Concise Bid Optimization Strategies with Multiple Budget Constraints

Arash Asadpour[1], Mohammad Hossein Bateni[2], Kshipra Bhawalkar[2], and Vahab Mirrokni[2]

[1] NYU Stern School of Business
aasadpou@stern.nyu.edu
[2] Google Inc.
{bateni,kshipra,mirrokni}@google.com

Abstract. A major challenge faced by the 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 these 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., predicted value (e.g., number of clicks) and cost per click for any bid—that are typically provided by ad-serving systems, we optimize the value given 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 constant-factor approximation can be obtained via a combination of solution enumeration and dependent LP-rounding techniques.

Finally, we evaluate the performance of our algorithms on real datasets in two regimes with 1- and 3-dimensional budget constraint. In the former case where uniform bidding 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—contrast with the single cluster permitted in uniform bidding. With only three dimensions in the budget constraint (one for total consumption, and another two for enforcing minimal diversity), the gap between the performance of our algorithm and an enhanced version of uniform bidding grows to an average of 5% to 6% (sometimes as high as 9%). Although the details of experiments for the multidimensional budget constraint to the full version of the paper are deferred to the full version of the paper, we report some highlights from the results.

1 Introduction

The Internet has become a major advertising medium, with billions of dollars at stake; according to the recent IAB report [18], Internet advertising revenues in the United States totaled $\$31.7$ billion in 2011 with sponsored search accounting for 46.5% of this revenue. Search engines provide simple ways to quickly set up an advertising campaign,

T.-Y. Liu et al. (Eds.): WINE 2014, LNCS 8877, pp. 263–276, 2014.

track expenses, monitor effectiveness of the campaigns, and tinker with campaign parameters, and this has made it relatively easy even for small advertisers to enter online advertising market. Even with all 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, 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. 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 partition them into a small number of clusters such that the advertiser is going to have the same bid on each cluster. Such concise bidding strategies are inspired by *uniform bidding strategies* that have been shown to achieve relatively good results [11]. 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.

Setting. 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 is based on having advertisers bid on the search query issued by the user, and the search engine run 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 [1, 22, 7, 19, 5, 6, 12], and advertiser's point of view [4, 11, 21, 23, 20, 8, 2]. 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. 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 is charged. Charging can be based on "impressions" (each time the ad being 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, 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] [11] for keywords; i.e., for each keyword w, advertisers get bid landscape functions value_w and $\overrightarrow{\mathsf{cost}}_w$ corresponding to different bids on keyword w.

For ease of presentation, here we mostly focus on the most common case of *cost-per-click* (CPC) charging and consider the bid landscapes for cost and number of clicks, however, unlike most previous work [11, 20], our results directly apply to more general pay-per-impression or pay-per-conversion models along with other value functions, and even to settings with nonconcave value or cost functions.

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.

Multidimensional Budget Constraints. Budget constraints play a major role in setting up an online advertising campaign, both from the auctioneer point of view and from 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, e.g., [13, 14]. 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 spent 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.

As an example, 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. 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

[1] Also referred to as "bid simulator" or "traffic estimator" [15, 16].

capture such general multidimensional budget constraints. We describe the details of these budget constraints in Section 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 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., partitioning 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.

Such concise bidding strategies are already studied in the context of *uniform bidding strategies*, introduced by the seminal paper of Feldman et al. [11]. In uniform bidding strategies, the advertiser bids uniformly on all keywords. Uniform bidding, although naïve at first glance, has been shown to achieve relatively good results, both in theory and practice [11, 20]. In fact, the simplicity of uniform bidding along with its reasonable performance make 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. Although effective in simplistic setting, uniform bidding mainly applies to a specific setting with a single budget constraint and concave cost and value functions. Search engines do give advertisers the ability to bid differently on each keyword. 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.

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.

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 2, and present approximation algorithms for this problem. The problem with super-constant number of budget constraints r does not admit a reasonable approximation algorithm as it is harder than *set packing*. The latter is known not to have any $\omega(r^{1-\epsilon})$ approximation unless $\mathsf{NP} \subseteq \mathsf{ZPP}$ [17]. In this paper, we focus on the problem with a constant number of budget constraints r.

Our main theoretical contribution in this paper is a constant-factor approximation for arbitrary number of clusters k. This constant-factor approximation algorithm is obtained using a dependent LP-rounding technique (performed in three phases) 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 to implement and is linear-time, however, its analysis uses a new technique 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 budge constraints. If a factor $1 + \epsilon$ violation of budget constraints 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 pretty involved and requires careful enumeration and modification of the residual instance.

Finally, we evaluate our algorithms on real data sets and compare their performance with the uniform bidding strategy. Even in a simple setting of maximizing the expected number of clicks subject to one budget constraint (for which uniform bidding is provably good), we show that using a small number of clusters can improve the expected number of clicks by 1% to 4%. We see more improvement with increasing number of clusters when the budget constraint is more tight. We also evaluate our algorithm in data sets with more budget constraints, and notice significant improvement (as high as 20%) 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. In the interest of space, some of the proofs will be deferred to the full version of the paper.

1.1 Related Work

As a central issue in online advertising, optimizing under budget constraints have been studied extensively both from publishers' (or search engines') point of view [19, 5, 6, 12], and from advertisers' point of view [4, 11, 21, 23, 20, 8, 2]. One well-studied problem from publisher's perspective is to deal with online allocation of ads in the presence of budget constraints [19, 5, 10, 9], and another line of research is dedicated to designing efficient mechanisms addressing incentive issues, and respecting budget constraints [6, 12]. More relevant to this paper, the bid optimization with budget constraints has also been studied from advertisers' perspective: either in a repeated auction setting [4], or in the context of broad-match ad auctions [8], or the case of long-term carryover effects [2].

This work is most related to the seminal paper of Feldman et al. [11] 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. [11] and Muthukrishnan et al. [20] 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 the cost and click landscapes follow a specific structure. Besides, those results apply only in the case of one budget constraint and (the proofs) do not easily generalize to settings with multiple budget constraints. Our results however apply to any set of monotone cost and value bid landscape functions (e.g., for the case of maximizing conversions), and more importantly handles multiple budget constraints. In addition, we compare our solution to the best solution with the same number of clusters, however, Feldman et al. compare their solution to the optimum of the knapsack problem with arbitrarily 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. can only get a $1 - 1/e$ approximation ratio. In fact, generalizing the results of Feldman et al. to multiple budget constraints is not possible, and we needed a new solution concept and a set of tools and techniques for this problem.

2 Preliminaries

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

2.1 Formal Problem Definition

In order to optimize their campaigns, we assume that advertisers get "bid landscapes" [11] as an input: For each keyword w, they get (i) a monotone nonnegative function value_w that maps any bid value to the expected value (e.g., number of clicks), and (ii) a nonnegative function $\overrightarrow{\mathsf{cost}}_w$ mapping any bid value to an r-dimensional cost vector incurred by the advertiser. These functions are left-continuous, but they do not necessarily satisfy Lipschitz smoothness conditions. (See [11] for an example of how these 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 the following:

Problem 1. Given are an integer k, a number r of budget constraints (resources), a real vector $\boldsymbol{L} \in \mathbb{R}^r$, a set $\mathcal{K}$ of keywords as well as value and cost landscape functions value_w : bids $\mapsto \mathbb{R}$ and $\overrightarrow{\mathsf{cost}}_w$: bids $\mapsto \mathbb{R}^r$ for each keyword $w \in \mathcal{K}$. Find a partition of $\mathcal{K}$ into k clusters $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_k$, and a set of bids b_i for $i \in [k]$ such that the expected resource consumption of the advertiser is no more than his budget vector L, i.e., $\sum_{i\in[k]} \sum_{w\in\mathcal{K}_i} \overrightarrow{\mathsf{cost}}_w(b_i) \leq \boldsymbol{L}$, and the advertiser's value (e.g., expected number of clicks), i.e., $\sum_{i\in[k]} \sum_{w\in\mathcal{K}_i} \mathsf{value}_w(b_i)$ is (approximately) maximized. We can also consider a *weighted* objective where clicks coming from different keywords may be of varying degrees of importance.

For ease of exposition, we use the shorthands $c_{ib}^{q} = \overrightarrow{\mathsf{cost}}_i^{(q)}(b)$ to refer to each budget constraint limit and $p_{ib} = \mathsf{value}_i(b)$ for values. We also refer to each of the r budget constraints as a resource constraint, i.e., the qth resource (or budget) constraint is the following: $\sum_{i \in [k]} \sum_{w \in \mathcal{K}_i} \overrightarrow{\mathsf{cost}}_i^{(q)}(b) \leq \boldsymbol{L}^q$. 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.[2]

2.2 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 or the independent set problem which is known to be inapproximable within a factor better than $n^{1-\epsilon}$ unless $NP \subset ZPP$ [17]. 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 only consider the case of $r = 1$ [11, 20, 8, 2].

Uniform resource limits. First note that we can assume without loss of generality that the resource usage limit vector $\boldsymbol{L} = \mathbf{1}$. To see this, note that we consider each resource separately. Therefore, if a resource limit q in $\boldsymbol{L}$ is positive, we can scale it to 1 while modifying $\mathsf{cost}_w^{(q)}$ appropriately for all $w \in \mathcal{K}$. 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.

Small set of potential bids. We next show that, 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. The proof of the following lemma can be found in the full version of the paper.

Lemma 1. *Given any $\delta > 0$, we can find (in polynomial time) a set B of size $\mathrm{poly}(n, \frac{1}{\delta})$ such that there exists a $(1-\delta)$-approximate solution (to the problem) all whose bids fall in B.*

In what follows, we consider two input regimes and present algorithms for each. For the general case, we present a constant factor approximation algorithm. If k, the number

---

[2] Note that different resource constraints are not for disjoint sets of keywords; had this been the case, the problem would have been decomposed into separate single-resource special cases. We emphasize, however, that in the example provided, different resource constraints correspond to different markets, each drawing from all keywords, albeit with different coefficients. These techniques can be potentially used to obtain lower bounds on each resource consumption, in effect, capturing diversification objectives better.

of permissible clusters, is a constant, the following theorem guarantees the existence of a polynomial-time approximation scheme (PTAS) based on dynamic programming for the problem. The proof is deferred to the full version of the paper.

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

3 Approximation Algorithm

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

3.1 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.

Consider a variable x_{ib} for each keyword $i \in \mathcal{K}$ and each relevant bid $b \in B$. In the integer linear program, x_{ib} denotes whether the advertiser should place a bid b on keyword i and 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 is a cluster with bid b. To make the LP more concise and readable, we use the shorthands $p_{ib} = \mathsf{value}_i(b)$ and $c^q_{ib} = \overrightarrow{\mathsf{cost}}^{(q)}_i(b)$.

$$\begin{aligned}
\mathsf{max}\ & \textstyle\sum_{i,b} p_{ib} x_{ib} & & (1)\\
\mathsf{s.t.}\ & \textstyle\sum_{i,b} c^q_{ib} x_{ib} \le L^{(q)}\ \forall q \in [r] & & \\
& \textstyle\sum_b y_b \le k \quad \forall i & & \\
& x_{ib} \le y_b \quad \forall i, b & & \\
& y_b \le 1 \quad \forall b & & \\
& \textstyle\sum_b x_{ib} \le 1 \quad \forall i & & (2)\\
& x_{ib}, y_b \ge 0 \quad \forall i, b. & & (3)
\end{aligned}$$

For the case when all $c^q_{ib} \le \epsilon L$, this LP has a small integrality gap, and the LP can be rounded to obtain an approximation ratio of $0.54 - \epsilon$. In fact, this can be done if $c^q_b = \frac{\sum_i c^q_{ib} x_{ib}}{y_b} \le \epsilon L^{(q)}$ for all b and q; see Section 3.2.

To solve the problem, we consider two cases. First, if there is a solution of value at least $\beta\mathsf{opt}$ (β to be determined later) for which the above condition holds, we can add $c^q_b \le \epsilon L^{(q)}$ conditions to the LP as follows.

$$\textstyle\sum_i c^q_{ib} x_{ib} \le \epsilon L^{(q)} y_b \qquad \forall b, q. \tag{4}$$

and then solve and round the LP to get a guarantee of $\beta(0.54 - \epsilon)$.

Second case happens when there is a solution of value at least $(1 - \beta)\mathsf{opt}$ that uses only large-cost clusters, i.e., each cluster of the solution has cost at least $\epsilon L^{(q)}$ for some q. In this case, there will be at most $r\epsilon^{-1}$ clusters in the solution, therefore, we can use Theorem 1 to get an approximation ratio of $1 - \beta - \epsilon$. Letting $\beta = \frac{1}{1.54}$, and outputing the best solution of the two methods yields an approximation ratio of $1 - \beta \approx 0.3506$ for the general case of the problem.

3.2 Rounding the LP

The above LP is rounded in three stages. In the first stage, we modify the fractional solution so that only k nonzero y_b variables remain. This is done carefully without losing more than $1-\epsilon$ factor in the objective value. At this point, we have an LP solution that is *almost* feasible—some constraints (2) may be violated. The second stage addresses this issue by modifying x_{ib} variables so that, at the end, all LP constraints are satisfied. The third 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 $1-\epsilon$.

Stage 1: Rounding y_b Variables. Variables y_b are considered one by one, and are rounded to either zero or one, and the value of the remaining (i.e., as of now unconsidered) variables y_b are adjusted accordingly. In particular, each variable y_b, when considered, is rounded up to one with probability y_b and is rounded down to zero with probability $1-y_b$. The remaining y_b variables are scaled such that their sum stays the same. During the process, x_{ib} variables are also scaled such that each x_{ib}/y_b remains a constant throughout. This process is a martingale, hence we have concentration bounds for $\sum_b c_b^q y_b$ (if all individual contributions are small). We scale down all x_{ib} variables by a factor $1+\epsilon$, so that the cost constraints are satisfied.

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 $l \in [3]$. Notice that, although LP itself is a certain value, each $\mathsf{LP}^{(l)}$ is a random variable. We have $\mathbf{E}\left[\mathsf{LP}^{(1)}\right] \geq \frac{1}{1+\epsilon}\mathsf{LP}$.

Stage 2: Modifying x_{ib} Variables to Get a Feasible Solution. Note that after rounding the y_b variables and scaling the x_{ib} variables appropriately, some constraints (2) may be violated. In particular, for certain keywords i, we may have $\sum_b x_{ib} > 1$. For each such keyword, we scale down all x_{ib} variables at the same rate to obtain $\sum_b x_{ib} = 1$. 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 due to these operations is not too much. More specifically, we prove the following.

Lemma 2. *We have* $\mathbf{E}\left[\mathsf{LP}^{(2)}\right] \geq \lambda \mathbf{E}\left[\mathsf{LP}^{(1)}\right]$ *for a positive constant value of* $\lambda = 0.539968$.

Here, we provide a weaker proof of the above lemma for $\lambda = \frac{1}{4}$. At the end of the proof we discuss the ideas to strengthen our analysis to $\lambda = 0.539968$. The complete proof is deferred to the full version of the paper.

Define random variable X_{ib} to be x_{ib}/y_b with probability y_b and zero otherwise. Let random variable X_i denote $\sum_b X_{ib}$. Furthermore, define random variable $P_i = \sum_b p_{ib} X_{ib}$, which is the contribution of keyword i to the objective value after Stage 1. Note that $\mathbf{E}[X_{ib}] = x_{ib}$, $\mathbf{E}[P_i] = \sum_b p_{ib} x_{ib}$, and $\mathbf{E}\left[\mathsf{LP}^{(1)}\right] = \sum_i \mathbf{E}[P_i]$.

Consider the contribution of all keywords i with $X_i > 2$ to the value of $\mathsf{LP}^{(1)}$. Let random variable $\hat{P}_i$ denote this contribution. By the following inequalities, we show that $\mathbf{E}[\hat{P}_i] \leq \mathbf{E}[P_i]/2$, which means that in expectation at least half of the value of $\mathsf{LP}^{(1)}$ comes from keywords i with $X_i \leq 2$.

$$\mathbf{E}\left[\hat{P}_i\right] = \mathbf{E}\left[P_i \middle| X_i > 2\right] \cdot \mathbf{Pr}\left[X_i > 2\right] = \mathbf{E}\left[\sum_b p_{ib} X_{ib} \middle| X_i > 2\right] \cdot \mathbf{Pr}\left[X_i > 2\right]$$

$$= \sum_b \mathbf{E}\left[p_{ib} X_{ib} \middle| X_i > 2\right] \cdot \mathbf{Pr}\left[X_i > 2\right] \tag{5}$$

$$= \sum_b \mathbf{E}\left[p_{ib} X_{ib} \middle| X_i > 2 \wedge X_{ib} > 0\right] \cdot \mathbf{Pr}\left[X_{ib} > 0\right] \cdot \mathbf{Pr}\left[X_i > 2 \middle| X_{ib} > 0\right]$$

$$= \sum_b \mathbf{E}\left[p_{ib} X_{ib} \middle| X_i > 2 \wedge X_{ib} = \frac{x_{ib}}{y_b}\right] \cdot \mathbf{Pr}\left[X_{ib} = \frac{x_{ib}}{y_b}\right] \cdot \mathbf{Pr}\left[X_i > 2 \middle| X_{ib} = \frac{x_{ib}}{y_b}\right]$$

$$= \sum_b p_{ib} \frac{x_{ib}}{y_b} \cdot y_b \cdot \mathbf{Pr}\left[X_i > 2 \middle| X_{ib} = \frac{x_{ib}}{y_b}\right]$$

$$= \sum_b p_{ib} x_{ib} \cdot \mathbf{Pr}\left[X_i^{-b} > 2 - \frac{x_{ib}}{y_b} \middle| X_{ib} = \frac{x_{ib}}{y_b}\right] \leq \sum_b p_{ib} x_{ib} \cdot \frac{\mathbf{E}\left[X_i^{-b} \middle| X_{ib} = \frac{x_{ib}}{y_b}\right]}{2 - \frac{x_{ib}}{y_b}}$$

$$\leq \sum_b p_{ib} x_{ib} \cdot \frac{1 - \frac{x_{ib}}{y_b}}{2 - \frac{x_{ib}}{y_b}} \leq \sum_b p_{ib} x_{ib} \cdot \frac{1}{2} = \frac{1}{2} \mathbf{E}\left[P_i\right]. \tag{6}$$

Now, note that in Stage 2, for each keyword i with $X_i > 1$ we scale down all X_{ib} 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 2. However, Inequality (6) shows that at least half of the value of $\mathbf{E}[\mathsf{LP}^{(1)}]$ is coming from such keywords. Hence, a quarter of $\mathbf{E}[\mathsf{LP}^{(1)}]$ is preserved after Stage 2, or, $\lambda \geq \frac{1}{4}$.

The above analysis is suboptimal for two reasons. First, it ignores the contribution of $p_{ib} x_{ib}$ 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). The deferred analysis takes advantage of these observations and some concentration bounds to achieve $\lambda = 0.539968$.

Stage 3: Rounding x_{ib} Variables. We can simply pick one bid b for each keyword i with probability proportional to x_{ib}. This independent rounding enjoys concentration properties (via Chernoff bounds) for total value as well as the cost vector. This follows from the assumption that no c_{ib}^q or p_{ib} is larger than ϵ or ϵopt, respectively, otherwise we would use the PTAS in Theorem 1. Therefore, we argue that, with high probability, both are within a factor $1 \pm \epsilon$ of the semi-integral LP solution. Then, 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 a $1 - \epsilon$ factor.

The discussions in this section so far can be summarized in the following theorem.

Theorem 2. *The above algorithm provides a 0.54-approximate integral solution to the LP* (1)-(3) *with additional constraints* (4). *This, combined with Theorem 1, can be used* *to obtain a 0.35-approximate solution for the general concise bidding problem, i.e., LP* (1)-(3).

In the full version of the paper, we provide an instance of the aforementioned LP with an integrality gap of 0.63.

4 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 and the cost he will have to pay when he bids a bid b. Such tools are provided for major sponsored search advertising systems [3, 15].

The datasets contain varying number of queries (from tens to tens of thousands). For each query we obtain estimates of clicks and cost for bids in the range $[\$0.10, \$2]$.

We then run our algorithms, an appropriate version of the uniform bidding algorithm and an algorithm that computes the optimal bid for each query against each of the datasets. We apply these algorithms at different budget values to see the impact of changing the budget in the relative performance of different algorithms.

4.1 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 [11].

We first see how the total amount of clicks that can be obtained grows as the number of permitted bids increases.

Figure 1 plots the performance of the bid allocation we find for different number of bids. It shows how the objective value of the fractional LP for each dataset grows with the number of clusters. This interesting point confirms the intuition that additional clusters allow for more refined bidding on various queries, hence better performance. We also note that, conveniently, with a few clusters, we can achieve the almost optimal solution, and there is no need for a complicated strategy.

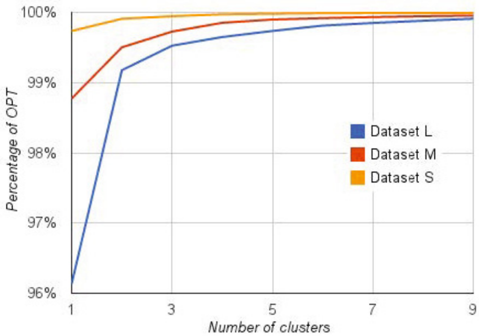

Fig. 1. Comparison of performance by number of clusters for all datasets

Next we consider the deterministic 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 performance of our algorithm for number of clusters $k = 1, 2, 3, 4$. We also report the performance of our implementation of uniform bidding that chooses an optimal set of two bids, with some

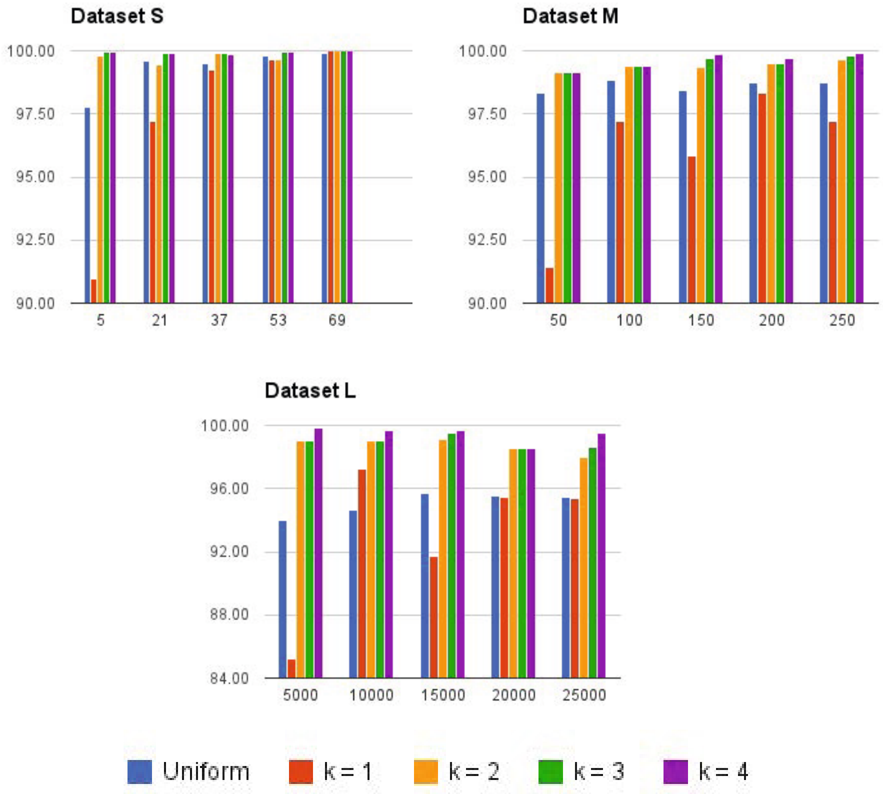

Fig. 2. Plots comparing performance of various algorithms on the three datasets. The x axis has budget in dollars, whereas the y axis shows the percentage of the (unrestricted) optimal solution.

probability chooses the first bid for all queries and otherwise chooses the other bid for all queries. This algorithm was described and analyzed in [11].

Plots in Figure 2 compare the performance of integer solutions produced by LP rounding with the uniform bidding benchmark.

We see that all algorithms perform 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. 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%.

4.2 A Multidimensional Budget Constraint

A virtue of our algorithm is that it is naturally taking 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. In the full version of this paper we report detailed results of evaluating our algorithms on experimental data in the presence of multidimensional budget constraint.

5 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 other monotone profit function) subject to a multidimensional budget constraint.

When the budget constraint has constant dimension, we propose a polynomial-time approximation scheme. 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 5% to 6% (and sometimes up to 9%).

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 the 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 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 preliminary study of this question for one-dimensional budget constraints (as reported in Figure 1) 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. Also, the reported result suggests 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 we face.

References

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

[2] Archak, N., Mirrokni, V., Muthukrishnan, S.: Budget optimization for online campaigns with positive carryover effects. In: Goldberg, P.W. (ed.) WINE 2012. LNCS, vol. 7695, pp. 86–99. Springer, Heidelberg (2012)
[3] Bing Ads. Bing ads intelligence (2013), http://advertise.bingads.microsoft.com/en-us/bingads-downloads/bingads-intelligence
[4] Borgs, C., Chayes, J.T., Immorlica, N., Jain, K., Etesami, O., Mahdian, M.: Dynamics of bid optimization in online advertisement auctions. In: World Wide Web, pp. 531–540 (2007)
[5] Devenur, N.R., Hayes, T.P.: The adwords problem: Online keyword matching with budgeted bidders under random permutations. In: ACM Conference on Electronic Commerce (EC), pp. 71–78 (2009)
[6] Dobzinski, S., Lavi, R., Nisan, N.: Multi-unit auctions with budget limits. Games and Economic Behavior 74(2), 486–503 (2012)
[7] Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. American Economic Review 97(1), 242–259 (2005)
[8] Even-Dar, E., Mirrokni, V.S., Muthukrishnan, S., Mansour, Y., Nadav, U.: Bid optimization for broad match ad auctions. In: World Wide Web, pp. 231–240 (2009)
[9] Feldman, J., Henzinger, M., Korula, N., Mirrokni, V.S., Stein, C.: Online stochastic packing applied to display ad allocation. In: de Berg, M., Meyer, U. (eds.) ESA 2010, Part I. LNCS, vol. 6346, pp. 182–194. Springer, Heidelberg (2010)
[10] Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., Pál, M.: Online ad assignment with free disposal. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 374–385. Springer, Heidelberg (2009)
[11] Feldman, J., Muthukrishnan, S., Pal, M., Stein, C.: Budget optimization in search-based advertising auctions. In: ACM Conference on Electronic Commerce (EC), pp. 40–49 (2007)
[12] Goel, G., Mirrokni, V.S., Paes Leme, R.: Polyhedral clinching auctions and the adwords polytope. In: STOC, pp. 107–122 (2012)
[13] Google Support, Google adwords dynamic search ads (2011), http://adwords.blogspot.fr/2011/10/introducing-dynamic-search-ads-beta.html
[14] Google Support. Google adwords automatic bidding tools (2013), http://support.google.com/adwords/bin/answer.py?hl=en&answer=2470106
[15] Google Support. Traffic estimator (2013), http://support.google.com/adwords/bin/answer.py?hl=en&answer=6329
[16] Google Support. Using the bid simulator (2013), http://support.google.com/adwords/bin/answer.py?hl=en&answer=2470105
[17] Hastad, J.: Clique is hard to approximate within $n^{1-\epsilon}$. Acta Mathematica 182(1) 105–142 (1999)
[18] IAB. Internet advertising bureau 2011 report (2011), http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_Report_FY_2011.pdf
[19] Mehta, A., Saberi, A., Vazirani, U.V., Vazirani, V.V.: Adwords and generalized online matching. Journal of ACM 54(5) (2007)
[20] Muthukrishnan, S., Pal, M., Svitkina, Z.: Stochastic models for budget optimization in search-based advertising. Algorithmica 58(4), 1022–1044 (2010)
[21] Rusmevichientong, P., Williamson, D.: An adaptive algorithm for selecting profitable keywords for search-based advertising services. In: ACM Conference on Electronic Commerce (EC), pp. 260–269 (2006)
[22] Varian, H.: Position auctions. Int. J. Industrial Opt. 25(6), 1163–1178 (2007)
[23] Zhou, Y., Chakrabarty, D., Lukose, R.: Budget constrained bidding in keyword auctions and online knapsack problems. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 566–576. Springer, Heidelberg (2008)