Online Bipartite Matching and Adwords

Vijay V. Vazirani
University of California, Irvine

Abstract

The purpose of this paper is to give a "textbook quality" proof of the optimal algorithm, called Ranking, for the online bipartite matching problem (OBM) and to highlight its role in matching-based market design. In particular, we discuss a generalization of OBM, called the adwords problem, which has had a significant impact in the ad auctions marketplace.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design

Keywords and phrases matching-based market design, online algorithms, ad auctions, competitive analysis

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

The online bipartite matching problem[1] (OBM) occupies a central place not only in online algorithms but also in matching-based market design, see details in Sections 1.1 and 1.2. The purpose of this paper is to give a "textbook quality" proof[2] of the optimal algorithm, called Ranking, for this problem and to highlight its role in matching-based market design. In particular, we discuss a generalization of OBM, called the adwords problem, which has had a significant impact in the ad auctions marketplace, see Section 1.2.

Ranking achieves a competitive ratio of $\left(1 - \frac{1}{e}\right)$ [17]. Its analysis, given in [17], was considered "difficult" and it also had an error. Over the years, several researchers contributed valuable ideas to simplifying its proof, see Section 1.1 for details. The proof given in this paper is based on these ideas. Additionally, we highlight a key property used in the proof, called the *No-Surpassing Property* and simplify further its proof. This property turns out to be the bottleneck to a substantial generalization which was attempted in [21], as described below.

The adwords problem, which is called General in this paper, is a generalization of OBM. It involves matching keyword queries, as they arrive online, to advertisers; the latter have daily budget limits and they make bids for the queries. The overall goal is to maximize the total revenue. This problem is notoriously difficult and has remained largely unsolved; see Section 1.1 for marginal progress made recently. Its special case, when bids are small compared to budgets, called Small, captures a key computational issue that arises in the context of ad auctions, for instance in Google's AdWords marketplace. An optimal algorithm for Small, achieving a competitive ratio of $\left(1 - \frac{1}{e}\right)$, was first given in [19]; for the impact of this result in the marketplace, see Section 1.2.

[1] For formal statements of problems discussed in this paper, see Section 2.
[2] e.g., the proof given in the chapter [8] of the upcoming edited book on matching-based market design.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In Open Problem Number 20 in [18], Mehta asks for a *budget-oblivious online algorithm* for SMALL. Such an algorithm does not know the daily budgets of advertisers; however, in a run of the algorithm, it knows when the budget of an advertiser is exhausted. However, its revenue is still compared to the optimal revenue generated by an offline algorithm with full knowledge of the budget. Its importance lies in its use in autobidding platforms [1, 6], which manage the ad campaigns of large advertisers; they dynamically adjust bids and budgets over multiple search engines to improve performance. The greedy algorithm, which matches an arriving query to the advertiser making the highest bid, is clearly budget-oblivious; its competitive ratio is 0.5. An improved algorithm, having a competitive ratio of 0.522, was recently obtained by Udwani [20], using the idea of an LP-free analysis, which involves writing appropriate linear inequalities to compare the online algorithm with the offline optimal algorithm.

Motivated by the recent simplification of the proof of (OBM), [21] attempted to extend RANKING all the way to SMALL. This attempt represents a more basic approach to SMALL than the one used in [19] (see Section 1.1) and the hope was that it would yield an algorithm with better properties, e.g., budget-obliviousness. [21] managed to extend RANKING to an intermediate problem, called SINGLE-VALUED, thereby giving an optimal, budget-oblivious algorithm; see Section 1.1 for competing results for this problem. Under SINGLE-VALUED, each advertiser can make bids of one value only, although the value may be different for different advertisers.

The analysis of SINGLE-VALUED given in [21] involved new ideas from two domains, namely probability theory and combinatorics, with the former playing a dominant role and the latter yielding a proof of the No-Surpassing Property for SINGLE-VALUED. Equipped with these new ideas, [21] next attempted an extension from RANKING to SMALL. Although the more difficult, probabilistic part, of the argument did extend, a counter-example was found to the combinatorial part, showing that the No-Surpassing Property does not hold for SMALL.

1.1 Related Works

We start by stating simplifications to the proof of OBM. At first, [11, 4], got the ball rolling, setting the stage for the substantial simplification given in [7], using a randomized primal-dual approach. [7] introduced the idea of splitting the contribution of each matched edge into primal and dual contributions and lower-bounding each part separately. Their method for defining prices p_j of goods, using randomization, was used by subsequent papers, including this one[3].

Interestingly enough, the next simplification involved removing the scaffolding of LP-duality and casting the proof in purely probabilistic terms[4], using notions from economics to split the contribution of each matched edge into the contributions of the buyer and the seller. This elegant analysis was given by [9]. A further simplification to the proof of the No-Surpassing Property for OBM is given in the current paper.

An important generalization of OBM is online b-matching. This problem is a special case

[3] For a succinct proof of optimality of the underlying function, e^{x-1}, see Section 2.1.1 in [12].

[4] Even though there is no overt use of LP-duality in the proof of [9], it is unclear if this proof could have been obtained directly, without going the LP-duality-route.

of General in which the budget of each advertiser is \$$b$ and the bids are $0/1$. [16] gave a simple optimal online algorithm, called BALANCE, for this problem. BALANCE awards the next query to the interested bidder who has been matched least number of times so far. [16] showed that as b tends to infinity, the competitive ratio of BALANCE tends to $\left(1 - \frac{1}{e}\right)$.

Observe that b-matching is a special case of Small, if b is large. Indeed, the first online algorithm [19] for Small was obtained by extending BALANCE as follows: [19] first gave a simpler proof of the competitive ratio of BALANCE using the notion of a *factor-revealing LP* [15]. Then they gave the notion of a *tradeoff-revealing LP*, which yielded an algorithm achieving a competitive ratio of $\left(1 - \frac{1}{e}\right)$. [19] also proved that this is optimal for b-matching, and hence Small, by proving that no randomized algorithm can achieve a better ratio for online b-matching; previously, [16] had shown a similar result for deterministic algorithms.

The algorithm of [19] is simple and operates as follows. The effective bid of each bidder j for a query is its bid multiplied by $(1 - e^{L_j/B_j})$, where B_j and L_j are the total budget and the leftover budget of bidder j, respectively; the query is matched to the bidder whose effective bid is highest. As a result, the algorithm of [19] needs to know the total budget of each bidder. Following [19], a second optimal online algorithm for Small was given in [5], using a primal-dual approach.

Another relevant generalization of OBM is online vertex weighted matching, in which the offline vertices have weights and the objective is to maximize the weight of the matched vertices. [2] extended Ranking to obtain an optimal online algorithm for this problem. Clearly, Single-Valued is intermediate between General and online vertex weighted matching. [2] gave an optimal online algorithm for Single-Valued by reducing it to online vertex weighted matching. This involved creating k_j copies of each advertiser j. As a result, their algorithm needs to use $\sum_{j \in A} k_j$ random numbers, where A is the set of advertisers.

We note that independent of [21], Albers and Schubert [3] had also obtained an optimal, budget-oblivious algorithm for Single-Valued; however, their technique was different and involved formulating a configuration LP and conducting a primal-dual analysis. Another advantage of the algorithms of [3] and [21], in contrast to [2], was that they need to use only $|A|$ random numbers.

For General, the greedy algorithm, which matches each query to the highest bidder, achieves a competitive ratio of $1/2$. Until recently, that was the best possible. In [13] a marginally improved algorithm, with a ratio of 0.5016, was given. It is important to point out that this 60-page paper was a tour-de-force, drawing on a diverse collection of ideas — a testament to the difficulty of this problem.

In the decade following the conference version (FOCS 2005) of [19], search engine companies generously invested in research on models derived from OBM and adwords. Their motivation was two-fold: the substantial impact of [19] and the emergence of a rich collection of digital ad tools. It will be impossible to do justice to this substantial body of work, involving both algorithmic and game-theoretic ideas; for a start, see the surveys [18, 12].

1.2 Significance and Practical Impact

Google's AdWords marketplace generates multi-billion dollar revenues annually and the current annual worldwide spending on digital advertising is almost half a trillion dollars. These revenues of Google and other Internet services companies enable them to offer crucial services, such as search, email, videos, news, apps, maps etc. for free – services that have virtually transformed our lives.

We note that Small is the most relevant case of adwords for the search ads marketplace e.g., see [6]. A remarkable feature of Google, and other search engines, is the speed with which they are able to show search results, often in milliseconds. In order to show ads at the same speed, together with search results, the solution for Small needed to be minimalistic in its use of computing power, memory and communication.

The online algorithm of [19] satisfied these criteria and therefore had a substantial impact in this marketplace. Furthermore, the idea underlying their algorithm was extracted into a simple heuristic, called *bid scaling*, which uses even less computation and is widely used by search engine companies today. As mentioned above, our Conditional Algorithm for Small is even more elementary and is budget-oblivious.

It will be useful to view the AdWords marketplace in the context of a bigger revolution, namely the advent of the Internet and mobile computing, and the consequent resurgence of the area of matching-based market design. The birth of this area goes back to the seminal 1962 paper of Gale and Shapley on stable matching [10]. Over the decades, this area became known for its highly successful applications, having economic as well as sociological impact. These included matching medical interns to hospitals, students to schools in large cities, and kidney exchange.

The resurgence led to a host of highly innovative and impactful applications. Besides the AdWords marketplace, which matches queries to advertisers, these include Uber, matching drivers to riders; Upwork, matching employers to workers; and Tinder, matching people to each other, see [14] for more details.

A successful launch of such markets calls for economic and game-theoretic insights, together with algorithmic ideas. The Gale-Shapley deferred acceptance algorithm and its follow-up works provided the algorithmic backbone for the "first life" of matching-based market design. The algorithm Ranking has become the paradigm-setting algorithmic idea in the "second life" of this area. Interestingly enough, this result was obtained in the pre-Internet days, over thirty years ago.

2 Preliminaries

Online Bipartite Matching (OBM): Let B be a set of n buyers and S a set of n goods. A bipartite graph $G = (B, S, E)$ is specified on vertex sets B and S, and edge set E, where for $i \in B,\ j \in S,\ (i, j) \in E$ if and only if buyer i *likes* good j. G is assumed to have a perfect matching and therefore each buyer can be given a unique good she likes. Graph G is revealed in the following manner. The n goods are known up-front. On the other hand, the buyers arrive one at a time, and when buyer i arrives, the edges incident at i are revealed.

We are required to design an online algorithm $\mathcal{A}$ in the following sense. At the moment a buyer i arrives, the algorithm needs to match i to one of its unmatched neighbors, if any; if all of i's neighbors are matched, i remains unmatched. The difficulty is that the algorithm does not "know" the edges incident at buyers which will arrive in the future and yet the size of the matching produced by the algorithm will be compared to the best *off-line matching*; the latter of course is a perfect matching. The formal measure for the algorithm is defined in Section 2.1.

General Adwords Problem (General): Let A be a set of m *advertisers*, also called *bidders*, and Q be a set of n *queries*. A bipartite graph $G = (Q, A, E)$ is specified on vertex sets Q and A, and edge set E, where for $i \in Q$ and $j \in A$, $(i, j) \in E$ if and only if bidder j is *interested in* query i. Each query i needs to be matched[5] to at most one bidder who is interested in it. For each edge (i, j), bidder j *knows* his bid for i, denoted by $\text{bid}(i, j) \in \mathbb{Z}_+$. Each bidder also has a *budget* $B_j \in \mathbb{Z}_+$ which satisfies $B_j \geq \text{bid}(i, j)$, for each edge (i, j) incident at j.

Graph G is revealed in the following manner. The m bidders are known up-front and the queries arrive one at a time. When query i arrives, the edges incident at i are revealed, together with the bids associated with these edges. If i gets matched to j, then the matched edge (i, j) is assigned a weight of $\text{bid}(i, j)$. The constraint on j is that the total weight of matched edges incident at it be at most B_j. The objective is to maximize the total weight of all matched edges at all bidders.

Adwords under Single-Valued Bidders (Single-Valued): Single-Valued is a special case of General in which each bidder j will make bids of a single value, $b_j \in \mathbb{Z}_+$, for the queries he is interested in. If i accepts j's bid, then i will be matched to j and the weight of this matched edge will be b_j. Corresponding to each bidder j, we are also given $k_j \in \mathbb{Z}_+$, the maximum number of times j can be matched to queries. The objective is to maximize the total weight of matched edges. Observe that the matching M found in G is a b-matching with the b-value of each query i being 1 and of advertiser j being k_j.

Adwords under Small Bids (Small): Small is a special case of General in which for each bidder j, each bid of j is small compared to its budget. Formally, we will capture this condition by imposing the following constraint. For a valid instance I of Small, define

$$\mu(I) = \max_{j \in A} \left\{ \frac{\max_{(i,j) \in E} \{\text{bid}(i, j) - 1\}}{B_j} \right\}.$$

Then we require that

$$\lim_{n(I) \to \infty} \mu(I) = 0,$$

where $n(I)$ denotes the number of queries in instance I.

2.1 The competitive ratio of online algorithms

We will define the notion of competitive ratio of a randomized online algorithm in the context of OBM.

▶ **Definition 1.** *Let $G = (B, S, E)$ be a bipartite graph as specified above. The competitive ratio of a randomized algorithm $\mathcal{A}$ for* OBM *is defined to be:*

$$c(\mathcal{A}) = \min_{G=(B,S,E)} \min_{\rho(B)} \frac{\mathbb{E}[\mathcal{A}(G, \rho(B))]}{n},$$

where $\mathbb{E}[\mathcal{A}(G, \rho(B))]$ is the expected size of matching produced by $\mathcal{A}$; the expectation is over the random bits used by $\mathcal{A}$. We may assume that the worst case graph and the order of arrival of buyers,

[5] Clearly, this is not a matching in the usual sense, since a bidder may be matched to several queries.

▶ **Algorithm 3.** (**Algorithm** Ranking)

1. **Initialization:** *Pick a random permutation, π, of the goods in S.*
2. **Online buyer arrival:** *When a buyer, say i, arrives, match her to the first unmatched good she likes in the order π; if none, leave i unmatched.*

Output the matching, M, found.

given by $\rho(B)$, are chosen by an adversary who knows the algorithm. It is important to note that the algorithm is provided random bits after *the adversary makes its choices.*

▶ Remark 2. For each problem studied in this paper, we will assume that the offline matching is complete. It is easy to extend the arguments, without changing the competitive ratio, in case the offline matching is not complete. As an example, this is done for OBM in Remark 16.

3 Ranking and its Analysis

Algorithm 3 presents an optimal algorithm for OBM. Note that this algorithm picks a random permutation of goods only once. Its competitive ratio is $(1 - \frac{1}{e})$, as shown in Theorem 15. Furthermore, as shown in [17], it is an optimal online bipartite matching algorithm: no randomized algorithm can do better, up to an $o(1)$ term.

We will analyze Algorithm 5 which is equivalent to Algorithm 3 and operates as follows. Before the execution of Step (1), the adversary determines the order in which buyers will arrive, say $\rho(B)$. In Step (1), each good j is assigned a *price* $p_j = e^{w_j - 1}$, where w_j, called the *rank* of j, is picked at random from $[0, 1]$; observe that $p_j \in [\frac{1}{e}, 1]$. In Step (2), buyers will arrive in the order $\rho(B)$, picked by the adversary, and will be matched to the cheapest available good. With probability 1 all n prices are distinct and sorting the goods by increasing prices results in a random permutation. Furthermore, since Algorithm 5 uses this sorted order only and is oblivious of the actual prices, it is equivalent to Algorithm 3. As we will see, the random variables representing actual prices are crucially important as well – in the analysis. We remark that for the generalizations of OBM studied in this paper, the prices are used not only in the analysis, but also by the algorithms.

3.1 Analysis of Ranking

We will use an *economic setting* for analyzing Algorithm 5 as follows. Each buyer i has *unit-demand* and *0/1 valuations* over the goods she likes, i.e., she accrues unit utility from each good she likes, and she wishes to get at most one of them. The latter set is precisely the set of neighbors of i in G. If on arrival of i there are several of these which are still unmatched, i will pick one having the smallest price [6]. Therefore the buyers will maximize

[6] As stated above, with probability 1 there are no ties.

▶ **Algorithm 5.** **(Algorithm Ranking: Economic Viewpoint)**

1. **Initialization:** $\forall j \in S$: *Pick* w_j *independently and uniformly from* $[0,1]$.
   Set price $p_j \leftarrow e^{w_j - 1}$.

2. **Online buyer arrival:** *When a buyer, say* i*, arrives, match her to the cheapest unmatched good she likes; if none, leave* i *unmatched.*

Output the matching, M*, found.*

their utility as defined below.

For analyzing this algorithm, we will define two sets of random variables, u_i for $i \in B$ and r_j, for $j \in S$. These will be called utility of buyer i and revenue of good j, respectively. Each run of Ranking defines these random variables as follows. If Ranking matches buyer i to good j, then define $u_i = 1 - p_j$ and $r_j = p_j$, where p_j is the price of good j in this run of Ranking. Clearly, p_j is also a random variable, which is defined by Step (1) of the algorithm. If i remains unmatched, define $u_i = 0$, and if j remains unmatched, define $r_j = 0$. Observe that for each good j, $p_j \in [\frac{1}{e}, 1]$ and for each buyer i, $u_i \in [0, 1 - \frac{1}{e}]$. Let M be the matching produced by Ranking and let random variable $|M|$ denote its size.

Lemma 4 pulls apart the contribution of each matched edge (i,j) into u_i and r_j. Next, we established in Lemma 13 that for each edge (i,j) in the graph, the total expected contribution of u_i and r_j is at least $1 - \frac{1}{e}$. Then, linearity of expectation allows us to reassemble the $2n$ terms in the right hand side of Lemma 4 so they are aligned with a perfect matching in G, and this yields Theorem 15.

▶ **Lemma 4.**

$$\mathbb{E}[|M|] \; = \sum_{i}^{n} \mathbb{E}\left[u_i\right] \; + \; \sum_{j}^{n} \mathbb{E}[r_j].$$

Proof. By definition of the random variables,

$$\mathbb{E}[|M|] \; = \mathbb{E}\left[\sum_{i=1}^{n} u_i \; + \; \sum_{j=1}^{n} r_j\right] \; = \sum_{i}^{n} \mathbb{E}\left[u_i\right] \; + \; \sum_{j}^{n} \mathbb{E}[r_j],$$

where the first equality follows from the fact that if $(i,j) \in M$ then $u_i + r_j = 1$ and the second follows from linearity of expectation. ◀

While running Algorithm 5, assume that the adversary has picked the order of arrival of buyers, say $\rho(B)$, and Step (1) has been executed. We next define several ways of executing Step (2). Let $\mathcal{R}$ denote the run of Step (2) on the entire graph G. Corresponding to each good j, let G_j denote graph G with vertex j removed. Define $\mathcal{R}_j$ to be the run of Step (2) on graph G_j.

Lemma 6 and Corollary 7 establish a relationship between the sets of available goods for a buyer i in the two runs $\mathcal{R}$ and $\mathcal{R}_j$; the latter is crucially used in the proof of Lemma 11. For ease of notation in proving these two facts, let us renumber the buyers so their order of arrival under $\rho(B)$ is $1, 2, \ldots n$. Let $T(i)$ and $T_j(i)$ denote the sets of unmatched goods at the time of arrival of buyer i (i.e., just before the buyer i gets matched) in the graphs G and G_j, in runs $\mathcal{R}$ and $\mathcal{R}_j$, respectively. Similarly, let $S(i)$ and $S_j(i)$ denote the set of unmatched goods that buyer i is incident to in G and G_j, in runs $\mathcal{R}$ and $\mathcal{R}_j$, respectively.

We have assumed that Step (1) of Algorithm 5 has already been executed and a price p_k has been assigned to each good k. With probability 1, the prices are all distinct. Let F_1 and F_2 be subsets of S containing goods k such that $p_k < p_j$ and $p_k > p_j$, respectively.

▶ **Lemma 6.** *For each i, $1 \leq i \leq n$, the following hold:*

1. $(T_j(i) \cap F_1) = (T(i) \cap F_1)$.
2. $(T_j(i) \cap F_2) \subseteq (T(i) \cap F_2)$.

Proof. Clearly, in both runs, $\mathcal{R}$ and $\mathcal{R}_j$, any buyer having an available good in F_1 will match to the most profitable one of these, without even considering the rest of the goods. Since $j \notin F_1$, the two runs behave in an identical manner on the set F_1, thereby proving the first statement.

The proof of the second statement is by induction on i. The base case is trivially true since $j \notin F_2$. Assume that the statement is true for $i = k$ and let us prove it for $i = k + 1$. By the first statement, we need to consider only the case that there are no available goods for the k^{th} buyer in F_1 in the runs $\mathcal{R}$ and $\mathcal{R}_j$. Assume that in run $\mathcal{R}_j$, this buyer gets matched to good l; if she remains unmatched, we will take l to be null. Clearly, l is the most profitable good she is incident to in $T_j(k)$. Therefore, the most profitable good she is incident to in run $\mathcal{R}$ is the best of l, the most profitable good in $T(k) - T_j(k)$, and j, in case it is available. In each of these cases, the induction step holds. ◀

In the corollary below, the first two statements follow from Lemma 6 and the third statement follows from the first two.

▶ **Corollary 7.** *For each i, $1 \leq i \leq n$, the following hold:*

1. $(S_j(i) \cap F_1) = (S(i) \cap F_1)$.
2. $(S_j(i) \cap F_2) \subseteq (S(i) \cap F_2)$.
3. $S_j(i) \subseteq S(i)$.

Next we define a new random variable, u_e, for each edge $e = (i, j) \in E$. This is called the *threshold* for edge e and is given in Definition 8. It is critically used in the proofs of Lemmas 11 and 13.

▶ **Definition 8.** *Let $e = (i, j) \in E$ be an arbitrary edge in G. Define random variable, u_e, called the* threshold *for edge e, to be the utility of buyer i in run $\mathcal{R}_j$. Clearly, $u_e \in [0, 1 - \frac{1}{e}]$.*

▶ **Property 9.** **(No-Surpassing for** OBM*)* *Let p_j be such that the bid of j, namely $1 - p_j$, is better than the best bid that buyer i gets in run $\mathcal{R}_j$. Then, in run $\mathcal{R}$, no bid to i will surpass $1 - p_j$.*

▶ **Lemma 10.** *The No-Surpassing Property holds for* OBM*.*

Proof. Suppose the bid of j, namely $1 - p_j$, is better than the best bid that buyer i gets in run $\mathcal{R}_j$. If so, i gets no bid from F_1 in $\mathcal{R}_j$; observe that they are all higher than $1 - p_j$. Now, by the first part of Corollary 7, i gets no bid from F_1 in run $\mathcal{R}$ as well, i.e., in run $\mathcal{R}$, no bid to i will surpass $1 - p_j$. ◀

▶ **Lemma 11.** *Corresponding to each edge $(i, j) \in E$, the following hold.*

1. *$u_i \geq u_e$, where u_i and u_e are the utilities of buyer i in runs $\mathcal{R}$ and $\mathcal{R}_j$, respectively.*
2. *Let $z \in [0, 1 - \frac{1}{e}]$. Conditioned on $u_e = z$, if $p_j < 1 - z$, then j will definitely be matched in run $\mathcal{R}$.*

Proof. **1).** By the third statement of Corollary 7, i has more options in run $\mathcal{R}$ as compared to run $\mathcal{R}_j$, and therefore $u_i \geq u_e$.

2). In run $\mathcal{R}$, if j is already matched when i arrives, there is nothing to prove. Next assume that j is not matched when i arrives. The crux of the matter is that by Lemma 10, the No-Surpassing Property holds. Therefore, in run $\mathcal{R}$, i will not have any option that is better than j and will therefore get matched to j. Since $1 - p_j > z$, $S_j(i) \cap F_1 = \emptyset$. Therefore by the first statement of Corollary 7, $S(i) \cap F_1 = \emptyset$. Since i will get no bid better than j in $\mathcal{R}$, the no-surpassing property indeed holds and i must get matched to j. ◀

▶ Remark 12. The random variable u_e is called *threshold* because of the second statement of Lemma 11. It defines a value such that whenever p_j is smaller than this value, j is definitely matched in run $\mathcal{R}$.

The intuitive reason for the next, and most crucial, lemma is the following. The smaller u_e is, the larger is the range of values for p_j, namely $[0, 1 - u_e)$, over which (i, j) will be matched and j will accrue revenue of p_j. Integrating p_j over this range, and adding $\mathbb{E}[u_i]$ to it, gives the desired bound. Crucial to this argument is the fact that p_j is independent of u_e. This follows from the fact that u_e is determined by run $\mathcal{R}_j$ on graph G_j, which does not contain vertex j.

▶ **Lemma 13.** *Corresponding to each edge $(i, j) \in E$,*

$$\mathbb{E}[u_i + r_j] \geq 1 - \frac{1}{e}.$$

Proof. By the first part of Lemma 11, $\mathbb{E}[u_i] \geq \mathbb{E}[u_e]$.

Next, we will lower bound $\mathbb{E}[r_j]$. Let $z \in [0, 1 - \frac{1}{e}]$ and let us condition on the event $u_e = z$. The critical observation is that u_e is determined by the run $\mathcal{R}_j$. This is conducted on graph G_j, which does not contain vertex j. Therefore u_e is independent of p_j.

By the second part of Lemma 11, $r_j = p_j$ whenever $p_j < 1 - z$. We will ignore the contribution to $\mathbb{E}[r_j]$ when $p_j \geq 1 - z$. Let w be s.t. $e^{w-1} = 1 - z$.

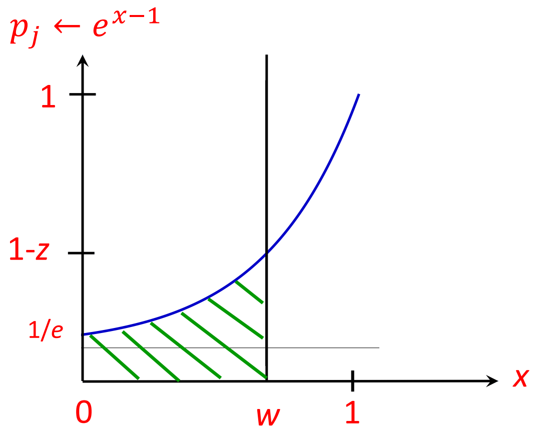

Figure 2 The shaded area is a lower bound on $\mathbb{E}[r_j \mid u_e = z]$.

Now p_j is obtained by picking x uniformly at random from the interval $[0, 1]$ and outputting e^{x-1}. In particular, when $x \in [0, w)$, $p_j < 1 - z$. If so, by the second part of Lemma 11, j is matched and revenue is accrued in r_j, see Figure 2. Therefore,

$$\mathbb{E}[r_j \mid u_e = z] \geq \int_0^w e^{x-1}\, dx \;=\; e^{w-1} - \frac{1}{e} \;=\; 1 - \frac{1}{e} - z.$$

Let $f_{u_e}(z)$ be the probability density function of u_e; clearly, $f_{u_e}(z) = 0$ for $z \notin [0, 1 - \frac{1}{e}]$. Therefore,

$$\mathbb{E}[r_j] \;=\; \mathbb{E}[\mathbb{E}[r_j \mid u_e]] \;=\; \int_{z=0}^{1-1/e} \mathbb{E}[r_j \mid u_e = z] \cdot f_{u_e}(z) dz$$

$$\geq \int_{z=0}^{1-1/e} \left(1 - \frac{1}{e} - z\right) \cdot f_{u_e}(z) dz \;=\; 1 - \frac{1}{e} - \mathbb{E}[u_e],$$

where the first equality follows from the law of total expectation and the inequality follows from fact that we have ignored the contribution to $\mathbb{E}[r_j \mid u_e]$ when $p_j \geq 1 - z$. Hence we get

$$\mathbb{E}[u_i + r_j] = \mathbb{E}[u_i] + \mathbb{E}[r_j] \geq 1 - \frac{1}{e}.$$

◀

▶ Remark 14. Observe that Lemma 13 is not a statement about i and j getting matched to each other, but about the utility accrued by i and the revenue accrued by j by being matched to various goods and buyers, respectively, over the randomization executed in Step (1) of Algorithm 5.

▶ **Theorem 15.** *The competitive ratio of* Ranking *is at least* $1 - \frac{1}{e}$.

Proof. Let P denote a perfect matching in G. The expected size of matching produced by RANKING is

$$\mathbb{E}\left[|M|\right] = \sum_{i}^{n} \mathbb{E}\left[u_i\right] + \sum_{j}^{n} \mathbb{E}[r_j] = \sum_{(i,j)\in P} \mathbb{E}[u_i + r_j] \geq n\left(1 - \frac{1}{e}\right),$$

where the first equality uses Lemma 4, the second follows from linearity of expectation and the inequality follows from Lemma 13 and the fact that $|P| = n$. The theorem follows. ◀

▶ Remark 16. In case G does not have a perfect matching, let P denote a maximum matching in G, of size k, say. Then summing $\mathbb{E}\left[u_i\right]$ and $\mathbb{E}[r_j]$ over the the vertices i and j matched by P, we get that the expected size of matching produced by RANKING is at least $k\left(1 - \frac{1}{e}\right)$.

References

1. Gagan Aggarwal, Ashwinkumar Badanidiyuru, and Aranyak Mehta. Autobidding with constraints. In *International Conference on Web and Internet Economics*, pages 17–30. Springer, 2019.
2. Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1253–1264, 2011.
3. Susanne Albers and Sebastian Schubert. Optimal algorithms for online b-matching with variable vertex capacities. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
4. Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *ACM Sigact News*, 39(1):80–87, 2008.
5. Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264, 2007.
6. Nikhil Devanur and Aranyak Mehta. Online matching in advertisement auctions. In Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani, editors, *Online and Matching-Based Market Design*. Cambridge University Press, 2022. [To appear] `https://www.ics.uci.edu/~vazirani/AdAuctions.pdf`.
7. Nikhil R Devanur, Kamal Jain, and Robert D Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 101–107. SIAM, 2013.
8. Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani. One-sided matching markets. In Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani, editors, *Online and Matching-Based Market Design*. Cambridge University Press, 2022. [To appear] `https://www.ics.uci.edu/~vazirani/Chapter2.pdf`.
9. Alon Eden, Michal Feldman, Amos Fiat, and Kineret Segal. An economic-based analysis of ranking for online bipartite matching. In *SIAM Symposium on Simplicity in Algorithms*, 2021.
10. David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

11 Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, volume 8, pages 982–991, 2008.

12 Zhiyi Huang and Thorben Trobst. Online matching. In Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani, editors, *Online and Matching-Based Market Design*. Cambridge University Press, 2022. [To appear] `https://www.ics.uci.edu/~vazirani/Ch4.pdf`.

13 Zhiyi Huang, Qiankun Zhang, and Yuhao Zhang. Adwords in a panorama. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1416–1426. IEEE, 2020.

14 Simons Institute. Online and matching-based market design, 2019. `https://simons.berkeley.edu/programs/market2019`.

15 Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM (JACM)*, 50(6):795–824, 2003.

16 Bala Kalyanasundaram and Kirk R Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1-2):319–325, 2000.

17 Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.

18 Aranyak Mehta. *Online matching and ad allocation*, volume 8. Now Publishers, Inc., 2013.

19 Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5), 2007.

20 Rajan Udwani. Adwords with unknown budgets. *arXiv preprint arXiv:2110.00504*, 2021.

21 Vijay V Vazirani. Online bipartite matching and adwords. *arXiv preprint arXiv:2107.10777*, 2021.