Greedy algorithms for online allocation problems with stochastic input

Aman Dhesi

Master's Thesis

In Partial Fullfillment of the Requirements
for the Master of Science in Engineering
Department of Computer Science
Princeton University

Adviser: Moses Charikar

June 2012

Abstract

We consider the general problem of allocating items to agents when items arrive online. Each agent specifies a positive real valuation for each item. When an item arrives, it must immediately be assigned to an agent and the decisions cannot be reversed. The goal of the algorithm is to maximize the sum of the values of the items received by the agents.

This problem has a long and rich history dating back to the seminal work of Karp, Vazirani and Vazirani who gave a randomized algorithm for the online bipartite matching problem with competitive ratio 1-1/e, and showed that it is optimal. More recently, Mehta, Saberi, Vazirani and Vazirani studied the problem of ad-allocation for search queries, for which they give an optimal (1-1/e)-competitive algorithm under some mild assumptions. Interest from web-search companies sparked off a line of research on these problems trying to beat the factor of 1-1/e in stochastic input models. Here, instead of the arrival order of the items being chosen by an adversary, it is assumed that there is some underlying probability distribution according to which the arrival order is generated.

Recently, Devanur, Jain and Kleinberg gave an exceedingly simple analysis of the original algorithm for online bipartite matching using the so-called Randomized Primal-Dual framework. We observe that their proof can be interpreted as a proof of the competitiveness of a greedy algorithm for the same problem, when the input is stochastic. We use this framework to show that natural greedy algorithms achieve a competitive ratio of 1-1/e for different variants of the online allocation problem with stochastic input. These variants include vertex-weighted bipartite matching and the ad-allocation problem. Then, we use this framework to give a simpler proof that the original algorithm for online bipartite matching achieves a competitive ratio strictly better than 1-1/e when the input is stochastic.

Finally, we discuss several open problems and state some conjectures. We believe that the Randomized Primal-Dual framework is a powerful tool for analyzing greedy algorithms for online allocation problems with stochastic input, and can be used to develop new insights and solve some of the open problems in this area.

Acknowledgments

I would like to thank my advisor Moses Charikar for his endless support and encouragement, both academic and otherwise. This work would not have been possible without his input and ideas. I would like to thank my officemates Arman Suleimenov and Dominic Kao for their support. I would also like to thank Omri Weinstein and Rajesh Ranganath for their advice and critique.

Contents

1 Introduction

In this work, we study a general class of online allocation problems in which indivisible items arriving one-by-one must be assigned to agents. The algorithm must make a decision when the item arrives and this decision cannot be reversed. The goal is to maximize "profit", which is defined differently in different instantiations of the general problem. Intuitively, "profit" measures the overall welfare or happiness of the involved agents, which depends on the set of items they are allocated. Each agent is constrained in some way, depending again on the particular problem. The constraints can be cardinality constraints, budget constraints and so on.

The motivating reason to study this class of problems is the allocation of advertisements to web search queries. When a user of a web search-engine types in a search keyword, she is shown a list of search results along with a list of advertisements that are pertinent to that keyword. Typically, advertisers provide bids for different keywords, and these bids are known offline. Advertisers also provide a fixed daily budget which is the maximum amount of money they are willing to spend on advertising in one day. The algorithm must allocate advertisments to arriving search keywords, and try to maximize the search engine's profit from the allocation process. We study this particular problem, called *online budgeted allocation* in Section 3. In practice, ofcourse, a search engine must take several other things into account, including the relevance of the ads to the user, click-through rates and so on.

The allocation problem can be modelled as a bipartite graph $G = (L, R, E)$ where the vertices in L (the agents) are known beforehand and the vertices in R (the items) arrive online or one-by-one. An edge (i, j) with weight b_{ij} denotes that agent i is willing to pay b_{ij} for item j. When a vertex j arrives, all the agents reveal their bids for it, and the algorithm must irreversibly allocate it to one of the agents.

Clasically, these online allocation problems have been studied in an adversarial model. In this setting, it is assumed that the arrival order of the items is selected by an adversary, and any algorithm must work for *any* arrival order. Clearly, this model is not realistic in the ad-allocation setting - search queries are not generated by an adversary. Recently, there has been a sequence of results developing and analyzing algorithms in a stochastic input model. Here, the input is assumed to be drawn from some underlying distribution. This is motivated by the fact that search engines have a veriety of statistical data about daily search queries. The most general such model that has been considered in literature is the *random order model*, in which the arrival order is a uniformly random permutation of the set of items chosen by an adversary.

We study several special cases of the general online allocation problem. We show that in the random order model, a simple *greedy* algorithm works well. Our results are based on a clever *randomized primal-dual* analysis technique of [5]. We propose the use of this technique for developing and analysing algorithms for online allocation problems in the random order model.

1.1 Definitions and Preliminaries

In the *online bipartite maching* problem, the graph $G = (L, R, E)$ is unweighted and the constraint on the agents (vertices in L) is that they can only be assigned a single item. Thus, the algorithm must construct a matching of the vertices of G. The goal is to maximize the size of the matching produced.

In the *online budgeted allocation (or AdWords)* problem, the agents in L have budgets B_i and specify bids b_{ij}. Each agent can be allocated any number of items, but the profit obtained is capped at B_i. The goal is to maximize the total profit obtained from all the agents.

In the *online vertex-weighted bipartite matching* problem, the graph $G = (L, R, E)$ has weights w_i on the agents in L. Equivalently, it has weights on edges such that all the edges incident to agent i have the same weight w_i. Every agent can be assigned a single item, so the algorithm must construct a matching. The goal is to maximize the weight of the matching produced.

Online algorithms and competitive analysis We use the notion of *competitive ratio* to measure the performance of online algorithms. Let $\mathcal{A}$ be an algorithm for a problem and let $\mathcal{I}$ be the set of all input instances. Let $\mathsf{ALG}[I]$ denote the performance of $\mathcal{A}$ on input I and let $\mathsf{OPT}[I]$ denote the best solution if the entire input was known in advance. Then, the competitive ratio of $\mathcal{A}$ is defined as $\inf_{I \in \mathcal{I}} \frac{\mathbb{E}[\mathsf{ALG}[I]]}{\mathsf{OPT}[I]}$.

Models of input In the *adversarial* input model, the graph is selected by a non-adaptive adversary, and then the items in R arrive online in an order also selected by the adversary. In the *stochastic i.i.d.* input model, it is assumed that there is a distribution on *types* of items, and items are drawn repeatedly i.i.d. from this distribution. This distribution can be either known or unknown to the algorithm. In the *random order* model, the input is chosen by an adversary, but the algorithm receives the items in a uniformly random order. Note that it can be shown that any algorithm that has a competitive ratio c in the random order model also has a competitive ratio at least c in the stochastic i.i.d models. This is one reason to focus our attention on the random order model.

1.2 Related Work

Karp et al studied the online bipartite matching problem and showed a randomized $1 - 1/e$ competitive algorithm and that no randomized algorithm can do better. In [12] Kalyanasunduram et al consider the problem of online b-matching, when the agents could be assigned upto b items. For this they gave a deterministic algorithm whose competitive ratio approaches $1 - 1/e$ as b grows large. Motivated by search advertising Mehta et al considered the online budgeted allocation or AdWords problem in [18], a generalization of the b-matching problem. For this, they gave a deterministic $1 - 1/e$ competitive algorithm under the *small bids* assumption, and showed that no randomized algorithm can do better. Goel et al in [10] considered the online budgeted allocation problem in the random order model and showed that a simple greedy algorithm is $1 - 1/e$ competitive under the small bids assumption. In the process, they also showed the first correct analysis for the result of [15] on online bipartite matching - the original proof had been observed to have a bug. Karande et al considered the online vertex-weighted bipartite matching problem in [1] and showed an optimal $1 - 1/e$ competitive algorithm in the adversarial model.

The $1-1/e$ barrier for online bipartite matching was first broken in the i.i.d. known dsitribution model in the work of Feldman et al [8]. They showed an algorithm based on exploiting the power of two choices to construct to matchings in the expected input instance is 0.67 competitive. This competitive ratio was improved to 0.699 in [4] and further improved to 0.702 as well as generalized to the unknown distribution model in [17]. In the random permutation model, the algorithm of [15] was shown to be 0.653 and 0.696 competitive in [14] and [16] respectively. For the online budgeted allocation problem in the random order model, Devanur et al in [6] show a $1 - \epsilon$ competitive algorithm for any $\epsilon > 0$ under some assumptions. Their approach which is based on learning dual variables was generalized to packing problems in [9]. Mirrokni et al showed in [19] that it is possible to simultaneously achieve good competitive ratios for both adversarial and random order models by analyzing the algorithm of [18] in the random order model.

The edge-weighted version of the online bipartite matching was considered in [7] and [11]. The former showed that in the "free disposal" model with large degrees, there is an optimal $1-1/e$ competitive algorithm. The latter showed that in the i.i.d. model there is a 0.667 competitive algorithm for the online bipartite matching problem with edge-weights.

1.3 Results and Organization

In Section 2 we introduce the *randomized primal-dual* framework of [5], by using it to analyze the RANKING algorithm of [15] for online bipartite matching and show that it's competitive ratio is $1-1/e$ in the adversarial model. By symmetry, this means that a simple GREEDY algorithm has a competitive ratio $1-1/e$ in the random order model. This forms the basis of our results.

In Section 3, we study the online budgeted allocation problem. We briefly mention the WEIGHTED-BALANCE algorithm of [18] in the adversarial model. In 3.2, we provide a simple proof via randomized primal-dual of a result of [10], showing that in the random order model, a simple GREEDY algorithm is $1-1/e$ competitive.

In Section 4, we turn our attention to the online vertex-weighted bipartite matching problem. In 4.1, we analyze the PERTURBED-GREEDY algorithm of [1] via randomized primal-dual, showing that it generalizes from the for RANKING in a straightforward way. In 4.2 we show via randomized primal-dual that in the random order model, a simply GREEDY algorithm is $1-1/e$-competitive.

In Section 5, we give a new proof using randomized primal-dual of the fact that RANKING beats $1-1/e$ for online bipartite matching in the random order model. The competitive ratio we are able to prove via this approach is inferior to those proved in [16] and [14]. However, we hope that our ideas can be extended to obtain a tighter analysis.

Finally, we conclude in Section 6 by mentioning some open problems and possible approaches to solving them.

2 Online Bipartite Matching

In the online bipartite matching problem, there is an unweighted bipartite graph $G=(L,R,E)$. The vertices in L are known beforehand and the vertices in R arrive online. When a vertex in R arrives, it must be matched to an unmatched neighbouring vertex in L, or not matched at all. A decision once made, cannot be reversed later. The goal is to maximize the expected size of the matching produced.

2.1 Online bipartite matching in the adversarial model

Recall that in the adversarial model, an (oblivious) adversary picks the graph $G = (L, R, E)$, and then we receive the vertices of R online, in the order specified by the adversary. In this section, we will show that the RANKING algorithm of [15] has a competitive ratio of $1 - 1/e$ for online bipartite matching. The RANKING algorithm is described below.

Algorithm 2.1 RANKING

1. Pick a random permutation σ of the vertices in L
2. When a vertex $j \in R$ arrives, match it to the first unmatched neighbour according to σ.

To facilitate the analysis, instead of picking a uniformly random permutation σ, we do the following. For every vertex $i \in L$, we pick $y_i \in_R [0, 1]$ randomly and independently. Now, the permutation given by the increasing order of y. Clearly, this process generates a uniformly random permutation, since each permutation is equally likely.

Notation We use RANKING(G, y) to denote the performance of RANKING on graph G when the randomness in the algorithm is fixed to be the vector y. We say that a vertex $j \in R$ is *matched at position* T if it is matched to a vertex i such that $y_i = T$.

First, we prove two properties of RANKING - *dominance* and *monotonicity*. These properties were implicitly used in all the previous proofs of Theorem 2.1 ([15], [2], [10]).

Lemma 2.1. ***(Dominance**, Lemma 3 in [2]) In* RANKING(G, y)*, suppose vertex* $j \in R$ *gets matched at position* T*. Define* G' *by introducing a new vertex* $i \in L$*. In* RANKING(G', y)*, if* $y_i \leq T$ *and* $(i, j) \in E$*, then* i *is matched.*

Proof. Suppose i was unmatched when j arrived. Then, if $y_i \leq T$ and $(i, j) \in E$, j will prefer to get matched to i than to the vertex at position T. So i will be matched. □

Lemma 2.2. ***(Monotonicity**, Lemma 4 in [2]) In* RANKING(G, y)*, suppose vertex* $j \in R$ *gets matched at position* T*. Define* G' *by introducing a new vertex* $i \in L$*. In* RANKING(G', y)*,* j *is matched at position* $\leq T$ *irrespective of* y_i *and the edges incident to* i*.*

Proof. First, note that for all vertices at position $< y_i$, $\mathsf{RANKING}(G, y)$ and $\mathsf{RANKING}(G', y)$ produce the same matching. To see this, let i' be any vertex with $y_{i'} < y_i$. If $\mathsf{RANKING}(G, y)$ matches some vertex j' to i', then in $\mathsf{RANKING}(G', y)$, j' will still prefer to get matched to i' than to i, and i' will still be available. Thus $\mathsf{RANKING}(G'y)$ will also match j' to i'.

If i is unmatched in $\mathsf{RANKING}(G', y)$, then clearly $\mathsf{RANKING}(G', y)$ is the same as $\mathsf{RANKING}(G, y)$. If i is matched in $\mathsf{RANKING}(G', y)$ to some vertex j', then in $\mathsf{RANKING}(G, y)$, either j' is unmatched or is matched to some vertex i' with $y_{i'} \geq y_i$. Then, by applying the same argument to i' it can be observed that the symmetric difference between $\mathsf{RANKING}(G, y)$ and $\mathsf{RANKING}(G', y)$ is a monotone alternating path starting at i. Thus, for every vertex $j \in R$, if it was matched at position T in $\mathsf{RANKING}(G, y)$ then it is matched at position $\leq T$ in $\mathsf{RANKING}(G', y)$.

□

Theorem 2.1. RANKING *has a competitive ratio of* $1 - 1/e$ *for online bipartite matching.*

Proof. Consider the following Primal and Dual LPs.

Primal

$\max \sum_{ij} x_{ij}$ subject to:

$\forall i : \sum_j x_{ij} \leq 1$

$\forall j : \sum_i x_{ij} \leq 1$

$\forall i, j : x_{ij} \geq 0$

Dual

$\min \sum_i \alpha_i + \sum_j \beta_j$ subject to:

$\forall (i, j) \in E : \alpha_i + \beta_j \geq 1$

$\forall i, j : \alpha_i, \beta_j \geq 0$

The primal LP is a relaxation to the offline version of the problem. x_{ij} denotes whether i is matched to j, the first constraint ensures that every vertex $i \in L$ is matched to only vertex and the second constraint ensures the same for every vertex $j \in R$. Our strategy is to maintain a solution to the primal LP according to the decisions made by RANKING. That is, whenever RANKING matches (i,j), we set $x_{ij}=1$. At the same time, we maintain a solution to the dual LP, and try to show that it is feasible and not too much larger than the primal LP. This will allow us to prove that the solution produced by the algorithm is competitive with respect to the dual optimum, and so is competitive with the optimum offline solution. When RANKING matches (i,j), we set $x_{ij}=1$ and make the following dual updates:

$$\alpha_i \leftarrow \frac{e^{y_i}}{e-1}, \quad \beta_j \leftarrow \frac{e-e^{y_i}}{e-1}$$

The primal solution is feasible, since it is a matching. When the algorithm matches (i,j), the increase in the primal objective function is 1 and the increase in the dual objective function is $\alpha_i+\beta_j = e/(e-1)$. Let P be the final primal objective value, and D be the final dual objective value, then $P \geq (1-1/e)D$. Thus, if we can show that the dual solution is feasible, then we are done.

Note however, that the the dual solution may not be feasible, but we will show that it is feasible *in expectation* over the choices of y. Fix any $(i,j)\in E$. Fix arbitrarily the values of $y_{i'}$ for all $i'\neq i$. Consider an imaginary execution of the algorithm with these values of y with i removed. Say j gets matched position L in this execution. By monotonicity j gets matched at position $\leq L$ in the actual run of the algorithm, irrespective of y_i, so $\beta_j \geq (e-e^L)/(e-1)$, conditioned on these values of y. By dominance, in the actual run of the algorithm, if $y_i \leq L$ then i is matched, so whenever $y_i \leq L, \alpha_i = e^{y_i}/(e-1)$. Thus, $\mathbb{E}[\alpha_i] \geq \int_0^l e^y/(e-1)dy = (e^L-1)/(e-1)$, conditioned on these values of y. So we have shown that $\mathbb{E}[\alpha_i+\beta_j] \geq (e-e^L)/(e-1)+(e^L-1)/(e-1)=1$. So $\mathbb{E}[\text{RANKING}] \geq (1-1/e)(\sum_i \mathbb{E}[\alpha_i]+\sum_j \mathbb{E}[\beta_j]) \geq (1-1/e)OPT$. □

2.2 Online bipartite matching in the random order model

Recall that in the random order model, the adversary picks the graph $G=(L,R,E)$ and the algorithm receives a random permutation of the vertices in R. In this section, we will show that in the random permutation model, the following simple greedy algorithm achieves a competitive ratio of $1-1/e$ for online bipartite matching.

The proof of the competitiveness of GREEDY follows from the observation of [15] that bipartite

Algorithm 2.2 GREEDY

1. Fix arbitrarily the order of the offline vertices.
2. When an online vertex j arrives, assign it to the first unmatched neighbour according to the (arbitrary) order selected.

matching is symmetric in the following sense. For any fixed order of the vertices in L and R, the matching constructed by GREEDY is the same whether L is considered the online side or R is. Thus, the expected size of the matching produced by GREEDY is the same whether the online vertices are randomly permuted or the offline vertices are. Note that RANKING is nothing but GREEDY applied to a random permutation of the offline side. So, the competitive ratio of GREEDY in the random order model is the same as that of RANKING in the adversarial model, which is $1 - 1/e$. We formally prove the symmetry property in the following lemma.

Lemma 2.3. *On any* $G = (L, R, E)$*,* GREEDY *produces the same matching whether* R *is considered online or* L*.*

Proof. Suppose GREEDY produces a matching M when L is considered online. Now, consider running an execution of GREEDY with R considered online. Let $r_1, r_2 ... r_n$ be the vertices in R. r_1 is matched to the same vertex in our simulation as in M, since no vertex above it has an edge to r_1. We will prove by induction that this holds true for all vertices in R. Suppose the matching produced by GREEDY on $r_1, r_2 ... r_{t-1}$ is consistent with M. Suppose r_t was matched to some vertex $i \in L$ in M. In our execution, when r_t arrives, i must be unmatched since by the inductive hypothesis, the matching produced on $r_1, ... r_{t-1}$ is consistent with M. So r_t either gets matched to i, or above i. Suppose there is some vertex i' above i in L such that r_t has an edge to i' and i' is unmatched when r_t arrives. Then, in M, i' is not matched to $r_1, ... r_{t-1}$, so it will get matched to r_t because it arrives before i (when L is considered online). This is a contradiction. Thus, the only possibility is that r_t is matched to i in our execution. This completes the proof. □

Theorem 2.2. GREEDY *has a competitive ratio of* $1 - 1/e$ *for online bipartite matching in the random order model.*

Proof. Follows from the above discussion and Lemma 2.3. □

2.3 Discussion

Observe that we can use the randomized primal-dual method to prove Theorem 2.2. In the random order model, we can think of assigning each online vertex a random $z_j \in_R [0,1]$ uniformly and independently. If we treat the vertices as arriving in increasing order of z, then each permutation of the vertices is equally likely. Together with appropriately defined versions of dominance and monotonicity, we can use the randomized primal-dual method and essentially repeat the analysis of Theorem 2.1 to prove Theorem 2.2.

This observation leads us to question whether similar ideas can be applied to other problems in the random order model. The answer is yes - In the next few sections, we will apply repeatedly this idea of analyzing greedy algorithms in the random order model using randomized primal-dual to different versions of the online allocation problem.

3 Online Budgeted Allocation

Recall that in the online budgeted allocation problem, there is a graph $G = (L, R, E)$ with weights b_{ij} on the edges. L, the set of bidders or agents, is known in advance. R consists of items that arrive online. Bidder i bids b_{ij} on the item j. Moreover, each bidder specifies a budget B_i. An algorithm for the online budgeted allocation problem must allocate, irreversibly, each arriving item to a bidder. The profit that the algorithm gets from assigning item j to bidder i is i's bid b_{ij}.

3.1 Online budgeted allocation in the adversarial model

In the adversarial model, the graph $G = (L, R, E)$ is chosen by an adversary and so is the order that the items arrive in. The problem of online budgeted allocation in the adversarial model was first introduced and studied by Mehta et al. in [18]. They gave a deterministic $1-1/e$ competitive algorithm under the assumption that all the bids are much smaller than the corresponding budgets, that is, $\max_{i,j} \frac{b_{ij}}{B_i}$ approaches 0. The algorithm, WEIGHTED-BALANCE, is described below.

Algorithm 3.1 WEIGHTED-BALANCE

1. For every bidder i, let r_i be the fraction of its budget already exhausted.
2. When item j arrives, assign it to the bidder i maximizing $b_{ij}(1 - e^{r_i - 1})$

Intuitively, it is good to allocate each arriving item to the bidder that has a large bid for it.

However, simply matching in a greedy way does only achieves a competitive ratio of 1/2. By being greedy, we may exhaust a bidder's budget too soon and thus not be able to allocate to it items arriving in the future. Thus, if a bidder's budget is close to being exhausted, the algrothim should be less likely to allocate more items to it. So there is a trade-off between the bidder's bid and the fraction of its budget already exhausted. The optimal trade-off is captures by the WEIGHTED-BALANCE algorithm.

The original proof of Mehta et al. used a "tradeoff-revealing LP" approach. In [3], Buchbinder et al. give a simpler proof using the primal-dual method. Note that the WEIGHTED-BALANCE algorithm is deterministic, and so is somewhat different from the algorithms we study, in which either the algorithm has randomness or the input does.

3.2 Online budgeted allocation in the random permutation model

In the random permutation model the graph $G = (L, R, E)$ is chosen by an adversary, but the algorithm receives a random permutation of the items R. The online budgeted allocation problem in the random order model was first studied by Goel et al in [10]. They first showed that in the random order model, GREEDY is $1 - 1/e$ competitive for online bipartite matching and then generalized this to online budgeted allocation, also under the "small bids" assumption. The assumption is that the bids b_{ij} are much smaller than the budgets B_i. More precisely, let $R_{max} = \max_{i,j} \frac{b_{ij}}{B_i}$. Then, the competitive ratio of GREEDY approaches $1 - 1/e$ as R_{max} approaches 0.

In this section, we re-prove their result using the randomized primal-dual framework. Our proof is simpler than that of [10], which makes use of a clever but fairly involved factor revealing LP. We believe it is also more amenable to being extended to other and more general problems. The GREEDY algorithm is described below.

Algorithm 3.2 GREEDY

When an item j arrives, assign it to the bidder i which has the maximum bid b_{ij} for j among the ones who haven't already exhausted their budget, even if this were to exceed i's budget.

To facilitate the analysis, we will pick for each $j \in R, z_j \in_R [0, 1]$ randomly and independently, and then assume that the items in R arrive in increasing order of z. Note that every permutation of the items is equally likely.

Notation We use GREEDY(G, z) to denote the performance of GREEDY on graph G when the input arrives according to the vector z. We say that a bidder $i \in L$ *exhausts its budget at position* T if an item j is allocated to it with $z_j = T$, and at that point, i has spent all its budget.

First we will prove the corresponding *dominance* and *monotonicity* properties for the online budgeted allocation problem, which were also implicitly used in [10].

Lemma 3.1. ***(Dominance)*** *In* $\mathsf{GREEDY}(G, z)$*, suppose bidder* i*'s budget gets exhausted at position* T*. Define* G' *by introducing a new item* $j \in R$*. In* $\mathsf{GREEDY}(G', z)$*, if* $z_j \leq T$ *and* $b_{ij} > 0$*, then* j *is allocated to some bidder* i' *with* $b_{i'j} \geq b_{ij}$*.*

Proof. Before j arrives, $\mathsf{GREEDY}(G, z)$ produces the same allocation as $\mathsf{GREEDY}(G', z)$. So when j arrives, i's budget is not exhausted since $z_j \leq T$. This means that j could be allocated to i and by definition of GREEDY it is matched to the highest available bidder, so $b_{i'j} \geq b_{ij}$. □

Lemma 3.2. *(**Monotonicity**, Lemma 3.1 in [10]) In* $\mathsf{GREEDY}(G, z)$*, suppose bidder* i*'s budget gets exhausted at position* T*. Define* G' *by introducing a new item* $j \in R$*. In* $\mathsf{GREEDY}(G', z)$*,* i*'s budget is exhausted at position* $\leq T$*, irrespective of* z_j *and the bids for item* j*.*

Proof. First, note that for all items at position $< z_j$, $\mathsf{GREEDY}(G, z)$ and $\mathsf{GREEDY}(G', z)$ produce the same allocation, since they arrive before j.

If j is not allocated in $\mathsf{GREEDY}(G', z)$, then clearly $\mathsf{GREEDY}(G', z)$ is the same as $\mathsf{GREEDY}(G, z)$. Suppose j is allocated in $\mathsf{GREEDY}(G', z)$ to some bidder i' and in $\mathsf{GREEDY}(G, z)$ i' budget gets exhausted at position T'. Then, in $\mathsf{GREEDY}(G', z)$, i''s budget gets exhausted at position $\leq T'$. This means that there may be some items in R that were allocated to i' in $\mathsf{GREEDY}(G, z)$ but not in $\mathsf{GREEDY}(G', z)$ since i' exhausted its budget before they arrived. Then, we can apply the same argument to those items to conclude that for every bidder $i \in L$, if its budget gets exhausted at position T in $\mathsf{GREEDY}(G, z)$, then its budget gets exhausted at position $\leq T$ in $\mathsf{GREEDY}(G', z)$. □

Theorem 3.1. *The competitive ratio of* GREEDY *approaches* $1 - 1/e$ *as* R_{max} *approaches 0.*

Proof. We will use the Randomized primal-dual framework to prove the theorem.

Primal

$\max \sum_{i,j} b_{ij} x_{ij}$ subject to:

$\forall i : \sum_j b_{ij} x_{ij} \leq B_i$

$\forall j : \sum_i x_{ij} \leq 1$

$\forall i, j : x_{ij} \geq 0$

Dual

$\min \sum_i \alpha_i B_i + \sum_j \beta_j$ subject to:

$\forall i, j : \alpha_i b_{ij} + \beta_j \geq b_{ij}$

$\forall i, j : \alpha_i, \beta_j \geq 0$

The primal LP is a relaxation to the offline version of the problem. x_{ij} denotes whether j is allocated to i, the first constraint ensures that no bidder i exceeds its budget and the second constraint ensures that each item j is allocated to at most one bidder. Following the pattern of previous proofs, we will update the dual variables as the algorithm progresses, ensuring that the increase in the dual objective function is not too much. We will allow the dual constraints to be violated, but will show that they are satisfied on average. Consider the following update rules. When item j arrives and is assigned to bidder i, set $x_{ij} = 1$, regardless of whether i's budget is exceeded or not, and set

$$\beta_j = b_{ij} \frac{e^{y_j}}{e-1}, \quad \alpha_i \leftarrow \alpha_i + \frac{b_{ij}}{B_i} \frac{e - e^{y_j}}{e-1}$$

First, notice that the primal solution thus obtained is almost feasible. Each budget constraint is violated at most once, and once the budget is exceeded no further items are assigned to that bidder, by definition of GREEDY. We will account for this slight violation in our analysis.

At every step, the increase in the primal objective is $\Delta P = b_{ij}$, and the increase in dual objective is $\Delta D \leq \Delta \alpha_i B_i + \beta_j = b_{ij} \frac{e - e^{y_j}}{e-1} + b_{ij} \frac{e^{y_j}}{e-1} = b_{ij} \frac{e}{e-1}$. So, the increase in the dual objective function is atmost $\frac{e}{e-1}$ times the increase in the primal objective function. This means that if the dual solution is feasible, then the primal solution is at most a factor $1 - 1/e$ less than the optimal offline solution.

Now, we show that the dual solution is feasible on average. That is, the expected values (over the choices of y) of all the dual variables together satisfy the dual constraints. Moreover, since for any choice of y, the primal solution is $1-1/e$-competitive with respect to the dual solution, this also holds in expectation. So the algorithm is $1-1/e$ competitive in expectation.

Fix any edge (i,j) and aribtrary values of y_j for all $j \neq j'$. Consider an imaginary execution of GREEDY, with j removed, and suppose that i's budget is exhausted when it is assigned item j' with $y_{j'} = L$. If i's budget is never exhausted, $L = 1$. Now, insert j back and vary y_j from 1 to 0. By dominance, when $y_j \leq L$, $\beta_j \geq b_{ij} \frac{e^{y_j}}{e-1}$, so $\mathbb{E}[\beta_j] \geq b_{ij} \frac{e^L-1}{e-1}$. By monotonicity, i's budget is exhausted when it is matched to some j' with $y_{j'} \leq L$. Let $j_1, j_2 ... j_r = j'$ be the items assigned to i, then $\alpha_i = \sum_{k=1}^{r} \frac{b_{ij_k}}{B_i} \frac{e-e^{y_{j_k}}}{e-1} \geq \frac{e-e^{y_{j'}}}{e-1} \geq \frac{e-e^L}{e-1}$. So $\mathbb{E}[\alpha_i b_{ij} + \beta_j] \geq b_{ij} \frac{e-e^L}{e-1} + b_{ij} \frac{e^L-1}{e-1} = b_{ij}$, thus showing that the dual is feasible on average. This completes the proof.

□

3.3 Discussion

We showed that the randomized primal-dual framework is useful for analyzing a GREEDY algorithm for the online budgeted allocation problem in the random order model. The result we show in Section 3.2 was first shown in [10], however, we hope that the simplicity of our analysis allows it to be extended to more general problems, as we discuss in the final section.

Note that in the random order model, GREEDY is not optimal. Recently, in [19], Mirrokni et al showed that the WEIGHTED-BALANCE algorithm of [18] has a competitive ratio of at least 0.76, albeit under some stronger assumptions.

4 Online Vertex-Weighted Matching

Recall that in the online vertex-weighted bipartite matching problem, there is a bipartite graph $G = (L, R, E)$ with weights w_i on the *offline* vertices L. The weights w are known in advance. The vertices in R arrive online and reveal the edges incident to them when they arrive. The algorithm must choose to match a vertex when it arrives, or to not match it all, and the decision cannot be reversed in the future. If an incoming vertex is matched to $i \in L$, then the weight of the match is

w_i. The goal is to maximize the total weight of the matching produced.

4.1 Online vertex-weighted bipartite matching in the adversarial model

In the adversarial model, an (oblivious) adversary picks the graph $G = (L, R, E)$, and then the algorithm receive the vertices of R online, in the order specified by the adversary. The online vertex-weighted bipartite matching problem in the adversarial model was first studied by Aggarwal et al in [1]. They devise an algorithm, PERTURBED-GREEDY and show that it is $1 - 1/e$ competitive. The PERTURBED-GREEDY algorithm is described below.

Algorithm 4.1 PERTURBED-GREEDY

1. For each offline vertex i, pick a random $y_i \in_R [0, 1]$ uniformly and indepependently.
2. When an online vertex j arrives, match it to the vertex i maximizing $w_i(1 - e^{y_i - 1})$.

Intuitively, matching to a high-weight vertex is good. However, it is easy to see that simply matching greedily to the highest weight neighbour does not do better than 1/2. Moreover, any algorithm for the weighted problem must also work for the unweighted case. Thus, the right algorithm should find a tradeoff between the greedy strategy and the RANKING algorithm. As we will show below (via Randomized primal-dual), the right tradeoff is embodied by PERTURBED-GREEDY.

Notation We use PERTURBED-GREEDY(G, y) to denote the performance of PERTURBED-GREEDY on graph G when the randomness in the algorithm is fixed to be the vector y. We say that a vertex $j \in R$ is matched at position T if it is matched to a vertex i such that $w_i(1 - e^{y_i - 1}) = T$. Notice that because of the way we have defined it, an incoming vertex tries to maximize its "position" rather than minimize it as in online bipartite matching.

First, we prove *dominance* and *monotonicity* properties of the PERTURBED-GREEDY algorithm, originally shown in [11].

Lemma 4.1. *(**Dominance**, Lemma 5 in [11]) In* PERTURBED-GREEDY(G, y)*, suppose vertex $j \in R$ gets matched at position T (note: have to define T appropriately). Define G' by introducing a new vertex $i \in L$. In* PERTURBED-GREEDY(G', y)*, if $w_i(1 - e^{y_i - 1}) \geq T$ and $(i, j) \in E$, then i is matched.*

Proof. Suppose when j arrives, i is unmatched. Then, if $w_i(1 - e^{y_i - 1}) \geq T$ and $(i, j) \in E$, j would prefer to get matched to i than to the vertex at position T. Hence, i will be matched. □

Lemma 4.2. *(**Monotonicity**, Lemma 5 in [11]) In* PERTURBED-GREEDY(G, y)*, suppose vertex $j \in R$ gets matched at position T. Define G' by introducing a new vertex $i \in L$. In* PERTURBED-GREEDY(G', y)*, j is matched at position $\geq T$ irrespective of y_i, w_i and the edges incident to i.*

Proof. First, note that for all vertices at position $> w_i(1 - e^{y_i - 1})$, PERTURBED-GREEDY(G, y) and PERTURBED-GREEDY(G', y) produce the same matching. To see this, let i' be any vertex at position $w_{i'}(1 - e^{y_{i'} - 1}) > w_i(1 - e^{1 - y_i})$. If PERTURBED-GREEDY$(G, y)$ matches some vertex j' to i', then in PERTURBED-GREEDY(G', y), j' will still prefer to get matched to i' than to i, and i' will still be available. Thus PERTURBED-GREEDY$(G'y)$ will also match j' to i'.

If i is unmatched in PERTURBED-GREEDY(G', y), then clearly PERTURBED-GREEDY(G', y) is the same as PERTURBED-GREEDY(G, y). If i is matched in PERTURBED-GREEDY(G', y) to some vertex j', then in PERTURBED-GREEDY(G, y), j' was either unmatched or matched to some vertex i' at position $w_{i'}(1 - e^{1 - y_{i'}}) < w_i(1 - e^{1 - y_i})$. Then, by applying the same argument to i' it can be observed that the symmetric difference between PERTURBED-GREEDY(G, y) and PERTURBED-GREEDY(G', y) is a monotone alternating path starting at i. Thus, for every vertex $j \in R$, if it was matched at position T in PERTURBED-GREEDY(G, y) then it is matched at position $\geq T$ in PERTURBED-GREEDY(G', y). □

Theorem 4.1. *(Theorem 1.1 in [1])* PERTURBED-GREEDY *has a competitive ratio of $1 - 1/e$ for online vertex-weighted bipartite matching.*

Proof. We will use the randomized primal-dual framework. Consider the following Primal and Dual LPs.

Primal

$\max \sum_{ij} w_i x_{ij}$ subject to:

$\forall i : \sum_j x_{ij} \leq 1$

$\forall j : \sum_i x_{ij} \leq 1$

$\forall i, j : x_{ij} \geq 0$

Dual

min $\sum_i \alpha_i + \sum_j \beta_j$ subject to:

$\forall (i,j) \in E : \alpha_i + \beta_j \geq w_i$

$\forall i, j : \alpha_i, \beta_j \geq 0$

The primal LP is just the weighted version of the LP used in Section 2, and so is a relaxation of the offline problem. Whenever the algorithm matches (i,j), we set $x_{ij} = 1$ and make the following dual updates:

$$\alpha_i = \frac{w_i e^{y_i}}{e-1}, \quad \beta_j = \frac{w_i(e - e^{y_i})}{e-1}$$

Clearly, the primal solution is feasible since the algorithm constructs a matching. When the algorithm matches (i,j), the increase in the primal objective function is w_i, while the increase in the dual objective function is $\alpha_i + \beta_j = w_i e/(e-1)$. Thus, if P and D denote the final primal and dual objective function values, then $P \geq (1 - 1/e)D$. So if we can show that the dual solution is feasible, we are done.

Now, fix any $(i,j) \in E$. Fix arbitrarily the values $y_{i'}$ for all $i' \neq i$. Consider an imaginary execution of the PERTURBED-GREEDY algorithm with these values of y and vertex i removed. Suppose in this execution, j were matched to i', and let L be such that $w_i(e-e^L)/(e-1) = w_{i'}(e-e^{y_{i'}})/(e-1)$ if such an L exists, otherwise $L = 0$. If j is unmatched, then $L = 1$. Now, by monotonicity, in the actual execution of the algorithm, irrespective of y_i, $\beta_j \geq w_i(e - e^L)/(e-1)$. By dominance, i is matched whenever $y_i \leq L$, so $\mathbb{E}[\alpha_i] \geq \int_0^L w_i e^y/(e-1)dy = w_i(e^L - 1)/(e-1)$. So $\mathbb{E}[\alpha_i + \beta_j] \geq w_i$. Thus, the dual solution is feasible in expectation, so $\mathbb{E}[\text{PERTURBED-GREEDY}] \geq (1 - 1/e)OPT$.

□

4.2 Online vertex-weighted bipartite matching in the random order model

In the random order model, the adversary picks the graph $G = (L, R, E)$ and the algorithm receives a random permutation of the vertices in R. In this section, we will show that a simple algorithm, GREEDY, achieves a competitive ratio of $1 - 1/e$ for online vertex-weighted bipartite matching in the random order model. The algorithm is described below.

Algorithm 4.2 GREEDY

When an online vertex j arrives, match it to the unmatched neighbour with highest weight w_i.

To facilitate the analysis, we will pick for each $j \in R$, $z_j \in_R [0, 1]$ randomly and independently, and then assume that the vertices in R arrive in increasing order of z. Note that every permutation of the vertices is equally likely.

Notation We use $\mathsf{GREEDY}(G, z)$ to denote the performance of GREEDY on graph G when the input arrives according to the vector z. We say that a vertex $i \in L$ is matched at position T if it is matched to a vertex j such that $z_j = T$.

First, we prove the analogous versions of Monotonicity and Dominance.

Lemma 4.3. ***(Dominance)*** *In* $\mathsf{GREEDY}(G, z)$*, suppose vertex* $i \in L$ *gets matched at position* T*. Define* G' *by introducing a new vertex* $j \in R$*. In* $\mathsf{GREEDY}(G', z)$*, if* $z_j \leq T$ *and* $(i, j) \in E$*, then* j *is matched to some vertex* i' *with* $w_{i'} \geq w_i$*.*

Proof. Before j arrives, $\mathsf{GREEDY}(G, z)$ and $\mathsf{GREEDY}(G', z)$ produce the same matching. So when j arrives, i is unmatched since $z_j \leq T$. Since $(i, j) \in E$, and by definition of GREEDY, j gets matched to the highest weight available vertex i', $w_{i'} \geq w_i$. □

Lemma 4.4. ***(Monotonicity)*** *In* $\mathsf{GREEDY}(G, z)$*, suppose vertex* $i \in L$ *gets matched at position* T*. Define* G' *by introducing a new vertex* $j \in R$*. In* $\mathsf{GREEDY}(G', z)$*,* i *is matched at position* $\leq T$ *irrespective of* z_j *and the edges incident to* j*.*

Proof. First, note that for all vertices at position $< z_j$, $\mathsf{GREEDY}(G, z)$ and $\mathsf{GREEDY}(G', z)$ produce the same matching, since they arrive before j.

If j is unmatched in $\mathsf{GREEDY}(G', z)$, then clearly $\mathsf{GREEDY}(G', z)$ is the same as $\mathsf{GREEDY}(G, z)$. If j is matched in $\mathsf{GREEDY}(G', z)$ to some vertex i', then in $\mathsf{GREEDY}(G, z)$, i' was either unmatched or matched to some vertex j' with $z_{j'} > z_j$. Then, by applying the same argument to j' it can be observed that the symmetric difference between $\mathsf{GREEDY}(G, z))$ and $\mathsf{GREEDY}(G', z)$ is a monotone alternating path starting at j. Thus, for every vertex $i \in L$, if it was matched at position T in $\mathsf{GREEDY}(G, z)$ then it is matched at position $\leq T$ in $\mathsf{GREEDY}(G', z)$. □

Theorem 4.2. GREEDY *has a competitive ratio of* $1 - 1/e$ *for online vertex-weighted bipartite matching in the random permutation model.*

Proof. We will use the same Primal-Dual LP pair as in the previous section. We reproduce it below.

Primal

$\max \sum_{ij} w_i x_{ij}$ subject to:

$\forall i : \sum_j x_{ij} \leq 1$

$\forall j : \sum_i x_{ij} \leq 1$

$\forall i, j : x_{ij} \geq 0$

Dual

$\min \sum_i \alpha_i + \sum_j \beta_j$ subject to:

$\forall (i, j) \in E : \alpha_i + \beta_j \geq w_i$

$\forall i, j : \alpha_i, \beta_j \geq 0$

When the algorithm matches (i, j) we set $x_{ij} = 1$ and make the following dual updates:

$$\alpha_i \leftarrow w_i(e - e^{z_j})/(e - 1), \quad \beta_j \leftarrow w_i e^{z_j}/(e - 1)$$

Note that the primal solution is feasible since it is a matching. When the algorithm matches (i, j), the increase in the primal objective function is w_i and the increase in the dual objective function is $w_i[(e - e^{z_j})/(e - 1) + e^{z_j}/(e - 1)] = w_i[e/(e - 1)]$. So if P and D denote the final primal and dual objective values respectively, then $P \geq (1 - 1/e)D$. So if we can show that the dual solution is feasible, then we have shown that the algorithm is (1-1/e)-competitive.

Fix any edge $(i, j) \in E$. Fix arbitrarily the values of $z_{j'}$ for all $j' \neq j$. Consider an imaginary execution of GREEDY with these values of z and with j removed. Suppose that in this execution i got matched at position L. Then by monotonicity, in the actual execution of the algorithm with these values of z, i is matched at position $\leq L$ irrespective of z_j. So $\alpha_i \geq w_i(e - e^L)/(e - 1)$. By dominance, when $z_j \leq 1$, j is matched to a vertex i' with $w_{i'} \geq w_i$. So conditioned on these values of $z_{j'}$, $\mathbb{E}[\alpha_i] \geq \int_0^L w_i e^z/(e - 1)dz = w_i(e^L - 1)/(e - 1)$. Thus, $\mathbb{E}[\alpha_i + \beta_j] \geq w_i$. This proves that the dual is feasible in expectation and so completes the proof.

□

4.3 Discussion

The RANKING algorithm for the unweighted online bipartite matching problem and the PERTURBED-GREEDY algorithm for the vertex-weighted version are closely related. In particular, RANKING uses a uniform distribution over permutations of the offline side, whereas PERTURBED-GREEDY uses a non-uniform distribution. The analysis of RANKING for online unweighted bipartite matching in Section 2.1 generalizes in a straightforward fashion to PERTURBED-GREEDY for the vertex-weighted case. In fact, the PERTURBED-GREEDY algorithm can be deduced from trying to repeat the randomized primal-dual analysis for the vertex-weighted case. Moreover, in the random order model the analysis of GREEDY in the unweighted case generalizes in an even simpler fashion to the weighted case. This hints that perhaps randomized primal-dual is the right tool to use when generalizing from the unweighted to the vertex-weighted case. It also raises the question whether there is a "reduction" from the vertex-weighted to the unweighted case, which would allow us to generalize all known results for the latter to the former.

5 Online bipartite matching: beating $1 - 1/e$

The online bipartite matching problem was studied in Section 2. In the random order model, a simple GREEDY algorithm has a competitive ratio $1-1/e$. In this section, we try to improve upon this factor of $1 - 1/e$ in the random order model. In particular, we will show that the RANKING algorithm has a competitive ratio strictly better than $1 - 1/e$. In contrast, it was shown in [15] that no randomized algorithm can do better than $1 - 1/e$ in the adversarial model.

The problem of online bipartite matching in the random order model was first considered by Karande et al ([14]) and Mahdian et al ([16]). They showed that the competitive ratio of RANKING in this model is at least 0.653 and 0.696 respectively. It was also shown in [14] that the competitive ratio of RANKING in the random permutation model was no more than 0.727. In [16], the authors use a strongly factor-revealing LP approach, where they write a family of factor-revealing LPs and then relax them to get a family of LPs with the property that the solution to any of them is a lower bound on the competitive ratio of the algorithm. Then, any of these LPs can be solved computationally to get a lower bound on the competitive ratio. On the other hand, the techniques of [14] are analytical. In fact, our proof in this section heavily uses their ideas. For completeness, we restate the RANKING algorithm below.

Algorithm 5.1 RANKING

1. Pick a random permutation σ of the vertices in L
2. When a vertex $j \in R$ arrives, match it to the first unmatched neighbour according to σ.

As before, for every vertex $i \in L$, we pick $y_i \in_R [0, 1]$ randomly and independently. Since now both sides of the bipartite graph are random permutations, we also pick for every $j \in R$, $z_j \in_R [0, 1]$.

Notation We use RANKING(G, y, z) to denote the performance of RANKING on graph G when the order of vertices in L (resp. R) is given by the vector y (resp. z) respectively. We say that a vertex $j \in R$ (resp. $i \in L$) is *matched at position* T if it is matched to a vertex i' (resp. j') such that $y_{i'} = T$ (resp. $z_{j'} = T$). For a given vector $y \in [0, 1]^n$, we use y_{-i} to denote the same vector with the i'th vertex removed.

First, we show a property of the RANKING algorithm that holds even in the adversarial model (although we haven't used it so far).

Lemma 5.1. *(Lemma 2 of [2]) Without loss of generality we can assume that in the worst case example for* RANKING $|L| = |R| =$ OPT.

Proof. Consider any graph $G = (L, R, E)$, and suppose there is a vertex $j \in R$ that is unmatched in OPT. We will show that by removing this vertex, the performance of RANKING only gets worse. Define G' as G with vertex j removed. Then, as shown in Lemma 2.2, the symmetric difference between the matchings RANKING(G, y) and RANKING(G', y) for any fixed y, is a single alternating path starting at j. Thus, the size of the matching produced by RANKING(G', y) is either the same or smaller than the size of the matching produced by RANKING(G). By symmetry, this also holds if there were a vertex $i \in L$ that was unmatched in OPT. Thus, by removing vertices that are not matched in OPT, the competitive ratio only gets worse and so we can assume that the graph has a perfect matching in the worst case. □

Next, we prove an important property that distinguishes RANKING in the random order model from RANKING in the adversarial model.

Lemma 5.2. *(Lemma 3 of [14]) Without loss of generality, we can assume that the worst case example for* RANKING *in the random order model is symmetric.*

Proof. We will show that for any graph $G = (L, R, E)$, if $\mathsf{RANKING}(G)$ has a competitive ratio c, then there is another graph G' that is symmetric and $\mathsf{RANKING}(G')$ also has a competitive ratio c. Let $G_1 = (L_1, R_1, E_1)$ and $G_2 = (L_2, R_2, E_2)$ be two copies of G. Consider $G' = (L_1 \cup R_2, L_2 \cup R_1, E_1 \cup E_2)$ which is symmetric. Since, G_1 and G_2 are disjoint, the competitive ratio of $\mathsf{RANKING}(G')$ is just the average of the competitive ratios of $\mathsf{RANKING}(G_1)$ and $\mathsf{RANKING}(G_2^T)$ where G_2^T is G_2 with its left and right bipartitions flipped. By symmetry the competitive ratios of $\mathsf{RANKING}(G_2^T)$ and $\mathsf{RANKING}(G_2)$ are the same. So the competitive ratio of $\mathsf{RANKING}(G')$ is the average of the competitive ratios of $\mathsf{RANKING}(G_1)$ and $\mathsf{RANKING}(G_2)$ which is c. □

We will characterize the matching produced by any algorithm in terms of different types of matches - *below* matches, *above* matches and *perfect* matches. For the purpose of the following definitions, fix an optimal matching OPT.

Definition 5.1. ***(Events,*** $B, \hat{B}, AA, P$***-events)*** *An event is a tuple (i, y, z) (or (y, z, j)) where $y, z \in [0, 1]^n$ and $i \in L$ ($j \in R$). In other words, it specifies the order of vertices in L and R, along with a particular vertex that we want to reason about. An event completely specifies the position and match of its vertex.*

Suppose $(i, j) \in$ OPT. An event (i, y, z) is a B-event if i is matched at position $> z_j$ in $\mathsf{RANKING}(G, y, z)$*. That is, i is matched at a position that is below where its optimal neighbour appears. In this case, by the Dominance property of* $\mathsf{RANKING}$*, j must be matched in* $\mathsf{RANKING}(G, y, z)$*.*

Similarly, an event (y, z, j) is a $\hat{B}$-event if j is matched at position $> y_i$ in $\mathsf{RANKING}(G, y, z)$*. That is, j is matched at a position that is below where its optimal neighbour appears. In this case, i must be matched in* $\mathsf{RANKING}(G, y, z)$*.*

An event (y, z, i) is an AA-event if i is matched at position $< z_j$ and j is matched at position $< y_i$ in $\mathsf{RANKING}(G, y, z)$*. In AA-events, by definition, both i and j are matched.*

An event (y, z, i) is a P-event if i is matched to j in $\mathsf{RANKING}(G, y, z)$*. An event (y, z, j) is a $\hat{P}$-event if j is matched to i in* $\mathsf{RANKING}(G, y, z)$*. Clearly, P and $\hat{P}$-events are the same, so we call them P-events. In P-events also, both i and j are matched.*

Define b to be the expected number (over the choices of y and z) of B-events scaled down by n and similarly define $\hat{b}, p$. The next lemma shows that without loss of generality, we can assume that in the worst case there is a symmetry in the B and $\hat{B}$ events.

Lemma 5.3. *Without loss of generality, we can assume that in the worst case, $b = \hat{b}$.*

Proof. From Lemma 5.2, we know that in the worst case the graph is symmetric across the bipartition. Let $i \in L$ and $i' \in R$ be a symmetric pair and suppose (i, y, z) be an B-event. Consider the event (z, y, i'). By Lemma 2.3, the matching produced is the same regardless of the online and offline sides. Then, if we consider L as the online side and by the symmetry of the graph, (z, y, i') is an $\hat{B}$-event. Thus, the probability of i being involved in a B-event is the same as that of i' being involved in an $\hat{B}$-event, so $b = \hat{b}$. □

Let $\mathbb{E}[\mathsf{RANKING}]$ denote the expected size of the matching produced by $\mathsf{RANKING}$. The next two lemmas will show two different lower bounds on $\mathbb{E}[\mathsf{RANKING}]$.

Lemma 5.4. $\mathbb{E}[\mathsf{RANKING}] = \frac{1}{2} + b + \frac{aa}{2} + \frac{p}{2}$

Proof. Let $G = (L, R, E)$ be such that $|L| = |R| = \mathsf{OPT} = n$. First we will show that for any y, z, the size of the matching $\mathsf{RANKING}(G, y, z)$ is exactly $\frac{n}{2} + \frac{B}{2} + \frac{\hat{B}}{2} + \frac{AA}{2} + \frac{P}{2}$ where by an abuse of notation we let $B, \hat{B}, AA$ and P be the number of B, $\hat{B}, AA$ and P-events in $\mathsf{RANKING}(G, y, z)$ respectively. Fix an optimal matching OPT. For every edge $(i, j) \in \mathsf{OPT}$, we know that at least one of i, j are matched since $\mathsf{RANKING}$ constructs a maximal matching. Thus, at least n vertices are matched, which gives the $\frac{n}{2}$ term. For every B-event (i, y, z), both i and j are matched, so we get an additional matched vertex, which means an additional $\frac{B}{2}$ matched edges and the same applies to AA and P-events, so we get an additional $\frac{AA}{2} + \frac{P}{2}$ matched edges. Note that it cannot be the case (i, y, z) is a B-event and (y, z, j) is a $\hat{B}$-event because of the greediness of $\mathsf{RANKING}$. Thus, we also get an additional $\frac{\hat{B}}{2}$ edges from the $\hat{B}$-events. Finally, note that these are the only ways in which both vertices i and j can be matched in $\mathsf{RANKING}(G, y, z)$.

By averaging over all choices of y, z we get that the expected competitive ratio of $\mathsf{RANKING}$ is $\frac{1}{2} + \frac{b}{2} + \frac{\hat{b}}{2} + \frac{aa}{2} + \frac{p}{2}$. By Lemma 5.3, we know that $b = \hat{b}$, so $\mathbb{E}[\mathsf{RANKING}] = \frac{1}{2} + b + \frac{aa}{2} + \frac{p}{2}$. □

It can be observed that in the worst case example for $\mathsf{RANKING}$ in the adversarial model, it is only the $\hat{B}$-events that make a contribution to increase the competitive ratio beyond 1/2. The number of B and P events decreases to 0. The key observation in [14] was that because of the symmetry property in the random order model, if there were no P-events, then there would be many B and $\hat{B}$-events and so the competitive ratio would be very large. This means that there must be some P-events. Their proof essentially shows a trade-off between the number of P and B-events and the competitive ratio. This is also the essence of the rest of our proof.

Next, via randomized primal-dual, we show that if there are a significant number of P and B matches, then the competitive ratio is better than $1 - 1/e$. The final step involves a trade-off between the following lemma and the previous one.

Lemma 5.5. $\mathbb{E}[\mathsf{RANKING}] \geq (1 - 1/e)(1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1})$

Proof. We will use the same Primal-Dual LP pair as in Section 2. We reproduce them here:

Primal

$\max \sum_{ij} x_{ij}$ subject to:

$\forall i : \sum_j x_{ij} \leq 1$

$\forall j : \sum_i x_{ij} \leq 1$

$\forall i, j : x_{ij} \geq 0$

Dual

$\min \sum_i \alpha_i + \sum_j \beta_j$ subject to:

$\forall (i, j) \in E : \alpha_i + \beta_j \geq 1$

$\forall i, j : \alpha_i, \beta_j \geq 0$

When the algorithm matches (i, j), we set $x_{ij} = 1$. For now, consider using the same dual update rules as before:

$$\alpha_i \leftarrow \frac{e^{y_i}}{e-1}, \quad \beta_j \leftarrow \frac{e - e^{y_i}}{e-1}$$

Now, fix any $(i, j) \in E$. Fix arbitrarily the values $z_{j'}$ for all j' and $y_{i'}$ for all $i' \neq i$. Consider an imaginary execution of $\mathsf{RANKING}$ with these values of y, z with i removed. Suppose that j was matched at position L in this imaginary execution.

By monotonicity, j is matched at position $\leq L$ in the actual execution of $\mathsf{RANKING}$ with these values of y, z, irrespective of the value of y_i. So $\beta_j \geq (e - e^L)/(e - 1)$, irrespective of y_i. Note however, that when $y_i \leq L$, it is possible that j is matched to i. Clearly, if such a perfect match

(i,j) occurs when $y_i = y$, then it also occurs in the range $[y, L]$, since in this range j prefers to get matched to i than at position L. So there is a (possibly empty) range of values for y_i above L in which a perfect match occurs. Call the length of the range $p_i(y_{-i}, z)$. This is the probability of vertex i being involved in a perfect match conditioned on the values of y_{-i} and z chosen. Then, we see that there is some gain in $\mathbb{E}[\beta_j|y_{-i}, z]$, given by

$$\mathbb{E}[\beta_j|y_{-i}, z] \geq \int_0^{L-p_i(y_{-i},z)} \frac{e-e^L}{e-1}dy + \int_{L-p_i(y_{-i},z)}^{L} \frac{e-e^y}{e-1}dy + \int_L^1 \frac{e-e^L}{e-1}dy$$

$$= \int_0^{L-p_i(y_{-i},z)} \frac{e-e^L}{e-1}dy + \int_{L-p_i(y_{-i},z)}^{L} \frac{(e-e^L)+(e^L-e^y)}{e-1}dy + \int_L^1 \frac{e-e^L}{e-1}dy$$

$$\geq \frac{e-e^L}{e-1} + \int_{L-p_i(y_{-i},z)}^{L} \frac{L-y}{e-1}dy = \frac{e-e^L+\frac{1}{2}p_i(y_{-i},z)^2}{e-1}$$

Now, by dominance, in the actual execution of RANKING with these values of y_{-i}, z, i is matched whenever $y_i \leq L$. According to dominance i is matched when $y_i \leq L$, so as before $\mathbb{E}[\alpha_i|y_{-i}, z] \geq (e^L-1)/(e-1)$. However, when $y_i \geq L$ it is still possible that i is matched . In particular, there is a (possibly empty) range of values of y_i where i gets matched *above* j and another range of values where i gets matched *below* j. Call the length of this range of above matches $aa_i(y_{-i}, z)$ and below matches $b_i(y_{-i}, z)$. Then, there is some gain in $\mathbb{E}[\alpha_i|y_{-i}, z]$, given by

$$\mathbb{E}[\alpha_i|y_{-i}, z] \geq \int_0^{L+aa_i(y_{-i},z)+b_i(y_{-i},z)} \frac{e^z}{e-1}dy$$

$$= \frac{e^{L+aa_i(y_{-i},z)+b_i(y_{-i},z)}-1}{e-1} \geq \frac{(e^L-1)+aa_i(y_{-i},z)+b_i(y_{-i},z)}{e-1}$$

So, we get

$$\mathbb{E}[\alpha_i+\beta_j|y_{-i}, z] \geq \frac{e-e^L+e^L-1}{e-1} + \frac{\frac{1}{2}p_i(y_{-i},z)^2+aa_i(y_{-i},z)+b_i(y_{-i},z)}{e-1}$$

$$= 1 + \frac{\frac{1}{2}p_i(y_{-i},z)^2+aa_i(y_{-i},z)+b_i(y_{-i},z)}{e-1}$$

Now, let p_i, b_i and aa_i denote the probability that vertex i is involved in a perfect match, below match or above match respectively, over all choices of y and z. Then, $p_i = \int_{y_{-i},z} p_i(y_{-i}, z)$, $b_i = \int_{y_{-i},z} b_i(y_{-i}, z)$ and $aa_i = \int_{y_{-i},z} aa_i(y_{-i}, z)$

$$\mathbb{E}[\alpha_i + \beta_j] = \int_{y_{-i},z} \mathbb{E}[\alpha_i + \beta_j | y_{-i}, z]$$

$$\geq \int_{y_{-i},z} 1 + \frac{\frac{1}{2}p_i(y_{-i},z)^2 + aa_i(y_{-i},z) + b_i(y_{-i},z)}{e-1}$$

$$\geq 1 + \frac{p_i^2}{2(e-1)} + \frac{aa_i}{e-1} + \frac{b_i}{e-1}$$

In the last step, we have used the fact that for a fixed mean the variance of a random variable is minimized by the peak distribution. This allowed us to get a lower bound by assuming that $p_i(y_{-i}, z) = p_i$ for all choices of y_{-i}, z.

It should be clear to the reader where we are headed. If we can show that $\mathbb{E}[\alpha_i + \beta_j] \geq 1 + c$ for all $(i,j) \in E$, then we can divide the dual update rules by $1 + c$ and get new dual update rules α'_i, β'_j such that $\mathbb{E}[\alpha'_i + \beta'_j] \geq 1$. Moreover, when the algorithm matches (i,j), the increase in the dual objective value is $\frac{e}{(e-1)(1+c)}$, allowing us to conclude that the competitive ratio is $(1-1/e)(1+c)$. However so far we have shown that $\mathbb{E}[\alpha_i + \beta_j] \geq 1 + c$ where c depends on i. To get rid of this dependence of c on i, we essentially need to average over all choices of i. As before, let $p = \sum_i p_i/n$, $aa = \sum_i aa_i/n$ and $b = \sum_i b_i/n$ be the expected number of P- and B-matches respectively, scaled down by n.

Define $\forall i, \alpha'_i = \sum_i \mathbb{E}[\alpha_i]/n$ and $\forall j, \beta'_j = \sum_i \mathbb{E}[\beta_j]/n$. Note that the dual objective value is $\sum_i \alpha'_i + \sum_j \beta'_j = n \sum_i \mathbb{E}[\alpha_i]/n + n \sum_j \mathbb{E}[\beta_j]/n = \sum_i \mathbb{E}[\alpha_i] + \sum_j \mathbb{E}[\beta_j] = \frac{e}{e-1}\mathbb{E}[\mathsf{RANKING}]$. For any $(i,j) \in E$, $\alpha'_i + \beta'_j = \sum_i \mathbb{E}[\alpha_i]/n + \sum_j \mathbb{E}[\beta_j]/n = \sum_{(i,j)\in OPT} \mathbb{E}[\alpha_i + \beta_j]/n \geq \sum_i (1 + \frac{p_i^2}{2(e-1)} + \frac{aa_i}{e-1} + \frac{b_i}{e-1})/n \geq 1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1}$. Now define

$$\alpha''_i = \frac{\alpha'_i}{1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1}} \; \forall i, \qquad \beta''_j = \frac{\beta'_j}{1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1}} \; \forall j$$

so that $\alpha''_i + \beta''_j \geq 1$. Thus, this is a dual feasible solution. Morever, the dual objective value is:

$$\sum_i \alpha''_i + \sum_j \beta''_j = \frac{\sum_i \alpha'_i + \sum_j \beta'_j}{1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1}}$$

$$= \frac{e}{(e-1)(1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1})}\mathbb{E}[\mathsf{RANKING}]$$

Since the dual solution is feasible, this means that

$$\mathbb{E}[\mathsf{RANKING}] \geq (1 - 1/e)(1 + \frac{p^2}{2(e-1)} + \frac{aa}{e-1} + \frac{b}{e-1})$$

□

Theorem 5.1. RANKING *has a competitive ratio of at least 0.643 for online bipartite matching in the random order model*

Proof. Let c be the competitive ratio of RANKING in the random order model. By Lemma 5.4, $c = \frac{1}{2} + b + \frac{p}{2} + \frac{aa}{2} \geq 1 - 1/e$, so $b + aa \geq b + \frac{aa}{2} \geq \frac{1}{2} - \frac{1}{e} - \frac{p}{2}$. Plugging this into the statement of Lemma 5.5, we get

$$c \geq 1 - 1/e + \frac{p^2}{2e} + \frac{1}{2e} - \frac{1}{e^2} - \frac{p}{2e} \geq 0.68 + \frac{p^2}{2e} - \frac{p}{2e}$$

$$c \geq \max\{0.68 + \frac{p^2}{2e} - \frac{p}{2e}, \quad 0.5 + \frac{p}{2}\}$$

In the final inequality, the second argument of the max comes from Lemma 5.4 by setting $b = 0$. Now, minimizing this quantity over the choice of p, we get $c \approx 0.643$ when $p \approx 0.286$. □

5.1 Discussion

The main idea in our proof that RANKING beats $1 - 1/e$ in the random order model, was the following. We showed (in Lemma 5.5) that depending on certain properties of the graph, the competitive ratio would be larger than $1 - 1/e$. On the other hand, we showed (in Lemma 5.4) that the graph must have these properties, otherwise again the competitive ratio would be larger than $1 - 1/e$. For the former, we used the randomized primal framework. In particular, the property we showed (and so did [14]) was that the graph must have a significant number of P-events.

Our result should be seen as a proof of concept. It is possible that by finding the right "property" that the graph must have (and showing that it does indeed have that property), we can obtain a tighter analysis. We believe that this is a promising approach.

6 Conclusions

In this thesis, we studied variants of the general online allocation problem, including bipartite maching, budgeted allocation and vertex-weighted bipartite matching. In particular, we studied these problems in the *random order model*, when the input arrives in a random permutation. We use the *randomized primal-dual* framework of [5] to show that for these problems, simple *greedy* algorithms achieve a competitive ratio of $1 - 1/e$. This framework was originally used to give an elegant analysis of the RANKING algorithm of [15] for the classical online bipartite matching problem in the adversarial model. We show that this framework is also useful in understanding these problems when the input comes as a random permutation, rather than adversarially. To achieve this, we exploited appropriately defined *monotonicity* and *dominance* properties of the greedy algorithms. Further, we use this framework to show that RANKING has a competitive ratio strictly better than $1 - 1/e$ for online bipartite matching in the random order model.

The three main problems that we consider have received much attention recently, and are well-understood in the adversarial model. In the random order model, our understanding of these problems is incomplete. Moreover, there are some interesting special cases of the general allocation problem that remain to be understood. We believe that the randomized primal-dual approach or some variant of it may be useful in attacking these problems. In the rest of this section, we discuss some of these open problems and directions for future work.

Online budgeted allocation (AdWords) The special case of online budgeted allocation when the bids are much smaller than the budgets is well understood. In the adversarial model, the WEIGHTED-BALANCE algorithm has a competitive ratio $1 - 1/e$. In the random order model, we give a simple proof of the result of [10] that GREEDY has a competitive ratio $1 - 1/e$. However, when the bids are unrestricted nothing better than a factor $1/2$ is known in either the adversarial or the random order model. It is not hard to see that GREEDY has a competitive ratio $1/2$, which is tight in the adversarial model. *We conjecture that* GREEDY *has a competitive ratio* $1 - 1/e$ *in the random order model.* We are not aware of a candidate algorithm that beats $1/2$ in the adversarial model.

Online vertex-weighted bipartite matching in random order model

The PERTURBED-GREEDY algorithm is known to have a competitive ratio $1 - 1/e$ in the adversarial model. We show that GREEDY has a competitive ratio $1 - 1/e$ in the random order model. Is it possible to beat $1 - 1/e$ in the random order model? In particular, what is the competitive ratio of PERTURBED-GREEDY in the random order model? In the adversarial model, the randomized primal-dual analysis of RANKING for the unweighted case easily generalizes to an analysis of PERTURBED-GREEDY in the weighted case. Is there a similar generalization of our analysis in Section 5 beating $1 - 1/e$ to the weighted case?

Online bipartite matching in random order model There is a gap in the best known lower bound (0.696) and upper bound (0.727) on the competitive ratio of RANKING for online bipartite matching in the random order model. Moreover, the best known lower bound has a compuational proof. We showed that the randomized primal-dual framework lends itself to show that RANKING has a competitive ratio better than $1-1/e$. Can this framework be used to provide a tight(er) analysis without resorting to computational approaches?

Generalized online matching with concave utilities in random order model Recently, Devanur et al considered a generalization of the online budgeted allocation problem where budgets are not hard constraints. Instead, the bidders have arbitrary concave functions that specify how much they are willing to pay. They show a $1 - 1/e$ competitive algorithm using convex programming duality, when fractional assignments are allowed. We believe that a suitable variant of our techniques can be used to show that in the random order model, a suitable variant of GREEDY has a competitive ratio $1 - 1/e$.

Online submodular welfare in random order model In the online submodular welfare problem, each bidder specifies a monotone submodular set function over the ground set of items. The goal is to produce an allocation that maximizes welfare among the bidders. A recent result of Kapralov et al ([13]) shows that in the adversarial model, 1/2 is the best possible competitive ratio. This raises the question if 1/2 can be beaten in the random order model.

References

[1] G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1253–1264. SIAM, 2011. URL http://dl.acm.org/citation.cfm?id=2133036.2133131.

[2] B. Birnbaum and C. Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, Mar. 2008. ISSN 0163-5700. doi: 10.1145/1360443.1360462. URL `http://doi.acm.org/10.1145/1360443.1360462`.

[3] N. Buchbinder, K. Jain, and J. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In L. Arge, M. Hoffmann, and E. Welzl, editors, *Algorithms ESA 2007*, volume 4698 of *Lecture Notes in Computer Science*, pages 253–264. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-75519-7. URL `http://dx.doi.org/10.1007/978-3-540-75520-3_24`. 10.1007/978-3-540-75520-3_24.

[4] M. de Berg and U. Meyer, editors. *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I*, volume 6346 of *Lecture Notes in Computer Science*, 2010. Springer. ISBN 978-3-642-15774-5.

[5] N. Devanur, K. Jain, and R. Kleinberg. Understanding karp-vazirani-vaziranis online matching via randomized primal-dual. 2000.

[6] N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, EC '09, pages 71–78, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-458-4. doi: 10.1145/1566374.1566384. URL `http://doi.acm.org/10.1145/1566374.1566384`.

[7] J. Feldman, N. Korula, V. S. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *WINE*, pages 374–385, 2009.

[8] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. *Foundations of Computer Science, IEEE Annual Symposium on*, 0:117–126, 2009. ISSN 0272-5428. doi: http://doi.ieeecomputersociety.org/10.1109/FOCS.2009.72.

[9] J. Feldman, M. Henzinger, N. Korula, V. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. In M. de Berg and U. Meyer, editors, *Algorithms ESA 2010*, volume 6346 of *Lecture Notes in Computer Science*, pages 182–194. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-15774-5. URL `http://dx.doi.org/10.1007/978-3-642-15775-2_16`. 10.1007/978-3-642-15775-2_16.

[10] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA

'08, pages 982–991, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=1347082.1347189.

[11] B. Haeupler, V. Mirrokni, and M. Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In N. Chen, E. Elkind, and E. Koutsoupias, editors, *Internet and Network Economics*, volume 7090 of *Lecture Notes in Computer Science*, pages 170–181. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-25509-0. URL http://dx.doi.org/10.1007/978-3-642-25510-6_15. 10.1007/978-3-642-25510-6_15.

[12] B. Kalyanasundaram and K. Pruhs. An optimal deterministic algorithm for online b-matching. *Theor. Comput. Sci.*, 233(1-2):319–325, 2000.

[13] M. Kapralov, I. Post, and J. Vondrák. Online and stochastic variants of welfare maximization. *CoRR*, abs/1204.1025, 2012.

[14] C. Karande, A. Mehta, and P. Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 587–596, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0691-1. doi: 10.1145/1993636.1993715. URL http://doi.acm.org/10.1145/1993636.1993715.

[15] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC '90, pages 352–358, New York, NY, USA, 1990. ACM. ISBN 0-89791-361-2. doi: 10.1145/100216.100262. URL http://doi.acm.org/10.1145/100216.100262.

[16] M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 597–606, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0691-1. doi: 10.1145/1993636.1993716. URL http://doi.acm.org/10.1145/1993636.1993716.

[17] V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: online actions based on offline statistics. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1285–1294. SIAM, 2011. URL http://dl.acm.org/citation.cfm?id=2133036.2133134.

[18] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5), Oct. 2007. ISSN 0004-5411. doi: 10.1145/1284320.1284321. URL http://doi.acm.org/10.1145/1284320.1284321.

[19] V. S. Mirrokni, S. O. Gharan, and M. Zadimoghaddam. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1690–1701. SIAM, 2012. URL http://dl.acm.org/citation.cfm?id=2095116.2095250.