Algorithmica (2010) 58: 1022–1044
DOI 10.1007/s00453-009-9311-6

Stochastic Models for Budget Optimization in Search-Based Advertising

S. Muthukrishnan · Martin Pál · Zoya Svitkina

Received: 30 April 2008 / Accepted: 15 April 2009 / Published online: 7 May 2009

Abstract Internet search companies sell advertisement slots based on users' search queries via an auction. Advertisers have to determine how to place bids on the keywords of their interest in order to maximize their return for a given budget: this is the *budget optimization* problem. The solution depends on the distribution of future queries. In this paper, we formulate *stochastic* versions of the budget optimization problem based on natural probabilistic models of distribution over future queries, and address two questions that arise.

Evaluation Given a solution, can we evaluate the expected value of the objective function?

Optimization Can we find a solution that maximizes the objective function in expectation?

Our main results are approximation and complexity results for these two problems in our three stochastic models. In particular, our algorithmic results show that simple *prefix* strategies that bid on all cheap keywords up to some level are either optimal or good approximations for many cases; we show other cases to be NP-hard.

Preliminary versions of this paper appeared in the WWW Workshop on Sponsored Search, 2007, and the Third International Workshop on Internet and Network Economics (WINE 2007).

This work was done while Z. Svitkina was visiting Google, Inc., New York, NY.

S. Muthukrishnan · M. Pál
Google, Inc., New York, NY, USA

S. Muthukrishnan
e-mail: muthu@google.com

M. Pál
e-mail: mpal@google.com

Z. Svitkina (✉)
Department of Computing Science, University of Alberta, Edmonton, Canada
e-mail: svitkina@ualberta.ca

Keywords Advertising auctions · Approximation algorithms · Stochastic optimization

1 Introduction

Internet search companies use auctions to sell advertising slots in response to users' search queries. To participate in these auctions, an advertiser selects a set of keywords that are relevant or descriptive of her business, and submits a bid for each of them. Upon seeing a user's query, the search company runs an auction among the advertisers who have placed bids for keywords matching the query and arranges the winners in slots. The advertiser pays only if a user clicks on her ad. Advertiser's bid affects the position of the ad, which in turn affects the number of clicks received and the cost incurred. In addition to the bids, the advertiser specifies a daily budget. When the cost charged for the clicks reaches the budget, the advertiser's ads stop participating in the auctions.

In what follows, we first model and abstract the budget optimization problem, and then present our stochastic versions, before describing the results.

1.1 Advertiser's Budget Optimization Problem

We adopt the viewpoint of an advertiser and study the optimization problem she faces. The advertiser has to determine the daily budget, a good set of keywords, and bids for these keywords so as to maximize the effectiveness of her campaign. The daily budget and the choice of keywords are business-specific, so they are assumed to be given in our problem formulation. Effectiveness of a campaign is difficult to quantify since clicks resulting from some keywords may be more desirable than others, and in some cases, just appearing on the results page for a user's query may have some utility. For most of the paper, we adopt a common measure of the effectiveness of a campaign, namely, the *number* of clicks obtained.[1] Further, seen from an individual advertiser's point of view, the budgets and bids of other advertisers are fixed for the day. We develop most of our discussions assuming that each keyword has a single winning bid amount, which is fixed throughout the day and known in advance. This models the case of an auction with a single slot, and disregards the possibility of other advertisers changing their bids or running out of budget.

Our setting is a simplification of the real mechanism for the sponsored search auctions, but it models important aspects of it. Currently, the search engines determine a minimum reserve price per auction as well as the number of ads that they show. As their estimates of the reserve price become more accurate, they will be able to show ads of all advertisers who bid higher than the reserve price. That would be closer to the situation we model. As another example, advertisers always have to consider some gross estimate of the behavior of the search engine market. They may simply partition the possible outcomes of an auction into two groups, depending on whether

[1] We extend our results to a more general model where clicks for different keywords may have different values in Sect. 6.1.

their ad lost the auction completely, or it did appear in some slot, and then consider the expected costs and numbers of clicks in the second case. In the future, it will be worthwhile to study more general models that capture the effects of positions in which the ads appear. We discuss this further in Sect. 6.2, where we mention the extensions that eliminate some of the assumptions that we make.

Under our assumptions, each keyword i has a single threshold bid amount, such that any bid below this amount loses the auction and does not get any clicks. Any bid above the threshold wins the auction, and gets clicks with cost per click equal to the threshold bid amount.[2] In this case the advertiser's decision for each keyword becomes binary: whether or not to bid on it above its threshold. We use a vector of decision variables $\mathbf{b}$, in which each component b_i can be integral ($\mathsf{b}_i \in \{0, 1\}$) or fractional ($\mathsf{b}_i \in [0, 1]$), and indicates whether or not there is a bid on keyword i. A fractional bid represents bidding for b_i fraction of the queries that correspond to keyword i, or equivalently bidding on each such query with probability b_i. Integer bid solutions are slightly simpler to implement than fractional bids and are more desirable when they exist. For any two vectors, we define the notation $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i \mathsf{x}_i \mathsf{y}_i$. Analogously, for three vectors we define $\langle \mathbf{x}, \mathbf{y}, \mathbf{z} \rangle = \sum_i \mathsf{x}_i \mathsf{y}_i \mathsf{z}_i$.

Finally, consider the effect of user behavior on the advertiser. We abstract it using the vector **clicks**, where clicks_i is the number of clicks for queries corresponding to keyword i. Each such click entails a cost cpc_i, which is assumed to be known. Now, the advertiser is *budget-constrained*, and some solutions may run out of budget, which decreases the total number of clicks obtained. In particular, the advertiser has a global daily budget B, which is used to get clicks for all of the keywords. When the budget is spent, the ads stop being shown, and no more clicks can be bought. We model the limited budget as follows. Consider a solution $\mathbf{b}$ that bids on some keywords. If the budget were unlimited, then bidding on those keywords would bring $\langle \mathbf{clicks}, \mathbf{b} \rangle$ clicks, which together would cost $\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle$. But when the budget B is smaller than this cost, the solution runs out of money before the end of the day, and misses the clicks that come after that point. If we assume that the queries and clicks for all keywords are distributed uniformly throughout the day and are well-mixed, then this solution reaches the budget after $B/\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle$ fraction of the day passes, missing $(1 - B/\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle)$ fraction of the possible clicks for each keyword. As a result, the number of clicks collected before the budget is exceeded is $\langle \mathbf{clicks}, \mathbf{b} \rangle \cdot B/\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle$ in expectation.

Based on the discussion so far, we can now state the optimization problem an advertiser faces.

Definition 1 BUDGET OPTIMIZATION PROBLEM (BO) An advertiser has a set T of keywords, with $|T| = n$, and a budget B. For each keyword $i \in T$, we are given clicks_i, the number of clicks that correspond to i, and cpc_i, the cost per click of these clicks. The objective is to find a solution $\mathbf{b} = (\mathsf{b}_1, \ldots, \mathsf{b}_n)$ with a bid $0 \le \mathsf{b}_i \le 1$ for each $i \in T$ to maximize

$$value(\mathbf{b}) = \gamma(\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle) \cdot \langle \mathbf{clicks}, \mathbf{b} \rangle, \tag{1}$$

[2]This assumes *second-price* auctions, where the winner's cost is the highest bid of others. All our results also apply to weighted second-price auctions, which are common in search-based advertising.

where $\langle \mathbf{clicks}, \mathbf{b} \rangle$ is the number of clicks available to the solution, and $\gamma(x) = \min(1, B/x)$, which is a function of the total cost of these clicks, is a scaling factor for the case that the budget is exceeded.

Let the average cost per click of a solution be $\mathsf{cpc}(\mathbf{b}) = \frac{\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle}{\langle \mathbf{clicks}, \mathbf{b} \rangle}$. Then the objective function can be expressed as

$$value(\mathbf{b}) = \begin{cases} \langle \mathbf{clicks}, \mathbf{b} \rangle, & \text{if } \langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle \le B, \\ B/\mathsf{cpc}(\mathbf{b}), & \text{if } \langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b} \rangle > B. \end{cases} \tag{2}$$

So maximizing $value(\mathbf{b})$ is equivalent to maximizing the number of clicks in case that we are under budget, and minimizing the average cost per click if we are over budget. We always assume that the keywords are numbered in the order of non-decreasing cpc_i, i.e. $\mathsf{cpc}_1 \le \mathsf{cpc}_2 \le \cdots \le \mathsf{cpc}_n$.

1.2 Stochastic Versions

Many variables affect the number of clicks that an advertiser receives in a day. Besides the advertiser's choice of her own budget and keywords which we take to be given, and the choices made by other advertisers which remain fixed, the main *variable* in our problem is the number of queries of relevance that users issue on that day, and the frequency with which the ads are clicked.[3] These quantities are not known precisely in advance. Our premise is that Internet search companies can analyze past data and provide probability distributions for parameters of interest. They currently do provide limited amount of information about the range of values taken by these parameters.[4] This motivates us to study the problem in the *stochastic* setting where the goal is to maximize the *expected* value of the objective under such probability distributions.

In the stochastic versions of our problem, the set of keywords T, the budget B, and the cost per click cpc_i for each keyword are fixed and given, just like in the BO problem of Definition 1. What is different is that the numbers of clicks clicks_i corresponding to different keywords are random variables having some joint probability distribution. But because general joint probability distributions are difficult to represent and to work with, we formulate the following natural stochastic models. (In contrast, the problem where **clicks** is known precisely is called the *fixed* model from here on.)

Proportional Model The relative proportions of clicks for different keywords remain constant. This is modeled by one global random variable for the total number of clicks in the day, and a fixed known multiplier for each keyword that represents that keyword's share of the clicks.

[3] The nature and number of queries vary significantly. An example in Google Trends shows the spikes in searches for shoes, flowers and chocolate: http://www.google.com/trends?q=shoes,flowers,chocolate.

[4] See, for example, the information provided to any AdWords advertiser. See also https://adwords.google.com/support.

Independent Keywords Model Each keyword comes with its own probability distribution for the number of clicks, and the samples are drawn from these distributions independently.

Scenario Model There is an explicit list of N *scenarios*. Each scenario specifies the number of clicks for each keyword, and has a probability of occurring. We think of N as reasonably small, and allow the running time of algorithms to depend (polynomially) on N.

The scenario model is important for two reasons. For one, market analysts often think of uncertainty by explicitly creating a set of a few model scenarios, possibly attaching a weight to each scenario. The second reason is that the scenario model gives us an important segue into understanding the fully general problem with arbitrary joint distributions. Allowing the full generality of an arbitrary joint distribution gives us significant modeling power, but poses challenges to the algorithm designer. Since a naive explicit representation of the joint distribution requires space exponential in the number of random variables, one often represents the distribution implicitly by a sampling oracle. A common technique, Sampled Average Approximation (SAA), is to replace the true distribution $\mathcal{D}$ by a uniform or non-uniform distribution $\hat{\mathcal{D}}$ over a set of samples drawn by some process from the sampling oracle, effectively reducing the problem to the scenario model. For some classes of problems, see e.g. [1–3], it is known that SAA approximates the original distribution to within an arbitrarily small error using polynomially many samples. While we are not aware of such bounds applicable to the budget optimization problem, understanding the scenario model is still an important step in understanding the general problem.

There are two issues that arise in each of the three stochastic models.

- STOCHASTIC EVALUATION PROBLEM (SE). Given a solution $\mathbf{b}$, can we evaluate $E[value(\mathbf{b})]$ for the three models above? Even this is nontrivial as is typical in stochastic optimization problems. It is also of interest in solving the budget optimization problem below.
- STOCHASTIC BUDGET OPTIMIZATION PROBLEM (SBO). This is the Budget Optimization problem with one of the stochastic models above determining **clicks**, with the objective to maximize

$$E[value(\mathbf{b})] = E[\gamma(\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b}\rangle) \cdot \langle \mathbf{clicks}, \mathbf{b}\rangle]. \tag{3}$$

The expectation is taken over the distribution of **clicks**.

1.3 Our Results

We present algorithmic and complexity-theoretic results for the SE and SBO problems.

For SE problems, our results are as follows. The problem is straight-forward to solve for the fixed and scenario models since the expression for the expected value of the objective can be explicitly written in polynomial time. For the proportional model, we give an exact algorithm to evaluate a solution, assuming that some elementary quantities (such as probability of a range of values) can be extracted from the given probability distribution in polynomial time. For the independent model, the number of possibilities for different click quantities may be exponential in the number of keywords, and the problem of evaluating a solution is likely to be #P-hard. We give a PTAS for this case. These evaluation results are used to derive algorithms for the SBO problem, though they may be of independent interest.

Our main results are for the SBO problem. In fact, all our algorithms produce a special kind of solutions called *prefix solutions*. A prefix solution bids on some prefix of the list of keywords sorted in the increasing order of cost per click (cpc_i), i.e., on the cheap ones. Formally, an *integer prefix solution* with bids b_i has the property that there exists some i^* such that $\mathsf{b}_i = 1$ for all $i \le i^*$, and $\mathsf{b}_i = 0$ for $i > i^*$. For a *fractional prefix solution*, there exists an i^* such that $\mathsf{b}_i = 1$ for $i < i^*$, $\mathsf{b}_i = 0$ for $i > i^*$, and $\mathsf{b}_{i^*} \in [0, 1]$. We show:

- For the proportional model, we can find an optimal fractional solution in polynomial time if the distribution of clicks can be described using polynomial number of points; else, we obtain a PTAS. We get this result by showing that the optimal fractional solution in this case is a prefix solution and giving an algorithm to find the best prefix.
- Our main technical contribution is the result for the independent model, where we prove that every integer solution can be transformed to a prefix solution by removing a set of expensive keywords and adding a set of cheap ones, while losing at most half of the value of the solution. Thus, some integer prefix is always a 2-approximate integer solution. When combined with our PTAS for the evaluation problem, this leads to a $2 + \varepsilon$ approximation algorithm. We also show that the best fractional prefix is not in general the optimal fractional solution in this case.
- For the scenario model, we show a negative result that finding the optimum, fractional or integer, is NP-hard. In this case, the best prefix solution can be arbitrarily far from the optimum.

1.4 Related Work

Together, our results represent a new theoretical study of stochastic versions of budget optimization problems in search-related advertising. The budget optimization problem was studied recently [4] in the fixed model, when **clicks** is known. On one hand, our study is more general, with the emphasis on the uncertainty in modeling the numbers of clicks for different keywords and the stochastic models we have formulated. We do not know of prior work in this area that formulates and uses our stochastic models. On the other hand, our study is less general as it does not consider the interaction between keywords that occurs when a user's search query matches two or more keywords, which is studied in [4].

Stochastic versions of many optimization problems have been considered, such as facility location, Steiner trees, bin-packing and LP (see, for example, the survey [3]). Perhaps the most relevant to our setting is the work on the stochastic knapsack problem, of which several versions have been studied. Dean et al. [5] consider a version of the problem in which item values are fixed, and item sizes are independent random variables. The realization of an item's size becomes known once it is placed in the knapsack, so an algorithm has to select items one at a time, until the knapsack capacity is exceeded. In [6] and [7], a version of the problem with fixed item values and random sizes is considered as well, but there the goal is to choose a valuable set of items whose probability of exceeding the knapsack capacity is small. Other authors [8–11] have studied versions with fixed item sizes but random values. Our SBO problem can be viewed as a version of stochastic knapsack, with keyword costs and numbers of clicks analogous to item sizes and values, respectively, and the budget playing the role of knapsack capacity. However, our formulation is different from others in several respects. First, there is no hard capacity constraint, in the sense that, unlike in knapsack, in our problem bidding on any set of keywords constitutes a feasible solution. The chosen set only affects the value of the objective, whose scaling factor γ decreases continuously as the cost of keywords increases past the budget. The second difference is that in our model, both the number of clicks and the cost of the keywords are random, but their ratio for each particular keyword (item) is fixed and known. Another difference is that previous work on stochastic knapsack considers independent distributions of item parameters, whereas two of our models (proportional and scenario) have correlated variables. Furthermore, although the greedy algorithm which takes items in the order of their value-to-size ratio is well-known and variations of it have been applied to knapsack-like problems, our analysis proving the 2-approximation result is new.

Recently, Chakrabarty et al. [12] considered an online knapsack problem with the assumption of small element sizes, and Babaioff et al. [13] considered an online knapsack problem with a random order of element arrival, both motivated by bidding in advertising auctions. The difference with our work is that these authors consider the problem in the online algorithms framework, and analyze the competitive ratios of the obtained algorithms. In contrast, our algorithms make decisions offline, and we analyze the obtained approximation ratios for the expected value of the objective. Also, our algorithms base their decisions on the probability distributions of the clicks, whereas the authors of [13] and [12] do not assume any advance knowledge of these distributions. The two approaches are in some sense complementary: online algorithms have the disadvantage that in practice it may not be possible to make new decisions about bidding every time that a query arrives, and stochastic optimization has the disadvantage of requiring the knowledge of the probability distributions.

Also motivated by advertising in search-based auctions, Rusmevichientong and Williamson [14] have studied the *keyword selection* problem, where the goal is to select a subset of keywords from a large pool for the advertiser to choose to bid. Their model is similar to our proportional model, but the proportions of clicks for different keywords are unknown. An adaptive algorithm is developed that learns the proportions by bidding on different prefix solutions, and eventually converges to near-optimal profits [14], assuming that various parameters are concentrated around their means. The difference with our work is that we consider algorithms that solve the problem in advance, and not by adaptive learning, and work for any arbitrary (but pre-specified) probability distributions.

There has been a lot of other work on search-related auctions in the presence of budgets, but it has primarily focused on the game-theoretic aspects [15, 16], strategy-proof mechanisms [17, 18], and revenue maximization [19, 20].

1.5 Map

We briefly discuss the fixed case first, and then focus on the three stochastic models in the following sections; in each case, we solve both evaluation and BO problems. Finally, we present some extensions of our work and state a few open problems.

2 Fixed Model

For the BO problem in the fixed model, a certain fractional prefix, which is easy to find, is the optimal solution. The algorithm is analogous to that for the fractional knapsack problem. We find the maximum index i^* such that $\sum_{i \le i^*} \mathsf{clicks}_i \mathsf{cpc}_i \le B$. If i^* is the last index in T, we set $\mathsf{b}_i = 1$ for all keywords i. Otherwise find a fraction $\alpha \in [0, 1)$ such that $\sum_{i \le i^*} \mathsf{clicks}_i \mathsf{cpc}_i + \alpha \cdot \mathsf{clicks}_{i^*+1} \mathsf{cpc}_{i^*+1} = B$, and set $\mathsf{b}_i = 1$ for $i \le i^*$, $\mathsf{b}_{i^*+1} = \alpha$, and $\mathsf{b}_i = 0$ for $i > i^* + 1$.

Theorem 2 *In the fixed model, the optimal fractional solution for the BO problem is the maximal prefix whose cost does not exceed the budget, which can be found in linear time.*

The integer version of this problem is NP-hard by reduction from KNAPSACK.

3 Proportional Model

In the proportional model of SBO, we are given $\mathbf{q}$, the *click frequency* for keywords, with $\sum_{i \in T} \mathsf{q}_i = 1$. The total number of clicks is denoted by a random variable C, and has a known probability distribution. The number of clicks for a keyword i is then determined as $\mathsf{clicks}_i = \mathsf{q}_i \cdot C$. The objective is to maximize the expected number of clicks, given by expression (3).

Theorem 3 *The optimal fractional solution for the SBO problem in the proportional model is a fractional prefix solution.*

Proof We use an interchange argument to show that any solution can be transformed into a prefix solution without decreasing its value. Consider a solution $\mathbf{b}$. If $\mathbf{b}$ is not a prefix solution, then there exist keywords i and j with $i < j$, $\mathsf{b}_i < 1$, and $\mathsf{b}_j > 0$. Choose the smallest such i and the largest such j. We will decrease the bid on j and increase the bid on i, thus making the solution closer to a prefix, while maintaining the cost and the number of clicks that it obtains. If $\mathsf{q}_i \cdot \mathsf{cpc}_i = 0$, set $\mathsf{b}_i = 1$ and continue. Otherwise pick the maximum $\delta_i, \delta_j > 0$ that satisfy

$$\delta_i \le 1 - \mathsf{b}_i, \quad \delta_j \le \mathsf{b}_j, \qquad \delta_i = \frac{\mathsf{q}_j \mathsf{cpc}_j}{\mathsf{q}_i \mathsf{cpc}_i} \delta_j.$$

We assign $\mathsf{b}'_i = \mathsf{b}_i + \delta_i$, $\mathsf{b}'_j = \mathsf{b}_j - \delta_j$, and $\mathsf{b}'_k = \mathsf{b}_k$ for $k \notin \{i, j\}$. The resulting solution $\mathbf{b}'$ satisfies $\langle \mathbf{q}, \mathbf{cpc}, \mathbf{b}' \rangle = \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b} \rangle$ and $\langle \mathbf{q}, \mathbf{b}' \rangle \ge \langle \mathbf{q}, \mathbf{b} \rangle$:

$$\langle \mathbf{q}, \mathbf{cpc}, \mathbf{b}' \rangle - \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b} \rangle = \mathsf{q}_i \mathsf{cpc}_i \delta_i - \mathsf{q}_j \mathsf{cpc}_j \delta_j = 0,$$
$$\langle \mathbf{q}, \mathbf{b}' \rangle - \langle \mathbf{q}, \mathbf{b} \rangle = \mathsf{q}_i \delta_i - \mathsf{q}_j \delta_j = \mathsf{q}_j \left(\frac{\mathsf{cpc}_j}{\mathsf{cpc}_i} - 1 \right) \delta_j \geq 0.$$

So for any specific value $C = c$, the objective function of the solution does not decrease, $value(\mathbf{b}') = \gamma(c\langle \mathbf{q}, \mathbf{cpc}, \mathbf{b}' \rangle) \cdot c\langle \mathbf{q}, \mathbf{b}' \rangle \geq value(\mathbf{b})$. Therefore, the expected value over C does not decrease either, $E[value(\mathbf{b}')] \geq E[value(\mathbf{b})]$. As a result of the transformation, either $\mathsf{b}'_i = 1$ or $\mathsf{b}'_j = 0$, so the process terminates after a finite number of steps, resulting in a prefix solution with expected value at least that of the original one. □

We now show how to solve the evaluation problem efficiently in the proportional model, and then use it to find the best prefix, which by Theorem 3 is the optimal fractional solution to SBO.

3.1 Evaluating a Solution

Assuming that the distribution for C is given in such a way that it is easy to evaluate $\Pr[C > c^*]$ and $\sum_{c \leq c^*} c \cdot \Pr[C = c]$ for any c^*, we show how to find $E[value(\mathbf{b})]$ for any given solution $\mathbf{b}$ without explicitly going through all possible values of C and evaluating the objective function for each one.

The solution $\mathbf{b}$ may be under or over budget depending on the value of C. Define a threshold $c^* = B/\langle \mathbf{q}, \mathbf{cpc}, \mathbf{b} \rangle$, so that if $C > c^*$, the solution exceeds the budget, and otherwise it does not. Notice that in the proportional model, $\mathsf{cpc}(\mathbf{b}) = \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b} \rangle / \langle \mathbf{q}, \mathbf{b} \rangle$ is independent of C. Then using expression (2) for $value(\mathbf{b})$, the objective becomes easy to evaluate:

$$E[value(\mathbf{b})] = \langle \mathbf{q}, \mathbf{b} \rangle \sum_{c \leq c^*} c \cdot \Pr[C = c] + \frac{B}{\mathsf{cpc}(\mathbf{b})} \Pr[C > c^*]. \tag{4}$$

Here the first part of the expression accounts for the cases when the budget is not exceeded, and the solution obtains all the available clicks. The second part accounts for the cases when the total cost exceeds B, and the solution spends the entire budget at the average rate of $\mathsf{cpc}(\mathbf{b})$ per click.

3.2 Finding the Optimal Prefix

It is nontrivial to find the best fractional prefix solution for the proportional case, and we mention two approaches that do not work. One simple way to find a prefix in the proportional model is to convert it to a fixed case problem by setting the number of clicks for each keyword to its expectation. This approach fails on an example with two keywords and two possible values of C: a likely value of zero and a small-probability large value. Another approach is some greedy procedure that lengthens the prefix while the solution improves. This does not work either, because the expected value of the solution as a function of the length of prefix can have multiple local maxima.

The best prefix can be found by producing a list of $O(n + t)$ prefixes (out of infinitely many possible ones) containing the optimum, and then evaluating each of them as described in Sect. 3.1. Here t is the number of possible values of C. If t is not polynomial in n, the probability that C falls between successive powers of $(1+\varepsilon)$ can be combined into buckets, yielding a PTAS for the problem. The algorithm divides the set of all prefixes into intervals in such a way that within a given interval, the objective function is continuous and differentiable, as it does not contain any sudden jumps resulting from reaching the budget or starting to bid on a new keyword. As a result, within any such interval, the function is simple to analyze, and attains at most one local maximum. So it suffices to evaluate the prefixes at the interval boundaries, as well as the unique best ones inside the intervals.

Theorem 4 *The optimal fractional solution to SBO problem in the proportional model can be found exactly in time $O(n+t)$, where t is the number of possible values of C, or approximated by a PTAS.*

Proof We *mark* some points in the space of possible prefixes. First, we mark all the integer prefixes. Then, for each value c of C that has non-zero probability, we mark the threshold prefix $\mathbf{b}$ that exactly spends the budget for $C=c$, i.e. such that $c\cdot\langle\mathbf{q},\mathbf{cpc},\mathbf{b}\rangle = B$. This partitions the space of prefixes into intervals. Notice that for any two prefix solutions $\mathbf{b}$ and $\mathbf{b}'$ inside of the same interval I, the set of positive-probability values of C that cause these solutions to exceed the budget is the same. Call this set $C_I^>$:

$$\begin{aligned} C_I^> &= \{c \mid \Pr[C=c]>0 \text{ and } c\cdot\langle\mathbf{q},\mathbf{cpc},\mathbf{b}\rangle > B\} \\ &= \{c \mid \Pr[C=c]>0 \text{ and } c\cdot\langle\mathbf{q},\mathbf{cpc},\mathbf{b}'\rangle > B\}. \end{aligned}$$

Now we show how to find the optimal prefix solution inside an interval defined by the marked points. Consider such an interval I, and suppose that all prefix solutions inside I bid $\mathsf{b}_j=1$ for $j<i$, $\mathsf{b}_j=0$ for $j>i$, and $\mathsf{b}_i\in(\alpha,\beta)$ for some $0\le\alpha<\beta\le 1$. Then the objective function for solutions in this interval becomes (analogously to (4))

$$\left(\sum_{j<i}\mathsf{q}_j+\mathsf{b}_i\mathsf{q}_i\right)\sum_{c\notin C_I^>} c\cdot\Pr[C=c] + B\cdot\Pr[C\in C_I^>]\cdot\frac{\sum_{j<i}\mathsf{q}_j+\mathsf{b}_i\mathsf{q}_i}{\sum_{j<i}\mathsf{q}_j\mathsf{cpc}_j+\mathsf{b}_i\mathsf{q}_i\mathsf{cpc}_i},$$

which we have to maximize over the possible values of the variable b_i. This can be done by taking the derivative of this expression with respect to b_i and setting it to zero, which has at most one solution on the interval (α,β). If this solution exists, we add it to a set of *interesting* points. To obtain the overall optimum, we evaluate all prefixes defined by *marked* and *interesting* points.

In the case that t is not polynomial in n, we obtain a PTAS for the problem. For a given constant ε, define a new random variable C' such that $\Pr[C'=0]=\Pr[C=0]$ and $\Pr[C'=(1+\varepsilon)^k]=\Pr[(1+\varepsilon)^k\le C<(1+\varepsilon)^{k+1}]$ for integer $k\ge 0$. Then run the above algorithm using C' for the total number of clicks, and return the optimal prefix found by it. We note that the number of distinct values in the support of C' is $O(\log C^{\max})$, where $C^{\max}$ is the maximum possible value of C, and thus polynomial in the input size. We now show that the solution returned by this algorithm is within a factor $(1+\varepsilon)$ of optimum.

Let $\mathbf{b}^*$ be the optimal solution, and let $\mathbf{b}$ be the solution found by the algorithm. We prove the following inequalities, which imply the result:

$$E_C[\mathit{value}(\mathbf{b})] \ge E_{C'}[\mathit{value}(\mathbf{b})] \ge E_{C'}[\mathit{value}(\mathbf{b}^*)] \ge \frac{1}{1+\varepsilon} E_C[\mathit{value}(\mathbf{b}^*)].$$

The middle one follows from optimality of $\mathbf{b}$ under the distribution of C'. For a value c, let us denote by $\lfloor c \rfloor$ the nearest value in the support of C' that is less than or equal to c. Thus, $\lfloor C \rfloor$ has the same distribution as C'. We also note that for any $\alpha \ge 1$, $\gamma(x/\alpha) = \min(1, \alpha B/x) \le \min(\alpha, \alpha B/x) = \alpha\gamma(x)$. So, to prove the first inequality, we have

$$\begin{aligned} E_{C'}[\mathit{value}(\mathbf{b})] &= E_C[\gamma(\lfloor C \rfloor \cdot \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b} \rangle) \cdot \lfloor C \rfloor \cdot \langle \mathbf{q}, \mathbf{b} \rangle] \\ &\le E_C\left[\frac{C}{\lfloor C \rfloor}\gamma(C \cdot \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b} \rangle) \cdot \lfloor C \rfloor \cdot \langle \mathbf{q}, \mathbf{b} \rangle\right] \\ &= E_C[\mathit{value}(\mathbf{b})]. \end{aligned}$$

And for the last inequality, using the fact that γ is monotone non-increasing,

$$\begin{aligned} E_{C'}[\mathit{value}(\mathbf{b}^*)] &= E_C[\gamma(\lfloor C \rfloor \cdot \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b}^* \rangle) \cdot \lfloor C \rfloor \cdot \langle \mathbf{q}, \mathbf{b}^* \rangle] \\ &\ge E_C\left[\gamma(C \cdot \langle \mathbf{q}, \mathbf{cpc}, \mathbf{b}^* \rangle) \cdot \frac{\lfloor C \rfloor}{C} C \cdot \langle \mathbf{q}, \mathbf{b}^* \rangle\right] \\ &\ge \frac{1}{1+\varepsilon} E_C[\mathit{value}(\mathbf{b}^*)]. \end{aligned}$$

□

4 Independent Model

In the independent model of SBO, there is a probability distribution for the number of clicks for each keyword $i \in T$. These distributions can be different for different keywords. The key distinguishing feature of this model is that for $i \ne j$, the variables clicks_i and clicks_j are independent. This model is more complex than the ones discussed so far. A three-keyword example shows the following.

Theorem 5 *In the independent model of the SBO problem, the optimal fractional solution may not be a prefix solution.*

Proof We give an example with three keywords in which the optimal solution bids on the first and third keywords, and gets more clicks than any (even fractional) prefix solution. The idea is that the second and third keywords cost about the same, but the third one is better because it always comes in the same quantity, whereas the second one has high variance. Let $\mathsf{cpc}_1 = 0$, $\mathsf{cpc}_2 = 1$, $\mathsf{cpc}_3 = 1$; $\mathsf{clicks}_1 = 1$ with probability 1, $\mathsf{clicks}_2 = (0 \text{ or } 1)$ with probability $\frac{1}{2}$ each, and $\mathsf{clicks}_3 = 1$ with probability 1; $B = 1$. The optimal solution is $\mathsf{b}_1 = \mathsf{b}_3 = 1$ and $\mathsf{b}_2 = 0$, which always gets 2 clicks.

The best prefix solution is $\mathsf{b}_1 = \mathsf{b}_2 = \mathsf{b}_3 = 1$, which gets 2 or 1.5 clicks with probability $\frac{1}{2}$ each, or only 1.75 clicks in expectation. The example can be modified so that the third keyword is strictly more expensive than the second one. □

In Sect. 4.2 we prove that some integer prefix is a 2-approximate integer solution. But finding the best integer prefix requires the ability to evaluate a given solution, which in this model is likely to be $\#P$-hard. In Sect. 4.1, we develop a PTAS, based on dynamic programming, for evaluating a solution. Combined, these two results imply a $(2+\varepsilon)$-approximation for the SBO problem in the independent model.

4.1 Evaluating a Solution in the Independent Model

In this section we present a PTAS for the SE problem in the independent model. We are given an instance of the SBO problem, and an (integer or fractional) solution $\mathbf{b}$. For a keyword $i \in T$, let $C_i = \{\mathsf{cl}_i \mid \Pr[\mathsf{clicks}_i = \mathsf{cl}_i] > 0\}$ be the set of values that clicks_i can take. For now we assume that $\sum_i |C_i|$ is polynomial in the size of the input, and later show how to remove this assumption. Let the vector $\mathbf{b}_{-i}$ be equal to $\mathbf{b}$, except with the entry b_i replaced by zero (and similarly for other vectors). By some algebraic manipulation, one can show the following.

Claim 6

$$E[\mathit{value}(\mathbf{b})] = \sum_{i\in T}\sum_{\mathsf{cl}_i\in C_i}\Pr[\mathsf{clicks}_i = \mathsf{cl}_i]\mathsf{b}_i\mathsf{cl}_i\sum_{d\geq 0}\gamma(d+\mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i)\Pr[\langle\mathbf{clicks},\mathbf{cpc},\mathbf{b}_{-i}\rangle = d].$$

In this expression, $\mathsf{b}_i\mathsf{cl}_i$ is the number of clicks from keyword i, and the expression in the third sum is the expected value of the scaling factor γ, with respect to d, the total cost of clicks from other keywords.

As a result, the problem of finding $E[\mathit{value}(\mathbf{b})]$ reduces to evaluating, for any given i and cl_i, the expression

$$s(i,\mathsf{cl}_i) = \sum_{d\geq 0}\gamma(d+\mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i)\Pr[\langle\mathbf{clicks},\mathbf{cpc},\mathbf{b}_{-i}\rangle = d]. \tag{5}$$

Lemma 7 *For any constant $\varepsilon > 0$, there is a polynomial-time algorithm that finds a value s' such that $s(i,\mathsf{cl}_i) \leq s' \leq (1+\varepsilon)s(i,\mathsf{cl}_i)$.*

Proof We build a dynamic programming table that estimates $\Pr[\langle\mathbf{clicks},\mathbf{cpc},\mathbf{b}_{-i}\rangle = d]$ as a function of d. Fix an ordering of elements in $T-\{i\}$ and construct a table P indexed by j and d, where $P(j,d)$ is the probability that the total cost of the first j elements is d. To make sure the table is of polynomial size, scale the costs so that the minimum non-zero value of $\mathsf{clicks}_j\cdot\mathsf{cpc}_j$ for any j is 1, and restrict the possible values of d to 0 and $(1+\frac{\varepsilon}{n})^k$ for non-negative integers k. If we let $M = \sum_{j\in T}\max\{\mathsf{cl}_j\cdot\mathsf{cpc}_j|\mathsf{cl}_j\in C_j\}$ be the maximum possible cost of all the clicks, then the number of values of d in the table is at most $\log_{1+\varepsilon/n} M = O(\frac{n}{\varepsilon}\log M)$, which is polynomial in the size of the input.

The table is initialized with $P(0,0)=1$ and other entries equal to zero. Then for each keyword $j \in T-\{i\}$, each possible number of clicks $\mathsf{cl}_j \in C_j$, and each entry $P(j-1,d)$ in the previous row, we update $P(j,\lfloor d+\mathsf{cl}_j \cdot \mathsf{cpc}_j \rfloor) = P(j,\lfloor d+\mathsf{cl}_j \cdot \mathsf{cpc}_j \rfloor) + \Pr[\mathsf{clicks}_j = \mathsf{cl}_j] \cdot P(j-1,d)$. Here the operator $\lfloor\ \rfloor$ represents rounding down to the next available value of d. After filling the table, the algorithm outputs the value of expression (5) as determined by probabilities in the last row of the table.

To bound the error incurred by rounding down the costs, consider a particular vector $\mathbf{cl}_{-i}$ that specifies a number of clicks for each $j \in T-\{i\}$. As a result of a series of updates, its probability $\Pr[\mathbf{clicks}_{-i} = \mathbf{cl}_{-i}] = \prod_{j\neq i} \Pr[\mathsf{clicks}_j = \mathsf{cl}_j]$ contributes to some entry of the last row of the table P, say to the one with $d' = (1+\frac{\varepsilon}{n})^k$. This d' is an estimate of $d^* = \langle \mathbf{cl}_{-i}, \mathbf{cpc}, \mathbf{b}\rangle$, the total cost of keywords other than i. Since we only rounded down, we have $d' \le d^*$. Now note that since the intervals between successive powers of $(1+\frac{\varepsilon}{n})$ are increasing, the biggest amount that we could have lost during any one rounding is $(1+\frac{\varepsilon}{n})^{k+1} - (1+\frac{\varepsilon}{n})^k = \frac{\varepsilon}{n}(1+\frac{\varepsilon}{n})^k = \frac{\varepsilon}{n}\cdot d'$. Since we performed the rounding during at most n updates, we have that the true value $d^* \le d' + n\cdot\frac{\varepsilon}{n}\cdot d' = (1+\varepsilon)d'$. So we have that the estimated cost d' is $\frac{d^*}{1+\varepsilon} \le d' \le d^*$.

What remains in order to show that the algorithm evaluates expression (5) accurately is to take γ into account. Since $\gamma(x)$ is monotone non-increasing,

$$\gamma\left(\frac{d^*}{1+\varepsilon} + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right) \ge \gamma\left(d' + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right) \ge \gamma\left(d^* + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right).$$

But notice that for any $\alpha \ge 1$, $\gamma(x/\alpha) = \min(1, \alpha B/x) \le \min(\alpha, \alpha B/x) = \alpha\gamma(x)$. So

$$\gamma\left(\frac{d^*}{1+\varepsilon} + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right) \le \gamma\left(\frac{d^* + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i}{1+\varepsilon}\right) \le (1+\varepsilon)\cdot\gamma\left(d^* + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right).$$

So we have that

$$\gamma\left(d' + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right) \in [1, 1+\varepsilon]\cdot\gamma\left(d^* + \mathsf{cl}_i\mathsf{cpc}_i\mathsf{b}_i\right).$$

So evaluating expression (5) using entries from the dynamic programming table instead of the true costs and probabilities incurs a multiplicative error of at most $(1+\varepsilon)$. □

If the input distributions are represented implicitly, such that $\sum_i |C_i|$ is not polynomial in the input size, then we first convert them into distributions with polynomial number of points by combining the probability mass between successive powers of $(1+\varepsilon')$ into buckets (rounding down). Then we run the above algorithm for discrete distributions so as to obtain a $(1+\varepsilon')$-approximation for the rounded instance. This will be a $(1+\varepsilon')^2$-approximation for the original instance, so if ε' is chosen such that $(1+\varepsilon')^2 \le (1+\varepsilon)$, we obtain the desired $(1+\varepsilon)$-approximation.

Theorem 8 *There is a PTAS for the stochastic evaluation problem in the independent keywords model.*

4.2 Prefix is a 2-Approximation

In this section we show that for any instance of the SBO problem in the independent model, there exists an integer prefix solution whose expected value is at least half that of the optimal integer solution. In particular, any integer solution $\mathbf{b}$ can be transformed into a prefix solution $\mathbf{b}_V$ without losing more than half of its value.

We have $E[value(\mathbf{b})] = \sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle) \cdot \langle \mathbf{cl}, \mathbf{b}\rangle$, where the vector $\mathbf{cl}$ ranges over possible values of $\mathbf{clicks}$. Let $S = \{i \mid \mathsf{b}_i = 1\}$ be the set of keywords that $\mathbf{b}$ bids on, and define $i^*(\mathbf{b})$ as the minimum index i^* such that keywords up to i^* account for at least half of this value:

$$\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle) \sum_{i \in S, i \le i^*} \mathsf{cl}_i \ge \frac{1}{2} E[value(\mathbf{b})]. \tag{6}$$

Theorem 9 *For any integer solution $\mathbf{b}$ to the SBO problem with independent keywords, there exists an integer prefix solution $\mathbf{b}_V$ such that $E[value(\mathbf{b}_V)] \ge \frac{1}{2} E[value(\mathbf{b})]$. In particular, the solution $\mathbf{b}_V$ bidding on the set $V = \{i \mid i \le i^*(\mathbf{b})\}$ has this property.*

The idea of the proof is to think of the above prefix solution as being obtained in two steps from the original solution $\mathbf{b}$. First, we truncate $\mathbf{b}$ by discarding all keywords after i^*. Then we fill in the gaps in the resulting solution in order to make it into a prefix. To analyze the result, we first show that all keywords up to i^* are relatively cheap, and that the truncated solution (called $\mathbf{b}_U$) retains at least half the value of the original one (Claim 10). Then we show that filling in the gaps preserves this guarantee. Intuitively, two good things may happen: either clicks for the new keywords don't come, in which case we get all the clicks we had before; or they come in large quantity, spending the budget, which is good because they are cheap. Lemma 11 analyzes what happens if new clicks spend α fraction of the budget.

Let $i^* = i^*(\mathbf{b})$. To analyze our proposed prefix solution $\mathbf{b}_V$, we break the set V into two disjoint sets U and N. $U = V \cap S = \{i \le i^* \mid i \in S\}$ is the set of cheapest keywords that get half the clicks of $\mathbf{b}$. The new set $N = V \setminus S = \{i \le i^* \mid i \notin S\}$ fills the gaps in U. Let $\mathbf{b}_U$ and $\mathbf{b}_N$ be the solutions that bid on keywords in U and N respectively.

Define the average cost per click of solution $\mathbf{b}$ as

$$\mathsf{cpc}^* = \frac{\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle)\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle}{\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle)\langle \mathbf{cl}, \mathbf{b}\rangle} = \frac{E[\min(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle, B)]}{E[value(\mathbf{b})]},$$

where the numerator is the average amount of money spent by $\mathbf{b}$, and the denominator is the average number of clicks obtained. A useful fact that follows is that

$$E[value(\mathbf{b})] \le \frac{B}{\mathsf{cpc}^*}. \tag{7}$$

We make two observations about $\mathbf{b}_U$ and i^*.

Claim 10 $E[value(\mathbf{b}_U)] \ge \frac{1}{2} E[value(\mathbf{b})]$ *and* $\mathsf{cpc}_{i^*} \le 2\mathsf{cpc}^*$.

Proof A keyword in U does not necessarily contribute exactly the same number of clicks when it appears as part of solution U as it does when it is part of the solution S. However, in U it contributes at least as much. Namely, for any $\mathbf{cl}$, $\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U\rangle) \geq \gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle)$. So

$$E[value(\mathbf{b}_U)] = \sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U\rangle)\langle \mathbf{cl}, \mathbf{b}_U\rangle$$
$$\geq \sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle) \sum_{i\in U} \mathsf{cl}_i \geq \frac{1}{2} E[value(\mathbf{b})]$$

by definitions of U and i^*.

For the second part, we note that, since i^* is defined as the *minimum* index satisfying (6), for $i < i^*$ we have $\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle) \sum_{i\in S, i<i^*} \mathsf{cl}_i < \frac{1}{2}E[value(\mathbf{b})]$, and therefore for $i \geq i^*$, $\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle) \times \sum_{i\in S, i\geq i^*} \mathsf{cl}_i \geq \frac{1}{2}E[value(\mathbf{b})]$. Now, using the fact that $\mathsf{cpc}_i \geq \mathsf{cpc}_{i^*}$ for $i \geq i^*$, we get

$$\mathsf{cpc}^* = \frac{\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle)\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle}{E[value(\mathbf{b})]}$$
$$\geq \frac{\sum_{\mathbf{cl}} \Pr[\mathbf{clicks} = \mathbf{cl}]\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}\rangle) \sum_{i\in S, i\geq i^*} \mathsf{cpc}_{i^*}\mathsf{cl}_i}{E[value(\mathbf{b})]} \geq \frac{1}{2}\mathsf{cpc}_{i^*},$$

where we factor out cpc_{i^*} and then use the minimality of i^* as discussed above. □

Now we state the main lemma.

Lemma 11 *Fix a value* $\mathbf{cl}$ *of* $\mathbf{clicks}$, *and let* $\alpha = \min(B, \langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N\rangle)/B$. *Then*

$$\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V\rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_V\rangle \geq \alpha \frac{B}{2\mathsf{cpc}^*} + (1-\alpha)\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U\rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_U\rangle.$$

Proof The idea here is that α is the fraction of the budget spent by the new keywords (ones from set N). So $(1-\alpha)$ fraction of the budget can be used to buy $(1-\alpha)$ fraction of clicks that $\mathbf{b}_U$ was getting, and α fraction is spent on keywords (whether from U or N) that cost at most $2\mathsf{cpc}^*$. A more formal analysis follows.

If $\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N\rangle \geq B$, then $\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V\rangle \geq B$, so $\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V\rangle) = B/\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V\rangle$, and

$$\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V\rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_V\rangle = B \cdot \frac{\langle \mathbf{cl}, \mathbf{b}_V\rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V\rangle} \geq \frac{B}{\mathsf{cpc}_{i^*}} \geq \frac{B}{2\mathsf{cpc}^*},$$

which proves the lemma for the case of $\alpha = 1$. Intuitively, in this case the whole budget is spent, and since all keywords in V cost at most $2\mathsf{cpc}^*$ (by Claim 10), V gets at least $\frac{B}{2\mathsf{cpc}^*}$ clicks. For the rest of the proof assume that $\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N\rangle < B$. Then $\alpha = \langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N\rangle/B < 1$.

Another simple case is $\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle \le B$. Then $\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle) = 1$, the budget is not exceeded, and V collects all the clicks from U and N:

$$\begin{aligned}\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_V \rangle &= \langle \mathbf{cl}, \mathbf{b}_N \rangle + \langle \mathbf{cl}, \mathbf{b}_U \rangle \\ &\ge \frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N \rangle}{2\mathsf{cpc}^*} + (1-\alpha)\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle)\langle \mathbf{cl}, \mathbf{b}_U \rangle.\end{aligned}$$

Now consider the case when $\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle > B$. Then $\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle) = B/\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle$. Here at most α fraction of the budget is used for the new keywords from N, which cost at most $2\mathsf{cpc}^*$ per click, and the remaining budget is able to buy $(1-\alpha)$ fraction of the clicks that $\mathbf{b}_U$ was getting.

Let $\mathsf{cpc}(\mathbf{b}_U) = \langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle / \langle \mathbf{cl}, \mathbf{b}_U \rangle$, and similarly for $\mathbf{b}_N$. Then

$$\frac{B}{\mathsf{cpc}(\mathbf{b}_U)} = \frac{B}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle} \langle \mathbf{cl}, \mathbf{b}_U \rangle \ge \gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle)\langle \mathbf{cl}, \mathbf{b}_U \rangle. \tag{8}$$

Now,

$$\begin{aligned}&\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_V \rangle \\ &\quad = \gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle) \cdot (\langle \mathbf{cl}, \mathbf{b}_U \rangle + \langle \mathbf{cl}, \mathbf{b}_N \rangle) \\ &\quad = \frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle} \cdot \frac{B}{\mathsf{cpc}(\mathbf{b}_U)} + \frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N \rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle} \cdot \frac{B}{\mathsf{cpc}(\mathbf{b}_N)} \\ &\quad = \frac{(1-\alpha)B}{\mathsf{cpc}(\mathbf{b}_U)} + \left[\frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle} - (1-\alpha)\right]\frac{B}{\mathsf{cpc}(\mathbf{b}_U)} + \frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N \rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle} \cdot \frac{B}{\mathsf{cpc}(\mathbf{b}_N)} \\ &\quad \ge \frac{(1-\alpha)B}{\mathsf{cpc}(\mathbf{b}_U)} + \left[\frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle} - (1-\alpha)\right]\frac{B}{2\mathsf{cpc}^*} + \frac{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_N \rangle}{\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle} \cdot \frac{B}{2\mathsf{cpc}^*} \\ &\quad \ge (1-\alpha)\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_U \rangle + \alpha \frac{B}{2\mathsf{cpc}^*},\end{aligned}$$

where the first inequality follows because $\mathsf{cpc}(\mathbf{b}_U) \le 2\mathsf{cpc}^*$, $\mathsf{cpc}(\mathbf{b}_N) \le 2\mathsf{cpc}^*$, and the quantity in square brackets is non-negative. The second inequality follows by combining terms with $\frac{B}{2\mathsf{cpc}^*}$ and using (8). □

Proof of Theorem 9 We now use the above results to prove the theorem. Let $\mathbf{clicks}_U$ be the vector $\mathbf{clicks}$ restricted to the set U. Here the independence of keywords becomes crucial. In particular, what we need is that the number of clicks that come for keywords in U is independent of the number of clicks for keywords in N. So the probability $\Pr[\mathbf{clicks}_V = \mathbf{cl}_V]$ is the product of $\Pr[\mathbf{clicks}_U = \mathbf{cl}_U]$ and $\Pr[\mathbf{clicks}_N = \mathbf{cl}_N]$. Notice that α of Lemma 11 depends only on keywords in N, and is independent of what happens with keywords in U. So here we call it α_N. We have

$$\begin{aligned}&E[\mathit{value}(\mathbf{b}_V)] \\ &\quad = \sum_{\mathbf{cl}_V} \Pr[\mathbf{clicks}_V = \mathbf{cl}_V] \cdot \gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_V \rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_V \rangle\end{aligned}$$

$$\geq \sum_{\mathbf{cl}_N} \sum_{\mathbf{cl}_U} \Pr[\mathbf{clicks}_N = \mathbf{cl}_N] \Pr[\mathbf{clicks}_U = \mathbf{cl}_U]$$

$$\times \left[\frac{\alpha_N B}{2\mathsf{cpc}^*} + (1-\alpha_N)\gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_U \rangle \right]$$

$$= \sum_{\mathbf{cl}_N} \Pr[\mathbf{clicks}_N = \mathbf{cl}_N]$$

$$\times \left[\frac{\alpha_N B}{2\mathsf{cpc}^*} + (1-\alpha_N) \sum_{\mathbf{cl}_U} \Pr[\mathbf{clicks}_U = \mathbf{cl}_U] \gamma(\langle \mathbf{cl}, \mathbf{cpc}, \mathbf{b}_U \rangle) \cdot \langle \mathbf{cl}, \mathbf{b}_U \rangle \right]$$

$$\geq \sum_{\mathbf{cl}_N} \Pr[\mathbf{clicks}_N = \mathbf{cl}_N] \cdot \frac{1}{2} E[value(\mathbf{b})] = \frac{1}{2} E[value(\mathbf{b})],$$

bounding both $\frac{B}{2\mathsf{cpc}^*}$ and $E[value(\mathbf{b}_U)]$ by $\frac{1}{2}E[value(\mathbf{b})]$ using inequality (7) and Claim 10. □

Theorem 9 combined with PTAS for the evaluation problem in the independent model (Theorem 8) gives a simple algorithm for finding a $(2+\varepsilon)$-approximate solution: evaluate each integer prefix using the PTAS and output the one with maximum value.

Theorem 12 *There is a* $(2+\varepsilon)$*-approximation algorithm for the SBO problem in the independent model, which runs in time polynomial in* n, $\frac{1}{\varepsilon}$, *and* $\log M$, *where* M *is the maximum possible cost of all clicks.*

5 Scenario Model

In the scenario model, we are given T, B and costs $\mathbf{cpc}$ as usual. The numbers of clicks are determined by a set of scenarios Σ and a probability distribution over it. The reason this model does not capture the full generality of arbitrary distributions is that we assume that the number of scenarios, N, is relatively small, in the sense that algorithms are allowed to run in time polynomial in N. On the other hand, if, for example, we express the independent model in terms of scenarios, their number would be exponential in the number of keywords.

The evaluation of a given solution in the scenario model does not present a problem, as it can be done explicitly in time polynomial in N, by evaluating each scenario and taking the expectation. Nevertheless, this is the most difficult model for SBO that we consider. We show two negative results.

Theorem 13 *The SBO problem is NP-hard in the scenario model.*

Proof We show that finding the optimal solution, either integer or fractional, to the SBO problem in the scenario model is NP-hard.

The reduction is from CLIQUE. We are given an instance of the CLIQUE problem with a graph G containing n nodes and m edges, and a desired clique size k. We use G and k to construct an instance I of SBO problem and a number V such that there is a solution to I with expected value of at least V if and only if G contains a clique of size k.

To specify I, let us construct a new bipartite graph $H = (L \cup R, E')$ whose right side R contains n nodes corresponding to nodes of G, and whose left side L contains m nodes corresponding to edges of G. A node in L corresponding to an edge (u, v) is connected to nodes in R that correspond to its endpoints u and v. We first describe the idea of the construction, and later show how to set the parameters to make it work. There will be three parameters, small positive values ε and δ, and a large value t.

All nodes of H are keywords, expensive ones on the left, with $\mathsf{cpc}_i = 1$ for $i \in L$, and cheap ones on the right, with $\mathsf{cpc}_i = \varepsilon$ for $i \in R$. The budget is $K = \binom{k}{2}$. The goal will be to force a solution to select K nodes from L that are incident to at most k nodes in R, which corresponds to finding a set of k vertices with K edges in G, i.e. a k-clique. The scenarios in the SBO problem are as follows. There is a high-probability scenario σ_0 in which one click comes for each keyword in L. This scenario is sufficiently likely (occurs with probability $1 - \delta$) that any integer solution to SBO has to bid for at least K of these keywords. Notice that since K is the budget, it does not make sense to bid on any more than K keywords. In addition to σ_0, there are n scenarios $\sigma_1 \cdots \sigma_n$, each occurring with probability δ/n. Scenario σ_i contains K/ε clicks for keyword $i \in R$ and a large number t of clicks for each of i's neighbors from L.

We now explain the intuition for why there is a good integrally-bidding solution for our SBO instance if and only if the graph G contains a k-clique. By the way we constructed the low-probability scenarios, if a solution does not bid on any neighbors of $i \in R$, then in scenario σ_i it would spend its whole budget on K/ε cheap clicks at cost ε each, thus obtaining many clicks. However, if it bids on any neighbors of i, then most of the budget will be spent on the expensive clicks from L, resulting in few clicks overall. So bidding on a keyword l from L effectively ruins the scenarios containing l's neighbors in R. Recall that the high probability of scenario σ_0 forces any good integral solution to bid on exactly K keywords from L. As a result, if G contains a k-clique, then it is possible to select K keywords corresponding to edges of G that ruin only k scenarios corresponding to nodes of G. However, if there is no k-clique, then bidding on any K keywords on the left ruins at least $k + 1$ scenarios, thus producing a solution with a lower value.

We now show how to set the parameters of the construction and prove that the reduction works even if fractional bidding on keywords is allowed. First, assume that G contains a k-clique. Then a solution $\mathbf{b}$ to I, with $\mathsf{b}_i = 1$ for all $i \in R$ and $\mathsf{b}_i = 1$ for the K keywords in L that correspond to edges of the clique, achieves the expected value of at least

$$V = (1 - \delta) \cdot K + \frac{\delta(n - k)}{n} \cdot \frac{K}{\varepsilon},$$

where the first term is the value from scenario σ_0, and the second term is the value from scenarios σ_i such that node i in G is not in the clique. Such scenarios are unaffected by the selected nodes in L and therefore get K/ε clicks each. There is additional value from scenarios σ_i for i in the clique, but we disregard it for this lower bound. Thus we get the following claim.

Claim 14 *If G contains a k-clique, then there exists a solution $\mathbf{b}$ to I such that $E[value(\mathbf{b})] \geq V$.*

To ensure that if there is no k-clique in G, then value V cannot be achieved by any bids, we set the parameters as follows.

1. Select ε such that $0 < \varepsilon < \frac{1}{k+1}$.
2. Select $\delta > 0$ small enough that $\frac{1-\delta}{2} - \frac{k\delta K}{n\varepsilon} > 0$. This is possible because the limit of the expression on the left as $\delta \to 0$ is $\frac{1}{2}$.
3. Let $\alpha = \frac{1}{2m}$.
4. Choose t large enough that $(k+1)\frac{K/\varepsilon+\alpha t}{K+\alpha t} < \frac{1}{\varepsilon}$. This is possible because $\lim_{t\to\infty}(k+1)\frac{K/\varepsilon+\alpha t}{K+\alpha t} = k+1 < \frac{1}{\varepsilon}$ by the choice of ε.

Claim 15 *If $E[value(\mathbf{b})] \geq V$ for some fractional solution $\mathbf{b}$ to the constructed instance I, then there must be at least K keywords $i \in L$ such that $\mathsf{b}_i \geq \alpha$.*

Proof Notice that $\sum_{i\in L} \mathsf{b}_i \geq (K - 1 + m\alpha)$ implies that $|\{i \in L | \mathsf{b}_i \geq \alpha\}| \geq K$, because b_i's are always at most 1.

What remains to show is that if $\sum_{i\in L} \mathsf{b}_i < K - 1 + m\alpha = K - \frac{1}{2}$, then $E[value(\mathbf{b})] < V$. This follows from the way we defined the parameters. Notice that $\delta\frac{K}{\varepsilon}$ is an upper bound on the value that any solution can obtain from scenarios $\sigma_1 \cdots \sigma_n$. Then

$$E[value(\mathbf{b})] < (1-\delta)\left(K - \frac{1}{2}\right) + \frac{\delta K}{\varepsilon} = V - \frac{1-\delta}{2} + \frac{k}{n}\frac{\delta K}{\varepsilon} < V,$$

where the last inequality follows from the choice of δ. □

Claim 16 *Let $X = \{i \in L \mid \mathsf{b}_i \geq \alpha\}$. If H contains at least $(k+1)$ nodes in R that have neighbors in X, then $E[value(\mathbf{b})] < V$.*

Proof For a node $i \in R$, let $\alpha_i = \sum_{j\in N_i} \mathsf{b}_j$, where $N_i \subseteq L$ is the set of neighbors of i. Assuming that there are at least $(k+1)$ nodes $i \in R$ such that $\alpha_i \geq \alpha$, we show that $E[value(\mathbf{b})] < V$. The value of a solution is always maximized by bidding on all keywords in R, because that maximizes the number of cheap clicks. So without loss of generality, we assume that $\mathsf{b}_i = 1$ for all $i \in R$.

In a given scenario σ_i, the number of clicks available to $\mathbf{b}$ is $\frac{K}{\varepsilon} + \alpha_i t$, where K/ε clicks come from keyword i, and $\alpha_i t$ come from its neighbors in L. The total cost in this scenario is $K + \alpha_i t$, where K is the cost of the cheap clicks and $\alpha_i t$ is the cost of the expensive ones. Using $(1-\delta)K$ as an upper bound on the value obtained from σ_0, and remembering that the budget is K, we have

$$
\begin{aligned}
E[value(\mathbf{b})] &\le (1-\delta)K + \frac{K\delta}{n}\sum_{i\in R}\frac{K/\varepsilon+\alpha_i t}{K+\alpha_i t} \\
&\le (1-\delta)K + \frac{K\delta}{n}\left[(k+1)\cdot\frac{K/\varepsilon+\alpha t}{K+\alpha t} + (n-k-1)\cdot\frac{1}{\varepsilon}\right] \\
&< V,
\end{aligned}
$$

where we use the fact that the fraction in the sum increases with decreasing α_i, bound α_i by α for $(k+1)$ of the terms and by 0 for the others, and use the choice of t for the final inequality. □

Clearly, if there is no k-clique in G, then every K edges in G will be incident on at least $k+1$ nodes. So from the preceding two claims, we may conclude that if G does not contain a k-clique, then no solution to I has expected value of V or more. Together with Claim 14, this proves that SBO problem in the scenario model is NP-hard. □

Our second negative result for the scenario model rules out the possibility of obtaining an approximation by using a prefix solution.

Theorem 17 *The gap between the optimal fractional prefix solution and the optimal* (*integer or fractional*) *solution to the SBO problem in the scenario model can be arbitrarily large.*

Proof We give an example in which the ratio between the value of the optimal solution and the value of any prefix solution can be arbitrarily large. The example contains n scenarios and $2n$ keywords, numbered 1 through $2n$. The cost per click of keywords increases exponentially, with $\mathsf{cpc}_i = c^i$, for some constant $c > 1$. There is a budget $B > 0$. Say that the n scenarios are numbered $\sigma = 1$ to n. In scenario σ, only keywords $2\sigma - 1$ and 2σ receive clicks, and they receive $B/c^{2\sigma-1}$ clicks each. The probabilities of scenarios increase exponentially, and they are equal to $\alpha c^{2\sigma-1}$ for scenario σ (α is chosen to make the probabilities sum to 1). The idea here is that in each scenario, there are two types of clicks, cheap and expensive (clicks for the even-numbered keywords are c times more expensive than for their preceding odd-numbered keywords), and there are enough cheap clicks to spend the whole budget. So for a particular scenario σ, the best thing to do is to bid only on the cheap keyword $2\sigma - 1$, which gets $B/c^{2\sigma-1}$ clicks. Bidding on both keywords exceeds the budget and decreases the number of clicks to $\frac{2}{c+1}(B/c^{2\sigma-1})$. Since the sets of keywords that receive clicks in different scenarios are disjoint, the optimal solution overall (which happens to be integer) is to bid on all the odd-numbered keywords, but not the even-numbered ones. This gets the maximum number of clicks for each scenario individually, and therefore gets the maximum number of clicks in expectation. The expected number of clicks for the optimal solution is therefore

$$\sum_{\sigma}\alpha c^{2\sigma-1}\cdot\frac{B}{c^{2\sigma-1}} = n\alpha B.$$

Now consider some prefix solution for this example, either integer or fractional, and the keyword i^* such that $\mathsf{b}_i = 1$ for $i < i^*$ and $\mathsf{b}_i = 0$ for $i > i^*$. Let $\sigma^* = \lceil i^*/2 \rceil$ be the scenario containing clicks for keyword i^*. Intuitively, the prefix solution ruins the scenarios numbered less than σ^* because it bids for both keywords in them, and it ruins the scenarios numbered greater than σ^* because it does not bid at all for the keywords in them. As a result, a prefix solution can do well in at most one scenario. It gets the small number of clicks, $\frac{2}{c+1}(B/c^{2\sigma-1})$, for scenarios $\sigma < \sigma^*$, and it gets 0 clicks for scenarios $\sigma > \sigma^*$. It may get up to $B/c^{2\sigma^*-1}$ clicks for σ^*. So the value of a prefix solution is at most

$$\sum_{\sigma<\sigma^*} \alpha c^{2\sigma-1} \cdot \frac{2}{(c+1)} \frac{B}{c^{2\sigma-1}} + \alpha c^{2\sigma^*-1} \cdot \frac{B}{c^{2\sigma^*-1}} \le n\alpha B \left(\frac{2}{c+1} + \frac{1}{n} \right),$$

which can be arbitrarily far from $OPT = n\alpha B$ as c and n increase. □

6 Extensions

6.1 Extension to Click Values

Here we show that our results easily generalize to the case when the clicks from different keywords have different values to the advertiser. For example, a weight associated with a keyword might represent the probability that a user clicking on the ad for that keyword will make a purchase.

For each keyword i, we are given a weight w_i which is the value of a click associated with this keyword, and we would like to maximize the weighted number of clicks obtained:

$$E[value(\mathbf{b})] = E\left[\gamma(\langle \mathbf{clicks}, \mathbf{cpc}, \mathbf{b}\rangle) \cdot \langle \mathbf{w}, \mathbf{clicks}, \mathbf{b}\rangle\right].$$

Obviously, the keywords with $w_i = 0$ can be just discarded. Now we make a substitution of variables, defining $\mathsf{clicks}'_i = w_i \mathsf{clicks}_i$ and $\mathsf{cpc}'_i = \mathsf{cpc}_i / w_i$. Substituting them into the objective function,

$$E[value(\mathbf{b})] = E\left[\gamma(\langle \mathbf{clicks}', \mathbf{cpc}', \mathbf{b}\rangle) \cdot \langle \mathbf{clicks}', \mathbf{b}\rangle\right],$$

we see that the problem reduces to the original unweighted SBO instance, with different keyword parameters. The proportional, independent, and scenario models of click arrival maintain their properties under this transformation, only some of the distributions for the numbers of clicks have to be scaled.

6.2 Extension to Multiple-Slot Auctions

We now sketch the extension of our results to the case when there are multiple slots, and in particular, we assume the Generalized Second Price (GSP) auction currently used by search-related advertising engines. When advertising slots are allocated by a second-price auction with multiple slots, the bid amount for a keyword determines the position of the corresponding ads, which affects the number of clicks obtained for this keyword and the cost per click of these clicks. When a user clicks on the ad in slot i, the advertiser at slot i is charged the bid amount of the advertiser at slot $i+1$ (or, typically, a slightly larger amount).

For a particular keyword and a k-slot auction in which it participates, there are k threshold bid amounts that determine the position into which the ad is placed. This in turn determines the probability of a click (clickthrough rate) and the cost per click for this keyword, both of which can be viewed as monotone non-decreasing step functions of the bid amount. This information can be visualized on a plot called a *landscape* (a detailed description of landscapes appears in [4]). Landscapes for multiple auctions for the same keyword can be combined into an *aggregate* landscape.

Some of our results extend to the model with such aggregate landscapes. Roughly speaking, a keyword with a landscape can be viewed as a list of several simple individual keywords with an additional constraint that a solution has to bid on some prefix of this list. Choosing such a prefix, which possibly consists of more than one of these simple keywords, then translates into placing a single bid for the real keyword. For the fixed and proportional models, the optimal solutions without landscapes are prefix solutions anyway (by Theorems 2 and 3), so if we solve the problem as in the one-slot case, the solution will automatically satisfy the prefix constraint for keywords with landscapes, which means that it will also be optimal for the multiple-slot problem. In the independent model, however, the approximation ratio of 2 for the prefix solutions (Theorem 9), that we prove for the one-slot case, does not extend to the case of landscapes. This is because some of the "keywords" are no longer independent, but are actually the different bidding options for the same keyword. In fact, a prefix solution can be arbitrarily bad compared to the optimal solution, by an example that is very similar to one in Theorem 17. The only difference is that, instead of keywords $2i-1$ and $2i$ being coupled by occurring in the same scenario, they are coupled by representing the landscape of the same keyword. The negative results (Theorems 13 and 17) about the scenario model of course still hold for the more general case of multiple-slot auctions.

7 Concluding Remarks

We have initiated the study of stochastic version of budget optimization. We obtained approximation results via prefix bids and showed hardness results for other cases. A lot remains to be done, both technically and conceptually. Technically, we need to extend the results to the case when there are interactions between keywords, that is, two or more of them apply to a user query and some resolution is needed. Also, we need to study online algorithms, including online budget optimization. Further, we would like to obtain some positive approximation results for the scenario model, which seems quite intriguing from an application point of view. The conceptual challenge is one of modeling. Are there other suitable stochastic models for search-related advertising, that are both expressive, physically realistic and computationally feasible?

Acknowledgements We thank Jon Feldman, Anastasios Sidiropoulos, and Cliff Stein for helpful discussions.

References

1. Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. **12**(2), 479–502 (2002)
2. Charikar, M., Chekuri, C., Pal, M.: Sampling bounds for stochastic optimization. In: Proc. 9th International Workshop on Randomization and Computation (2005)
3. Swamy, C., Shmoys, D.B.: Approximation algorithms for 2-stage stochastic optimization problems. SIGACT News **37**(1), 33–46 (2006)
4. Feldman, J., Muthukrishnan, S., Pal, M., Stein, C.: Budget optimization in search-based advertising auctions. In: Proc. 9th ACM Conf. on Electronic Commerce (2007)
5. Dean, B.C., Goemans, M.X., Vondrak, J.: Approximating the stochastic knapsack problem: the benefit of adaptivity. In: Proc. 45th IEEE Symp. on Foundations of Computer Science, pp. 208–217 (2004)
6. Kleinberg, J., Rabani, Y., Tardos, E.: Allocating bandwidth for bursty connections. SIAM J. Comput. **30**(1), 191–217 (2000)
7. Goel, A., Indyk, P.: Stochastic load balancing and related problems. In: Proc. 40th IEEE Symp. on Foundations of Computer Science (1999)
8. Carraway, R.L., Schmidt, R.L., Weatherford, L.R.: An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. Nav. Res. Logist. **40**, 161–173 (1993)
9. Henig, M.I.: Risk criteria in a stochastic knapsack problem. Oper. Res. **38**(5), 820–825 (1990)
10. Sniedovich, M.: Preference order stochastic knapsack problems: methodological issues. J. Oper. Res. Soc. **31**(11), 1025–1032 (1980)
11. Steinberg, E., Parks, M.: A preference order dynamic program for a knapsack problem with stochastic rewards. J. Oper. Res. Soc. **30**(2), 141–147 (1979)
12. Chakrabarty, D., Zhou, Y., Lukose, R.: Budget constrained bidding in keyword auctions and online knapsack problems. In: WWW2007 Workshop on Sponsored Search Auctions (2007)
13. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: A knapsack secretary problem with applications. In: Proc. 10th APPROX (2007)
14. Rusmevichientong, P., Williamson, D.P.: An adaptive algorithm for selecting profitable keywords for search-based advertising services. In: Proc. 8th ACM Conf. on Electronic Commerce, pp. 260–269 (2006)
15. Ostrovsky, M., Edelman, B., Schwarz, M.: Internet advertising and the generalized second price auction: selling billions of dollars worth of keywords. Am. Econ. Rev. **97**(1) (2007)
16. Aggarwal, G., Goel, A., Motwani, R.: Truthful auctions for pricing search keywords. In: Proc. 8th ACM Conf. on Electronic Commerce, pp. 1–7 (2006)
17. Borgs, C., Chayes, J., Immorlica, N., Mahdian, M., Saberi, A.: Multi-unit auctions with budget-constrained bidders. In: Proc. 7th ACM Conf. on Electronic Commerce, pp. 44–51 (2005)
18. Borgs, C., Chayes, J., Etesami, O., Immorlica, N., Jain, K., Mahdian, M.: Bid optimization in online advertisement auctions. In: 16th International World Wide Web Conference (2007)
19. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. J. ACM **54**(5) (2007)
20. Mahdian, M., Nazerzadeh, H., Saberi, A.: Allocating online advertisement space with unreliable estimates. In: Proc. 9th ACM Conf. on Electronic Commerce (2007)