

# On Revenue Maximization in Second-Price Ad Auctions <sup>\*</sup>

Yossi Azar<sup>1</sup>, Benjamin Birnbaum<sup>2</sup>, Anna R. Karlin<sup>2</sup>, and C. Thach Nguyen<sup>2</sup>

<sup>1</sup> Microsoft Research and Tel-Aviv University  
azar@tau.ac.il

<sup>2</sup> University of Washington  
<birnbaum,karlin,ncthach>@cs.washington.edu

**Abstract.** Most recent papers addressing the algorithmic problem of allocating advertisement space for keywords in sponsored search auctions assume that pricing is done via a first-price auction, which does not realistically model the Generalized Second Price (GSP) auction used in practice. Towards the goal of more realistically modeling these auctions, we introduce the *Second-Price Ad Auctions* problem, in which bidders' payments are determined by the GSP mechanism. We show that the complexity of the Second-Price Ad Auctions problem is quite different than that of the more studied First-Price Ad Auctions problem. First, unlike the first-price variant, for which small constant-factor approximations are known, it is NP-hard to approximate the Second-Price Ad Auctions problem to any non-trivial factor. Second, this discrepancy extends even to the 0-1 special case that we call the *Second-Price Matching* problem (2PM). In particular, offline 2PM is APX-hard, and for online 2PM there is no deterministic algorithm achieving a non-trivial competitive ratio and no randomized algorithm achieving a competitive ratio better than 2. This stands in contrast to the results for the analogous special case in the first-price model, the standard bipartite matching problem, which is solvable in polynomial time and which has deterministic and randomized online algorithms achieving better competitive ratios. On the positive side, we provide a 2-approximation for offline 2PM and a 5.083-competitive randomized algorithm for online 2PM. The latter result makes use of a new generalization of a classic result on the performance of the “Ranking” algorithm for online bipartite matching.

## 1 Introduction

The rising economic importance of online sponsored search advertising has led to a great deal of research focused on developing its theoretical underpinnings. (See, e.g., [1] for a survey). Since search engines such as Google, Yahoo! and MSN depend on sponsored search for a significant fraction of their revenue, a key problem is how to optimally allocate ads to keywords (user searches) so as to maximize search engine revenue [2–11]. Most of the research on the dynamic version of this problem assumes that once the participants in each

---

<sup>\*</sup> This research was supported by NSF Grant CCF-0635147, a grant from Yahoo! Research, and an NSF Graduate Research Fellowship.

keyword auction are determined, the pricing is done via a first-price auction; in other words, bidders pay what they bid. This does not realistically model the standard mechanism used by search engines, called the Generalized Second Price mechanism (GSP) [12, 13].

In an attempt to model reality more closely, we study the *Second-Price Ad Auctions* problem, which is the analogue of the above allocation problem when bidders’ payments are determined by the GSP mechanism. As in other work [4–6, 10, 11], we make the simplifying assumption that there is only one slot for each keyword. In this case, the GSP mechanism for a given keyword auction reduces to a second-price auction – given the participants in the auction, it allocates the advertisement slot to the highest bidder, charging that bidder the bid of the second-highest bidder.<sup>1</sup>

In the Second-Price Ad Auctions problem, there is a set of keywords  $U$  and a set of bidders  $V$ , where each bidder  $v \in V$  has a known daily budget  $B_v$  and a non-negative bid  $b_{u,v}$  for every keyword  $u \in U$ . The keywords are ordered by their arrival time, and as each keyword  $u$  arrives, the algorithm (i.e., the search engine) must choose a bidder to allocate it to. The search engine is not required to choose the highest-bidding bidder; in order to optimize the allocation of bidders to keywords, search engines typically use a “throttling” algorithm that chooses which bidders to select to participate in an auction for a given keyword [7].<sup>2</sup>

In the previously-studied first-price version of the problem, allocating a keyword to a bidder meant choosing a single bidder  $v$  and allocating  $u$  to  $v$  at a price of  $b_{u,v}$ . In the Second-Price Ad Auctions problem, two bidders are selected instead of one. Of these two bidders, the bidder with the higher bid (where bids are always reduced to the minimum of the actual bid and bidders’ remaining budgets) is allocated that keyword’s advertisement slot at the price of the other bid. (In the GSP mechanism for  $k$  slots,  $k + 1$  bidders are selected, and each of the top  $k$  bidders pays the bid of the next-highest bidder.)

This process results in an allocation and pricing of the advertisement slots associated with each of the keywords. The goal is to select the bidders participating in each auction to maximize the total profit extracted by the algorithm. For an example instance of this problem, see Figure 1.

## 1.1 Our Results

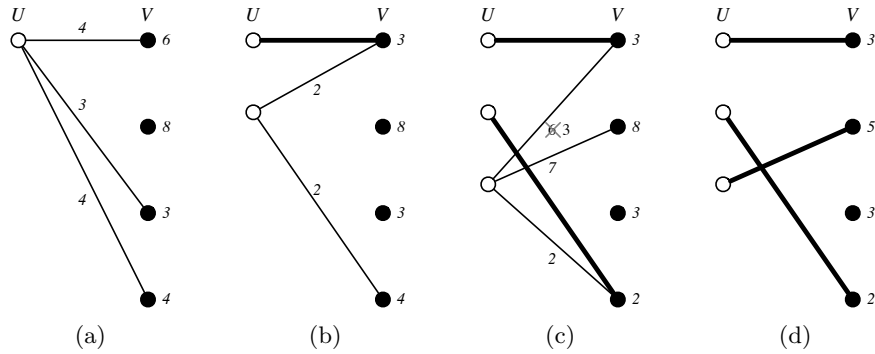
We begin by considering the *offline* version of the Second-Price Ad Auctions problem, in which the algorithm knows all of the original bids of the bidders (Section 3). Our main result here is that it is NP-hard to approximate the optimal solution to this problem to within a factor better than  $\Omega(m)$ , where  $m$  is the number of keywords, even when the bids are small compared to budgets.

<sup>1</sup> This simplification makes 2PAA not a complete model for sponsored search. A second price auction for a single slot is in fact a Vickery auction, but this is not true in the case of multi-slot GSP auction. However, 2PAA is still much nearer to the practice than the previously studied first price models. Also, our hardness results clearly hold for the multi-slot case as well

<sup>2</sup> In this paper, we assume the search engine is optimizing over revenue although it is certainly conceivable that a search engine would consider other objectives.

This strong inapproximability result is matched by the trivial algorithm that selects the single keyword with the highest second-best bidder and allocates only that keyword to its top two bidders. It stands in sharp contrast to the standard First-Price Ad Auctions problem, for which there is a  $4/3$ -approximation to the offline problem [6, 11] and an  $e/(e - 1)$ -competitive algorithm to the online problem when bids are small compared to budgets [5, 10].

We then turn our attention to a theoretically appealing special case that we call *Second-Price Matching*. In this version of the problem, all bids are either 0 or 1 and all budgets are 1. This can be thought of as a variant on maximum bipartite matching in which the input is a bipartite graph  $G = (U \cup V, E)$ , and the vertices in  $U$  must be matched, in order, to the vertices in  $V$  such that the profit of matching  $u \in U$  to  $v \in V$  is 1 if and only if there is at least one additional vertex  $v' \in V$  that is a neighbor of  $u$  and is unmatched at that time.



**Fig. 1.** An example of the Second-Price Ad Auctions problem: The nodes in  $U$  are keywords and the nodes in  $V$  are bidders. The number immediately to the right of each bidder represents its remaining budget, and the number next to each edge connecting a bidder to a keyword represents the bid of that bidder for that keyword. Figure 1(a) shows the situation when the first keyword arrives. For this keyword, the search engine selects the first bidder, whose bid is 4, and the third bidder, whose bid is 3. The keyword is allocated to the first bidder at a price of 3, thereby reducing that bidder's budget by 3. Figure 1(b) shows the situation when the second keyword arrives. The first and fourth bidders are selected, and the keyword is allocated to the fourth bidder at a price of 2, thereby reducing its remaining budget to 2. As each keyword arrives, the bid of a bidder for that keyword is adjusted to the minimum of its original bid and its remaining budget. Thus, for example, when the third keyword arrives, as shown in Figure 1(c), the bid of the first bidder for that keyword is adjusted from its original value of 6 down to 3 since that is its remaining budget. The two bidders then selected are the first and the second, and the keyword is allocated to the second bidder at a price of 3.

One can justify the second-price version of the problem by observing that when we sell an item, we can only charge the full value of the item when there is more than one interested buyer.

Recall that the first-price analogue to the Second-Price Matching problem, the maximum bipartite matching problem, can be solved optimally in polynomial time. The online version has a trivial 2-competitive deterministic greedy algorithm and an  $e/(e-1)$ -competitive randomized algorithm due to Karp, Vazirani and Vazirani [14], both of which are best possible.

In contrast, we show that the Second-Price Matching problem is APX-hard (Section 4.1). We also give a 2-approximation algorithm for the offline problem (Section 4.2). We then turn to the online version of the problem. Here, we show that no deterministic online algorithm can get a competitive ratio better than  $m$ , where  $m$  is the number of keywords in the instance, and that no randomized online algorithm can get a competitive ratio better than 2 (Section 5.1). On the other hand, we present a randomized online algorithm that achieves a competitive ratio of  $4\sqrt{e}/(\sqrt{e}-1) \approx 5.08$  (Section 5.2). To obtain this competitive ratio, we prove a generalization of the result due to Karp, Vazirani, and Vazirani [14] and Goel and Mehta [8] that the *Ranking* algorithm for online bipartite matching achieves a competitive ratio of  $e/(e-1)$ .

## 1.2 Related Work

As discussed above, the related First-Price Ad Auctions problem<sup>3</sup> has received a fair amount of attention. Mehta et al. [10] present an algorithm for the online version that achieves an optimal competitive ratio of  $e/(e-1)$  for the case when the bids are much smaller than the budgets, a result also proved by Buchbinder et al. [5]. When there is no restriction on the values of the bids relative to the budgets, the best known competitive ratio is 2 [18]. For the offline version of the problem, a sequence of papers [18, 3, 16, 4, 11, 6] culminating in a paper by Chakrabarty and Goel, and independently, a paper by Srinivasan, show that the offline problem can be approximated to within a factor of  $4/3$  and that there is no polynomial time approximation algorithm that achieves a ratio better than  $16/15$  unless  $P = NP$  [6].

The most closely related work to ours is the paper of Goel, Mahdian, Nazarzadeh and Saberi [7], which builds on the work of Abrams, Medelvitich, and Tomlin [2]. Goel et al. look at the online allocation problem when the search engine is committed to charging under the GSP scheme, with multiple slots per keyword. They study two models, the “strict” and “non-strict” models, both of which differ from our model even for the one slot case by allowing bidders to keep bidding their original bid, even when their budget falls below this amount. Thus, in these models, although bidders are not charged more than their remaining budget when allocated a keyword, a bidder with a negligible amount of remaining budget can keep his bids high indefinitely, and as long as this bidder is never allocated another slot, this high bid can determine the prices other bidders pay on many keywords. Under the assumption that bids are small compared to

<sup>3</sup> This problem has also been called the *Adwords* problem [10] and the *Maximum Budgeted Allocation* problem [4, 6, 11].

budgets, Goel et al. build on the linear programming formulation of Abrams et al. to present an  $e/(e-1)$ -competitive algorithm for the non-strict model and a 3-competitive algorithm for the strict model.

The significant, qualitative difference between these positive results and the strong hardness we prove for our model suggests that these aspects of the problem formulation are important. We feel that our model, in which bidders are not allowed to bid more than their remaining budget, is more natural because it seems inherently unfair that a bidder with negligible or no budget should be able to indefinitely set high prices for other bidders.

## 2 Model and Notation

We define the Second-Price Ad Auctions (2PAA) problem formally as follows. The input is a set of ordered keywords  $U$  and bidders  $V$ . Each bidder  $v \in V$  has a budget  $B_v$  and a nonnegative bid  $b_{u,v}$  for every keyword  $u \in U$ . We assume that all of bidder  $v$ 's bids  $b_{u,v}$  are less than or equal to  $B_v$ .

Let  $B_v(t)$  be the remaining budget of bidder  $v$  immediately after the  $t$ -th keyword is processed (so  $B_v(0) = B_v$  for all  $v$ ), and let  $b_{u,v}(t) = \min(b_{u,v}, B_v(t))$ . (Both quantities are defined inductively.) A solution (or *second-price matching*) to 2PAA chooses for the  $t$ -th keyword  $u$  a pair of bidders  $v_1$  and  $v_2$  such that  $b_{u,v_1}(t-1) \geq b_{u,v_2}(t-1)$ , allocates the slot for keyword  $u$  to bidder  $v_1$  and charges bidder  $v_1$  a price of  $p(t) = b_{u,v_2}(t-1)$ , the bid of  $v_2$ . (We say that  $v_1$  acts as the *first-price bidder* for  $u$  and  $v_2$  acts as the *second-price bidder* for  $u$ .) The budget of  $v_1$  is then reduced by  $p(t)$ , so  $B_{v_1}(t) = B_{v_1}(t-1) - p(t)$ . For all other bidders  $v \neq v_1$ ,  $B_v(t) = B_v(t-1)$ . The final value of the solution is  $\sum_t p(t)$ , and the goal is to find a solution of maximum value.

In the offline version of the problem, all of the bids are known to the algorithm beforehand, whereas in the online version of the problem, keyword  $u$  and the bids  $b_{u,v}$  for each  $v \in V$  are revealed only when keyword  $u$  arrives, at which point the algorithm must irrevocably map  $u$  to a pair of bidders without knowing the bids for the keywords that will arrive later.

The special case referred to as Second-Price Matching (2PM) is where  $b_{u,v}$  is either 0 or 1 for all  $(u, v)$  pairs and  $B_v = 1$  for all  $v$ . We will think of this as the variant on maximum bipartite matching (with input  $G = (U \cup V, E)$ ) described in Section 1.1. Note that in 2PM, a keyword can only be allocated for profit if its degree is at least two. Therefore, we assume without loss of generality that for all inputs of 2PM, the degree of every keyword is at least two.

For an input to 2PAA, let  $R_{min} = \min_{u,v} B_v/b_{u,v}$ , and let  $m = |U|$  be the number of keywords.

## 3 Hardness of Approximation of 2PAA

In this section, we present our main hardness result for the Second-Price Ad Auctions problem. For a constant  $c \geq 1$ , let 2PAA( $c$ ) be the version of 2PAA in which we are promised that  $R_{min} \geq c$ .

**Theorem 1.** *Let  $c \geq 1$  be a constant integer. For any constant  $c' > c$ , it is NP-hard to approximate 2PAA( $c$ ) to a factor of  $m/c'$ .*

Hence, even when the bids are guaranteed to be smaller than the budget by a large constant factor, it is NP-hard to approximate 2PAA to a factor better than  $\Omega(m)$ . As noted in the introduction, this hardness is matched by a trivial algorithm.

*Proof.* Here we provide the reduction. Some of the details of the proof are left for the full version of the paper.

Fix a constant  $c' > c$ , and let  $n_0$  be the smallest integer such that for all  $n \geq n_0$ ,

$$c' \cdot \frac{c(n^5 + n + 2)}{cn^2 + n + 2} \geq c(n^3 + cn^2 + n + 2) \quad (1)$$

and

$$\frac{n/2 + 1}{2} \geq c. \quad (2)$$

Note that since  $n_0$  depends only on  $c'$ , it is a constant.

We reduce from PARTITION, in which the input is a set of  $n \geq n_0$  items, and the weight of the  $i$ -th item is given by  $w_i$ . If  $W = \sum_{i=1}^n w_i$ , then the question is whether there is a partition of the items into two subsets of size  $n/2$  such that the sum of the  $w_i$ 's in each subset is  $W/2$ . It is known that this problem (even when the subsets must both have size  $n/2$ ) is NP-hard [24].

Given an instance of PARTITION, we create an instance of 2PAA( $c$ ) as follows. (This reduction is illustrated in Figure 2.)

- First, create  $n + 2$  keywords  $c_1, \dots, c_n, e_1, e_2$ . Second, create an additional set

$$G = \{g_{i,k} : 1 \leq i \leq n^2 \text{ and } 1 \leq k \leq c\}$$

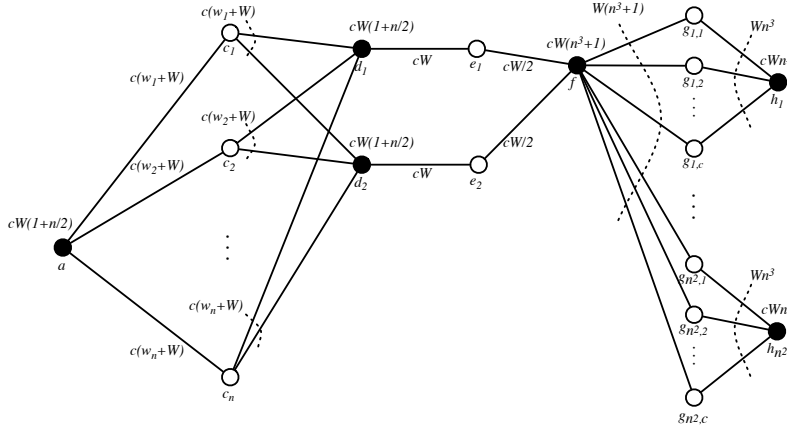
of  $cn^2$  keywords. The keywords arrive in the order

$$c_1, \dots, c_n, e_1, e_2, g_{1,1}, \dots, g_{1,c}, \dots, g_{n^2,1}, \dots, g_{n^2,c}.$$

- Create  $n^2 + 4$  bidders  $a, d_1, d_2, f, h_1, \dots, h_{n^2}$ . Set the budgets of  $a, d_1$ , and  $d_2$  to  $cW(1 + n/2)$ . Set the budget of  $f$  to  $cW(n^3 + 1)$ . For  $1 \leq i \leq n^2$ , set the budget of  $h_i$  to  $cWn^3$ .
- For  $1 \leq i \leq n$ , bidders  $a, d_1$ , and  $d_2$  bid  $c(w_i + W)$  on keyword  $c_i$ .
- For  $j \in \{1, 2\}$ , bidder  $d_j$  bids  $cW$  on keyword  $e_j$ . Bidder  $f$  bids  $cW/2$  on both  $e_1$  and  $e_2$ .
- For  $1 \leq i \leq n^2$  and  $1 \leq k \leq c$ , keyword  $g_{i,k}$  receives a bid of  $W(n^3 + 1)$  from bidder  $f$  and a bid of  $Wn^3$  from bidder  $h_i$ .

This reduction can clearly be performed in polynomial time. Furthermore, it can easily be checked that (2) implies that no bidder bids more than  $1/c$  of its budget on any keyword.

We claim (and prove in the full version of the paper) that if the PARTITION instance is a “yes” instance, then there exists a feasible solution to the 2PAA( $c$ ) instance of value at least  $cW(n^5 + n + 2)$ , and if the PARTITION instance is a



**Fig. 2.** The 2PAA( $c$ ) instance of the reduction. Each bidder's budget is shown above its node, and the bids of bidders for keywords is shown near the corresponding edge.

“no” instance, then every feasible solution to the 2PAA( $c$ ) instance must have value less than  $cW(n^3 + cn^2 + n + 2)$ . Since there are  $cn^2 + n + 2$  keywords in the 2PAA( $c$ ) instance, equation (1) implies that an  $m/c'$ -approximation algorithm can distinguish between the two cases, and hence such an algorithm cannot exist unless  $P = NP$ .  $\square$

## 4 Offline Second-Price Matching

In this section, we turn our attention to the offline version of the special case of Second-Price Matching (2PM).

### 4.1 Hardness of Approximation

To prove that 2PM is APX-hard, we reduce from vertex cover, using the following result.

**Theorem 2 (Chlebík and Chlebíková [25]).** *It is NP-hard to approximate Vertex Cover on 4-regular graphs to within  $53/52$ .*

The precise statement of our hardness result is the following theorem

**Theorem 3.** *It is NP-hard to approximate 2PM to within a factor of  $364/363$ .*

### 4.2 A 2-Approximation Algorithm

Consider an instance  $G = (U \cup V, E)$  of the 2PM problem. We provide an algorithm that first finds a maximum matching  $f : U \rightarrow V$  and then uses  $f$

to return a second-price matching that contains at least half of the keywords matched by  $f$ .<sup>4</sup> Given a matching  $f$ , call an edge  $(u, v) \in E$  such that  $f(u) \neq v$  an *up-edge* if  $v$  is matched by  $f$  and  $f^{-1}(v)$  arrives before  $u$ , and a *down-edge* otherwise. Recall that we have assumed without loss of generality that the degree of every keyword in  $U$  is at least two. Therefore, every keyword  $u \in U$  that is matched by  $f$  must have at least one up-edge or down-edge. Theorem 4 shows that the following algorithm, called `ReverseMatch`, is a 2-approximation for 2PM.

**ReverseMatch Algorithm:**

*Initialization:*

Find an arbitrary maximum matching  $f : U \rightarrow V$  on  $G$ .

*Constructing a 2nd-price matching:*

Consider the matched keywords in reverse order of their arrival.

For each keyword  $u$ :

If keyword  $u$  is adjacent to a down-edge  $(u, v)$ :

Assign keyword  $u$  to bidder  $f(u)$  (with  $v$  acting as the second-price bidder).

Else:

Choose an arbitrary bidder  $v$  that is adjacent to keyword  $u$ .

Remove the edge  $(f^{-1}(v), v)$  from  $f$ .

Assign keyword  $u$  to bidder  $f(u)$  (with  $v$  acting as the second-price bidder).

**Theorem 4.** *The ReverseMatch algorithm is a 2-approximation.*

## 5 Online Second-Price Matching

In this section, we consider the online 2PM problem, in which the keywords arrive one-by-one and must be matched by the algorithm as they arrive. We start, in Section 5.1, by giving a simple lower bound showing that no deterministic algorithm can achieve a competitive ratio better than  $m$ , the number of keywords. Then we move to randomized online algorithms and show that no randomized algorithm can achieve a competitive ratio better than 2. In Section 5.2, we provide a randomized online algorithm that achieves a competitive ratio of  $4\sqrt{e}/(\sqrt{e} - 1) \approx 5.083$ .

### 5.1 Lower Bounds

The following theorem establishes our lower bound on deterministic algorithms, which matches the trivial algorithm of arbitrarily allocating the first keyword to arrive, and refusing to allocate any of the remaining keywords. The adversary for this lower bound offers the first keyword two bidders; whichever bidder the algorithm chooses for this first keyword will be needed as a second-price bidder for the rest of the keywords.

**Theorem 5.** *For any  $m$ , there is an adversary that creates a graph with  $m$  keywords that forces any deterministic algorithm to get a competitive ratio no better than  $1/m$ .*

<sup>4</sup> Note that  $f$  is a partial function.

We next show that no online (randomized) online algorithm for 2PM can achieve a competitive ratio better than 2. To do this we invoke Yao’s Principle [26] and construct a distribution of inputs for which the best deterministic algorithm achieves an expected performance of (asymptotically) 1/2 the value of the optimal solution.

Our distribution is constructed as follows. The first keyword arrives, and it is adjacent to two bidders. Then the second keyword arrives, and it is adjacent to one of the two bidders adjacent to the first keyword, chosen uniformly at random, as well as a new bidder; then the third keyword arrives, and it is adjacent to one of the bidders adjacent to the second keyword, chosen uniformly at random, as well as a new bidder; and so on, until the  $m$ -th keyword arrives.

We claim (and prove in the full version) that no deterministic algorithm can achieve an expected performance better than 1/2 on this distribution. This yields the following theorem.

**Theorem 6.** *The competitive ratio of any randomized algorithm for 2PM must be at least 2.*

## 5.2 A Randomized Competitive Algorithm

In this section, we provide an algorithm that achieves a competitive ratio of  $2\sqrt{e}/(\sqrt{e}-1) \approx 5.083$ . The result builds on a new generalization of the result that the Ranking algorithm for online bipartite matching achieves a competitive ratio of  $e/(e-1) \approx 1.582$ . This was originally shown by Karp, Vazirani, and Vazirani [14], though a mistake was recently found in their proof by Krohn and Varadarajan and corrected by Goel and Mehta [8].

The online bipartite matching problem is merely the first-price version of 2PM, i.e., the problem in which there is no requirement for there to exist a second-price bidder to get a profit of 1 for a match. The Ranking algorithm chooses a random permutation on the bidders  $V$  and uses that to choose matches for the keywords  $U$  as they arrive. This is described more precisely below.

### Ranking Algorithm:

*Initialization:*

Choose a random permutation (ranking)  $\sigma$  of the bidders  $V$ .

*Online Matching:*

Upon arrival of keyword  $u \in U$ :

Let  $N(u)$  be the set of neighbors of  $u$  that have not been matched yet.

If  $N(u) \neq \emptyset$ , match  $u$  to the bidder  $v \in N(u)$  that minimizes  $\sigma(v)$ .

Karp, Vazirani, and Vazirani, and Goel and Mehta prove the following result.

**Theorem 7 (Karp, Vazirani, and Vazirani [14] and Goel and Mehta [8]).** *The Ranking algorithm for online bipartite matching achieves a competitive ratio of  $e/(e-1) + o(1)$ .*

In order to state our generalization of this result, we define the notion of a *left  $k$ -copy* of a bipartite graph  $G = (U \cup V, E)$ . Intuitively, a left  $k$ -copy of  $G$  makes  $k$  copies of each keyword  $u \in U$  such that the neighborhood of a copy of  $u$  is the same as the neighborhood of  $u$ . More precisely, we have the following definition.

**Definition 8.** Given a bipartite graph  $G = (U_G \cup V, E_G)$ , a **left  $k$ -copy** of  $G$  is a graph  $H = (U_H \cup V, E_H)$  for which  $|U_H| = k|U_G|$  and for which there exists a map  $\zeta : U_H \rightarrow U_G$  such that

- for each  $u_G \in U_G$  there are exactly  $k$  vertices  $u_H \in U_H$  such that  $\zeta(u_H) = u_G$ , and
- for all  $u_H \in U_H$  and  $v \in V$ ,  $(u_H, v) \in E_H$  if and only if  $(\zeta(u_H), v) \in E_G$ .

Our generalization of Theorem 7 describes the competitive ratio of Ranking on a graph  $H$  that is a left  $k$ -copy of  $G$ . Its proof builds on the proof of Theorem 7 presented by Birnbaum and Mathieu [27].

**Theorem 9.** Let  $G = (U_G \cup V, E_G)$  be a bipartite graph that has a maximum matching of size  $OPT_{1P}$ , and let  $H = (U_H \cup V, E_H)$  be a left  $k$ -copy of  $G$ . Then the expected size of the matching returned by Ranking on  $H$  is at least

$$kOPT_{1P} \left( 1 - \frac{1}{e^{1/k}} + o(1) \right) .$$

Using this result, we are able to prove that the following algorithm, called RankingSimulate, achieves a competitive ratio of  $4\sqrt{e}/(\sqrt{e} - 1)$ .

**RankingSimulate Algorithm:**

*Initialization:*

Set  $M$ , the set of *matched* bidders, to  $\emptyset$ .

Set  $R$ , the set of *reserved* bidders, to  $\emptyset$ .

Choose a random permutation (ranking)  $\sigma$  of the bidders  $V$ .

*Online Matching:*

Upon arrival of keyword  $u \in U$ :

Let  $N(u)$  be the set of neighbors of  $u$  that are not in  $M$  or  $R$ .

If  $N(u) = \emptyset$ , do nothing.

If  $|N(u)| = 1$ , let  $v$  be the single bidder in  $N(u)$ .

With probability  $1/2$ , match  $u$  to  $v$  and add  $v$  to  $M$ , and

With probability  $1/2$ , add  $v$  to  $R$ .

If  $|N(u)| \geq 2$ , let  $v_1$  and  $v_2$  be the two distinct bidders in  $N(u)$  that minimize  $\sigma(v)$ .

With probability  $1/2$ , match  $u$  to  $v_1$ , add  $v_1$  to  $M$ , and add  $v_2$  to  $R$ , and

With probability  $1/2$ , match  $u$  to  $v_2$ , add  $v_1$  to  $R$ , and add  $v_2$  to  $M$ .

Let  $G = (U_G \cup V, E_G)$  be the bipartite input graph to 2PM, and let  $H = (U_H \cup V, E_H)$  be a left 2-copy of  $H$ . In the arrival order for  $H$ , the two copies of each keyword  $u_G \in U$  arrive in sequential order. The proof of the following lemma is straightforward.

**Lemma 10.** Fix a ranking  $\sigma$  on  $V$ . For each bidder  $v \in V$ , let  $X_v$  be the indicator variable for the event that  $v$  is matched by Ranking on  $H$ , when the ranking is  $\sigma$ .<sup>5</sup> Let  $X'_v$  be the indicator variable for the event that  $v$  is matched by RankingSimulate on  $G$ , when the ranking is  $\sigma$ . Then  $\mathbb{E}(X'_v) = X_v/2$ .

<sup>5</sup> Note that once  $\sigma$  is fixed,  $X_v$  is deterministic.

With Theorem 9 and Lemma 10, we can now prove the main result of this section.

**Theorem 11.** *The competitive ratio of RankingSimulate is  $2\sqrt{e}/(\sqrt{e} - 1) \approx 5.083$ .*

*Proof.* For a permutation  $\sigma$  on  $V$ , let  $\text{RankingSimulate}(\sigma)$  be the matching of  $G$  returned by  $\text{RankingSimulate}$ , and let  $\text{Ranking}(\sigma)$  be the matching of  $H$  returned by  $\text{Ranking}$ . Lemma 10 implies that, conditioned on  $\sigma$ ,  $\mathbb{E}(|\text{RankingSimulate}(\sigma)|) = |\text{Ranking}(\sigma)|/2$ . By Theorem 9,

$$\mathbb{E}(|\text{RankingSimulate}(\sigma)|) = \frac{1}{2} \mathbb{E}(|\text{Ranking}(\sigma)|) \geq OPT_{1P} \left(1 - 1/e^{1/2} + o(1)\right) .$$

Fix a bidder  $v \in V$ . Let  $P_v$  be the profit from  $v$  obtained by  $\text{RankingSimulate}$ . Suppose that  $v$  is matched by  $\text{RankingSimulate}$  to keyword  $u \in U_G$ . Recall that we have assumed without loss of generality that the degree of  $u$  is at least 2. Let  $v' \neq v$  be another bidder adjacent to  $u$ . Then, given that  $v$  is matched to  $u$ , the probability that  $v'$  is matched to any keyword is no greater than  $1/2$ . Therefore,  $\mathbb{E}(P_v | v \text{ matched}) \geq 1/2$ . Hence, the expected value of the second-price matching returned by  $\text{RankingSimulate}$  is

$$\begin{aligned} \sum_{v \in V} \mathbb{E}(P_v) &= \sum_{v \in V} \mathbb{E}(P_v | v \text{ matched}) \Pr(v \text{ matched}) \\ &\geq \frac{1}{2} \sum_{v \in V} \Pr(v \text{ matched}) \\ &= \frac{1}{2} \mathbb{E}(|\text{RankingSimulate}(\sigma)|) \\ &\geq \frac{1}{2} OPT_{1P} \left(1 - 1/e^{1/2} + o(1)\right) \\ &\geq \frac{1}{2} OPT_{2P} \left(1 - 1/e^{1/2} + o(1)\right) , \end{aligned}$$

where  $OPT_{2P}$  is the size of the optimal second-price matching on  $G$ . □

## References

1. Lahaie, S., Pennock, D.M., Saberi, A., Vohra, R.V.: Sponsored search auctions. In Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V., eds.: *Algorithmic Game Theory*. Cambridge University Press (2007) 699–716
2. Abrams, Z., Mendelevitch, O., Tomlin, J.: Optimal delivery of sponsored search advertisements subject to budget constraints. In: *EC '07*. (2007)
3. Andelman, N., Mansour, Y.: Auctions with budget constraints. In: *SWAT '04* (LNCS 3111), Springer (2004) 26–38
4. Azar, Y., Birnbaum, B., Karlin, A.R., Mathieu, C., Nguyen, C.T.: Improved approximation algorithms for budgeted allocations. In: *ICALP '08* (LNCS 5125), Springer (2008) 186–197
5. Buchbinder, N., Jain, K., Naor, J.S.: Online primal-dual algorithms for maximizing ad-auctions revenue. In: *ESA '07* (LNCS 4698), Springer (2007) 253–264

6. Chakrabarty, D., Goel, G.: On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. In: FOCS '08. (2008) 687–696
7. Goel, A., Mahdian, M., Nazerzadeh, H., Saberi, A.: Advertisement allocation for generalized second pricing schemes. In: Workshop on Sponsored Search Auctions. (2008)
8. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: SODA '08. (2008) 982–991
9. Mahdian, M., Nazerzadeh, H., Saberi, A.: Allocating online advertisement space with unreliable estimates. In: EC '07. (2007) 288–294
10. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. *J. ACM* **54**(5) (2007) 22
11. Srinivasan, A.: Budgeted allocations in the full-information setting. In: APPROX '08 (LNCS 5171), Springer (2008) 247–253
12. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review* **97** (2007) 242–259
13. Varian, H.R.: Position auctions. *International Journal of Industrial Organization* **25** (2007) 1163–1178
14. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: STOC '90. (1990) 352–358
15. Dobzinski, S., Schapira, M.: An improved approximation algorithm for combinatorial auctions with submodular bidders. In: SODA '06. (2006) 1064–1073
16. Feige, U., Vondrak, J.: Approximation algorithms for allocation problems: Improving the factor of  $1 - 1/e$ . FOCS '06 (2006) 667–676
17. Khot, S., Lipton, R.J., Markakis, E., Mehta, A.: Inapproximability results for combinatorial auctions with submodular utility functions. In: WINE '05 (LNCS 3828, Springer (2005) 92–101
18. Lehmann, B., Lehmann, D., Nisan, N.: Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* **55**(2) (2006) 270–296
19. Mirrokni, V., Schapira, M., Vondrak, J.: Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In: EC '08. (2008)
20. Vondrak, J.: Optimal approximation for the submodular welfare problem in the value oracle model. In: STOC '08. (2008) 67–74
21. Chekuri, C., Khanna, S.: A ptas for the multiple knapsack problem. In: SODA '00. (2000) 213–222
22. Fleischer, L., Goemans, M.X., Mirrokni, V.S., Sviridenko, M.: Tight approximation algorithms for maximum general assignment problems. In: SODA '06. (2006) 611–620
23. Shmoys, D., Tardos, E.: An approximation algorithm for the generalized assignment problem. *Mathematical Programming* **62** (1993) 461–474
24. Garey, M.R., Johnson, D.S.: *Computers and Intractability*. W. H. Freeman and Company (1979)
25. Chlebík, M., Chlebíková, J.: Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science* **354**(3) (2006) 320–338
26. Yao, A.: Probabilistic computations: Toward a unified measure of complexity. In: FOCS '77. (1977) 222–227
27. Birnbaum, B., Mathieu, C.: On-line bipartite matching made simple. *SIGACT News* **39**(1) (2008) 80–87