Bid Optimization for Broad Match Ad Auctions

Eyal Even Dar
Google Research
76 9th Ave
New York, NY 10011
evendar@google.com

Vahab S. Mirrokni
Google Research
76 9th Ave
New York, NY 10011
mirrokni@google.com

S. Muthukrishnan
Google Research
76 9th Ave
New York, NY 10011
muthu@google.com

Yishay Mansour
Google Research and Tel-Aviv University
76 9th Ave
New York, NY 10011
muthu@google.com

Uri Nadav[*]
Tel-Aviv University
Tel-Aviv, 69978
uri.nadav@gmail.com

ABSTRACT

Ad auctions in sponsored search support "broad match" that allows an advertiser to target a large number of queries while bidding only on a limited number. While giving more expressiveness to advertisers, this feature makes it challenging to optimize bids to maximize their returns: choosing to bid on a query as a broad match because it provides high profit results in one bidding for related queries which may yield low or even negative profits.

We abstract and study the complexity of the *bid optimization problem* which is to determine an advertiser's bids on a subset of keywords (possibly using broad match) so that her profit is maximized. In the query language model when the advertiser is allowed to bid on all queries as broad match, we present a linear programming (LP)-based polynomial-time algorithm that gets the optimal profit. In the model in which an advertiser can only bid on keywords, ie., a subset of keywords as an exact or broad match, we show that this problem is not approximable within any reasonable approximation factor unless P=NP. To deal with this hardness result, we present a constant-factor approximation when the optimal profit significantly exceeds the cost. This algorithm is based on rounding a natural LP formulation of the problem. Finally, we study a budgeted variant of the problem, and show that in the query language model, one can find two budget constrained ad campaigns in polynomial time that implement the optimal bidding strategy. Our results are the first to address bid optimization under the broad match feature which is common in ad auctions.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*; H.4 [**Information Systems Applications**]: Miscellaneous

[*]Part of this research was conducted while visiting Google Research.

General Terms

Algorithm, Theory, Economics

Keywords

Sponsored Search, Ad Auctions, Optimal Bidding, Bid Optimization

1. INTRODUCTION

Sponsored search is a large and thriving market with three distinct players. *Users* go to search engines such as Yahoo! or Google and pose queries; in the process, they express their intention and preferences. *Advertisers* seek to place advertisements and target them to users' intentions as expressed by their queries. Finally, *search engines* provide a suitable mechanism for doing this. Currently, the mechanism relies on having advertisers bid on the search issued by the user, and the search engine to run an *auction* at the time the user poses the query to determine the advertisements that will be shown to the user. As is standard, the advertiser only pays 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.

In this paper, we assume the perspective of the advertiser. The advertisers need to target their ad campaigns to users' queries. Thus, they need to determine the set S of queries of their interest. Once that is determined, they need to strategize in the auction that takes place for each of the queries in S. A lot of research has focused on the game theory and optimization behind these auctions, both from the search engine [1, 16, 6, 2, 10, 4] and advertiser [3, 8, 5, 11] points of view. There has been relatively little prior research on how advertisers target their campaign, i.e., how they determine the set S.

The criterion for choosing S is for the advertiser to pick a set of *keyphrases* that searchers may use in their query when looking for their products. The central challenge then is to match the advertisers keyphrases with the potential queries issued by the users. It is difficult if not impossible for the advertisers to identify all possible variations of keyphrases that a user looking for their product may use in their query. As an example, consider a vendor who chooses the keyphrase *tennis shoes*. Users searching for them may use singular or plural, synonyms and other variations ("clay court footwear"), may misspell ("tenis shoe"), use extensions ("white tennis shoes") or reorder the words ("shoes lawn tennis"). In fact, users may even search using words not found in the keyphrase ("Wimbledon gear", "US Open Shoes", "hard court soles"), and may still be of interest to the advertiser. These artifacts such as plurals, synonyms, misspellings, extensions, and reorderings are very common, and the problems get compounded since typical ad campaigns comprise several keyphrases, each with its own set of artifacts.

Major search engines help advertisers address this challenge by providing a structured bidding language. While the specific details differ from search engine to search engine [17, 20, 19], at the highest level, the bidding language supports two *match types*: exact and broad. In *exact* matchtype (called "exact" in MSN AdCenter and Google, and "standard" in Yahoo), ad would be eligible to appear when a user searches for the specific keyphrase without any other terms in the query, and words in the keyphrase need to appear in that order. In *broad* matchtype (called "broad" in MSN, related to "phrase" and "broad" in Google, and "advanced match type" in Yahoo), the system automatically makes advertisers eligible on relevant variations of their keyphrases including for the various artifacts listed earlier, even if the search terms are not in the keyphrase lists. Thus, the search engines automate the aspect of detecting artifacts and matching the query to keyphrases of interest to advertisers.[1] Thus the task of advertisers becomes determining the keyphrases and choosing the match type on each.

The question we address here is, how does an advertiser bid in presence of these match types? Say each query q has a value $v(q)$ per click for the advertiser that is known to the advertiser and is private. Further, we let $c(q)$ be the expected price per click and let $n(q)$ be the expected number of clicks. These are statistical estimates provided by the search engines [18, 23, 21]. Then, we consider two optimization problems: (i) in one variant, we assume that the advertiser wishes to maximize their *expected profit*, that is, $\sum_q (v(q) - c(q))n(q)$, and (ii) in the other variant, given a budget B for the advertiser, we assume that the advertiser wishes to maximize their *expected value*, that is, $\sum_q v(q)n(q)$ subject to the condition that the expected spend $\sum_q c(q)n(q)$ does not exceed the budget.

The technical challenge arises due to *query dependencies*. When one bids on a keyphrase for query q, as a result of a broad match, it may apply to query q' as well. The advertiser has different values $v(q)$ and $v(q')$ on these because users for q and q' differ on their intentions and therefore on their respective values to the advertiser. So, the advertiser may make good profit on q and may wish to bid on that query, but is then forced to implicitly bid on q' as well, and may even make negative profit on q'! Under what circumstances is it now desirable for the advertiser to bid for q?

Note that query dependence is a fundamental aspect of sponsored search since advertisers can realistically only choose and strategize on a small set of keyphrases because of the effort involved, and have to typically rely on the search engine to carefully apply their strategy to variants of their keyphrases. But beyond that, even an ad campaign that is willing to exert a lot of effort and use a large number of keyphrases or relies on a search engine to provide rich bidding languages [9] will still find it impossible to include all search variations of the keyphrases as exact matches, and must necessarily rely on broad match for the variations that search users develop and prefer over time. Thus, the advertisers bid implicitly on queries on which they can not directly control the tradeoff between the cost and the value.

Query dependence introduces a complex optimization problem of trading off the benefits of bidding on a keyphrase against the impact of bidding on its dependent queries. In the sponsored search world, there is a keen awareness of this complexity of bidding, and most search engines and third-party bidding agents provide detailed tips and guidelines for advertisers [24, 22]. Beyond these guidelines, what is missing is a clear theoretical understanding of the tradeoffs and the complexity of the bidding problem that advertisers face.

We initiate principled study of bidding in presence of broad matches. Specifically, our contributions are as follows.

1. We abstract two models — query and keyword language models — to study bidding optimization problems.

   In the query language model, the advertiser bids directly on user queries and wishes to determine which query if any to bid on, to maximize expected profit. This models both the theoretical extreme where an advertiser can bid on any of the queries the search engine will see, and the practical reality where the advertiser has a select set of queries in mind and wishes only to optimize within that set. In the keyword language model, advertisers may bid only on a subset of queries, and broad match implicitly derives bids as needed. This directly models the common reality.

2. We present efficient, polynomial time algorithms for the bid optimization problem under these two models.

   In query bidding, we get a polynomial-time algorithm that maximizes the profit, using a reduction to the well-known Min-Cut problem in graphs. This is in contrast to the poor performance of natural greedy algorithms for this problem. We also study the budgeted variant of the problem, and propose a novel strategy using *two* distinct budgeted ad campaign that gets the optimal profit. We do so by studying the structure of the basic feasible solutions of a corresponding linear programming formulation of the problem.

   For keyword bidding, we show that even limited instances are NP-Hard to not only optimize, but even to approximate; to deal with this hardness result, we present a constant-factor approximation when advertisers profit following an optimal bid is considerably greater than her cost. This result is based on applying a randomized rounding method on the optimal fractional solutions of the linear programming relaxation of the problem.

These represent the first known theoretical results for the problem of bid optimization in presence of broad matches, a problem advertisers face now since this feature is offered

[1]These match types may be further modified by ensuring that the ad be *not* shown on occurrence of certain keywords in the query; this feature (called "negative" in MSN and Google or "excluded" by Yahoo) and other targeting criteria associated with keyphrase campaigns do not change the discussion and the results here.

by the major search engines. Prior research in bid optimization for advertisers [3, 5, 13] primarily focused on determining suitable bids for exact match types and does not study the query dependence and implicit bids; [8, 11] studied the problem of maximizing the number of clicks, and not the profit which is the more standard metric. At the technical core, our challenge is to tradeoff positive profit from bidding on a keyphrase that applies to one query q against possibly a negative profit from the implied bids of broad match on queries q'. This query dependence is a novel feature in sponsored search auctions, not explicitly studied in prior literature, and our results for this problem may have applications beyond, in the general auction theory area.

Finally, we report experimental results on a small family of instances of the bid optimizations problem, and compute the optimal bidding using the integer linear programming formulation. Our main observation in these experiments is that by considering only the broad match, we do not lose much in the maximum profit of the solution. This supports our hope that under reasonable circumstances (similar to the ones in our experiments), considering only broad match is effective, and in turn, that would enable advertisers to focus on campaigns with small lists of keyphrases.

2. MODEL

We consider the optimization problems that an advertiser faces while bidding in an auction for queries with a broad match feature.

The Advertiser. We consider a single advertiser who is interested in showing her ad to users after they search for queries from a set Q. The advertiser has some utility from having a user click on her ad. In reality, clicks associated with different queries may have different utility to the advertiser; The advertiser has a value of $v(q)$ units of monetary value associated with a 'click' that follows a query $q \in Q$.

We assume a posted price model where prices are posted and the search volume of every query as well as its click through rate (i.e., the probability that users would click her ad) are known to the advertiser. Namely, every query q is associated with a pair of parameters, known to the advertiser, $(c(q), n(q))$, where $c(q)$ is the per click cost of q, and $n(q)$ is the expected number of clicks that would result from winning q (the expected number of clicks can be determined from the search volume of q and the advertiser's specific click through rate for q).

Thus, when an advertiser wins a query q, her overall profit [2] from winning, denoted $w(q)$ is

$$w(q) = (v(q) - c(q))\, n(q)\ .$$

Note that although each query has a positive value, winning it may result in an overall negative profit.

Bidding languages. A bidding language is a way for an advertiser to specify her value or willingness to pay for queries. Eventually, the auctioneer needs to have a bid for every possible query [3]. The choice of a bidding language is critical for the auction mechanism. At the one extreme, it may be infeasible to allow an advertiser to specify explicitly her value for every possible query. On the other hand, a language that is too restrictive would not allow an advertiser to communicate her preferences properly.

In order to study the complexity of the optimal bidding in the broad match framework while taking into account the intersections among broad matches for different keywords, we first consider a bidding language in which an advertiser can specify a bid for every query q but only as a broad-match. We refer to this language as the *query language.*

To allow the most accurate description of an advertisers value per query, the ultimate way is to let the advertiser specify all possible queries with exact or broad match, and a monetary bid for each of them. If an advertiser is allowed to bid on each type of query as an exact match as well as broad match, she can decide for each query independent of the other queries, and the complexity of the bidding problem is not captured in such a bidding language.

To capture the complexity of the optimal bidding problem and the fact that advertisers may only bid on a subset of queries, we study the *keyword language* that allows advertisers to place a bid only on (single) keywords or short phrases. More precisely, in the keyword language, we assume that advertisers are allowed to bid only on a subset $S \subset Q$ of queries.

A further improvement of this language would allow the advertiser to specify, besides a value bid for $s \in S$, whether s is to be matched exactly or broadly.

A bid $b \in \mathbb{R}_+^{|Q|}$ in some bidding language is associated with a set of 'winning queries' denoted by $\varphi(b) = \{q \in Q \mid b(q) \geq c(q)\}$. A subset T of queries which is a winning set of some bid b is referred to as a *feasible winning set.* The utility associated with a winning set T is

$$u(T) = \sum_{q \in T} (v(q) - c(q))\, n(q),$$

where $v(\cdot)$ and $n(\cdot)$ are advertiser specific.

A feasible winning set with optimal utility is referred to as an optimal winning set.

The Auction. For every query, the auctioneer should decide the bid of every advertiser. This decision is easy for queries on which the advertiser bids explicitly (as an exact match). However, for the queries that the advertiser has not bid directly, but only through a broad match framework, the auctioneer should compute an appropriate bid for the advertiser to participate in the auction.

A natural way for setting such a value is to aggregate the bid values of all the phrases matched by the query. While there are several choices for the aggregation method, in this paper, we consider the max aggregation operator — when a query q matches phrases $w_1, \ldots, w_k$ (as a broad match) from the advertiser list of phrases, its bid is interpreted as $b(q) = \max_i b(w_i)$.

We can now state formally the bid optimization problem. Given advertiser's specific data (A set Q, value for queries v, search volume and click through rates $n(\cdot)$) and a bidding language $\mathcal{L}$, an optimal bid b^*, is a feasible bid in the language $\mathcal{L}$ that maximizes the advertisers' utility from winning a set $\varphi(b)$ of queries. Formally,

$$b^* \in argmax_{b \in \mathcal{L}} \{u(\varphi(b))\}. \qquad (2.1)$$

[2]In this paper, we use terms utility and profit interchangably.

[3]A bid of 0 for a query may be regarded as the default in a case where the advertiser is not explicitly interested in a query q and nor in queries that q match broadly.

Query dependencies. We say that a query q depends on a query q' if winning query q' implies winning query q. In the broad match auction in which the bid interpretation strategy is done using the max operator, this happens if q matches q' broadly, and its cost $c(q)$ is less than that of $c(q')$. In other words, if a bid b wins q', it must be that $b(q') \geq c(q')$, but the interpreted bid for q is then at least $b(q') \geq c(q)$ since $c(q') \geq c(q)$, hence the bid b must be winning q as well. As a result, the cost structure incurs a set of pairs (q', q) where the first entry of each pair $q' \in S$ is a valid phrase in the bidding language and the second entry is a valid query in the set of queries Q such that winning query q' implies winning query q. This set of pairs is denoted by $\mathcal{C}$ and formally:

$$\mathcal{C} = \{(q', q) | q' \in S, q \in Q, q \text{ matches } q' \text{ broadly }, c(q') \geq c(q)\}.$$

Moreover, we define $D(q) = \{q' | (q', q) \in \mathcal{C}\}$, and $N(q) = \{q' | (q, q') \in \mathcal{C}\}$.

Budget-constrained Ad Campaigns. A variant of the optimal bidding problem in the broad match framework is to find a set of queries to bid on that maximizes the total value of the queries won by the advertiser subject to a budget constraint, i.e, our goal is to bid on a subset T of queries to maximize $\sum_{q \in T} v(q)n(q)$ subject to the budget constraint $\sum_{q \in T} c(q)n(q) \leq B$. To handle such a budget constraint, we assume that one can run a *budget-constrained ad campaign* by bidding on a subset T of keywords and setting a budget B. Assuming $B' = \sum_{q \in T} c(q)n(q)$, there are two possibilities in this budget-constrained ad campaign: (i) If $B' \leq B$, the auction is run in a normal way and the value from this ad campaign for the advertiser is $\sum_{q \in T} v(q)n(q)$, (ii) On the other hand, if $B' > B$, we assume that the queries arrive at the same rate and as a result, for each query, we get $\frac{B'}{B}$ fraction of the value of an ad campaign without the budget constraint. In other words, the value that the advertiser gets is $\sum_{q \in T} v(q)n(q)\frac{B'}{B}$. We can also interpret the above assumption by a throttling method in which, in order to cope with the budget constraint, at each step, we let the advertiser participate in the auction with probability $\frac{B'}{B}$.

3. BIDDING IN THE QUERY LANGUAGE

In this section, we study the query language that allows placing a bid on every query. We observe that in the query language, the task of computing an optimal bid is equivalent to that of computing an optimal winning set: Given an optimal feasible set T set a bid $b(q) = c(q)$ for every query $q \in T$ with positive weight and $b(q) = 0$ otherwise.

LEMMA 3.1. *A bid b derived from an optimal winning set T, as described above, is an optimal bid.*

PROOF. By construction, the bid b wins all the queries with positive weight from T, and every other query must belong to T (otherwise T would not be feasible). $\square$

We therefore consider algorithms for computing an optimal feasible winning set. First, we consider a greedy algorithm, denoted by *Max-Margin Greedy.* Initially, *Max-Margin Greedy* sets the winning set to be empty. Then, iteratively, it adds a bid on a query with the highest marginal benefit to the winning set utility. Unfortunately, *Max-Margin Greedy* fails to compute an optimal winning set due to the following example.

Example. Consider a set Q of queries which contains n keywords and another $\binom{n}{2}$ queries, each of which is a pair of keywords. The cost of each query is set to \$1. Hence, the query dependencies is such that winning a keyword implies winning the set of $n-1$ queries made of pairs of keywords in which this keyword appears. The value of a keyword is set to \$2; The value of a each pair is set to $1 - 1.5/n$. So, every keyword attains a positive utility of \$1, and every pair causes a loss of \$$1.5/n$. Initially, *Max-Margin Greedy*'s bid is empty. At this point *Max-Margin Greedy* is stuck — every single query it adds to the winning set results in a negative overall utility. Thus, this instance, *Max-Margin Greedy* would yield 0 utility. An optimal solution wins all the queries and has a utility of $n \times (2-1) - \binom{n}{2}(1 - \frac{1.5}{n} - 1) = \frac{n-3}{4}$.

One can explore other variants of greedy algorithms for this problem. For example, a natural greedy algorithm is *Max-Rate Greedy* algorithm: Initially set the winning set to the empty set, and then iteratively, add a bid on a query with the highest ratio of marginal profit over the marginal cost, or the query with the highest ratio of marginal value over marginal cost. We note that all these iterative greedy algorithms pefrom poorly for the above example. Even a significant look-ahead will not resolve this bad example.

We turn to the next algorithm *OptBid1* for computing an optimal winning set. *OptBid1* is a solution to the following integer linear program:

$$\begin{aligned} \text{ILP} \quad &: \quad \max \sum_{q_i \in Q} X_{q_i} w(q_i) \\ \text{For every pair } (q_j, q_i) \in \mathcal{C} \quad &: \quad X_{q_i} - X_{q_j} \geq 0 \\ \forall q_i \in Q \quad &: \quad X_{q_i} \in \{0, 1\} \qquad (3.1) \end{aligned}$$

For every query q, an integral variable X_q is a 0-1 variable which is equal to 1 if and only if q belongs to the winning set of queries. In order to solve the above ILP, we relax it to a linear program where instead of integer 0-1 variables, we have fractional X_q variables with values between 0 and 1 ($0 \leq X_q \leq 1$). Here, we observe that the integrality gap of this linear programming relaxation is 1, i.e., for any instance of this linear program, there exists an optimal solution X^* in which all the values are integer $X^*_q \in \{0, 1\}$ for all $q \in Q$.

LEMMA 3.2. *The integrality gap of the linear programming relaxation of the ILP 3.1 is one.*

PROOF. The lemma follows from the fact that the constraint matrix of the LP relaxation of ILP 3.1 is totally uni-modular[4]. A sufficient condition for a matrix to be totally uni-modular is that every row has either two non-zero entries, one is 1 and the other -1, or a single non zero entry with value 1 or -1. An integer program whose constraint matrix is totally uni-modular and whose right hand side is integer can be solved by linear programming since all its basic feasible solutions are integer (see [12] pp. 316). $\square$

The above lemma implies the following polynomial-time algorithm *OptBid1* for optimal bidding in the query language: compute a basic feasible solution X^* of the LP relaxation of ILP 3.1, and find a bidding strategy corresponding to the winning set of X^*, i.e., $\{q \in Q | X^*_q = 1\}$.

[4]A matrix A is totally uni-modular if every square submatrix of it is uni-modular, i.e., every submatrix has a determinant of 0, -1 or +1.

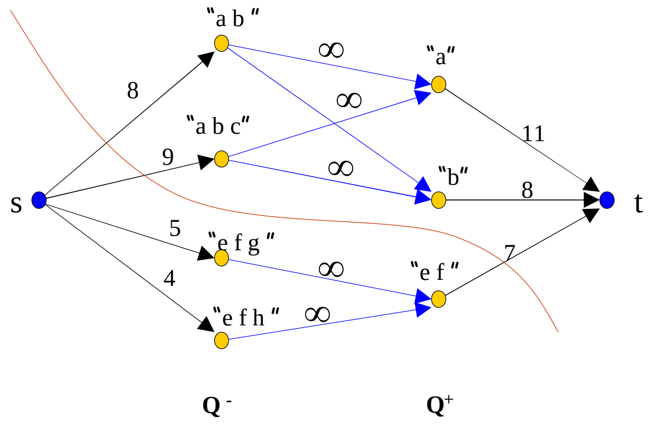

Figure 1: An example of running algorithm OptBid2 on the set of queries (with the following profit: $\{(a, 11), (b, 8), (ab, -8), (abc, -9), (ef, 7), (efg, -5), (efh, -4)\}$), and dependency graph as illustrated. ObtBid2 will choose the winning set $\{a, b, ab, abc\}$. The optimal bid is $\{(a, 11), (b, 8)\}$.

The running time of Algorithm *OptBid1* is that of solving a linear program with $\Omega(|Q|^2)$ constraints, which although polynomial in $|Q|$, might be inefficient. Next, we present a faster algorithm, *OptBid2*.

For the purpose of presenting Algorithm *OptBid2*, we define a weighted flow graph $G = (V, E)$, derived from the input. The vertex set of G is $V = \{s, t\} \cup Q^+ \cup Q^-$, where s is a source node, t is a target node and Q^+ and Q^- are the sets of queries with positive/non-positive weights respectively, *i.e.*, $Q^+ \equiv \{q \mid w(q) > 0\}$. The source vertex s is connected to each vertex $q \in Q^-$ with an edge of weight $|w(q)| = |(v(q) - c(q))n(q)|$. The target vertex t is connected with each vertex $p \in Q^+$ with an edge of weight $w(p)$. Two vertices $q \in Q^-, p \in Q^+$ are connected with an edge of weight ∞ if and only if $(p, q) \in \mathcal{C}$.

Algorithm *OptBid2*

1. Compute a min-cut of G. Let S, T be the two sides of the cut.
2. Assume, without loss of generality, that $t \in T$. Return $T \setminus \{t\}$, that is, the set of queries that are on the same side of the cut as t is an optimal winning set.

The running time of Algorithm *OptBid2* is that of min-cut, *i.e.*, $O(|Q|^3)$ [14].

THEOREM 3.3. *Algorithm* OptBid2 *finds an optimal winning set.*

PROOF. We show T is an optimal winning set using the dual program of ILP.

$$\begin{aligned}
\text{DUAL} &: \min \sum_{q \in Q^+} Z_q \\
\forall q \in Q^+ &: \sum_{q' : (q, q') \in \mathcal{C}} Y_{q,q'} + Z_q \geq w(q) \\
\forall q' \in Q^- &: \sum_{q : (q, q') \in \mathcal{C}} Y_{q,q'} \leq -w_{q'} \text{ (Notice that } w_{q'} \leq 0) \\
\forall (q, q') \in \mathcal{C} &: Y_{q,q'} \geq 0 \\
\forall q \in Q^+ &: Z_q \geq 0
\end{aligned}$$

Let $f = (f_e)_{e \in E}$ be a maximum flow in G, with value c. For every $(q, q') \in \mathcal{C}$, set $Y_{q,q'} := f_{q,q'}$ and for every $q \in Q^+$, set $Z_q := w(q) - \sum_{q' | (q,q') \in \mathcal{C}} Y_{q,q'}$. It is straightforward to verify that this is a feasible solution of DUAL with value $\sum_{q \in Q^+} w(q) - c$.

Now, observe that T is a feasible set in the query language. For every pair $(q, q') \in \mathcal{C}$, we have that if $q \in T$ then also $q' \in T$. Otherwise, the edge (q, q'), with weight ∞, would be part of the cut. Thus, the value of the min cut is

$$c = \sum_{q \in Q^- \cap T} |w(q)| + \sum_{q \in Q^+ \setminus T} w(q).$$

and therefore,

$$\begin{aligned}
u(T) &= \sum_{q \in T} w(q) = \sum_{q \in Q^+} w_q - \sum_{q \in Q^+ \setminus T} w(q) + \sum_{q \in Q^- \cap T} w(q) \\
&= \sum_{q \in Q^+} w_q - c.
\end{aligned}$$

We already found a feasible solution for the dual of ILP, with the same value. We therefore conclude, using the weak duality theorem, that T is an optimal solution of ILP. □

4. BIDDING IN THE KEYWORD LANGUAGE

In this section, we study optimal bidding for the keyword language, where the advertiser is restricted to bid on a subset of (possibly short) queries $S \subset Q$.

Note that in the case that all queries have positive utility, the optimal bid is trivial by simply placing a high bid for every query in S. In addition, finding the optimal bid when all queries are associated with a negative utility is trivial (a bid of \$0 for every phrase in S is optimal). Moreover, in the case of uniform value from every query, the optimal bid is easy — a uniform bid equal to the uniform value guarantees winning every query with positive weight and losing every query with negative weight, which is of course optimal. In realistic settings, some queries have positive utility and some have negative utility. In this case the problem of finding the optimal bid becomes intractable. More precisely, as we show now, even when the set of queries Q is made up from single keywords and pairs of keywords, this problem becomes hard to approximate within a factor of $|S|^{1-\epsilon}$, for every $\epsilon > 0$:

THEOREM 4.1. *In the keyword language broad match framework, it is NP-hard to approximate the optimal value of the optimal bidding problem within a factor of $|S|^{1-\epsilon}$, for any $\epsilon > 0$.*

PROOF. We give a factor-preserving reduction to the independent set problem. Given a graph G with n nodes, and m edges, we construct the following instance of our problem: put a singleton keyword for each node v of G with weight $w_v = -(deg(v) - 1)$, and put a query consisting of a pair of keywords corresponding to each edge e of G with weight 1. The maximum value we can get from picking a keyword is 1, and we get this value if all of its neighbors do not appear in the output. It can be seen that the optimum solution is an independent set of nodes (since otherwise, we get zero or negative from a picked node), and as a result, the maximum value is the same as the size of the independent set. □

4.1 A Constant-Factor Approximation

In this section, in light of the above hardness result, we design a constant-factor approximation algorithm for a special