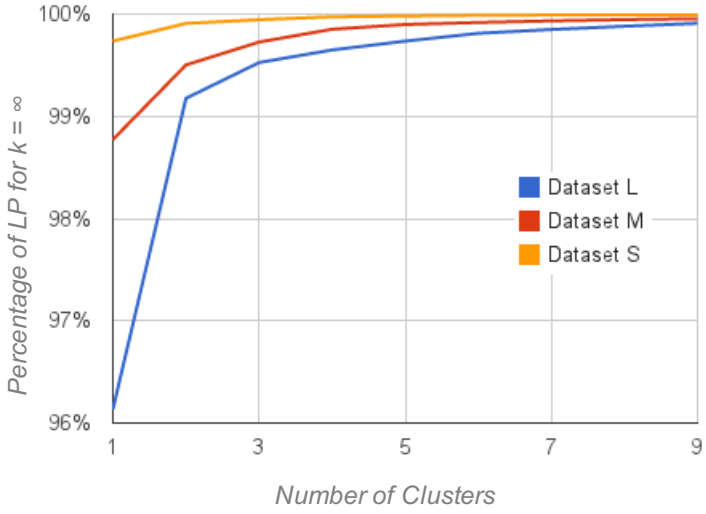


Figure 4 **Comparison of the performance of fractional solutions according to the number of clusters for all datasets. The y-axis shows the percentage of each solution with respect to the optimal LP solution for $k = \infty$.**

intuition that additional clusters allow for more refined bidding on various queries, hence better performance.

We emphasize that the LP solution is a concave function in k, the number of clusters. This is due to the more general observation that the solution of any maximization linear program in the standard form is a concave function of the right hand side of the constraints (see for instance, Luenberger and Ye 2016). It can be observed from the experiments that with only a few clusters, we can achieve the almost optimal unrestricted solution (i.e., where $k = \infty$). In fact, in all three data sets the LP for $k = 2$ achieves more than 99% of the unrestricted optimal solution, which shows that there is no need for a complicated strategy.

Next we consider the integral solutions constructed by our algorithm for each of these datasets. The values in the figures are normalized with respect to the optimal solution that chooses an individual bid for each query. We report the performance of our algorithm for number of clusters bounded by $k = 1, 2, 3, 4$. The algorithm assigns each query to at most one of the clusters. Note that in particular, it may leave some queries unassigned, thus effectively bidding zero on those queries. We refer to these algorithms as int.k where k is the maximum number of bids we choose.

We also report the performance of the uniform bidding algorithm of Feldman et al. (2007), which chooses an optimal set of two bids, then with some probability chooses the first bid for all queries and otherwise, chooses the other bid for all queries. The two chosen bids correspond to the two adjacent corner points of the two-dimensional convex hull of the aggregated $(\mathsf{value}, \mathsf{cost})$ landscape on which the highest value point under the given budget constraint lies. They proved a $(1 - 1/e)$-approximation ratio for this algorithm under a single budget constraint.

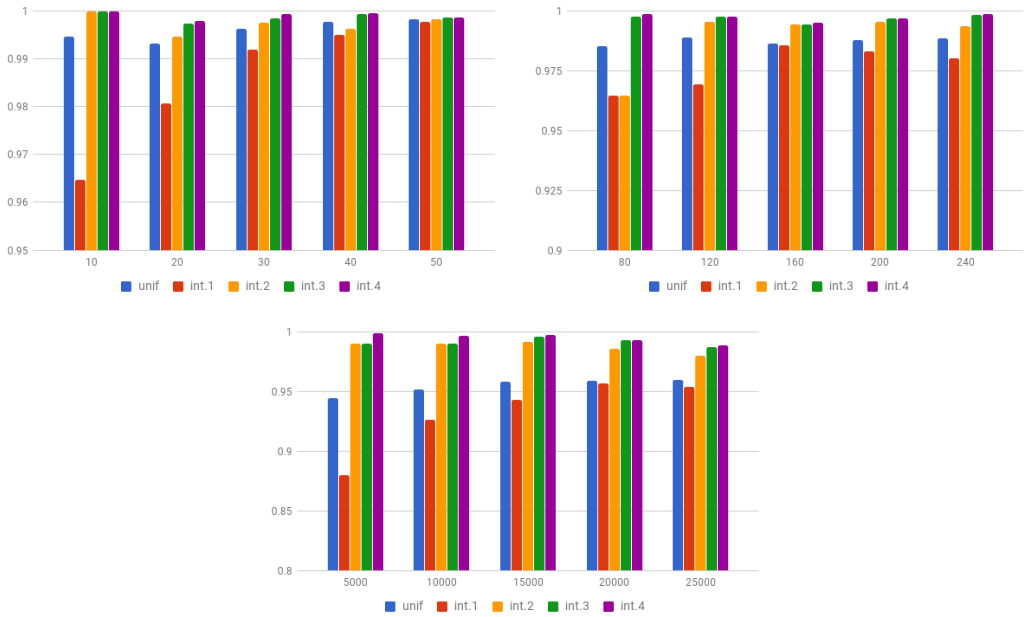


Figure 5 **These plots are comparing the performance of various algorithms on the three datasets. The x axis has budget in dollars, whereas the y axis shows the percentage of the fractional optimal solution with $k = \infty$. In the top left (resp. top right) the results for the data set S (resp. M) are reported, where the bottom graph shows the results for the data set L.**

Plots in Figure 5 compare the performance of integer solutions produced by LP rounding with the uniform bidding benchmark. We see that all algorithms perform almost equally well at high budgets or when the instance is fairly small, where even the optimal fractional solution is almost integral. On the other hand at lower values of budget and specially in the larger instances, we see superiority of our algorithms as the number of clusters grows. This is due to the fact that a large instance with low flexibility on the budget may introduce some pathological structures with which uniform bidding lacks the degree of freedom to deal. In particular, for the large dataset L, we see improvements of 4% to 6% in the expected number of clicks compared to the uniform bidding strategy when we increase the number of clusters to four. For other datasets which are smaller the average improvement in number of clicks is about 1%.

5.2. A multidimensional budget constraint

A useful property of our algorithm is that it is designed to take care of the setting where multiple budget constraints are present. These additional budget constraints may arise when an advertiser, considering the semantic meaning of different queries, wishes to restrict how much of her budget is spent on specific queries or domains.

We consider three different datasets each with up to six total budget constraints. The first constraint in each set puts a limit on the total cost of all queries. The other budget constraints

correspond to a subset of the queries and constraint the amount of money that may be spent on these. The additional budget constraints are overlapping, have approximately the same number of queries (around 40%), and have the same budget constraint.

We compare the performance of our algorithm, int.k, as defined before, to a uniform bidding algorithm that chooses a single nonzero bid b. This algorithm is further allowed to randomize between bidding zero and a bid b on all queries while respecting the budget constraints in expectation if that yields a higher number of expected clicks. Whenever appropriate, we also report the performance of frac.opt which is the optimal fractional solution with $k = \infty$ for the given data set and budget.

The three datasets we considered are termed S', M', and L'. They are the same sets of queries as S, M, and L, only with additional budget constraints. In Figure 6, we consider the objective value for various algorithms as the number of constraints increases while the budget is held fixed. All values are normalized with the maximum objective value obtainable — by frac.opt (i.e., $k = \infty$) with only one additional constraints. We see that the performance of each algorithm suffers as the number of constraints increases. It can be observed that in all three data sets the performance of our algorithm is superior to that of the adjusted uniform bidding by between 10% to 20%. Our algorithm that chooses up to 3 or 4 bids performs fairly close to optimal. The algorithm int.1 is between 4% to 20% lower than the optimum solution, while the algorithm int.2 ranges from almost optimum to 5% below optimum. The gap between frac.opt and other solutions generally increases with more constraints. The relative performance of uniform bidding is inferior to our algorithm even with one additional budget constraint as can be observed in Figure 6.

In Figure 7, we focus on a single algorithm (int.3) and consider how its performance varies for different budgets as the number of constraints is varied. All of the values are normalized relative to the maximum objective value obtainable — by frac.opt with no additional constraints and highest budget. Note that all additional constraints have the same budget.

We note that the algorithm obtains lower objective value (relative to the unrestricted optimal solution) in cases with lower budgets. This is to be expected, since the problem is more heavily constrained with the lower budget and as a result, the lower degree of freedom makes the job of the rounding more difficult. We also see a slight decrease in objective value as the number of constraints increases. This decrease is driven by additional constraints further constraining how bids for the queries can be set and affecting the rounding procedure.

Finally, in Figure 8 we look at the performance ratio of our algorithm with respect to the unrestricted optimal solution for a given number of constraints and a given budget. We see a large improvement over the uniform bidding algorithm and also as the number of bids is increased. Our algorithms obtain about 5% to 35% better approximation to the optimal compared to uniform

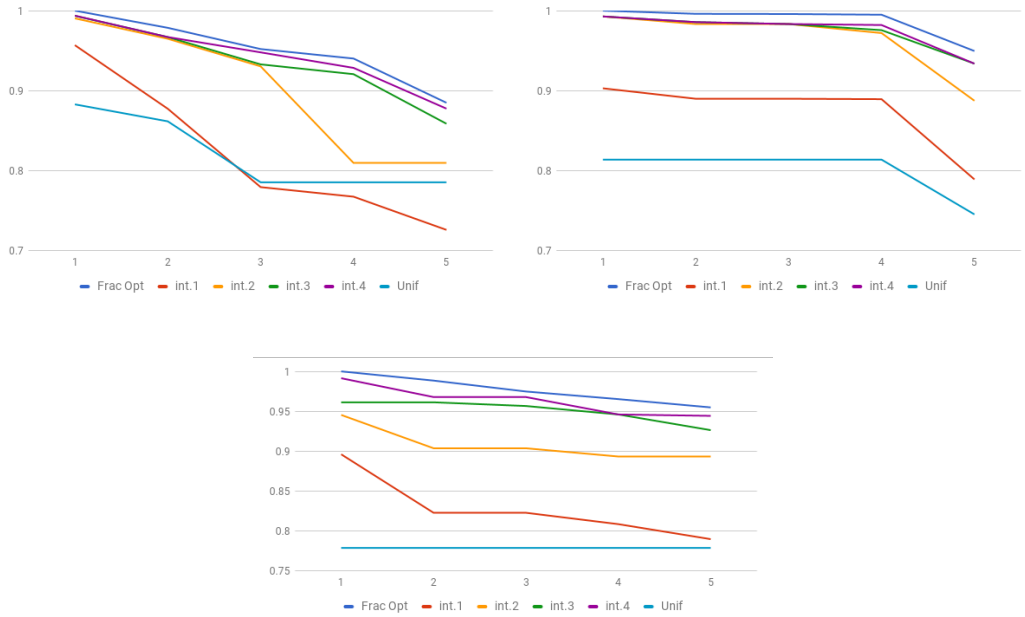


Figure 6 **These plots are comparing the performance of our algorithms for a fixed budget on the three datasets. The x axis has number of additional constraints, whereas the y axis shows the performance ratio with respect to the fractional optimal solution with $k = \infty$ and only one budget constraint. In the top left (resp. top right) the results for the data set S' (resp. M') are reported, where the bottom graph shows the results for the data set L'.**

bidding with a single bid. We observe that the gap between uniform bidding and our algorithms closes at higher budgets where the problem becomes more flexible. The size of the data set does not pay a significant role. This is because both the size of the extra constraints and size of the budget scale with the size of the data set.

The result of experimental studies can be summarized in the following observations.

– The LP formulation incurs a loss below 4% for $k = 1$ compared to $k = \infty$. The loss drops to below 1% for $k = 2$. This confirms that even very concise bidding strategies may capture the full value of the advertisers.

– In the case of single budget constraint, our concise bidding strategy outperforms uniform bidding, specially in the presence of low budgets in large data sets (a 4% to 6%) improvement.

– With multiple budget constraints, the gain of concise bidding strategies is even higher and can be up to 35%.

– The loss from the rounding (the rounding gap) increases with higher number of constraints.

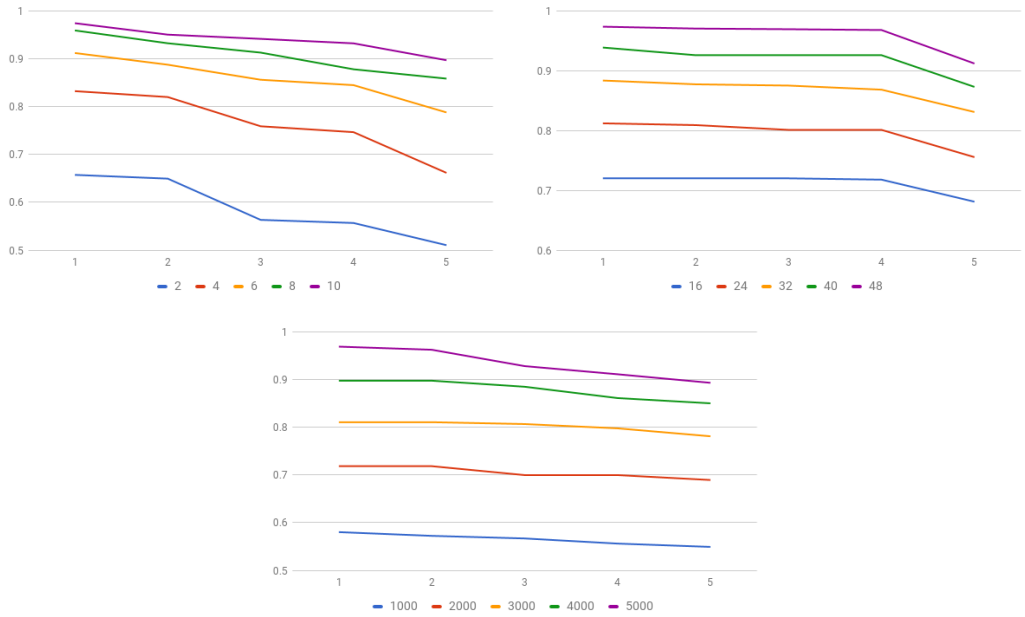


Figure 7 These plots are comparing the performance of various algorithms on the three datasets with up to three different bids. The x axis has number of additional constraints, whereas the y axis shows the performance ratio with respect to the fractional optimal solution with $k = \infty$ and only one budget constraint. In the top left (resp. top right) the results for the data set S' (resp. M') are reported, where the bottom graph shows the results for the data set L'.

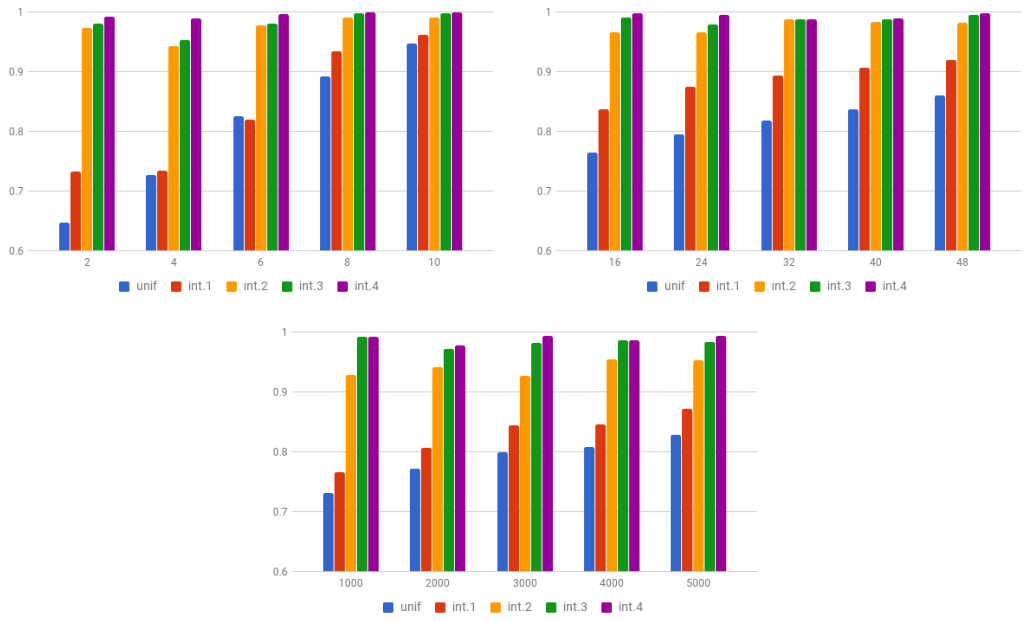


Figure 8 These plots are comparing the performance of various algorithms on the three datasets with four total budget constraints. The x axis has the budget in dollars, whereas the y axis shows the percentage of the fractional optimal solution with $k = \infty$ and four budget constraints at the given budget.

6. Conclusion and Future Directions

We formulate the problem of finding the concise bidding strategy for advertisers in order to obtain the maximum number of clicks (or maximize any other monotone profit function) subject to a multidimensional budget constraint.

When the budget constraint has constant dimensions, we propose a polynomial-time approximation scheme (PTAS). Otherwise, we present an LP rounding algorithm that is both fast and simple to implement. While the approximation guarantee for this algorithm is ≈ 0.54, it performs much better in practice. In particular, even for the case of a one-dimensional budget constraint, our algorithm beats the state of the art algorithm (uniform bidding) by 1% to 6%. Conveniently this is achieved by very concise bidding strategies that use only two or three different bids (where uniform bidding uses one). The gap between the performance of our algorithm and the enhanced uniform bidding widens in the case of having a small number (e.g., two) extra dimensions in the budget constraint to guarantee diversity for advertisement targeting. In this case, our algorithms outperforms the state of the art by an average of up to 35%.

From a managerial point of view, not only the platform of concise bidding improves the efficiency of the advertisers (and consequently, the desirability of the service provided by the publishers such as search engines), but it also provides a natural classification of the keywords (in k groups) in terms of their importance.

One obvious future direction would be to improve the analysis of our LP rounding. We conjecture that the integrality gap is $1 - \frac{1}{e}$ and that our current rounding approach indeed achieves this approximation factor. Another possible research direction is to investigate the effect of k (the maximum number of possible bids provided to the advertiser) on the optimum solution of the problem. Currently, we assume that the value of k is given, and based on that we provide a set of—at most—k bids to the advertiser to choose their bid from. However, it is not clear how the value of k itself should be determined. The trade-off here is between simplicity (i.e., lower values of k that lead to a more concise set of possible bids) and performance (higher values of k which lead to a broader set of feasible solutions and consequently, improve the optimum solution). One approach to this question would be to examine the value of the optimum solutions for different values of k. A study of this question for one-dimensional budget constraints (as also reported in the experiments in Figure 4) suggests that the expected marginal gain from allowing one more possible bid (i.e., adding one unit to k) is *diminishing*. In other words, the expected profit is a concave function of k. (As discussed in Section 5 this effect holds for the fractional optimum solution due to the LP being a maximization in the standard form and k is on the right hand side. However, this is not clear to be correct for the integral optimal solution or for our randomized rounding solution.) However, the reported results in Figure 4 suggest that the marginal gains rapidly diminish and most of the gain

is captured by going from $k=1$ to $k=2$, where we go from "forcing the solution to use the single bid available" to "allowing the solution to optimize over two available bids". Formalizing these observations would be very helpful in providing better insight about the nature of the problem and the challenges the advertisers and the publishers face in strategizing their campaigns and ad allocation mechanisms.

Acknowledgement

The authors would like to thank the Department editor, Associate editor, and two anonymous referees whose comments and feedback greatly helped us improve the paper.

References

Aggarwal, G., A. Goel, R. Motwani. 2006. Truthful auctions for pricing search keywords. *ACM Conference on Electronic Commerce (EC)*. 1–7.

Archak, N., V. S. Mirrokni, S. Muthukrishnan. 2012. Budget optimization for online campaigns with positive carryover effects. *Workshop on Internet & Network Economics*. 86–99.

Asadpour, A., M. X. Goemans, A. Madry, S. Oveis Gharan, A. Saberi. 2017. An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. *Operations Research* **65**(4) 1043–1061.

Asadpour, A., A. Saberi. 2010. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM J. Comput.* **39**(7) 2970–2989.

Bateni, M., J. Feldman, V. Mirrokni, S. C. Wong. 2014. Multiplicative bidding in online advertising. *ACM Conference on Economics and Computation (EC)*. 715–732.

Bing Ads. 2013. Bing ads intelligence. http://advertise.bingads.microsoft.com/en-us/bingads-downloads/bingads-intelligence.

Birkhoff, G. 1946. Three observations on linear algebra. *Univ. Nac. Tacumán Rev.* (5) 147–151.

Borgs, C., J. T. Chayes, N. Immorlica, K. Jain, O. Etesami, M. Mahdian. 2007. Dynamics of bid optimization in online advertisement auctions. *World Wide Web*. 531–540.

Chung, F., L. Lu. 2004. Coupling online and offline analyses for random power law graphs. *Internet Mathematics* **1**.

Devanur, N. R., T. P. Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. *ACM Conference on Electronic Commerce (EC)*. 71–78.

Dobzinski, S., R. Lavi, N. Nisan. 2012. Multi-unit auctions with budget limits. *Games and Economic Behavior* **74**(2) 486–503.

Edelman, B., M. Ostrovsky, M. Schwarz. 2005. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review* **97**(1) 242–259.

Even-Dar, E., V. S. Mirrokni, S. Muthukrishnan, Y. Mansour, U. Nadav. 2009. Bid optimization for broad match ad auctions. *World Wide Web*. 231–240.

Feldman, J., M. Henzinger, N. Korula, V. S. Mirrokni, C. Stein. 2010. Online stochastic packing applied to display ad allocation. *European Conference on Algorithms (ESA)*. 182–194.

Feldman, J., N. Korula, V. S. Mirrokni, S. Muthukrishnan, M. Pal. 2009. Online ad assignment with free disposal. *Workshop on Internet & Network Economics*. 374–385.

Feldman, J., S. Muthukrishnan, M. Pal, C. Stein. 2007. Budget optimization in search-based advertising auctions. *ACM Conference on Electronic Commerce (EC)*. 40–49.

Goel, G., V. S. Mirrokni, R. Paes Leme. 2012. Polyhedral clinching auctions and the adwords polytope. *STOC*. 107–122.

Google Support. 2011. Google adwords dynamic search ads. http://adwords.blogspot.fr/2011/10/introducing-dynamic-search-ads-beta.html.

Google Support. 2013a. Google adwords automatic bidding tools. http://support.google.com/adwords/bin/answer.py?hl=en&answer=2470106.

Google Support. 2013b. Traffic estimator. http://support.google.com/adwords/bin/answer.py?hl=en&answer=6329.

Google Support. 2013c. Using the bid simulator. http://support.google.com/adwords/bin/answer.py?hl=en&answer=2470105.

Google Support. 2014. Setting bid adjustments. https://support.google.com/adwords/answer/2732132.

Hastad, J. 1999. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica* **182**(1) 105–142.

IAB. 2017. Internet advertising revenue report: 2016 full year results. https://www.iab.com/wp-content/uploads/2016/04/IAB_Internet_Advertising_Revenue_Report_FY_2016.pdf.

Kulik, A., H. Shachnai, T. Tamir. 2011. Approximations for monotone and non-monotone submodular maximization with knapsack constraints. *CoRR* **abs/1101.2940**.

Luenberger, D. G., Y. Ye. 2016. *Linear and Nonlinear Programming*, *International Series in Operations Research & Management Science*, vol. 228. 4th ed. Springer.

Mehta, A., A. Saberi, U. V. Vazirani, V. V. Vazirani. 2007. Adwords and generalized online matching. *Journal of ACM* **54**(5).

Muthukrishnan, S., M. Pal, Z. Svitkina. 2010. Stochastic models for budget optimization in search-based advertising. *Algorithmica* **58**(4) 1022–1044.

Panconesi, A., A. Srinivasan. 1997. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.* **26**(2) 350–368.

Rusmevichientong, P., D. Williamson. 2006. An adaptive algorithm for selecting profitable keywords for search-based advertising services. *ACM Conference on Electronic Commerce (EC)*. 260–269.

Singh, M., N. K. Vishnoi. 2014. Entropy, optimization and counting. *Symposium on Theory of Computing STOC*. 50–59.

Srinivasan, A. 2001. Distributions on level-sets with applications to approximation algorithms. *Foundations of Computer Science (FOCS)*. 588–597.

Varian, H. 2007. Position auctions. *Int. J. Industrial Opt.* **25**(6) 1163–1178.

von Neumann, J. 1953. A certain zero-sum two-person game equivalent to an optimal assignment problem. *Ann. Math. Studies* (28) 5–12.

Vondrak, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. *ACM Symposium on Theory of Computing (STOC)*. 67–74.

Zhou, Y., D. Chakrabarty, R. M. Lukose. 2008. Budget constrained bidding in keyword auctions and online knapsack problems. *Workshop on Internet & Network Economics*. 566–576.

Appendix A: Reduction From Set-packing

Given an instance of set packing with sets $S_1, S_2, \ldots, S_r$, the goal is to pick the maximum number of disjoint sets. The reduction to our problem goes as follows. Let the set of possible bids be $\{0, 1\}$ and add a keyword corresponding to each set S_i. For every element e, add a 1-dimensional budget constraint to ensure that only one keyword corresponding to the sets containing e may be "selected": the cost is always zero for bid 0, but is 1 for bid 1 in the corresponding dimension. (All budget constraints are normalized in that the upper bounds are 1.) The value of a keyword is zero for bid 0, but one for bid 1. Therefore, the number of keywords using bid 1 is equal to the number of "selected" cells. Now, the maximum value with respect to the budget constraints is equivalent to the maximum number of disjoint sets among $S_1, S_2, \ldots, S_r$.

Appendix B: PTAS for The Case of Constant k

In this section, we show how to solve the bid clustering problem in the case of a constant k almost optimally. In particular, we provide a polynomial-time approximation scheme (PTAS) with running time exponential in k and polynomial in the size of input. The algorithm starts by enumerating *all* possible bid values for the k clusters. More specifically, we "guess" a set $\{b_1, b_2, \ldots, b_k\} \subseteq B$ of bids to be used in the solution. For each set of cluster bids, then, we employ the dynamic programming method to find the best solution. The output is going to be the best solution among all those found through dynamic programming.

Discretization of cost. Before stating the dynamic program, we discretize the the bid-to-cost functions. Consider a (not necessarily constant) precision parameter $\eta > 0$. We round each component of $\overrightarrow{\mathsf{cost}}_w(b)$ down to the next integral multiple of $\eta L/n$, where L denotes the resource

limit in the corresponding budget constraint. In particular, given the multidimensional resource limit vector $\vec{L}$, the rounded $\mathsf{cost}_w^{(q)}(b)$, for each dimension $1 \le q \le r$, bid $b \in B$ and keyword $w \in \mathcal{K}$, will be the largest integer multiple of $\eta L^{(q)}/n$ that does not exceed the original, unrounded $\mathsf{cost}_w^{(q)}(b)$. Since costs are only rounded down, the optimal solution remains feasible. The solution that the dynamic program finds, however, may not be feasible as its (original unrounded) cost might exceed $\vec{L}$. This violation, though, cannot be more than $n \cdot \eta\vec{L}/n$. Therefore, the cost of the solution is at most $(1+\eta)\vec{L}$. Later on we show how to eliminate this violation in cost. As mentioned earlier, we assume without loss of generality that $\vec{L} = \vec{1}$, since budget constraints can be normalized. Let $\mathcal{L} = \left\{\vec{c} = (c^{(1)}, c^{(2)}, \ldots, c^{(r)}) : \vec{0} \le \vec{c} \le \vec{1} \text{ and } c^{(q)} \cdot n/\eta \in \mathbb{N}\right\}$ be the set of all valid cost vectors (after discretization).

B.1. PTAS with small violation of budget

At this point, we have a set of only k bids that we can use. We employ a dynamic programming algorithm to fill the two-dimensional table $\texttt{profit}[i, \vec{j}]$ for $0 \le i \le n$ and $\vec{j} \in \mathcal{L}$; note that the table could be thought of as being $(r+1)$-dimensional since the second index (whose range is $\mathcal{L}$) is itself a (discrete) r-dimensional vector. The cell $\texttt{profit}[i, \vec{j}]$ is intended to hold the value of the best solution using the first i keywords (fix an arbitrary ordering $w_1, w_2, \ldots, w_n$ at the beginning) consuming a budget of exactly $\vec{j}$ (in the discretized cost space). We first initialize $\texttt{profit}[i, \vec{0}] = 0$ for any $0 \le i \le n$ and $\texttt{profit}[0, \vec{j}] = -\infty$ for any $\vec{j} \in \mathcal{L} \setminus \vec{0}$. Then, we consider each $1 \le i \le n$ in increasing order, and go over every $b \in \{b_1, \ldots, b_k\}$ to compute

$$\texttt{profit}[i, \vec{j}] = \max_{\substack{b \in \{b_1, \ldots, b_k\} \\ \overrightarrow{\mathsf{cost}}_{w_i}(b) \le \vec{j}}} \left\{ \mathsf{value}_{w_i}(b) + \texttt{profit}[i-1, \vec{j} - \overrightarrow{\mathsf{cost}}_{w_i}(b)] \right\}.$$

We assume that a max expression with empty domain evaluates to $-\infty$.

Recall that Lemma 1 restricted the set of candidate bids to consider at the expense of at most $\delta\mathsf{OPT}$ loss in the objective. Moreover, the cost landscapes were discretized with respect to an error parameter η. A statement regarding the correctness for Algorithm 2 appears next. The proof is straightforward and is given in Appendix C.

LEMMA 6. *Fix set of bids $B' \in B^k$. Then for all $0 \le i \le n$ and $\vec{j} \in \mathcal{L}$, the dynamic programming cell $\texttt{profit}[i, \vec{j}]$ computed by Line 12 holds the intended value: it is within a factor $1-\delta$ of the maximum value that can be achieved from the first i keywords using a (discretized) resource consumption vector $\vec{j}$ with the bid set B'.*

The size of the DP table is $O((n+1)^{1+r}/\eta^r)$ since the discretization of $\vec{L} = \vec{1}$ gives a range of size $1 + n/\eta$ for each component.

ALGORITHM 2: `BidClustering` for small k with small budget violation

Input: $\mathcal{K}, k, \eta, \delta, \vec{L} = \vec{1}, \mathsf{value}_w$ and $\overrightarrow{\mathsf{cost}}_w$ for $w \in \mathcal{K}$

Output: value of best bidding strategy

Find the set of important bids B as described in Lemma 1 with precision parameter δ.
Discretize cost functions $\overrightarrow{\mathsf{cost}}_w$ with error parameter η
best $\leftarrow -\infty$
for $B' = (b_1, b_2, \ldots, b_k) \in B^k$ **do**
  for $i \in [n]$ **do**
    Let $\mathtt{profit}[i, \vec{0}] \leftarrow 0$
  for $\vec{j} \in \mathcal{L} \setminus \vec{0}$ **do**
    Let $\mathtt{profit}[0, \vec{j}] \leftarrow -\infty$
  for $i \leftarrow 1$ **to** n **do**
    for $\vec{j} \in \mathcal{L}$ **do**
      $$\mathtt{profit}[i, \vec{j}] \leftarrow \max_{\substack{b \in B' \\ \overrightarrow{\mathsf{cost}}_{w_i}(b) \le \vec{j}}} \left\{ \mathsf{value}_{w_i}(b) + \mathtt{profit}[i-1, \vec{j} - \overrightarrow{\mathsf{cost}}_{w_i}(b)] \right\}$$
  Let cur-max $\leftarrow \max_{\vec{j} \in \mathcal{L}} \mathtt{profit}[n, \vec{j}]$
  if best $<$ cur-max **then**
    Let best $\leftarrow$ cur-max
return best

COROLLARY 2. *Algorithm 2 finds a solution whose cost is at most* $(1+\eta)\vec{L}$*, and whose value is at least* $(1-\delta)\mathsf{OPT}$*. The algorithm runs in polynomial time (in terms of* n *as well as* $1/\eta$ *and* $1/\delta$*) if* k *and* r *are constants.*

In the following section we show how to avoid the $1+\eta$ violation of budgets.

B.2. PTAS without any violation of budget

In the following, we call a (keyword, bid) pair an "item." Our goal is to pick a set of "compatible" items that satisfy certain resource constraints and maximize the value. A set of items are called compatible if they consist of distinct keywords. Recall that we are given a target constant precision parameter $\epsilon > 0$, and we want to obtain an approximation factor of $1-\epsilon$.

For each resource q, pick (i.e., try all possibilities for) the set G_q of $\Delta = 3r/\epsilon$ items that have the highest usage of resource q in the optimal solution. (Notice that there are polynomially many choices to make here since r and ϵ are constants.) Let $G = \cup_q G_q$ be the union of all these items. Clearly $3r/\epsilon = \Delta \le |G| \le r\Delta = 3r^2/\epsilon$.

For each resource q, let $s^{(q)}$ be the Δ-th largest usage of resource q among items in G, and let $g^{(q)}$ be the total usage of resource q in G. We then denote by $\tilde{L}^{(q)} = L^{(q)} - g^{(q)}$ the "residual" budget for resource q (after all items in G are accounted for). We define $\vec{\tilde{L}}$ and $\vec{g}$ similarly to $\vec{L}$.

Roughly speaking, we plan to run Algorithm 2, with parameter $\eta = \epsilon/12r$, on the "residual instance" using the residual budget. The residual instance does not include keywords whose bids are already fixed by G. However, in order to cope with the potential budget violation of the resulting solution, we first modify the instance slightly. The modification produces a new budget limit vector $\vec{\hat{L}}$ that does not exceed $\vec{\tilde{L}}$ as well as a (possibly empty) subset $G' \subseteq G$ of items. The solution returned by Algorithm 2 is then augmented by $G \setminus G'$. We define $\vec{\hat{L}}$ and G' carefully such that (1) the difference $\vec{\tilde{L}} - \vec{\hat{L}}$ and the resource consumption of the excluded items G' can together offset the potential violation from Algorithm 2, and (2) the value lost by excluding items G' is pretty small. We consider two cases for each resource q.

1. If $s^{(q)} < \eta \tilde{L}^{(q)}$, we set $\hat{L}^{(q)} = (1-\eta)\tilde{L}^{(q)}$.
2. If $s^{(q)} \geq \eta \tilde{L}^{(q)}$, we set $\hat{L}^{(q)} = \tilde{L}^{(q)}$ and add to G' the least-value item among the top Δ consumers of resource q in G. We lose no more than than OPT/Δ in value, hence a total of $\frac{r}{\Delta}\mathsf{OPT}$ when summed over all resources.

The discussion above leads to the following proof for Theorem 1.

Proof of Theorem 1. We show that Algorithm 3 is a PTAS for the problem. First, note that the algorithm runs in polynomial time. There are two enumeration steps: one is the same as that in Algorithm 2 and has $|B|^k$ options (where $|B| = O(n \log n/\epsilon)$); the second one picks at most $r\Delta = 3r^2/\epsilon$ (keyword, bid) pairs, hence $O((nk)^{r\Delta})$ choices. The rest of the algorithm is polynomial-time since Algorithm 2 was proved to be so, noting that $\eta = \epsilon/12r$ is a constant.

To establish correctness, we first argue that the final solution satisfies the budget constraints. Notice that we consider several possibilities for B' and G, and take the best solution obtained from each. Consider a specific choice of B' and G. Corollary 2 guarantees that the solution S returned on Line 15 costs no more than $(1+\eta)\vec{\hat{L}}$. To show that $S \cup G \setminus G'$ costs at most $\vec{L}$, we consider each resource q separately. If q falls in case one above, i.e., $s^{(q)} < \eta\tilde{L}^{(q)}$, we have $(1+\eta)\hat{L}^{(q)} = (1-\eta^2)\tilde{L}^{(q)}$, hence adding items G (let alone $G \setminus G'$) to S will not violate budget limit of resource q, because the usage of G is $g^{(q)} = L^{(q)} - \tilde{L}^{(q)}$. If, one the other hand, q falls in case two above, i.e., $s^{(q)} \geq \eta\tilde{L}^{(q)}$, we have $(1+\eta)\hat{L}^{(q)} \leq (1+\eta)\tilde{L}^{(q)}$. Adding G to this solution may create a budget violation of at most $\eta\tilde{L}^{(q)} \leq s^{(q)}$, which is canceled out by removing items G'.

Finally we prove the desired bound on the value of the final solution and finish the correctness proof. It suffices to do this for only one of the possibilities we consider for B' and G. Lemma 1 states that the best solution with bids from B' has value at least $(1-\delta)\mathsf{OPT}$. Take such a solution OPT_1. (Recall that, when it is clear from context, we may use $G, G', \mathsf{OPT}_1, \ldots$ to denote the solution as well as its value.) Focus on the set B' of (discretized) bids used in OPT_1. Moreover, assume that G contains the top Δ consumers of each resource among the items in OPT. This is possible since $|G| \geq r\Delta$. We let $\mathsf{OPT}_2 = \mathsf{OPT}_1 \setminus G$, and write $\mathsf{OPT}_2 + G = \mathsf{OPT}_1$ for the values.

ALGORITHM 3: `BidClustering` for small k with no budget violation

Input: $\mathcal{K}, k, \epsilon, \vec{L} = \vec{1}, \mathsf{value}_w$ and $\overrightarrow{\mathsf{cost}}_w$ for $w \in \mathcal{K}$

Output: value of best bidding strategy

Find the set of important bids B as described in Lemma 1, with $\delta = \epsilon/6$.
best $\leftarrow -\infty$
for $B' = (b_1, b_2, \ldots, b_k) \in B^k$ **do**
  for each set G of (w, b) pairs with distinct w's such that $w \in \mathcal{K}, b \in B', |G| \leq 3r^2/\epsilon$ **do**
    if G satisfies all resource constraints **then**
      Let $\vec{\tilde{L}}$ be residual budget after bidding according to G
      Let $G' \leftarrow \emptyset$
      for each resource q **do**
        Let $s^{(q)} \leftarrow$ the $3r/\epsilon$-th largest usage of resource q among those in G
        if $s^{(q)} \geq \frac{\epsilon}{12r}\tilde{L}^{(q)}$ **then**
          Add to G' the least-profit item in G among the top $3r/\epsilon$ consumers of resource q
          Let $\hat{L}^{(q)} \leftarrow \tilde{L}^{(q)}$
        else
          Let $\hat{L}^{(q)} \leftarrow (1 - \frac{\epsilon}{12r})\tilde{L}^{(q)}$
      Run DP of Algorithm 2 with budget $\vec{\hat{L}}$ excluding the keywords used in G and with cost violation parameter $\eta = \frac{\epsilon}{12r}$
      Let cur-max $\leftarrow$ value of DP plus that of $G \setminus G'$
      if best $<$ cur-max **then**
        Let best $\leftarrow$ cur-max
return best

We construct another solution $\mathsf{OPT}_3 \subseteq \mathsf{OPT}_2$ that satisfies the modified residual budget constraints $\vec{\hat{L}}$ and is almost as good as OPT_2 in terms of value: i.e., $\mathsf{OPT}_3 \geq \mathsf{OPT}_2 - 4r\eta\mathsf{OPT}$. This is done by setting $\mathsf{OPT}_3 = \mathsf{OPT}_2$ and then carefully removing items from OPT_3 until the said conditions hold. As OPT_2 by definition satisfies $\vec{\tilde{L}}$, we only need to modify it because of resources q where $\hat{L}^{(q)} = (1-\eta)\tilde{L}^{(q)}$. Consider one such resource q whose modified residual budget constraint is violated by OPT_3. By definition we have $s^{(q)} < \eta\tilde{L}^{(q)}$. In other words, every item in OPT_2 consumes less than $\eta\tilde{L}^{(q)}$. Greedily pack them in pieces with resource consumption in $[\eta\tilde{L}^{(q)}, 2\eta\tilde{L}^{(q)})$. (Possibly ignore one piece of resource consumption less than $\eta\tilde{L}^{(q)}$ at the end.) The assumption on budget violation implies that OPT_3 uses at least $(1-\eta)\tilde{L}^{(q)}$ units of resource q, hence we get at least $\lfloor \frac{1-\eta}{2\eta} \rfloor > \frac{1-3\eta}{2\eta} \geq \frac{1}{4\eta}$ pieces out of the greedy packing procedure. (The last inequality follows from $3\eta \leq \frac{1}{2}$.) Therefore, removing the smallest-value piece from OPT_3 cannot reduce its value by more than $4\eta\mathsf{OPT}_3 \leq 4\eta\mathsf{OPT}$, however, it ensures that OPT_3 satisfies the limit on resource q. Once

all resources are treated in this way, we obtain a solution OPT_3 for the modified residual budget constraints $\vec{\hat{L}}$ whose value is $\mathsf{OPT}_3 \geq \mathsf{OPT}_2 - 4r\eta\mathsf{OPT}$. By Corollary 2, Line 15 finds a solution OPT_4 whose value is at least $\mathsf{OPT}_4 \geq (1-\delta)\mathsf{OPT}_3$.

We already showed that $G' \leq \frac{r}{\Delta}\mathsf{OPT}$. Thus at the end of Algorithm 3 we obtain a solution of value at least

$$\begin{aligned}
\mathsf{OPT}_4 + G - G' &\geq (1-\delta)\mathsf{OPT}_3 + G - \frac{r}{\Delta}\mathsf{OPT} \\
&\geq (1-\delta)(\mathsf{OPT}_2 - 4r\eta\mathsf{OPT}) + G - \frac{r}{\Delta}\mathsf{OPT} \\
&= (1-\delta)\mathsf{OPT}_2 + G - (4r\eta + \frac{r}{\Delta})\mathsf{OPT} \\
&\geq \mathsf{OPT}_2 + G - (\delta + 4r\eta + \frac{r}{\Delta})\mathsf{OPT} \\
&= \mathsf{OPT}_1 - (\delta + 4r\eta + \frac{r}{\Delta})\mathsf{OPT} \\
&\geq (1-\delta)\mathsf{OPT} - (\delta + 4r\eta + \frac{r}{\Delta})\mathsf{OPT} \\
&= \mathsf{OPT} - (2\delta + 4r\eta + \frac{r}{\Delta})\mathsf{OPT}.
\end{aligned}$$

Recalling $\delta = \epsilon/6$, $\eta = \epsilon/12r$ and $\Delta = 3r/\epsilon$ finishes the proof. □

Similar to other dynamic programing approaches, it is straightforward that maintaining an auxiliary table denoting how each table entry was updated can be used to reconstruct the solution.

The reader may observe that the only exponential part of the running time that cannot be readily parallelized is the inner loop of the dynamic program. The other enumeration steps produce independent subproblems, hence can be run on different machines, provided that we have many thereof.

Appendix C: Omitted Proofs For Lemmas

Proof of Lemma 1. Without loss of generality, assume that there exists a large bid value b^∞ that is infeasible for all keywords—i.e., $\overrightarrow{\mathsf{cost}}_w(b^\infty) \not\leq \vec{1}$ for any keyword w (recall that $\vec{L} = \vec{1}$). Let $b_w^\infty \leq b^\infty$ be the minimum such bid value for a keyword w; this is well-defined due to left-continuity of cost functions and the above assumption. No advertiser can bid b_w^∞ or higher on w, thus, his bid for this keyword falls in $[0, b_w^\infty)$.

Let w' be the most valuable keyword if the advertiser is to bid on only one:

$$w' = \arg\max_{w \in \mathcal{K}} \sup_{b:\overrightarrow{\mathsf{cost}}_w(b) \leq \vec{1}} \mathsf{value}_w(b).$$

Then let $U \leq \mathsf{value}_{w'}(b_w^\infty)$ be a lower bound on the advertiser's value. We assume that, if the advertiser's value from a keyword is less than $\zeta U/3n$, it is indeed zero. Notice that the total contribution we lose is less than $\zeta U/3 \leq \zeta\,\mathsf{OPT}/3$.

Focus on one keyword w. Let $U_w = \sup_{b<b_w^\infty} \mathsf{value}_w(b)$ be the maximum value we can derive from keyword w. The advertiser's value from w, if not zero, falls in the range $[\zeta U/3n, U_w]$. Build the set B_w of important bids for w as follows. If $U_w < \zeta U/3n$, we let $B_w = \emptyset$. Otherwise, let $b_{w,0}$ be the smallest b such that $\mathsf{value}_w(b) \geq \zeta U/3n$; this is well-defined due to left-continuity of value_w. For each $i \geq 0$, if possible, let $b_{w,i+1}$ be the smallest $b < b_w^\infty$ such that $\mathsf{value}_w(b) \geq (1+\zeta/3)\mathsf{value}_w(b_{w,i})$. Add all $b_{w,i}$ values to B_w. The definition guarantees that for any possible bid b for w (that generates a value in $[\zeta U/3n, \mathsf{value}_w(b_w^\infty)]$), there exists $b' \in B_w$ such that $b' \leq b$ and $\mathsf{value}(b') \geq \mathsf{value}(b)/(1+\zeta/3)$. Therefore, replacing b by b' in the optimal solution does not increase the cost, and does not change the value of keyword w by more than a $1/(1+\zeta/3)$ factor.

Letting $B = \{0\} \cup \bigcup_{w\in\mathcal{K}} B_w$, we can replace each bid in the optimal solution by one in B leading to no cost increase and preserving at least $1-\zeta$ of the original value; more precisely, it guarantees $(1-\zeta/3)\mathsf{OPT}/(1+\zeta/3) \geq (1-\zeta/3)^2\mathsf{OPT}$ where the $1-\zeta/3$ factor is the result of losing a value of less than $\frac{\zeta}{3n}U$ from each keyword, and the $1/(1+\zeta/3)$ factor comes from switching to bids in B.

It only remains to bound the size of B. Notice that the $\mathsf{value}_w(b_{w,i})$ increases exponentially for each keyword w. Since $\mathsf{value}_w(b_w^\infty) \leq U$ for w, we get $|B_w| \leq 1 + \log_{1+\zeta/3}(3n/\zeta)$, hence an upper bound of $O(n\zeta^{-1}\log n)$ on the size of B. □

Proof of Claim 1. If the "if condition" in Line 6 holds, then

$$\begin{aligned}\sum_{b=n_{j-1}}^{n_j} \pi_b &= \frac{1}{1-\lambda}\tilde{f}_{j,n_{j-1}} + \sum_{b=n_{j-1}+1}^{n_j} \frac{1-\lambda-\mu}{(1-\lambda)(1-\mu)}\tilde{f}_{j,b} \\ &= \frac{1}{1-\lambda}\mu + \frac{1-\lambda-\mu}{(1-\lambda)(1-\mu)}(1-\mu) \\ &= 1,\end{aligned}$$

where the last equality uses the fact that due to Eq. (15) we have $\sum_{b=n_{j-1}}^{n} \tilde{f}_{j,b} = 1$. On the other hand, if the "if condition" does not hold, then

$$\begin{aligned}\sum_{b=n_{j-1}}^{n_j} \pi_b &= 0 + \sum_{b=n_{j-1}+1}^{n_j} \frac{1}{1-\mu}\tilde{f}_{j,b} \\ &= \frac{1}{1-\mu}(1-\mu) = 1.\end{aligned}$$

Therefore, in both cases $\sum_{b=n_{j-1}}^{n_j} \pi_b = 1$ when the algorithm enters Line 12. Also, clearly all π_b's are non-negative. Thus, the distribution imposed by $(\pi_b : n_{j-1} \leq b \leq n_j)$ is proper. □

Proof of Lemma 2. We prove a stronger result. We show that the probability that our rounding chooses a bid b in the j-th iteration of the "for loop" of Line 4 is $\tilde{f}_{j,b}$, i.e., the rounding is margin-preserving not only on the vertices, but also on each edge. Note that this is enough for the proof of

the lemma, because for each non-border bid b so that $n_{j-1} < b < n_j$ we have $y_b = \tilde{f}_{j,b}$ and for any border bid $b = n_j$ we have $y_b = \tilde{f}_{j,b} + \tilde{f}_{j+1,b}$. (See Eq. 16.)

We denote the event that a bid b is matched to j with "$b \leftrightarrow j$". We show that $\mathbf{Pr}[b \leftrightarrow j] = \tilde{f}_{j,b}$ for any $n_{j-1} \le b \le n_j$. (We emphasize that due to the construction of our rounding, $\mathbf{Pr}[b \leftrightarrow j] = \tilde{f}_{j,b} = 0$ for all $b < n_{j-1}$ or $b > n_j$.) The proof is buy induction on j. The basis of the induction is straightforward by noting that n_0 is defined to be 1, Y_1 is 0 in the beginning of the algorithm, and $\lambda = 0$ in the first run of the loop. Hence, by the formulas of Line 7 and 8, for all $n_0 = 1 \le b \le n_1$ we have $\mathbf{Pr}[b \leftrightarrow j] = \tilde{f}_{j,b}$, which proves the induction basis.

For the inductive step, fix $j > 1$. Note that the "if condition" of Line 6 in the beginning of the j-th iteration of the "for loop" is equivalent with the event "$n_{j-1} \not\leftrightarrow j-1$". However, by the induction hypothesis, $\mathbf{Pr}[n_{j-1} \leftrightarrow j-1] = \tilde{f}_{j-1,n_{j-1}} = \lambda$ for the λ defined in the j-th iteration. Hence, for $b = n_{j-1}$ we have

$$\begin{aligned}
\mathbf{Pr}[b \leftrightarrow j] &= \mathbf{Pr}[b \leftrightarrow j \mid b \not\leftrightarrow j-1] \cdot \mathbf{Pr}[b \not\leftrightarrow j-1] + \mathbf{Pr}[b \leftrightarrow j \mid b \leftrightarrow j-1] \cdot \mathbf{Pr}[b \leftrightarrow j-1] \\
&= \frac{1}{1-\lambda} \tilde{f}_{j,b} \cdot (1-\lambda) + 0 \cdot \lambda \\
&= \tilde{f}_{j,b}.
\end{aligned}$$

Also, for any given b so that $n_{j-1} < b \le n_j$ we have

$$\begin{aligned}
\mathbf{Pr}[b \leftrightarrow j] &= \mathbf{Pr}[b \leftrightarrow j \mid b \not\leftrightarrow j-1] \cdot \mathbf{Pr}[b \not\leftrightarrow j-1] + \mathbf{Pr}[b \leftrightarrow j \mid b \leftrightarrow j-1] \cdot \mathbf{Pr}[b \leftrightarrow j-1] \\
&= \frac{1-\lambda-\mu}{(1-\lambda)(1-\mu)} \tilde{f}_{j,b} \cdot (1-\lambda) + \frac{1}{1-\mu} \tilde{f}_{j,b} \cdot \lambda \\
&= \tilde{f}_{j,b}.
\end{aligned}$$

Hence, for any j, b we have $\mathbf{Pr}[b \leftrightarrow j] = \tilde{f}_{j,b}$ which, as discussed before, results in the proof for the rounding to be margin-preserving. □

Proof of Lemma 3. We first prove the following claim, which formalizes the fact that our rounding method for the y_b's (i.e., Stage i of the algorithm) is equivalent to the Bayesian rounding of Asadpour and Saberi (2010). We note that the characterization of the conditional expectation in the following claim is the same as the update rule (3) of Algorithm 2 in Asadpour and Saberi (2010). We use the same notation of $b \leftrightarrow j$ as in the proof of Lemma 2.

Claim 5 **(An equivalent update for the edges.)**. *For given bids b, b' and $j < j'$ such that $n_{j-1} \le b \le n_j$, and $n_{j'-1} \le b' \le n_{j'}$ for some $j' > j$, if $e_0, e_1, \cdots, e_l$ are the edges of the unique path from $e_0 = (j', b')$ to $e_l = (j, b)$, then we have* $\mathbf{Pr}[b' \leftrightarrow j' \mid b \leftrightarrow j] = \left(1 + (-1)^l \prod_{i=1}^{l-1} \frac{\tilde{f}_{e_i}}{1-\tilde{f}_{e_i}}\right) \tilde{f}_{j',b'}$.

Proof of Claim 5. We consider two cases; one where $b < n_j$ and the other where $b = n_j$. We only present the proof for the case $b < n_j$ as the other case follows from the same analysis. The proof is by induction on j'. Consider $j' = j+1$. The rounding algorithm in the $j+1$-st iteration of the "for loop" falls into the category of $Y_{n_j} = 0$ category (i.e., $n_j \not\leftrightarrow j$, since we are conditioning on $b \leftrightarrow j$ and we have assumed $b < n_j$). Therefore, $\mathbf{Pr}[n_j \leftrightarrow j+1 \mid b \leftrightarrow j] = \tilde{f}_{j+1,n_j}/(1-\tilde{f}_{j,n_j})$ according to our algorithm. However, the path from the edge $e_0 = (j+1, n_j)$ to $e_2 = (j,b)$ is e_0, e_1, e_2 where $e_1 = (j, n_j)$ and therefore, the claim follows for $b' = n_j$. For $n_j < b' \le n_{j+1}$, according to the algorithm we have

$$\begin{aligned}\mathbf{Pr}[b' \leftrightarrow j+1 \mid b \leftrightarrow j] &= \frac{1-\tilde{f}_{j,n_j}-\tilde{f}_{j+1,n_j}}{(1-\tilde{f}_{j,n_j})(1-\tilde{f}_{j+1,n_j})}\tilde{f}_{j+1,b'} \\ &= \left(1-\frac{\tilde{f}_{j,n_j}\tilde{f}_{j+1,n_j}}{(1-\tilde{f}_{j,n_j})(1-\tilde{f}_{j+1,n_j})}\right)\tilde{f}_{j+1,b'},\end{aligned}$$

which coincides with the statement of the lemma, for the path $(j+1,b'),(j+1,n_j),(j,n_j),(j,b)$. This establishes the basis of the induction.

Consider $j' > j+1$ and $n_{j'-1} \le b' \le n_j$. The path from (j',b') to (j,b) passes through $(j'-1, n_{j'-1})$. For $n_{j'-1} \le b' \le n_j$ we have

$$\begin{aligned}\mathbf{Pr}[b' \leftrightarrow j' \mid b \leftrightarrow j] = \mathbf{Pr}[b' \leftrightarrow j' \mid n_{j'-1} \leftrightarrow j'-1 \wedge b \leftrightarrow j]\,\mathbf{Pr}[n_{j'-1} \leftrightarrow j'-1 \mid b \leftrightarrow j] \\ +\mathbf{Pr}[b' \leftrightarrow j' \mid n_{j'-1} \not\leftrightarrow j'-1 \wedge b \leftrightarrow j]\,\mathbf{Pr}[n_{j'-1} \not\leftrightarrow j'-1 \mid b \leftrightarrow j].\end{aligned}$$

However, note that our rounding is memoryless. In other words, the condition "$n_{j'-1} \leftrightarrow j'-1$" (or $n_{j'-1} \not\leftrightarrow j'-1$) is the only factor important for our rounding algorithm in determining what happens to $b \ge n_{j'}$. Therefore,

$$\begin{aligned}\mathbf{Pr}[b' \leftrightarrow j' \mid b \leftrightarrow j] = \mathbf{Pr}[b' \leftrightarrow j' \mid n_{j'-1} \leftrightarrow j'-1]\,\mathbf{Pr}[n_{j'-1} \leftrightarrow j'-1 \mid b \leftrightarrow j] \\ +\mathbf{Pr}[b' \leftrightarrow j' \mid n_{j'-1} \not\leftrightarrow j'-1]\,\mathbf{Pr}[n_{j'-1} \not\leftrightarrow j'-1 \mid b \leftrightarrow j]. \qquad (28)\end{aligned}$$

Note that the condition $n_{j'-1} \not\leftrightarrow j'-1$ is equivalent to having $Y_{n_{j'-1}} = 0$ in the "if condition" in the j'-th iteration of the for loop. Let $e_0 = (n_{j'-1}, j'-1), e_1, \cdots, e_l = (j,b)$ be the path between $(n_{j'-1}, j'-1)$ and (j,b). By the induction hypothesis we have

$$\mathbf{Pr}[n_{j'-1} \leftrightarrow j'-1 \mid b \leftrightarrow j] = \left(1+(-1)^l \prod_{i=1}^{l-1} \frac{\tilde{f}_{e_i}}{1-\tilde{f}_{e_i}}\right)\tilde{f}_{j'-1,n_{j'-1}}.$$

Also, due to the choice of the rounding in the j'-th iteration of the loop we have

$$\mathbf{Pr}[b' \leftrightarrow j' \mid n_{j'-1} \leftrightarrow j'-1] = \begin{cases} 0 & \text{if } b' = n_{j'-1}, \\ \dfrac{1}{1-\tilde{f}_{j',n_{j'-1}}}\tilde{f}_{j',b'} & \text{if } n_{j'-1} < b' \le n'_j, \quad \text{and} \end{cases}$$

$$\mathbf{Pr}\left[b' \leftrightarrow j' \mid n_{j'-1} \not\leftrightarrow j'-1\right] = \begin{cases} \dfrac{1}{1-\tilde{f}_{j'-1,n_{j'-1}}}\tilde{f}_{j',b'} & \text{if } b' = n_{j'-1}, \\ \dfrac{1-\tilde{f}_{j'-1,n_{j'-1}}-\tilde{f}_{j',n_{j'-1}}}{(1-\tilde{f}_{j'-1,n_{j'-1}})(1-\tilde{f}_{j',n_{j'-1}})}\tilde{f}_{j',b'} & \text{if } n_{j'-1} < b' \le n'_j. \end{cases}$$

By plugging these values into Eq. (28) and straightforward algebra we conclude that

$$\mathbf{Pr}\left[b' \leftrightarrow j' \mid b \leftrightarrow j\right] = \begin{cases} \left(1+(-1)^{l+1}\dfrac{\tilde{f}_{j'-1,n_{j'-1}}}{1-\tilde{f}_{j'-1,n_{j'-1}}}\prod_{i=1}^{l-1}\dfrac{\tilde{f}_{e_i}}{1-\tilde{f}_{e_i}}\right)\tilde{f}_{j',b'}, & \text{if } b' = n_{j'-1} \text{ and} \\ \left(1+(-1)^{l+2}\dfrac{\tilde{f}_{j'-1,n_{j'-1}}\tilde{f}_{j',n_{j'-1}}}{(1-\tilde{f}_{j'-1,n_{j'-1}})(1-\tilde{f}_{j',n_{j'-1}})}\prod_{i=1}^{l-1}\dfrac{\tilde{f}_{e_i}}{1-\tilde{f}_{e_i}}\right)\tilde{f}_{j',b'} & \text{if } n_{j'-1} < b' \le n'_j. \end{cases}$$

Noting that for $n_{j'-1} < b'$ the unique path connecting (b', j') to (b, j) is $(b', j'), (n_{j'-1}, j'), e_0, e_1, \cdots, e_l$ completes the proof. □

Next, we use the result of Claim 5 to prove that the rounded value of y for any two bids are negatively correlated.

Claim 6. *For any two given bids b and b' we have $\mathbf{Pr}\left[Y_{b'} = 1 \wedge Y_b = 1\right] \le \mathbf{Pr}\left[Y_b = 1\right]\mathbf{Pr}\left[Y_{b'} = 1\right]$ and $\mathbf{Pr}\left[Y_{b'} = 0 \wedge Y_b = 0\right] \le \mathbf{Pr}\left[Y_b = 0\right]\mathbf{Pr}\left[Y_{b'} = 0\right]$.*

Proof of Claim 6. W.l.o.g suppose that $b < b'$. Note that we only need to prove that $\mathbf{Pr}\left[Y_{b'} = 1 \mid Y_b = 1\right] \le \mathbf{Pr}\left[Y_{b'} = 1\right]$, since if it holds then we have

$$\begin{aligned} \mathbf{Pr}\left[Y_{b'=0} \mid Y_b = 0\right] &= 1 - \mathbf{Pr}\left[Y_{b'} = 1 \mid Y_b = 0\right] \\ &= 1 - \mathbf{Pr}\left[Y_b = 0 \mid Y_{b'} = 1\right]\frac{\mathbf{Pr}\left[Y_{b'} = 1\right]}{\mathbf{Pr}\left[Y_b = 0\right]} \\ &= 1 - (1 - \mathbf{Pr}\left[Y_b = 1 \mid Y_{b'} = 1\right])\frac{\mathbf{Pr}\left[Y_{b'} = 1\right]}{\mathbf{Pr}\left[Y_b = 0\right]} \\ &\le 1 - (1 - \mathbf{Pr}\left[Y_b = 1\right])\frac{\mathbf{Pr}\left[Y_{b'} = 1\right]}{\mathbf{Pr}\left[Y_b = 0\right]} \\ &= 1 - \mathbf{Pr}\left[Y_{b'} = 1\right] \\ &= \mathbf{Pr}\left[Y_{b'} = 0\right]. \end{aligned}$$

Consider two cases; one where b is a non-border bid and the other where it is such. If b is a non-border bid, then there exists j such that $n_{j-1} < b < n_j$. We consider the following cases.

1. *Bid b' is a non-border and $n_{j-1} < b' < n_j$.* Due to the construction of the rounding, at most one of b or b' will be picked. Therefore, we have $\mathbf{Pr}\left[Y_{b'} = 1 \wedge Y_b = 1\right] = 0$, which immediately results in the desired upper bound.
2. *Bid b' is a non-border bid such that for some $j' > j$ we have $n_{j'-1} < b' < n_{j'}$.* Since both bids are non-border, the events "$Y_{b'} = 1$" and "$Y_b = 1$" are equivalent to "$b' \leftrightarrow j'$" and "$b \leftrightarrow j$", respectively. However, according to the result of Claim 5, we have

$$\mathbf{Pr}\left[b' \leftrightarrow j' \mid b \leftrightarrow j\right] = \left(1 - \prod_{l=j}^{j'-1}\frac{\tilde{f}_{l,n_l}\tilde{f}_{l,n_{l+1}}}{(1-\tilde{f}_{l,n_l})(1-\tilde{f}_{l,n_{l+1}})}\right)\tilde{f}_{j',b'} \le \tilde{f}_{j',b'} = \mathbf{Pr}\left[b' \leftrightarrow j'\right].$$

3. *Bid b' is a border bid and $b' = n_j$.* In this case, using Claim 5 we have

$$\begin{aligned}\mathbf{Pr}\left[Y'_b = 1 \mid Y_b = 1\right] &= \mathbf{Pr}\left[b' \leftrightarrow j \mid b \leftrightarrow j\right] + \mathbf{Pr}\left[b' \leftrightarrow j+1 \mid b \leftrightarrow j\right] \\ &= 0 + \left(1 + \frac{\tilde{f}_{j,n_j}}{1-\tilde{f}_{j,n_j}}\right)\tilde{f}_{j+1,b'} \\ &= \frac{\tilde{f}_{j+1,b'}}{1-\tilde{f}_{j,b'}} \leq \tilde{f}_{j+1,b'} + \tilde{f}_{j,b'} = y_{b'},\end{aligned}$$

where the inequality is a direct result of $\tilde{f}_{j+1,b'} + \tilde{f}_{j,b'} \leq 1$.

4. *Bid b' is a border bid and $b' = n_{j'}$ for $j' > j$.* In this case, using Claim 5 we have

$$\begin{aligned}\mathbf{Pr}\left[Y'_b = 1 \mid Y_b = 1\right] &= \mathbf{Pr}\left[b' \leftrightarrow j' \mid b \leftrightarrow j\right] + \mathbf{Pr}\left[b' \leftrightarrow j'+1 \mid b \leftrightarrow j\right] \\ &= \left(1 - \prod_{l=j}^{j'-1} \frac{\tilde{f}_{l,n_l}\tilde{f}_{l+1,n_l}}{(1-\tilde{f}_{l,n_l})(1-\tilde{f}_{l+1,n_l})}\right)\tilde{f}_{j',b'} \\ &\quad + \left(1 + \frac{\tilde{f}_{j',n_{j'}}}{1-\tilde{f}_{j',n_{j'}}} \prod_{l=j}^{j'-1} \frac{\tilde{f}_{l,n_l}\tilde{f}_{l+1,n_l}}{(1-\tilde{f}_{l,n_l})(1-\tilde{f}_{l+1,n_l})}\right)\tilde{f}_{j'+1,b'} \\ &= \tilde{f}_{j',b'} + \tilde{f}_{j'+1,b'} + \left(\prod_{l=j}^{j'-1} \frac{\tilde{f}_{l,n_l}\tilde{f}_{l+1,n_l}}{(1-\tilde{f}_{l,n_l})(1-\tilde{f}_{l+1,n_l})}\right)\underbrace{\left(\frac{\tilde{f}_{j',b'}\tilde{f}_{j'+1,b'}}{1-\tilde{f}_{j',b'}} - \tilde{f}_{j',b'}\right)}_{\leq 0} \\ &\leq \tilde{f}_{j',b'} + \tilde{f}_{j'+1,b'} = y_{b'},\end{aligned}$$

where the inequality is again a direct result of $\tilde{f}_{j+1,b'} + \tilde{f}_{j,b'} \leq 1$.

Hence, if b is a non-border bid, then for every $b' > b$ we have $\mathbf{Pr}\left[Y_{b'} = 1 \mid Y_b = 1\right] \leq \mathbf{Pr}\left[Y_{b'} = 1\right]$. In order to complete the proof of the claim, we need to consider the case where b is a border bid. Let $b = n_j$ for some j. In that case, we use the fact that $\mathbf{Pr}\left[Y_{b'} = 1 \mid Y_b = 1\right] = (\mathbf{Pr}\left[Y_{b'} = 1 \mid b \leftrightarrow j\right]\tilde{f}_{j,b} + \mathbf{Pr}\left[Y_{b'} = 1 \mid b \leftrightarrow j+1\right]\tilde{f}_{j+1,b})/(\tilde{f}_{j,b} + \tilde{f}_{j+1,b})$ and apply Claim 5 to the terms $\mathbf{Pr}\left[Y_{b'} = 1 \mid b \leftrightarrow j\right]$ and $\mathbf{Pr}\left[Y_{b'} = 1 \mid b \leftrightarrow j+1\right]$. The rest of the proof follows with similar algebraic calculations as in the case of b being a non-border bid. □

In order to complete Lemma 3 we use induction. Consider a set of m bids $b_1 \leq b_2 \leq \cdots \leq b_m$. For $m = 2$ the statement of the lemma is equivalent to that of Claim 6. Consider $m > 2$. We have $\mathbf{Pr}\left[\wedge_{i=1}^{m}(Y_{b_i} = 1)\right] = \mathbf{Pr}\left[\wedge_{i=1}^{m-1}(Y_{b_i} = 1)\right] \cdot \mathbf{Pr}\left[(Y_{b_m} = 1) \mid \wedge_{i=1}^{m-1}(Y_{b_i} = 1)\right]$. Since the rounding is memoryless (as discussed in the proof of Claim 5), we have $\mathbf{Pr}\left[(Y_{b_m} = 1) \mid \wedge_{i=1}^{m-1}(Y_{b_i} = 1)\right] = \mathbf{Pr}\left[(Y_{b_m} = 1) \mid (Y_{b_{m-1}} = 1)\right]$ which is bounded from above by $\mathbf{Pr}\left[Y_{b_m} = 1\right]$, according to Claim 6. Also, $\mathbf{Pr}\left[\wedge_{i=1}^{m-1}(Y_{b_i} = 1)\right] \leq \prod_{i=1}^{m-1}\mathbf{Pr}\left[Y_{b_i} = 1\right]$ using the induction hypothesis. Therefore, $\mathbf{Pr}\left[\wedge_{i=1}^{m}(Y_{b_i} = 1)\right] \leq \prod_{i=1}^{m}\mathbf{Pr}\left[Y_{b_i} = 1\right]$. The bound for $\mathbf{Pr}\left[\wedge_{i=1}^{m}(Y_{b_i} = 0)\right]$ follows similarly. This completes the proof of the lemma. □

Proof of Lemma 5. Recall that

$$\lambda \geq \int_0^1 f(t)dt + \int_1^\infty \frac{1}{t} f(t)dt. \tag{29}$$

To minimize the right-hand side, observing that the multiplier of $f(t)$ is nonincreasing, we should push the mass of $f(t)$ as much as possible to larger values of t. More specifically, let $f^*(t)$ be the density function that minimizes the right-hand side. Without loss of generality, assume that $f^*(t) = 0$ for $t < 1$. Then, for any $t \geq 1$ we should have $\int_t^\infty f^*(t')dt' = G(t)$. Otherwise, moving a total mass of $G(t) - \int_t^\infty f^*(t')dt'$ from $f^*(t)$ for $t' < t$ to $f^*(t)$ reduces the right-hand side of (29), which is a contradiction. Therefore, we have $f^*(t) = -\frac{dG(t)}{dt}$, plugging which into (29) yields the statement of the lemma. □

Proof of Lemma 6. Lemma 1 guarantees that restricting to the set B of bids leads to a loss of at most $1-\delta$ in value. Enumeration of k bids among these ensures we guess the correct values once. It remains to analyze the dynamic program itself.

The argument is via mathematical induction on i. Those cells taking up values at initialization clearly have the correct values. Now consider another cell $\texttt{profit}[i, \vec{j}]$. The advertiser needs to place some bid (from $\{b_1, \ldots, b_k\}$) on keyword w_i. If some bid b is feasible (i.e., within budget), we look up a DP cell previously computed correctly (via induction hypothesis) to find the value of this option. The maximum of all these options gives the best solution for $\texttt{profit}[i, \vec{j}]$. □

Appendix D: An exponential-size reduction to submodular maximization

In this section, we describe a reduction from the bid clustering problem to the submodular maximization problem subject to several knapsack constraints. It will result only in an exponential-time $1 - 1/e$-approximation for the problem.

To derive the reduction, we first define a value function over subsets of an exponential-size ground set—i.e., the domain is doubly exponential—and show that it is submodular. This implies that we can essentially use an approximation algorithm for the submodular maximization problem. However, the current approaches to solving such tasks start with a certain enumeration procedure which in our case would require exponential running time.

Recall that $\mathcal{K}$ is the set of keywords, and that B from Lemma 1 is the set of all important bids we need to consider; we need to pay a factor $1 - \epsilon$ in the objective for this. Define the ground set $\mathcal{G} = \{(b, S) : b \in B, S \subseteq \mathcal{K}\}$. The elements of the ground set are the clusters we can use to construct the clustering (and bidding strategy). An element (b, S) signifies, roughly speaking, a decision to bid b on a set S of keywords.

We next define two functions, namely $\overrightarrow{\mathsf{cost}}$ and value^*, on the subsets of ground set $\mathcal{G}$. Each subset of $\mathcal{G}$, ideally with disjoint clusters of keywords, denotes a bidding strategy. For each $\mathcal{S} \subseteq \mathcal{G}$, let $\overrightarrow{\mathsf{cost}}(\mathcal{S}) = \sum_{(b,S)\in\mathcal{S}} \sum_{w\in S} \overrightarrow{\mathsf{cost}}_w(b)$. If the clusters in $\mathcal{S}$ have disjoint sets of keywords, this function indeed computes the cost of the bidding strategy. Otherwise, we need to have a resolution of $\mathcal{S}$ to a bidding strategy. In this work, we assume that each keyword receives the highest bid among all elements of $\mathcal{S}$: i.e., we bid

$$\mathsf{bid}(w,\mathcal{S}) = \max\left\{0, \max_{\substack{(b,S)\in\mathcal{S}\\ w\in S}} b\right\}$$

on w. In this case, $\overrightarrow{\mathsf{cost}}$ always gives an upper bound on the cost, giving the algorithm an incentive to produce disjoint sets of keywords. Further, we define $\mathsf{value}^*(\mathcal{S}) = \sum_{w\in\mathcal{K}} \mathsf{value}_w(\mathsf{bid}(w,\mathcal{S}))$.

Clearly $\overrightarrow{\mathsf{cost}}$ is an additive function. Let us next demonstrate that value^* is submodular and monotone. Recall that a function $f : 2^A \mapsto \mathbb{R}$ is submodular if and only if $f(B\cup\{x\}) - f(B) \geq f(B'\cup\{x\}) - f(B')$ if $B \subseteq B' \subseteq A - \{x\}$. (This is called the increasing marginal value property.)

LEMMA 7. *The function* value^* *is submodular and monotone.*

Proof of Lemma 7. Monotonicity is clear since $\mathsf{bid}(w,\mathcal{S})$ is monotone (a maximum function) and value_w is nondecreasing. For submodularity consider sets $\mathcal{S},\mathcal{S}' \subseteq \mathcal{G}$ and keyword $(b,S) \in \mathcal{G}$ such that $A \subseteq A' \subseteq \mathcal{G} - (b,S)$. Note that the changes in value^* depend only on value_w terms for $w \in S$. Since value_w functions are nondecreasing, it suffices to establish, for each $w \in S$, that $\mathsf{bid}(w,\mathcal{S}\cup\{(b,S)\}) - \mathsf{bid}(w,\mathcal{S}) \geq \mathsf{bid}(w,\mathcal{S}'\cup\{(b,S)\}) - \mathsf{bid}(w,\mathcal{S}')$. Clearly, each side of the inequality is a nonnegative term. Suppose the right-hand side is positive, otherwise the inequality holds. This implies that the new bid for w comes from the new element (b,S), hence $\mathsf{bid}(w,\mathcal{S}'\cup\{(b,S)\}) = \mathsf{bid}(w,\mathcal{S}\cup\{(b,S)\}) = b$. The inequality then follows from $\mathsf{bid}(w,\mathcal{S}) \leq \mathsf{bid}(w,\mathcal{S}')$. □

To find the best solution $\mathcal{S}$, now we can essentially use an algorithm for submodular maximization with cardinality (i.e., at most k clusters) and (multiple) knapsack constraints ($\overrightarrow{\mathsf{cost}}(\mathcal{S}) \leq \vec{L}$) (Vondrak 2008, Kulik et al. 2011). Resolving the conflicts for keywords appearing in multiple elements does not increase the cost, and does not change the value. This readily gives a bidding strategy that has an approximation ratio of $1 - \frac{1}{e} - \epsilon$.

The problem with the above algorithm is that not only are such algorithms complicated, but they have a running time which depends on the size of the ground set. This makes them difficult to use in practice. In particular, these algorithms start with an enumeration procedure which in our case requires exponential running time.