Online Matching*
(DRAFT: Not for Distribution)

Zhiyi Huang Thorben Tröbst

August 11, 2021

1 Introduction

Building on three decades of research, online matching has evolved into a rich theory at the intersection of economics, operations research, and computer science. This development was in large part due to many applications that arose from the evolution of the internet and mobile computing: Google matches search queries to advertisers, Uber matches drivers to riders, Upwork matches employers to freelancers, Tinder matches people to other people, etc.

This chapter will present online matching models for these applications, and will study the corresponding algorithms from the economic viewpoint introduced in Chapter **??**. Section 2 will cover models inspired by online advertising. Section 3 will focus on more general arrival models motivated by other problems such as ride-sharing. We will assume familiarity with the Online Bipartite Matching Problem and its associated notions defined in Chapter **??**.

2 Models for Online Advertising

The online matching models that we will study in this section are a substantial abstraction of ad auctions that take place on search engines and other websites every day.

Example 1. Consider a search engine provider which shows ads to users when they perform searches. The offline vertices in this case are advertisements, e.g., 100 ads for a new book, 200 ads for a new chair, etc. The online vertices are search requests by users of the search engine.[1] Each time a user performs a search, an ad relevant to their search terms should be displayed instantaneously; relevance corresponds to edges in the graph.

*This chapter will appear in *Online and Matching-Based Market Design.* Federico Echenique, Nicole Immorlica and Vijay V. Vazirani, Editors. Cambridge University Press.

[1]In the terminology of Chapter **??**, search requests would be *buyers* and advertisers *goods* even though their "real" roles in the ad market are the exact opposite.

The example leaves open how the advertisers pay the search engine provider. If each ad is worth the same amount, the algorithm and analysis from Chapter **??** apply. This section will consider more complex compensation schemes.

Remark 2. In this chapter we will generally take a worst-case approach. However, in practice one may sometimes make certain additional assumptions such as requiring that search requests are sampled from a fixed distribution or that advertisers are able to pay for a large number of ads. Models with these additional assumptions are studied in Chapter **??**.

2.1 Ranking with Vertex Weights

We start by studying a slight generalization of the Online Bipartite Matching Problem in which each good j has a *value* of $v_j \geq 0$, and the goal is to maximize the value of matched goods. This means that advertisers pay different rates to have their ads displayed in Example 1.

Recall Ranking and its analysis from the economic viewpoint introduced in Chapter **??**. We will make two changes to the algorithm: the price of good j is scaled by its value, and buyers buy goods maximizing their utility, i.e., the difference between a goods value and its price. Formally this is Algorithm 1.

Algorithm 1: Ranking with Vertex Weights

for $j \in S$ **do**
    Set price $p_j \leftarrow v_j e^{w_j - 1}$ where $w_j \in [0, 1]$ is sampled uniformly.
for *each buyer i who arrives* **do**
    Let $N(i)$ be the unmatched neighbors of i.
    if $N(i) \neq \emptyset$ **then**
        Match i to $j \in N(i)$ maximizing $v_j - p_j$.

Definition 3. For good $j \in S$, the revenue r_j is p_j if it was sold and 0 otherwise. For buyer $i \in B$, the utility u_i is $v_j - p_j$ if it bought some good j and 0 otherwise.

The analysis of the vertex-weighted Ranking algorithm remains largely the same as in Chapter **??**. For the sake of completeness, we will state the necessary lemmas and sketch their proofs. Once again we consider a random variable u_e which represents the utility of buyer i if edge $e = (i, j)$ is removed from the graph.

Lemma 4. *Let $(i, j) \in E$ be arbitrary. Then:*

1. $u_i \geq u_e$.
2. *If $p_j < v_j - u_e$, then j will be matched.*

Proof. **1)** Having an additional edge can only increase the utility of buyer i since i is free not to choose it.

2) If j has not been matched when i arrives, then j will be matched to i. □

Lemma 5. *Let* $(i, j) \in E$ *be arbitrary. Then* $\mathbb{E}[u_i + r_j] \geq (1 - 1/e)v_j$.

Proof. As in Chapter **??**, the idea is to condition on the value of u_e. By Lemma 4, we know that $\mathbb{E}[u_i] \geq \mathbb{E}[u_e]$. Moreover, for any $z \in [0, (1-1/e)v_j]$ let w be such that $v_j e^{w-1} = v_j - z$. Then by Lemma 4

$$\begin{aligned} \mathbb{E}[r_j \mid u_e = z] &\geq \int_0^w v_j e^{x-1} \, \mathrm{d}x \\ &= \left(1 - \frac{1}{e}\right) v_j - z. \end{aligned}$$

Note that this inequality holds trivially also if $z > (1 - 1/e)v_j$.

Finally, by the law of total expectation we know that

$$\mathbb{E}[r_j] = \mathbb{E}_z[\mathbb{E}[r_j \mid u_e = z]] = \left(1 - \frac{1}{e}\right) v_j - \mathbb{E}[u_e]$$

and thus $\mathbb{E}[u_i + r_j] \geq (1 - 1/e)v_j$ by linearity of expectation. □

Theorem 6. *Algorithm 1 is* $(1 - 1/e)$*-competitive.*

Proof. Let M be a matching in G which maximizes the total value of the matched goods. The value matched by Ranking is exactly $\sum_{i \in B} u_i + \sum_{j \in S} r_j$. Further by Lemma 5 and linearity of expectation we get

$$\begin{aligned} \mathbb{E}\left[\sum_{i \in B} u_i + \sum_{j \in S} r_j\right] &\geq \sum_{(i,j) \in M} \mathbb{E}[u_i + r_j] \\ &\geq \sum_{(i,j) \in M} \left(1 - \frac{1}{e}\right) v_j. \end{aligned}$$ □

Chapter **??** showed that the factor of $1 - 1/e$ is optimal, even in the case of unit values. Thus, Ranking provides an optimal online algorithm for the Vertex-Weighted Online Bipartite Matching Problem.

Remark 7. Readers familiar with primal-dual algorithms will recognize this type of argument: what we have essentially shown is that the variables $\mathbb{E}[u_i]$ and $\mathbb{E}[v_j]$ form an $(1 - 1/e)$-approximately feasible solution to the dual of the matching problem. This primal-dual viewpoint on online matching is equivalent to the economic viewpoint presented in this chapter.

2.1.1 Deriving the Optimal Prices

In the previous section we used the functions $g_j(x) := v_j e^{x-1}$ to assign prices $p_j \leftarrow g_j(w_j)$ to the goods j. But how did we know that these are the right prices? The idea is to carry out the analysis of Algorithm 1 via generic g_j and try to maximize the bound on the competitive ratio.

Lemma 4 is unaffected by the choice of g_j but in Lemma 5 our goal will be to prove that $\mathbb{E}[u_i + r_j] \geq \Gamma v_j$ for the largest possible Γ. In the proof we assume that g_j is increasing and that its range contains $[(1-\Gamma)v_j, v_j]$.

Under these assumptions, we then observe in the proof of Lemma 5 that

$$\mathbb{E}[r_j \mid u_e = z] \geq \int_0^w g_j(x)\,\mathrm{d}x$$

where $z \in [0, \Gamma v_j]$ and w is such that $g_j(w) = v_j - z$. In order to finish the proof we need that

$$\int_0^w g_j(x)\,\mathrm{d}x \geq \Gamma v_j - z = \Gamma v_j - v_j + g_j(w).$$

So our goal is to find some g_j such that Γ can be chosen as large as possible. Let us assume that for the optimal g_j and maximal Γ we have equality, i.e.

$$v_j - g_j(w) + \int_0^w g_j(x)\,\mathrm{d}x = \Gamma v_j. \tag{1}$$

Taking the derivative with respect to w of both sides yields the elementary differential equation $-g'_j(w) + g_j(w) = 0$ which is solved by functions of the form $g_j(x) = Ce^x$. Substituting this back into (1) yields $v_j - C = \Gamma v_j$ and so we would like to make C as small as possible. But since the range of g_j should include v_j, the minimum choice for C is v_j/e which yields $g_j(x) = v_j e^{x-1}$ and $\Gamma = 1 - 1/e$.

2.2 Fractional and Multi-Matching Models

Aside from Ranking, there is another framework for solving online matching problems. If we allow matching multiple buyers to a single good, an effective strategy is to *balance* the number of buyers each good is matched to. Depending on whether buyers are matched fractionally or integrally to the goods, the algorithms based on this idea are called Water-Filling or Balance.

2.2.1 Water-Filling

As a warm-up, we first introduce Water-Filling and its analysis in the Online Bipartite Matching Problem. In this setting, each buyer i is matched fractionally when it arrives, i.e., the algorithm determines non-negative values $(x_{i,j})_{(i,j)\in E}$ subject to $\sum_{j\in N(i)} x_{i,j} \leq 1$ and for any good j, $\sum_{i\in N(j)} x_{i,j} \leq 1$. The goal is to maximize $\sum_{(i,j)\in E} x_{i,j}$.

Remark 8. Allowing fractional matchings makes the problem strictly easier. By a classical result in matching theory, in any bipartite graph, the size of a maximum *fractional* matching equals the size of a maximum *integral* matching.

The Water-Filling algorithm is inspired by the following physical metaphor. Imagine that each buyer brings a watering can filled with 1 unit of water and each good corresponds to a tank that holds at most one unit of water. How should we distribute each buyer's water into their neighboring tanks such that the total amount of water is maximized? The intuitive answer is that each buyer should pour water into the tanks with the lowest water level so as to equalize the water level in its connected tanks. See Figure 1 for an example.

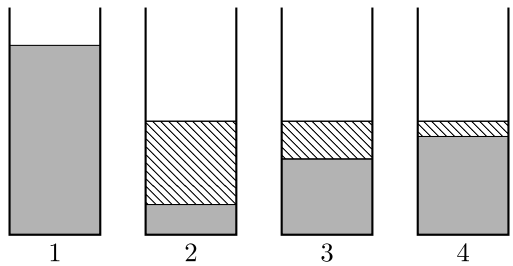

Figure 1: Shown is an iteration of Water-Filling with four "tanks" (offline vertices). The gray region represents the water level at the beginning of the iteration whereas the shaded region represents an additional unit of water.

As with Ranking, we will take an economic viewpoint on Water-Filling. Each good will have a price e^{w-1}, similar as in Ranking, but this time it depends on the "water level" w, i.e., how much of the good has been matched off so far. When buyer i arrives, it buys fractional edges $x_{i,j}$ from its neighbors $N(i)$ so as to maximize the total utility. Formally, let $w_j \coloneqq \sum_{i' \in N(i)} x_{i',j}$ be the total amount of good j which has been sold so far. Buyer i solves the convex program

$$\begin{aligned} \max_{(x_{i,j})_{j \in N(i)}} \quad & \sum_{j \in N(i)} \int_{w_j}^{w_j + x_{i,j}} \left(1 - e^{w-1}\right) \mathrm{d}w \\ \text{s.t.} \quad & \sum_{j \in N(i)} x_{i,j} \leq 1, \\ & x_{i,j} \geq 0 \qquad \forall j \in N(i) \end{aligned} \tag{2}$$

which is equivalent to the aforementioned continuous process of water-filling.

Accordingly, buyer i pays $\int_{w_j}^{w_j + x_{i,j}} e^{w-1} \, \mathrm{d}w$ for each good j. We remark that the utility $1 - e^{w-1}$ ensures that the water levels do not exceed 1. This results in Algorithm 2 (the payments are only required for the analysis).

Algorithm 2: Water-Filling for the Online Fractional Bipartite Matching Problem

---

for *each buyer i who arrives* **do**
  Compute an optimum solution $(x_{i,j})_{j \in N(i)}$ to the Eqn. (2).

Under this economic interpretation of Water-Filling, its analysis follows the same framework as the analysis of Ranking presented in Chapter **??** and in the previous section. Let r_j be the total revenue of good j, i.e., the sum of all payments made for fractions of j by the buyers. Let u_i be the utility of buyer i, i.e., the objective function value of (2) in buyer i's iteration.

Lemma 9. *Let* $(i, j) \in E$ *be arbitrary. Then* $r_j + u_i \geq 1 - 1/e$.

Proof. On the one hand, let w^* be the water level of good j at the end of Algorithm 2. Then $r_j = \int_0^{w^*} e^{w-1}\,\mathrm{d}w$.

On the other hand, let w_j be the amount of good j which is allocated before buyer i arrives. Then u_i is the objective function value of (2) and thus must be at least $1 - e^{w_j + x_{i,j} - 1}$. Otherwise the solution could be improved by increasing $x_{i,j}$ by some suitably small $\epsilon > 0$ and decreasing $x_{i,j'}$ for some other $j' \in N(i)$ to maintain that $\sum_{k \in N(i)} x_{i,k} \leq 1$.

Finally, using $w_j + \Delta x_{i,j} \leq w^*$, we get

$$\begin{aligned} r_j + u_i &\geq \int_0^{w^*} e^{w-1}\,\mathrm{d}w + 1 - e^{w_j + \Delta x_{i,j} - 1} \\ &\geq e^{w^*-1} - \frac{1}{e} + 1 - e^{w^*-1} = 1 - \frac{1}{e} \end{aligned}$$

□

Theorem 10. *Algorithm 2 is* $(1 - 1/e)$*-competitive.*

Proof. It is clear from the updates of $x_{i,j}$, r_j and u_i during the algorithm, that

$$\sum_{(i,j) \in E} x_{i,j} = \sum_{j \in S} r_j + \sum_{i \in B} u_i.$$

On the other hand, let $(y_{i,j})_{(i,j) \in E}$ be any other fractional matching. Then

$$\begin{aligned} \sum_{(i,j) \in E} y_{i,j} &\leq \frac{1}{1 - 1/e} \sum_{(i,j) \in E} y_{i,j}(r_j + u_i) \\ &\leq \frac{1}{1 - 1/e} \left(\sum_{j \in S} r_j + \sum_{i \in B} u_i \right) \end{aligned}$$

where we used that $\sum_{j \in S} y_{i,j} \leq 1$ for all i and likewise when we sum over buyers. Together with the previous equality this shows that the total size of the fractional matching x is at least a $1 - 1/e$ fraction of any fractional matching. □

As previously discussed, the Online Fractional Bipratite Matching Problem is easier than the integral problem which also admit a $(1 - 1/e)$-competitive algorithm. The main improvement here is that Water-Filling is deterministic whereas Ranking is randomized. There is also a matching upper bound of $1 - 1/e$ for the fractional problem, and thus Water-Filling is optimal; see Exercise 3.

2.2.2 Balance

Water-Filling is applicable only when fractional allocations are allowed, which is rarely the case in practice. Nonetheless, a rounded variant of Water-Filling called Balance works well for problems in which the offline vertices may be matched many times. Importantly, it includes the online advertising in Example 1 since advertisers typically want their ads displayed thousands or even millions of times.

Formally, the Online Bipartite b-Matching Problem is a variant of Online Bipartite Matching in which each good may be matched up to $b \in \mathbb{N}$ times.[2] A natural approach for this problem is to match each online vertex to a neighboring offline vertex which is currently least matched. From an economic viewpoint, it means that the price of a good increases monotonically with the fill level. The optimal choice for the pricing function turns out to be $(1+1/b)^{w-b}$ where w is the number of buyers matched to the given good. When b is large, it is approximately $e^{\frac{w}{b}-1}$, i.e., the pricing function in Water-Filling. See Algorithm 3.

Algorithm 3: Balance for the Online Bipartite b-Matching Problem

Initialize the price p_j of each good j to $(1+1/b)^{-b}$.
for *each buyer i who arrives* **do**
  Let $N(i)$ be the neighbors of i.
  Match i to $j \coloneqq \arg\min_{j \in N(i)} p_j$.
  $p_j \leftarrow p_j \cdot (1+1/b)$

As before, let r_j be the total revenue collected by good j and let u_i be the utility obtained by buyer i.

Lemma 11. *Let $(i,j) \in E$ be arbitrary. Then $r_j/b + u_i \geq 1 - (1+1/b)^{-b}$.*

Proof. Let w be the total number of buyers who end up buying good j. Then

$$\begin{aligned} r_j &= \sum_{k=0}^{w-1} \left(1+\frac{1}{b}\right)^{k-b} \\ &= b\left(\left(1+\frac{1}{b}\right)^{w-b} - \left(1+\frac{1}{b}\right)^{-b}\right). \end{aligned}$$

On the other hand, the price of good j was at most $(1+1/b)^{w-b}$ when i arrived, and so $u_i \geq 1 - (1+1/b)^{w-b}$. Thus $r_j/b + u_i \geq 1 - (1+1/b)^{-b}$. □

Theorem 12. *Algorithm 3 is $(1-(1+1/b)^{-b})$-competitive.*

[2] The algorithm and analysis generalizes to having distinct b_j for different goods j.

Proof. Let M be a maximal b-matching in G. Then

$$|M| \le \sum_{(i,j)\in M} \frac{r_j/b + u_i}{1-(1+1/b)^{-b}}$$
$$\le \frac{1}{1-(1+1/b)^{-b}} \left(\sum_{j\in S} r_j + \sum_{i\in B} u_i \right).$$

Since the sum of revenues and utilities is the size of the matching generated by the algorithm, this concludes the proof. □

Since $(1+1/b)^{-b}$ converges to $1/e$ as $b \to \infty$, the competitive ratio converges to $1-1/e$ for large capacities. There is a matching lower bound of $1-(1+1/b)^{-b}$ for *deterministic* algorithms in the Online Bipartite b-Matching Problem. Of course, if we consider randomized algorithms, Ranking is $(1-1/e)$-competitive.

2.3 Edge-Weighted Online Matching with Free Disposal

We will now turn to the Edge-Weighted Online Bipartite Matching Problem in which the value of a good may also depend on the buyer it is matched to, i.e., each edge $(i,j) \in E$ has a value $v_{i,j} \ge 0$. This is relevant in the modern markets of *targeted* advertising. Advertisers are willing to pay more if an ad is shown to a user from its target demographic. For example, a restaurant will be willing to pay more for having its ads shown to users that live close by. This is also known as the Display Ads problem.

Theorem 13. *There is no constant factor competitive online algorithm for the Edge-Weighted Online Bipartite Matching Problem.*

Proof. Consider a family of instances that consist of a single good and n buyers. For any $1 \le k \le n$ and some $V > 1$, let there be an instance in which the first k buyers have values $1, V, V^2, \ldots, V^{k-1}$ for the good, and the remaining buyers have value 0. Suppose that the algorithm assigns the good to buyer i with probability p_i when $k \ge i$. Then $\sum_{i=1}^{n} p_i \le 1$. Hence, there is some i such that $p_i \le \frac{1}{n}$. But then the algorithm's expected value for instance $k = i$ is at most $\frac{1}{n}V^{k-1} + \frac{n-1}{n}V^{k-2}$. Comparing with the optimal value V^{k-1}, the competitive ratio is at most $\frac{1}{n} + \frac{n-1}{nV}$, which tends to 0 as n and V tend to infinity. □

Theorem 13 necessitates an additional assumption if we want to make the edge-weighted problem tractable. Observe that the constraint of displaying an advertiser's ad at most once is artificial. An advertiser would be happy to have their ad displayed multiple times if the extra displays are free. Hence, we consider a relaxed model where an offline vertex (advertiser) can be matched more than once, but only the most valuable one counts in the objective. Equivalently, it can be stated as the *free disposal* assumption: offline vertices may drop edges to previous online vertices in favor of better ones. It allows us to design constant competitive online algorithms; e.g. see Exercise 4 for a greedy approach.

We will modify the Water-Filling algorithm from Section 2.2 for the fractional setting by changing the prices of goods suitably. In particular, even if a good was already fully matched, it might continue getting matched to better and better edges. Thus the price can no longer depend only on the current fill-level of the good; it must further take into account the values of the edges it is matched along.

To keep track of this, we define a function $w_{x,j} : \mathbb{R}_{\geq 0} \to [0,1]$ for any fractional matching x and any good $j \in S$. The value of $w_{x,j}(t)$ represents the fraction of good j which is assigned in x on edges of value at least t, i.e.:

$$w_{x,j}(t) \coloneqq \sum_{i: v_{i,j} \geq t} x_{i,j}.$$

Consider now what happens when buyer i arrives and wishes to buy good j. Assume that j is already fully matched to various buyers and let i' be the buyer with the least value. We can then think of i as filling up the rectangle $[v_{i',j}, v_{i,j}] \times [0,1]$ with the graph of $w_{x,j}(t)$. See Figure 2.

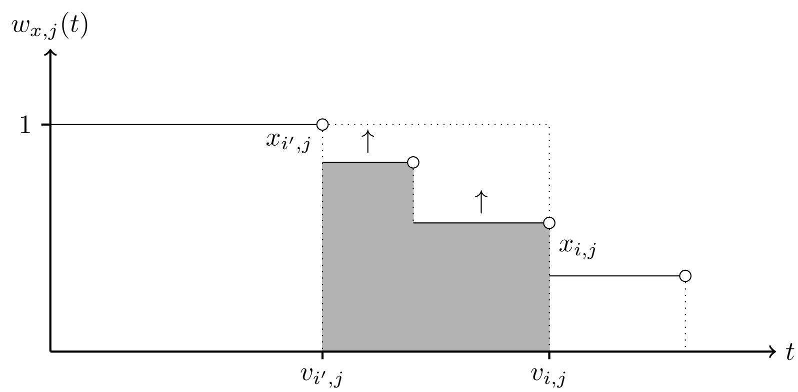

Figure 2: Shown is the function $w_{x,j}(t)$ for some fixed good j and assignment x. Switching some of the allocation from buyer i' to i increases the level of $w_{x,j}(t)$ everywhere between the values of $v_{i',j}$ and $v_{i,j}$.

This motivates a price of the form $\int_{v_{i',j}}^{v_{i,j}} g(t)\,\mathrm{d}t$ and it turns out that setting $g(t) \coloneqq e^{t-1}$ is optimal as it was for Water-Filling. Matching j fractionally to i, however, will also unmatch the same amount of j from i'. Hence, to keep the utility of i' unchanged, an additional price of $v_{i',j}$ is necessary to "pay off" buyer i'. Thus the total price per unit of the edge (i,j) should be

$$v_{i',j} + \int_{v_{i',j}}^{v_{i,j}} e^{w_{x,j}(t)-1}\,\mathrm{d}t = \int_{0}^{v_{i,j}} e^{w_{x,j}(t)-1}\,\mathrm{d}t$$

resulting in Algorithm 4.

Algorithm 4: Water-Filling with Edge Weights and Free Disposal

1 Match all goods j fully to dummy buyers along value 0 edges.
2 **for** *each buyer i who arrives* **do**
3   Let the price p_j of each good j be $\int_0^{v_{i,j}} e^{w_{x,j}(t)-1}\,\mathrm{d}t$.
4   **while** *i is not fully matched and there is at least one good j with $v_{i,j} > p_j$* **do**
5   Let $j = \arg\max_{j \in S} v_{i,j} - p_j$ and let i' be a buyer of j along a least value edge.
6   Reallocate an infinitesimal fraction of j by setting $x_{i,j} \leftarrow x_{i,j} + \mathrm{d}x$ and $x_{i',j} \leftarrow x_{i',j} - \mathrm{d}x$.

Remark 14. We are using the word "algorithm" loosely here to describe a continuous process. It is possible to use convex programming as in Water-Filling or other approximation methods to give an actual algorithm.

Revenues r_j and utilites u_i are defined as in the previous subsection. When the matching is adjusted by $\mathrm{d}x$ in line 6, we interpret this as buyer i paying $(p_j - v_{i',j})\mathrm{d}x$ to good j and $v_{i',j}\mathrm{d}x$ to buyer i' so that $u_{i'}$ remains unchanged. Thus r_j increases by $(p_j - v_{i',j})\mathrm{d}x$ and u_i increases by $(v_{i,j} - p_j)\mathrm{d}x$.

Lemma 15. *At any stage during the algorithm, we have*

$$\sum_{j \in S} r_j + \sum_{i \in B} u_i = \sum_{(i,j) \in E} x_{i,j} v_{i,j}.$$

Proof. This holds in the beginning since all variables are zero. Consider some iteration of the while loop in Algorithm 4. Then, by definition both sides of the equation increase exactly by $(v_{i,j} - v_{i',j})\mathrm{d}x$. □

Lemma 16. *Let $(i,j) \in E$ be arbitrary. Then at the end of the algorithm*

$$r_j + u_i \geq \left(1 - \frac{1}{e}\right) v_{i,j}.$$

Proof. Let x be the final allocation. Then during the iteration in which buyer i arrived, the price p_j would have been at most $\int_0^{v_{i,j}} e^{w_{x,j}(t)-1}\,\mathrm{d}t$ since the prices are non-decreasing during the algorithm. Thus the utility of buyer i at the end of their iteration (which is the same as their utility at the end of the algorithm) must be at least $u_i \geq v_{i,j} - \int_0^{v_{i,j}} e^{w_{x,j}(t)-1}\,\mathrm{d}t$.

On the other hand, the total revenue r_j collected by good j during the algorithm is

$$\int_0^{\infty} \int_0^{w_{x,j}(t)} e^{s-1}\,\mathrm{d}s\,\mathrm{d}t = \int_0^{\infty} \left(e^{w_{x,j}(t)-1} - \frac{1}{e}\right)\mathrm{d}t.$$

Thus

$$r_j + u_i \geq v_{i,j} - \int_0^{v_{i,j}} e^{w_{x,j}(t)-1}\,\mathrm{d}t + \int_0^{\infty} \left(e^{w_{x,j}(t)-1} - \frac{1}{e}\right) \mathrm{d}t$$
$$\geq v_{i,j} - \int_0^{v_{i,j}} \frac{1}{e}\,\mathrm{d}t = \left(1 - \frac{1}{e}\right) v_{i,j}. \qquad \square$$

Theorem 17. *Algorithm 4 is* $(1-1/e)$*-competitive.*

Proof. This follows just as the proofs of Theorems 6, 10, and 12. □

2.4 Online Matching with Stochastic Rewards

In the previous models, the revenue of an ad depends only on whether the user was *shown* the ad. However, advertisers are more interested in whether a user actually *clicked* on the ad. This has led to the *pay-per-click* model in online advertising, in which the reward for making a match depends on some external source—the users.

The Online Bipartite Matching Problem with Stochastic Rewards assumes stochastic users' behaviors. When an algorithm attempts to match buyer i to good j, this will succeed with some probability $p_{i,j} \in [0,1]$, which we assume to be known to the algorithm. The goal is to maximize the expected number of successfully matched edges.

The previous sections analyzed the competitive ratios of online algorithms with respect to the optimal matching in G in hindsight, i.e., the best an offline algorithm can do. This benchmark, however, is no longer suitable in the presence of stochasticity; see Exercise 5. Instead, we define the competitive ratio by comparing the expected size of the algorithm's matching to the solution of the Budgeted Allocation Problem (see Chapter **??**), which we will refer to as OPT.

$$\begin{aligned}
\max_{(x_{i,j})_{(i,j)\in E}} \quad & \sum_{(i,j)\in E} p_{i,j} x_{i,j} \\
\text{s.t.} \quad & \sum_{(i,j)\in E} p_{i,j} x_{i,j} \leq 1 \quad \forall j \in S, \\
& \sum_{(i,j)\in E} x_{i,j} \leq 1 \quad \forall i \in B, \\
& x \geq 0.
\end{aligned}$$

We shall think of $x_{i,j}$ as the probability that the algorithm tries to match edge (i,j). Let $L_j \coloneqq \sum_{i\in B} p_{i,j} x_{i,j}$ denote the *load* of any offline vertex $j \in S$.

Lemma 18. *The expected size of a matching computed by any algorithm $\mathcal{A}$ is at most OPT.*

Proof. For any $i \in B$, $\sum_{j\in S} x_{i,j}$ is the expected number of $\mathcal{A}$'s attempts to match i, and is therefore at most 1. Moreover, the load L_j measures the expected number of successful matches to the vertex j, which is at most 1. Thus x is a feasible solution to the Budgeted Allocation Problem. Finally, the expected size of $\mathcal{A}$'s matching is $\sum_{(i,j)\in E} p_{i,j} x_{i,j}$, and thus is at most OPT. $\square$

We focus on *vanishing and equal probabilities*, i.e. all $p_{i,j}$ are equal some p which tends to zero. Besides the ideas in the previous sections, the main additional insight required to handle stochastic rewards is to view the matching process through an alternative angle.

Fix any online algorithm $\mathcal{A}$ and any instance of the Online Bipartite Matching Problem with Stochastic Rewards. Consider the stochastic process in which $\mathcal{A}$ is shown online vertices one by one and attempts to match them. There are now two different perspectives on who "decides" whether an attempted match (i, j) is successful. First, one might imagine that the online vertex i flips a coin to decide whether the match is successful. On the other hand, one could also imagine that the offline vertex j has already flipped a coin ahead of time for each matching attempt to determine whether it will be successful.

In the latter case, there is $k \in \mathbb{N}$ for each good j such that j will be matched successfully if and only if $\mathcal{A}$ attempts at least k matches to j. Moreover, k is distributed geometrically, i.e., $\mathbb{P}[k = l] = p(1-p)^{l-1}$ for any $l \in \mathbb{N}$. As $p \to 0$, the distribution of the total load pk that can be assigned to j before a successful match is an exponential distribution. As a result we get the following theorem.

Theorem 19. *As $p \to 0$, the Online Bipartite Matching Problem with Stochastic Rewards becomes equivalent to the Online Bipartite Matching Problem with Stochastic Budgets, a variant of the Online Bipartite b-Matching Problem in which the b-value of each good is unknown to the algorithm and independently sampled from an exponential distribution.*

A variant of Balance works well in this setting. This is Algorithm 5.

Algorithm 5: Stochastic Balance for the Online Bipartite Matching Problem with Stochastic Budgets

```
for each online vertex i who arrives do
    Let N(i) be i' neighbors that have not yet exceeded their budgets.
    Match i to a neighbor j ∈ N(i) with the least number of match
      attempts thus far.
```

Unfortunately, bounding the competitive ratio of Stochastic Balance is rather involved and lies beyond the scope of this chapter.

Theorem 20. *Algorithm 5 is $\frac{1}{2}(1+(1-p)^{2/p})$-competitive for the Online Bipartite Matching Problem with Stochastic Budgets and is thus $\frac{1}{2}(1+e^{-2})$-competitive for the Online Bipartite Matching Problem with Stochastic Rewards as $p \to 0$.*

Unlike the simpler settings which we studied in the previous sections, these competitive ratios are not tight. In particular, these competitive ratios are with respect to the solution to the Budgeted Allocation Problem. It is possible to define better benchmarks, and to improve the competitive ratio in this way.

3 Arrival Models for Other Applications

We now turn to other applications such as ride-sharing. These settings call for more general arrival models of vertices and edges, and in some cases consider non-bipartite graphs.

3.1 Fully Online Matching

The Fully Online Matching Problem is a generalization of the Online Bipartite Matching Problem. Here we are given a (potentially non-bipartite) graph $G = (V, E)$, which is unknown to the algorithm initially. Notably, there are no offline vertices that are known upfront. Two kinds of events occur in some order:

- *Arrival of a vertex i:* Vertex i and its incident edges with existing vertices are revealed. This is the earliest time that i can be matched.
- *Deadline of a vertex i:* This is the latest time that i can be matched. We assume that all neighbors of i arrive before its deadline.

The goal is as usual to compute a matching of maximum size.

Example 21. Consider a ride-sharing platform. Each rider specifies the pickup location and destination, and remains on the platform for several minutes. Two riders can be paired up to share a ride if their pickup locations and destinations are near, and their requests have overlapping time windows. We may use vertices to represent riders, and edges to denote the pairs of riders that can share rides.

As usual a greedy approach yields a $\frac{1}{2}$-competitive algorithm; see Exercise 6. Both Ranking and Water-Filling from Section 2 generalize to the integral and fractional variants of the Fully Online Matching Problem respectively.

3.1.1 Ranking

The main difference compared to Chapter **??** and Section 2 is the lack of a classification of the vertices into buyers and goods a priori. When a vertex arrives, it first acts as a good, i.e. a price is randomly sampled and the vertex passively waits for someone to buy it. Once it reaches the deadline, however, it will act as a buyer, immediately buying the cheapest neighboring vertex which has not yet been matched. Formally this is Algorithm 6.

Algorithm 6: Ranking for the Fully Online Matching Problem

for *each vertex i who arrives* **do**
  Set price $p_i \leftarrow e^{w_i - 1}$ where $w_i \in [0, 1]$ is sampled uniformly.
for *each vertex i who departs* **do**
  Let $N(i)$ be the unmatched neighbors of i. **if** $N(i) \neq \emptyset$ **then**
    Match i to $j \in N(i)$ minimizing p_j.

When an edge (i, j) is added to the matching by Algorithm 6 where i has the earlier deadline, i acts as a buyer and j acts as a good. Accordingly, the *gain* of i is $g_i = e^{w_i - 1}$ and the gain of j is $g_j = 1 - e^{w_i - 1}$ (corresponding to revenue and utility respectively).

Definition 22. The *marginal price* p_i^* of a vertex i w.r.t. fixed prices of the other vertices is the highest price below which i would be bought before its deadline. The *marginal rank*, denoted as w_i^* satisfies $e^{w_i^* - 1} = p_i^*$. If i always acts as a good, define $p_i^* = w_i^* = 1$.

Next focus on an arbitrary edge $(i, j) \in E$ with i having an earlier deadline.

Lemma 23. *Either i acts as a good and gains $e^{w_i - 1}$, or it gains at least $1 - p_j^*$.*

Proof. It holds trivially in the boundary case when $p_j^* = w_j^* = 1$.

If $p_j > p_j^*$, j is unmatched by its deadline, which is after i's deadline. The claim holds because otherwise i should have chosen j on i's deadline.

Finally consider $p_j < p_j^*$. Compare the matching with the case of $p_j > p_j^*$ up to i's deadline. By an induction the set of vertices on the side of j is always a superset of the previous case by at most one extra vertex. Hence, if a vertex on i's side has a different match, it is because either the extra vertex on j's side buys it, or it buys the extra vertex on j's side due to a cheaper price. In particular, this holds for vertex i and proves the lemma. □

One can use this machinery of marginal prices to show that Ranking beats the greedy algorithm. The proof of this result is rather involved, however, and lies beyond the scope of this chapter.

Theorem 24. *The competitive ratio of* Ranking *in fully online matching on general graphs is at least* 0.521.

Instead we will restrict our attention to bipartite graphs. This is relevant in ride hailing, in which users act as either drivers or passengers.

Theorem 25. Ranking *is* 0.544*-competitive for the Fully Online Bipartite Matching Problem.*

Proof. Consider any edge (i, j) in the optimal matching, with i's deadline being earlier. By the definitions of j's marginal price p_j^* and marginal rank w_j^*, the expected gain of j is at least $\int_0^{w_j^*} e^{w_i - 1}\, \mathrm{d}w_j$. By Lemma 23, on the other hand, we have $g_i \geq \min\left\{e^{w_i - 1}, 1 - p_j^*\right\}$.

If $e^{w_i - 1} \geq 1 - p_j^*$, the analysis is similar to the analysis for the Online Bipartite Matching Problem:

$$g_i + \mathsf{E}\left[g_j\right] \geq 1 - e^{w_j^* - 1} + \int_0^{w_j^*} e^{w_j - 1}\, \mathrm{d}w_j = 1 - \frac{1}{e}.$$

Otherwise, we have $g_i \geq e^{w_i - 1}$. Taking expectation over w_j, a simple calculation now shows

$$\mathsf{E}\left[g_i\right] + \mathsf{E}\left[g_j\right] \geq \int_0^1 \min\left\{1 - \frac{1}{e}, e^{w_j - 1}\right\} \mathrm{d}w_j \approx 0.55418.$$

Therefore, the expected size of the matching by Ranking is at least

$$\sum_{i \in V} \mathsf{E}[g_i] \geq \sum_{(i,j) \in \mathrm{OPT}} \big(\mathsf{E}[g_i] + \mathsf{E}[g_j]\big) \geq 0.554 \cdot |\mathrm{OPT}|. \qquad \square$$

Finally, we remark that a similar yet more involved analysis gives the tight competitive ratio $\Omega \approx 0.567$ of Ranking in the Fully Online Bipartite Matching Problem, where Ω is the unique solution to $\Omega e^{\Omega} = 1$.

3.1.2 Water-Filling

We can extend the Water-Filling approach to the Fully Online Fractional Matching Problem in a similar way as we did with Ranking: each vertex i acts as a good until its deadline at which point it acts as a buyer. The price of a neighboring good $j \in N(i)$ is then $p(w_j) := \frac{1}{\sqrt{2}} w_j + 1 - \frac{1}{\sqrt{2}}$ where w_j is how much of j has been matched so far. Accordingly, each buyer will solve the convex program

$$\begin{aligned} \max_{(\Delta x_{i,j})_{j \in N(i)}} \quad & \sum_{j \in N(i)} \left(\Delta x_{i,j} - \int_{w_j}^{w_j + \Delta x_{i,j}} p(w) \,\mathrm{d}w \right) \\ \text{s.t.} \quad & \sum_{j \in N(i)} \Delta x_{i,j} \leq 1 - w_i, \\ & \Delta x_{i,j} \geq 0 \quad \forall j \in N(i) \end{aligned} \tag{3}$$

to maximize their utility and pay $\int_{w_j}^{w_j + \Delta x_{i,j}} p(w) \,\mathrm{d}w$ to each neighbor $j \in N(i)$; see Algorithm 7.

Remark 26. This is the first time that the prices do not depend exponentially on the fill level w. The exact price function is as always chosen to yield an optimal analysis.

Algorithm 7: Water-Filling for the Fully Online Fractional Matching Problem

---

for *each buyer i who departs* **do**
  Compute an optimum solution $(\Delta x_{i,j})_{j \in N(i)}$ to the CP (3).
  $x \leftarrow x + \Delta x$

As in Section 3.1.1, we let g_i be the *gain* of vertex i, i.e. the sum of its revenue while acting as a good and its utility while acting as a buyer.

Lemma 27. *Let $(i, j) \in E$ be arbitrary with i's deadline being earlier. Then at the end of Algorithm 7, we have $g_i + g_j \geq 2 - \sqrt{2}$.*

Proof. Consider the matched levels w_i and w_j right after i's deadline. Either $w_i = 1$ or $w_j = 1$; otherwise, i could have been matched more to j during its departure. Note that if $w_j = 1$, we have $g_j = \int_0^1 p(w)\,\mathrm{d}w = 2 - \sqrt{2}$, proving the above inequality.

Now assume $w_i = 1$ and $w_j < 1$. Further consider the matched level w'_i right before i's deadline, i.e. how much of i was bought while it was acting as a good. Vertex i gains $\int_0^{w'_i} p(w)\,\mathrm{d}w$ in revenue from those previous matches. In addition, it gains at least $1 - p(w_j)$ per unit of good that i buys on departure as it could have always bought j instead. Thus

$$g_i \geq \int_0^{w'_i} p(w)\,\mathrm{d}w + \big(1 - w'_i\big)\big(1 - p(w_j)\big).$$

Further by $g_j = \int_0^{w_j} p(w)\,\mathrm{d}w$, and the choice of price $p(w) = \frac{1}{\sqrt{2}}w + 1 - \frac{1}{\sqrt{2}}$ we have

$$\begin{aligned} g_i + g_j &\geq \int_0^{w'_i} p(w)\,\mathrm{d}w + \big(1 - w'_i\big)\big(1 - p(w_j)\big) + \int_0^{w_j} p(w)\,\mathrm{d}w \\ &= \frac{1}{2\sqrt{2}}\big(w'_i + w_j - 2 + \sqrt{2}\big)^2 + 2 - \sqrt{2} \;\geq\; 2 - \sqrt{2}. \end{aligned}$$ □

As a corollary we immediately obtain the following theorem.

Theorem 28. *The competitive ratio of* Water-Filling *for the Fully Online Fractional Matching Problem is* $2 - \sqrt{2} \approx 0.585$.

Unlike for the Online Biparite Matching Problem, we have thus established a better competitive ratio for Water-Filling than for Ranking. One might conjecture that both algorithms ultimately end up being $(1-1/e)$-competitive as was the case in many other settings we have studied so far. However, it is known that there is no 0.6317-competitive algorithm (note that $1-1/e \approx 0.6321$), even for the Fully Online Fractional Matching Problem.

3.2 General Vertex Arrivals

The model of general vertex arrivals is similar to the fully online setting in that it also lets all vertices arrive online and allows non-bipartite graphs. However, matches must be made when the vertices *arrive* and there is no notion of departing.

This arrival model is even more difficult to handle and the analysis of algorithms with competitive ratios strictly above 1/2 is beyond the scope of this chapter. However, we will outline a proof that Water-Filling can once again be used to beat the greedy algorithm for the fractional variant of this problem.

The algorithm is mostly the same as Algorithm 7 except that vertices are matched on arrival and that the price function $p : [0,1] \to [0,1]$ will have to take on a more complicated form. For now we will just assume that p is non-decreasing and satisfies $p(1) = 1$.

In addition, if vertex i buys fractional edges $(\Delta_{i,j})_{j \in N(i)}$ then we require that

$$\sum_{j \in N(i)} \left(\Delta_{i,j} - \int_{w_j}^{w_j + x_{i,j}} p(w)\,\mathrm{d}w \right) \geq \int_0^{w_i + \sum_{j \in N(i)} \Delta x_{i,j}} p(w)\,\mathrm{d}w.$$

In other words, the gain (in the form of utility) that i achieves by buying other vertices should never be less than the gain (in the form of revenue) that it would have made if it had been matched to the same level later. This may simply be added as a constraint to the convex program (3).

Lemma 29. *For any vertex i, we have $g_i \geq \int_0^{w_i} p(w)\mathrm{d}w$.*

Proof. Assume that upon arrival i was matched to level w_i' and then later to w_i. The total utility it achieved during its arrival must have been at least $\int_0^{w_i'} p(w)\mathrm{d}w$ by assumption. Moreover, the total revenue it made later would have been exactly $\int_{w_i'}^{w_i} p(w)\mathrm{d}w$. Together, the claim follows. □

Lemma 30. *For any edge $(i,j) \in E$ where i arrives before j, we have $g_j \geq w_j\big(1 - p(w_i)\big)$ after j arrives.*

Proof. Since j fractionally matches its neighbors to maximize its gain and p is non-decreasing, j gets at least $1 - p(w_i)$ in utility per unit of match on its arrival. □

To analyze the competitive ratio of Water-Filling, assume that there exists an auxiliary non-increasing function $f : [0,1] \to [0,1]$ with $f(1) = 0$ such that for any $w^* \in [0,1]$

$$\int_0^{f(w^*)} p(w)\mathrm{d}w \leq f(w^*)\big(1 - p(w^*)\big). \tag{4}$$

Lemma 31. *For any edge $(i,j) \in E$ where i arrives before j, $w_i = 1$ or $w_j \geq f(w_i)$ after j arrives.*

Proof. The lemma is trivially true if $w_i = 1$ or $w_j = 1$. Since f is non-increasing, we may restrict ourselves to the time just after j arrives and assume that $w_i, w_j < 1$. Then we must have $g_j = \int_0^{w_j} p(w)\,\mathrm{d}w$ and $w_j > 0$ as otherwise j could have bought more of i.

Together with Lemma 30, this implies

$$\int_0^{w_j} p(w)\mathrm{d}w \geq w_j\big(1 - p(w_i)\big).$$

But since p is non-decreasing, Equation (4) implies that $x_j \geq f(x_i)$. □

Theorem 32. *There is a price function p such that the competitive ratio of* Water-Filling *is at least* 0.523.

Proof. Consider any edge $(i, j) \in E$, where i arrives before j. By Lemmas 29, 30, and 31, the total gain of i and j is at least

$$\alpha_i + \alpha_j \geq \int_0^{w_i} p(w)\mathrm{d}w + f(w_i)\big(1 - p(w_i)\big).$$

Hence, we wish to find non-decreasing p and non-increasing f so as to maximize

$$\Gamma := \min_{w_i \in [0,1]} \left(\int_0^{w_i} p(w)\mathrm{d}w + f(w_i)\big(1 - p(w_i)\big) \right).$$

Note that this would imply a Γ-competitive algorithm.

Unfortunately, we do not know of a simple closed-form solution for which it is easy to show that $\Gamma > \frac{1}{2}$. Nonetheless, by appropriately discretizing f and p, it is possible to numerically solve for both functions while optimizing Γ. Using this approach one can find numerical solutions with $\Gamma \geq 0.523$. □

3.3 Edge Arrivals

Finally, let us briefly touch on the most general model of edge arrivals. The vertices are now entirely offline and instead, it is the edges which arrive in adversarial order. Once an edge is revealed to the algorithm, it must decide whether it wants to include it in the matching or discard it forever.

The greedy algorithm remains 1/2-competitive even with edge arrivals. Unlike the other models, however, no algorithm can do any better.

Theorem 33. *The competitive ratio of any algorithm is at best* $1/2$ *under edge arrivals even for bipartite graphs.*

Proof. For a sufficiently large n, consider a bipartite graph $G = (B, S, E)$ with n vertices on each side. Let $B = \{b_1, b_2, \dots, b_n\}$ and $S = \{s_1, s_2, \dots, s_n\}$. The edges arrive in n batches. The first batch has only one edge (b_1, s_1). The second batch has two edges (b_1, s_2) and (b_2, s_1). Generally, the k-th batch has k edges $(b_1, s_k), (b_2, s_{k-1}), \dots, (b_k, s_1)$ for $k \leq n$. We consider a family of n instances, by following the above construction up to k batches for any $k \leq n$.

Observe that for any $k \leq n$, the graph consisting of the first k batches has a unique perfect matching that consists entirely of the edges in the k-th batch. Therefore, the algorithm faces a dilemma in every state: it needs to choose enough edges in the current stage to ensure the competitive ratio should it end right away; should the instance continue, however, all these chosen edges would hurt the final matching size.

Concretely, for any $i, j \leq n$ such that $i + j \leq n + 1$, let $x_{i,j}$ denote the probability that edge (b_i, s_j) is chosen by the algorithm. Assume that the algorithm is Γ-competitive for some constant $\Gamma \geq 1/2$, then (x, Γ) is a feasible solution of

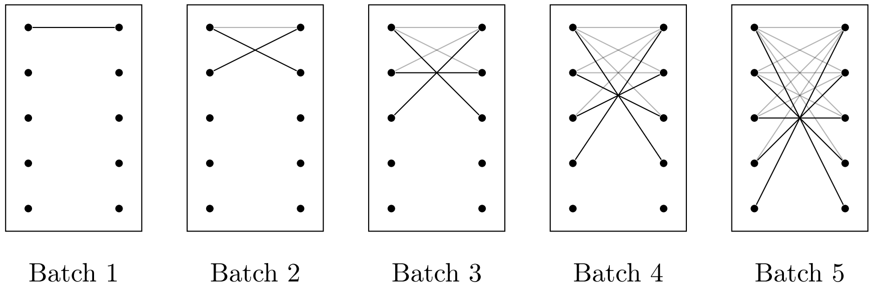

Figure 3: Shown is a hard instance for edge arrivals with 10 vertices. The edges arrive in 5 separate batches.

the following LP.

$$\max_{(x_{i,j})_{(i,j)\in E},\Gamma} \quad \Gamma \tag{5a}$$

$$\text{s.t.} \quad \sum_{j\in N(i)} x_{i,j} \le 1 \qquad \forall i \le n, \tag{5b}$$

$$\sum_{i\in N(j)} x_{i,j} \le 1 \qquad \forall j \le n, \tag{5c}$$

$$\sum_{i=1}^{k} \sum_{j=1}^{k+1-i} x_{i,j} \ge \Gamma \cdot k \qquad \forall k \le n, \tag{5d}$$

$$x_{i,j} \ge 0 \quad \forall (i,j) \in E. \tag{5e}$$

Constraint (5d) expresses the fact that the algorithm needs to be Γ-competitive even if we stop after the k-th batch.

To prove the claim it suffices to show that this LP is bounded by 1/2 as $n \to \infty$. Consider the dual LP:

$$\begin{aligned}
\min_{(\alpha_i)_{i=1}^n,(\beta_j)_{j=1}^n,(\gamma_k)_{k=1}^n} \quad & \sum_{i=1}^n \alpha_i + \sum_{j=1}^n \beta_j \\
\text{s.t.} \quad & \alpha_i + \beta_j \ge \sum_{k=i+j}^{n} \gamma_k \quad \forall i \le n, j \in N(i), \\
& \sum_{k=1}^n k \cdot \gamma_k \ge 1, \\
& \alpha_i, \beta_j, \gamma_k \ge 0 \qquad \forall i,j,k \le n.
\end{aligned}$$

By weak duality, we can bound (5a) and thus the competitive ratio Γ by providing a feasible dual assignment. For all i, j, k, set

$$\alpha_i := \beta_i := \begin{cases} \frac{n-2(i-1)}{n(n+1)} & \text{if } i \leq \frac{n}{2}+1, \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_k := \frac{2}{n(n+1)}.$$

Then one may check that the dual objective function is $1/2 + O(1/n)$. □

4 Exercises

1. Give a complete proof of Lemma 4. In particular, carry out the proof by induction mentioned in the proof sketch.

2. Give a deterministic Water-Filling algorithm for the Online Fractional Vertex-Weighted Bipartite Matching Problem and show that it is $(1-1/e)$-competitive. *Hint:* The prices should be scaled in the same way as we did for Ranking in Section 2.1.

3. Show that the family of upper-triangular graphs used in Chapter **??** to show the $1-1/e$ bound on the competive ratio for the Online Bipartite Matching Problem also yield a $1-1/e$ bound for the Online Fractional Bipartite Mathing Problem.

4. Show that a variant of the greedy algorithm is 1/2-competitive for the Online Edge-Weighted Bipartite Matching Problem with Free Disposal. Specifically, this greedy algorithm should match buyer i to good j maximizing $v_{i,j} - v_{i',j}$ where i' is the buyer that j is currently matched to.

5. Give a family of bipartite graphs $G = (S, B, E)$, edge probabilities $p_{i,j}$, and an arrival order of the online vertices B such that the ratio between the best online algorithm with stochastic rewards and the expected size of a maximum matching in the graph tends to 0. *Hint:* It suffices to consider a single arriving vertex.

6. Give greedy algorithms for the Fully Online Matching Problem and the Online Biparite Matching Problem with Edge Arrivals and show that they are 1/2-competitive.

5 Bibliographic Notes

Online matching was first introduced in the seminal work by Karp et al. (1990). The analysis of the algorithms in this chapter is largely based on Devanur et al. (2013) and the economic viewpoint on this analysis is due to Eden et al. (2021). Ranking with vertex weights was introduced by Aggarwal et al. (2011) and our proof is based on Vazirani (2021). Kalyanasundaram and Pruhs (2000) gave the

BALANCE algorithm for b-matching, which is equivalent to WATER-FILLING as b tends to infinity. The analysis of the Online Edge-Weighted Bipartite Matching Problem with Free Disposal is based on work by Devanur et al. (2016). Mehta and Panigrahi (2012) defined the setting with stochastic rewards, and proved the results covered in this chapter.

Huang et al. (2018) and Huang et al. (2019) introduced the fully online matching model and analyzed both RANKING and WATER-FILLING in this setting, proving the results from Section 3.1. Wang and Wong (2015) gave the general vertex arrival model. The primal algorithm and analysis shown here is due to Tang (2020). Lastly, Gamlath et al. (2019) proved the hardness result for the edge arrival model, Theorem 33.

References

Aggarwal, Gagan, Goel, Gagan, Karande, Chinmay, and Mehta, Aranyak. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocations. Pages 1253–1264 of: *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM.

Devanur, Nikhil R, Jain, Kamal, and Kleinberg, Robert D. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. Pages 101–107 of: *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete algorithms*. SIAM.

Devanur, Nikhil R, Huang, Zhiyi, Korula, Nitish, Mirrokni, Vahab S, and Yan, Qiqi. 2016. Whole-page optimization and submodular welfare maximization with online bidders. *ACM Transactions on Economics and Computation (TEAC)*, **4**(3), 1–20.

Eden, Alon, Feldman, Michal, Fiat, Amos, and Segal, Kineret. 2021. An Economics-Based Analysis of RANKING for Online Bipartite Matching. Pages 107–110 of: Le, Hung Viet, and King, Valerie (eds), *4th Symposium on Simplicity in Algorithms, SOSA 2021*. SIAM.

Gamlath, Buddhima, Kapralov, Michael, Maggiori, Andreas, Svensson, Ola, and Wajc, David. 2019. Online matching with general arrivals. Pages 26–37 of: *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science*. IEEE.

Huang, Zhiyi, Kang, Ning, Tang, Zhihao Gavin, Wu, Xiaowei, Zhang, Yuhao, and Zhu, Xue. 2018. How to match when all vertices arrive online. Pages 17–29 of: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*.

Huang, Zhiyi, Peng, Binghui, Tang, Zhihao Gavin, Tao, Runzhou, Wu, Xiaowei, and Zhang, Yuhao. 2019. Tight competitive ratios of classic matching algorithms in the fully online model. Pages 2875–2886 of: *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM.

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

Karp, Richard M, Vazirani, Umesh V, and Vazirani, Vijay V. 1990. An optimal algorithm for on-line bipartite matching. Pages 352–358 of: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*.

Mehta, Aranyak, and Panigrahi, Debmalya. 2012. Online matching with stochastic rewards. Pages 728–737 of: *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science*. IEEE.

Tang, Zhihao Gavin. 2020. *Personal communication*.

Vazirani, Vijay V. 2021. *Randomized Online Algorithms for Adwords*.

Wang, Yajun, and Wong, Sam Chiu-wai. 2015. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. Pages 1070–1081 of: *International Colloquium on Automata, Languages, and Programming*. Springer.