
Online Selection Problems against Constrained Adversary

Zhihao Jiang¹ Pinyan Lu² Zhihao Gavin Tang² Yuhao Zhang³

Abstract

Inspired by a recent line of work in online algorithms with predictions, we study the constrained adversary model that utilizes predictions from a different perspective. Prior works mostly focused on designing simultaneously robust and consistent algorithms, without making assumptions on the quality of the predictions. In contrary, our model assumes the adversarial instance is consistent with the predictions and aim to design algorithms that have best worst-case performance against all such instances. We revisit classical online selection problems under the constrained adversary model. For the single item selection problem, we design an optimal algorithm in the adversarial arrival model and an improved algorithm in the random arrival model (a.k.a., the secretary problem). For the online edge-weighted bipartite matching problem, we extend the classical Water-filling and Ranking algorithms and achieve improved competitive ratios.

1. Introduction

In the classical online algorithm literature, the performance of an algorithm is evaluated by its competitive ratio in the worst case. This principle has guided the design of online algorithms for decades and achieved great successes in many areas, including caching, scheduling, online matching, etc.. However, the worst case analysis is often too pessimistic when applied in practice.

A recent trend has been interested in incorporating machine-

learned advice to online algorithm designs. In particular, the algorithm is given some extra information about the instance (a.k.a. *prediction*) that are typically calculated by machine learning algorithms. The goal is then to design a simultaneously *consistent* and *robust* algorithm without knowing the quality of the prediction, meaning that the algorithm has good performance when the prediction is accurate and the worst-case performance of the algorithm is also preserved. This framework is first proposed by Purohit et al. (Purohit et al., 2018) for the ski-rental problem and the non-clairvoyant scheduling problem. Later, it is extended to online caching (Lykouris & Vassilvitskii, 2018; Rohatgi, 2020; Jiang et al., 2020; Antoniadis et al., 2020a), online covering problems (Bamas et al., 2020b), energy minimization (Bamas et al., 2020a), and secretary and online bipartite matching problems (Antoniadis et al., 2020b;a).

In this paper, we study the *constrained adversary* model, to utilize the predictions from a different perspective. In contrary to the robustness-consistency framework, we make assumptions on the quality of the prediction. Instead of a *point prediction* without any accuracy guarantee, we consider a *prediction interval* $[p_l, p_u]$ that covers the value of the real instance. I.e., the input instance must be consistent with the prediction interval. Our goal is to design an algorithm that has the best worst-case competitive ratio against this constrained adversary.

Our model is motivated by the *confidence intervals* from statistics. In statistics, each confidence interval is associated with a coverage probability, meaning the probability that the true value is covered by the interval. It is straightforward to combine statistics methods with our framework with predictions. As an illustration, we can start with a confidence interval of arbitrary coverage probability $1 - \delta$ and use it as the prediction interval of our algorithm. Straightforwardly, our competitive analysis stands with the probability at least $1 - \delta$.

1.1. Our Models and Contributions

Under the constrained adversary model, we revisit several online selection problems, including the secretary problem and the online matching problem, studied by Antoniadis et al. (Antoniadis et al., 2020b), and adapt the prediction model therein.

^{*}Equal contribution ¹Department of Management Science and Engineering, Stanford University, Stanford, California, USA ²School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China ³John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Zhihao Jiang <faebdc@stanford.edu>, Pinyan Lu <lu.pinyan@mail.shufe.edu.cn>, Zhihao Gavin Tang <tang.zhihao@mail.shufe.edu.cn>, Yuhao Zhang <zhang_yuhao@sjtu.edu.cn>.

Single Item Selection. We illustrate our constrained adversary model in the basic online single-item selection problem. Let there be n items arriving sequentially. In every round i , the value of the i -th item is revealed, and the algorithm decides immediately and irrevocably whether to select it. If the algorithm decides to select an item, the process ends; otherwise, the process continues to the next round. The goal is to maximize the selected value and we compare against the largest value of all items in hindsight. Suppose we are given a prediction interval $[p_\ell, p_u]$ that covers the largest value among all items. In other words, the optimal value must fall in the interval. Define the ratio between the upper and lower bounds of the interval, i.e. $\frac{p_u}{p_\ell}$, to be the *accuracy* of our prediction interval and denote it by Δ . The closer Δ is to 1, the better the quality of the prediction.

We make a comparison to the prediction model of Antoniadis et al. (Antoniadis et al., 2020b). They assume a point prediction p that predicts the optimal value and use $\eta = |\text{OPT} - p|$ to evaluate the error of the prediction. Given a prediction interval $[p_\ell, p_u]$, an equivalent interpretation is to set the prediction to be $p = \frac{p_\ell + p_u}{2}$ and assume that the error $\eta \leq \frac{p_u - p_\ell}{2}$.

We define $\Pi(p_\ell, p_u)$ to be the family of all instances that has the optimal value between p_ℓ and p_u and aim to design algorithms that maximize the worst-case competitive ratio against the constrained adversary, i.e.

$$\max_{\text{ALG}} \min_{I \in \Pi(p_\ell, p_u)} \frac{\text{ALG}(I)}{\text{OPT}(I)}.$$

We show that the optimal competitive ratio only depends on the accuracy of the prediction interval. And there is an intuitive analog to the robustness-consistency framework for the extreme values of Δ . Indeed, the competitive ratio when $\Delta = 1$ corresponds to the consistency, and when $\Delta \rightarrow \infty$ it corresponds to the robustness, although we adapt the max-min benchmark and the robustness-consistency model focuses on the trade-offs between the two values.

Remarkably, when Antoniadis et al. (Antoniadis et al., 2020b) applied the robustness-consistency framework to the single item selection problem, the random order arrival assumption is necessary, since otherwise, no algorithm can be robust. Generally, the robustness-consistency framework only extends those classical online algorithm problems that admit non-trivial theoretical guarantees. In contrast, by restricting the power of the adversary, the constrained adversary model poses a systematic way for studying online algorithms with predictions and raises technically interesting questions, even when the classical setting without prediction is trivial.

In Section 3, we design an optimal competitive algorithm for the adversarial arrival setting, for any fixed prediction accuracy Δ .

In Section 4, we extend our model to the random arrival order setting, i.e. the secretary problem. This is the most technical part of our paper. The main challenge is that the classical algorithm for the secretary problem is designed to maximize the probability of catching the largest item, while we aim to maximize the expectation of the value of the selected item. We consider a family of time-dependent threshold algorithms that achieve optimal competitive ratios in the extreme cases: 1) the competitive ratio is 1 when $\Delta = 1$, i.e. the prediction is exact; 2) the competitive ratio is $\frac{1}{e}$ when $\Delta \rightarrow \infty$, matching the best possible ratio of the secretary problem. For values in between the extreme cases, we lower bound the competitive ratio of our algorithm by a non-linear programming and use computer-assisted analysis.

Online Bipartite Matching. We further extend our results to the online edge-weighted bipartite matching problem. Consider an underlying edge-weighted bipartite graph $G = (L \cup R, E, w)$. The offline vertices L are known in advance and the vertices in R arrive online. Upon the arrival of a vertex, its incident edges are realized as well as their weights. The algorithm decides to match it to an unmatched neighbor or leave it unmatched. Our goal is to maximize the total weight of the selected matching and we compare against the optimal matching in hindsight. We adapt the prediction model of (Antoniadis et al., 2020b) to our setting. The algorithm is given a prediction interval per offline vertex and is guaranteed that in the optimal matching, the weight of the incident edge of v falls in its prediction interval, for every $v \in L$. The prediction accuracy of each vertex is defined to be the ratio between the upper and lower bounds of its prediction interval, and the prediction accuracy of the instance is defined to be the maximum accuracy over all vertices. As observed by Antoniadis et al. (Antoniadis et al., 2020b), when the predictions are accurate, the online edge-weighted bipartite matching problem degenerates to the vertex-weighted version of the problem and the optimal competitive ratio is $1 - \frac{1}{e}$ (Aggarwal et al., 2011).

In Section 5, we generalize the algorithm of Aggarwal (Aggarwal et al., 2011) to our constrained adversary setting. Prior to our work, the most popular model for studying online edge-weighted bipartite matching is to introduce the free-disposal of edges, so that non-trivial theoretical guarantees is achievable. Our model provides an alternative option. Our competitive ratio matches the optimal $1 - \frac{1}{e}$ ratio when the prediction accuracy equals 1 and converges to the optimal competitive ratio achieved in the single item setting up to lower order terms when the prediction accuracy goes to infinity.

Compare to Sun et al.(2017)’s work Our matching model is closely related to the bounded weight online bi-

partite matching problem studied by Sun et al. (Sun et al., 2017). They assume all given edge weight is bounded in an interval $[\alpha, \beta]$ while we have a prediction interval for every offline vertex. Because we can omit all the edges outside the prediction interval, it's the same to say that each offline vertex v has an individual edge weight range $[\alpha_v, \beta_v]$. In Sun et al.'s algorithm, they directly run ranking on a randomly generalized subgraph with a guessed edge-weight range $[\alpha \cdot 2^i, \alpha \cdot 2^{i+1}]$, and they suffer a factor $(1 - 1/e)$ (from RANKING) $\cdot O(1/\log \frac{\beta}{\alpha})$ (from guessing) $= O(1/\log \frac{\beta}{\alpha})$. However, our algorithm combined our exactly optimal single item selection algorithm and RANKING via a primal-dual method which can get a better competitive ratio than the product of the two factors. To sum up, the two models are the same in the single item case, and we make the competitive ratio exactly optimal (they can archive $O(1/\log \frac{\beta}{\alpha})$ by guessing). In the multi-item case, our algorithm can be transferred to the bounded weight setting, and we improve the competitive ratio of Sun et al.'s algorithm at any $\Delta > 1$ ($\Delta = \frac{\beta}{\alpha}$) not only because we use the optimal single-item selecting idea but also because we combine the RANKING idea in a better way.

1.2. Related Work

The most related work is the recent NIPS paper by (Antoniadis et al., 2020b), that includes a comprehensive review of the related literature. We only discussed the most related ones here.

Algorithms with predictions. The constrained adversary model is previously considered by Purohit (Purohit, 2019) for the ski-rental problem in his TTIC talk. In their model, the instance is guaranteed to be at least γ days and improved competitive algorithm is designed. This is the only work that studies the constrained adversary model in online algorithms, to the best of our knowledge. Most works in the algorithms with predictions literature aim to balance the consistency and robustness of the algorithm. Medina and Vassilvitskii (Munoz & Vassilvitskii, 2017) studied the revenue optimization problem, with a prediction of the bid value. Purohit et al. (Purohit et al., 2018) considered the ski rental problem and the Non-clairvoyant scheduling problem, and they gave the competitive ratios as a function of the prediction error. For the ski-rental setting, Wei and Zhang (Wei & Zhang, 2020) provided the tight robustness-consistency trade-off; Gollapudi et al. (Gollapudi & Panigrahi, 2019) further extended it to the multiple predictors, and they provided and evaluated experimentally tight algorithms. In contrast to viewing the prediction generation as a black box, Anand et al. (Anand et al., 2020) customizing machine learning algorithms directly for optimization tasks in the ski rental problem. Lykouris and Vassilvitskii (Lykouris & Vassilvitskii, 2018) studied the caching problem with pre-

dictions and succeeded in incorporating the prediction to the Marker algorithm (Fiat et al., 1994). Later the result has been improved by Rohatgi (Rohatgi, 2020), Jiang et al. (Jiang et al., 2020) and Antoniadis et al. (Antoniadis et al., 2020a). Lattanzi et al. (Lattanzi et al., 2020) and Mitzenmacher (Mitzenmacher, 2020) studied the online scheduling problem with predictions in different perspectives. Notice that Mitzenmacher introduce a new quality measure called the price of misprediction to evaluate his algorithm. Bhaskara et al. (Bhaskara et al., 2020) considered the online linear optimization problem with prediction. For readers who are interested, Roughgarden's book (Roughgarden, 2021) includes a chapter *algorithms with predictions*.

Secretary Problem. The secretary problem is a fundamental model in online stopping and online optimization. It is first formulated by Dynkin (Dynkin, 1963), in which Dynkin designed a algorithm with $1/e$ probability of selecting the best candidate. The $1/e$ ratio is proven to be the best possible ratio from different perspectives (Ferguson et al., 1989; Correa et al., 2019). To consider more than one accepted candidate, the matroid secretary problem has been studied by Babaioff et al. (Babaioff et al., 2018), and they presented an $O(\log k)$ -competitive algorithm for general matroids (where k is the rank of the matroid). A closely related setting is the known i.i.d setting, in which the value of each item is drawn independently from a known distribution and the objective is to maximize the probability of selecting the best item. Gilbert and Mosteller (Gilbert & Mosteller, 1966) proposed an algorithm with winning probability 0.58 and their policy has been proved to be optimal in several follow-up works (Samuels, 1982; Berezovskiy & Gnedin, 1984; Gnedin, 1996).

Online Bipartite Matching. The online bipartite matching model is proposed by Karp et al. (Karp et al., 1990), and they designed the celebrated RANKING algorithm with optimal competitive ratio $(1 - 1/e)$. Later, Aggarwal (Aggarwal et al., 2011) studied the vertex-weighted generalization and obtained the same optimal competitive ratio $(1 - 1/e)$. Non-trivial competitive algorithms are impossible in the natural edge-weighted extension of the online bipartite matching problem. As discussed in the introduction, the most popular model is to allow free disposal of edges. A recent breakthrough is by Fahrback et al. (Fahrback et al., 2020), which achieved the first non-trivial competitive ratio 0.5086. Besides the weight extension, Huang et al. (Huang et al., 2018; 2019a; 2020a) studied the fully online version of the online matching problem, which allows all vertices to arrive online and the underlying graph to be non-bipartite. They prove RANKING can archive 0.5671 and 0.5211 competitive ratio in the bipartite and general case. Recently, Huang et al. (Huang et al., 2020b) proposed a better algorithm that beat RANKING in the bipartite case, which improves the com-

petitive ratio to be 0.569. Stochastic models are also considered in the online bipartite matching literature. Karande et al. (Karande et al., 2011) and Mahdian et al. (Mahdian & Yan, 2011) studied the random arrival model for the unweighted version of the online bipartite matching problem, and they proved the competitive ratio of Ranking is at least 0.696. Huang et al. (Huang et al., 2019b) considered the vertex-weighted version with random arrivals and proposed an algorithm with competitive ratio 0.653.

2. Preliminaries

We set up formal notations to describe the online selection problems studied by our paper. In the single item setting, let there be n items and let $\{v_i\}_{i \in [n]}$ be their values. The algorithm is given a prediction interval $[p_\ell, p_u]$ upfront and is guaranteed that the optimal value falls in the interval. We shall refer to p_ℓ as the prediction and define the accuracy of the prediction to be $\Delta \stackrel{\text{def}}{=} \frac{p_u}{p_\ell}$.

In the online bipartite matching problem, there is an underlying *edge-weighted* bipartite graph $G = (L \cup R, E, w)$. Let $[p_v, \Delta_v p_v]$ be the prediction interval of v , for each offline vertex $v \in L$. In the optimal matching M^* , every vertex $v \in L$ is matched to an edge with weight between p_v and $\Delta_v p_v$. We refer to p_v as the prediction of v and Δ_v as the prediction accuracy of vertex v . The prediction accuracy Δ of the instance is defined to be $\max_{v \in L} \Delta_v$.

For all results achieved in this paper, our competitive ratios depend solely on the prediction accuracy of the instance.

3. Single Item: Adversarial Arrivals

In this section, we study the single item selection problem with worst case arrival order. Without loss of generality, we assume the prediction interval to be $[1, \Delta]$ and the optimal value $v^* = \max_i v_i$ is guaranteed to be in $[1, \Delta]$. We start with a deterministic fractional algorithm and then provide a lossless online rounding of the fractional algorithm. Finally, we prove the optimality of our algorithm.

Algorithm 1 Online Singlatioe Item Selection with Prediction

- 1: Fix a non-decreasing function $g : [0, 1] \rightarrow [1, \Delta]$.
 - 2: Let $x = 0$. $\{x$ is the portion that have been selected. $\}$
 - 3: Upon the arrival of an item i with value v_i :
 - 4: **If** $v_i \geq g(x)$,
 - 5: $x \leftarrow g^{-1}(v_i)$ {We select $(g^{-1}(v_i) - x)$ fraction of i , where $g^{-1}(v) \stackrel{\text{def}}{=} \max\{x \mid g(x) \leq v\}$.}
-

Fractional Algorithm. Our algorithm admits an intuitive interpretation. At any step of the instance, we maintain a threshold based on the currently selected portion of our al-

gorithm and then continuously select the current item whenever its value exceeds the threshold. Indeed, our threshold increases when the selected portion increases. Refer to Algorithm 1 for a formal definition.

The choice of function g is optimized based on the analysis of our algorithm. For the purpose of a clearer presentation, we first present the explicit formula of g and prove a mathematical fact of it, which is essential to our competitive analysis.

Lemma 3.1. *There exists a non-decreasing function $g(\cdot)$ that $\forall v \in [1, \Delta]$, $\int_0^{g^{-1}(v)} g(x) dx = \Gamma \cdot v$, where $\Gamma = \frac{1}{\ln \Delta + 1}$.*

Proof. Let

$$g(x) = \begin{cases} 1 & x \in [0, \Gamma] \\ \Delta^x \cdot e^{x-1} & x \in [\Gamma, 1] \end{cases} \quad (1)$$

It is straightforward to verify $g(\cdot)$ is a continuous non-decreasing function because $g(\Gamma) = \Delta^\Gamma \cdot e^{\Gamma-1} = 1$. Then, we have that $g^{-1}(v) \in [\Gamma, 1]$ for $v \in [1, \Delta]$, to conclude the lemma, it suffices to show that $\int_0^{x^*} g(x) dx = \Gamma \cdot g(x^*)$ for all $x^* \in [\Gamma, 1]$:

$$\begin{aligned} \int_0^{x^*} g(x) dx &= \Gamma + \int_\Gamma^{x^*} \Delta^x \cdot e^{x-1} dx \\ &= \Gamma + \int_\Gamma^{x^*} e^{(\ln \Delta + 1)x - 1} dx \\ &= \Gamma + \Gamma \cdot e^{(\ln \Delta + 1)x - 1} \Big|_\Gamma^{x^*} = \Gamma \cdot g(x^*). \end{aligned}$$

□

Theorem 3.1. *Algorithm 1 with the function g chosen in Eqn (1) is $\Gamma = \frac{1}{\ln \Delta + 1}$ -competitive.*

Proof. Let $v^* \in [1, \Delta]$ be the optimal value. Suppose the maximum item arrives as the n -th item. Let x_i be the selected portion of our algorithm after the arrival of the i -th item for all $i \in [n]$. We have that 1) x_i is non-decreasing, and 2) if $x_i > x_{i-1}$, the value of the i -th item equals $g(x_i)$. Hence, our gain from the i -th item equals $(x_i - x_{i-1}) \cdot g(x_i)$. Moreover, we have that $x_n \geq g^{-1}(v^*)$, as otherwise we would continue selecting the n -th item. Therefore,

$$\begin{aligned} \text{ALG} &= \sum_{i=1}^n (x_i - x_{i-1}) \cdot g(x_i) \geq \int_0^{x_n} g(x) dx \\ &\geq \int_0^{g^{-1}(v^*)} g(x) dx = \Gamma \cdot v^*, \end{aligned}$$

where the last equation follows from Lemma 3.1. This concludes the competitive ratio of our algorithm. □

Integral Algorithm. Next, we give a natural online rounding of the above fractional algorithm.

1. Draw a number x uniformly at random from $[0, 1]$.
2. Set a threshold $g(x)$ and select the first item whose value is at least the threshold.

Theorem 3.2. *The randomized integral algorithm with the function g chosen in Eqn (1) is $\Gamma = \frac{1}{\ln \Delta + 1}$ -competitive.*

Proof. Let $v^* \in [1, \Delta]$ be the optimal value. Observe that the algorithm ends up with accepting an item if and only if the random threshold $g(x)$ is at most v^* . Let $v(x)$ be the value of the selected item when the threshold is chosen to be $g(x)$ for $x \in [0, g^{-1}(v^*)]$. Then, we have

$$\mathbb{E}[\text{ALG}] = \int_0^{g^{-1}(v^*)} v(x) dx \geq \int_0^{g^{-1}(v^*)} g(x) dx = \Gamma \cdot v^*,$$

where the inequality follows from the definition of our threshold-based algorithm and the last equation follows from Lemma 3.1. \square

Optimality. Finally, we prove that the $\Gamma = \frac{1}{\ln \Delta + 1}$ competitive ratio is tight for all online (randomized) algorithms. Indeed, we prove a stronger statement that no *fractional* algorithm can have better competitive ratio.

Theorem 3.3. *For any $\Delta \geq 1$, no fractional algorithm can achieve a competitive ratio better than $\frac{1}{\ln \Delta + 1}$.*

Proof. Fix arbitrary n , let $\epsilon = \frac{\Delta-1}{n}$. Let there be n items, where the value of the i -th item is $v_i = 1 + (i-1) \cdot \epsilon$. Consider the following family of instances. In all cases, the items arrive from the smallest weight to the largest. However, there are n possible stopping times, that the instance stops right after the arrival of the i -th item, for each $i \in [n]$. Fix an arbitrary fractional algorithm. Let x_i be the selected portion of the algorithm after the arrival of the i -th item. Let Γ be the competitive ratio of the algorithm. It must satisfy that the algorithm is Γ -competitive for all the n instances defined above. Therefore, Γ is bounded by the objective of the following linear program.

$$\begin{aligned} \max : & \Gamma_n \\ \text{subject to: } & \sum_{j=1}^i (x_j - x_{j-1}) \cdot v_j \geq \Gamma_n \cdot v_i, \quad \forall i \in [n]; \\ & 0 = x_0 \leq x_1 \leq \dots \leq x_n \leq 1. \end{aligned}$$

We then solve the above optimization problem. Multiplying $\left(\frac{1}{v_i} - \frac{1}{v_{i+1}}\right)$ to the first family of constraints and summing

them up, we have

$$\begin{aligned} & \sum_{i=1}^n \left(\frac{1}{v_i} - \frac{1}{v_{i+1}}\right) \sum_{j=1}^i (x_j - x_{j-1}) \cdot v_j \\ & \geq \Gamma_n \cdot \sum_{i=1}^n \left(\frac{1}{v_i} - \frac{1}{v_{i+1}}\right) \cdot v_i, \end{aligned}$$

where we use v_{n+1} to denote infinity for notation simplicity. Rearranging the left hand side and the right hand side of the equation gives that

$$\begin{aligned} \text{LHS} &= x_n + \sum_{j=1}^{n-1} x_j \cdot \left(v_j \cdot \sum_{i=j}^n \left(\frac{1}{v_i} - \frac{1}{v_{i+1}}\right) \right. \\ & \quad \left. - v_{j+1} \cdot \sum_{i=j+1}^n \left(\frac{1}{v_i} - \frac{1}{v_{i+1}}\right) \right) \\ &= x_n \leq 1; \\ \text{RHS} &= \sum_{i=1}^n \frac{v_{i+1} - v_i}{v_{i+1}} = \sum_{i=1}^{n-1} \frac{\epsilon}{v_{i+1}} + 1 \\ & \xrightarrow{n \rightarrow \infty} \int_1^{\Delta} \frac{1}{v} dv + 1 = \ln \Delta + 1. \end{aligned}$$

Observe that Γ is at most the optimal solution Γ_n^* of the above linear program for all n . When n goes to infinity, Γ_n^* tends to $\frac{1}{\ln \Delta + 1}$, that concludes the proof. \square

4. Single Item: Random Arrivals

In this section, we study the secretary problem with prediction. The total number of items n is known upfront by the algorithm. Again, with out loss of generality, assume the prediction interval is $[1, \Delta]$ and the optimal value is in $[1, \Delta]$. Within this section, we assume there is a time horizon from 0 to 1 and the arriving time t_i of each item i is drawn independently and uniformly from $[0, 1]$. Moreover, we are aware of the current time t when an item arrives. This model is equivalent to the random arrival assumption by the following folklore reduction. We first draw n independent numbers uniformly from $[0, 1]$. Then, when the i -th item arrives, let the current time be the i -th largest number among the random numbers.

Recall that without the prediction (i.e., $\Delta = \infty$), the optimal competitive algorithm for the classical secretary problem is to reject all items before time $\frac{1}{e}$ and accept the first best-so-far item afterwards¹. With a prediction in hand, an immediate improvement over this algorithm is to add an extra condition for accepting items: accept an item only if its value is at least 1. After all, we are guaranteed the existence

¹This is equivalent to the algorithm that rejects the first $\text{Bin}(n, \frac{1}{e})$ items and accept the first best-so-far item afterwards.

of an item with value at least 1. The modified algorithm achieves an improved competitive ratio that depends on the accuracy Δ . On the other hand, the algorithm does not fully exploit the prediction as we never accept items before time $\frac{1}{e}$. Consider the case when an item with value Δ shows up at the beginning. Any reasonable algorithm should accept it since we know there is no better item.

Based on the above two observations, our algorithm is designed to incorporate them in a smoother way. Consider the following time-dependent threshold algorithm:

Time-dependent Threshold Algorithm. Fix a non-increasing function $h : [0, 1] \mapsto [1, \Delta]$. When an item arrives at time t_i , the algorithm accept it if it is the best-so-far and its value is at least the time-dependent threshold $h(t_i)$. I.e, $v_i \geq \max_{j < i} v_j$ and $v_i \geq h(t_i)$.

The algorithm described above belongs to this family of algorithms and can be characterized by the function $h(t) = \Delta$ for $t < \frac{1}{e}$ and $h(t) = 1$ for $t \geq \frac{1}{e}$. Next, we present a general analysis for all time-dependent threshold algorithms and do the optimization of function h afterwards.

4.1. Analysis

For any non-increasing function h , let $h^{-1}(v)$ to be the inverse function of h , i.e.

$$h^{-1}(v) = \min_{t \in [0, 1]} \{h(t) \leq v\}.$$

Let v^* be the maximum value among all items. We prove two lower bounds of algorithm's selected value in the following lemma:

Lemma 4.1. *Let i be item with the largest value v^* , and j be largest one released before t_i .*

1. *If $t_i \geq h^{-1}(v^*)$ and $t_j < h^{-1}(v^*)$, the algorithm gets v^* .*
2. *If $t_i \geq h^{-1}(v^*)$ and $t_j > h^{-1}(v^*)$, the algorithm gets at least $h(t_j)$.*

Proof. In the case when $t_i \geq h^{-1}(v^*)$ and $t_j < h^{-1}(v^*)$, we claim that no item can be accepted before time $h^{-1}(v^*)$. Because v^* is the max value among all items and the time-requirement function is non-increasing, all items arrive before $h^{-1}(v^*)$ do not have large enough value to be accepted. Then, all items arrive in $[h^{-1}(v^*), t_i]$ can not be accepted because these items are not larger than v_j . Finally, since item i satisfies both conditions of our algorithm, we select it and get v^* .

In the case when $t_i \geq h^{-1}(v^*)$ and $t_j > h^{-1}(v^*)$, no item can be accepted before $h^{-1}(v^*)$ for the same reason. Then, we have two possible cases: 1) the algorithm accepts an

item before or at t_j , 2) the algorithm does not accept any item before or at t_j . In the first case, because of the time-dependent threshold, the algorithm gets at least $h(t_j)$. In the second case, since all items arriving between (t_j, t_i) cannot be the best-so-far, the algorithm rejects them all. Consequently, our algorithm selects item i and gets $v^* \geq h(t_j)$. \square

Equipped with the above lemma, we are ready to prove the competitive ratio of any time-dependent threshold algorithm.

Lemma 4.2. *Time-dependent threshold algorithm with function h has competitive ratio*

$$\Gamma = \inf_{v^* \in [1, \Delta]} \left\{ \ln\left(\frac{1}{h^{-1}(v^*)}\right) \cdot h^{-1}(v^*) + \frac{1}{v^*} \int_{h^{-1}(v^*)}^1 \ln \frac{1}{x} \cdot h(x) dx \right\}.$$

Proof. Let i be item with the largest value v^* , and j be largest one released before t_i . For any $t \in [h^{-1}(v^*), 1]$, with probability

$$\Pr [t_j < h^{-1}(v^*) \mid t_i = t] = \frac{h^{-1}(v^*)}{t},$$

t_j is smaller than $h^{-1}(v^*)$ and the algorithm gets v^* by the first statement of Lemma 4.1. For any $t_j = t' \in [h^{-1}(v^*), t]$, the algorithm gets at least $h(t_j)$ by the second statement of Lemma 4.1. Combining them together, we have that

$$\begin{aligned} & \mathbb{E}[\text{ALG}] \\ & \geq \int_{h^{-1}(v^*)}^1 \left(\frac{h^{-1}(v^*)}{t_i} \cdot v^* + \int_{h^{-1}(v^*)}^{t_i} \frac{h(t_j)}{t_i} dt_j \right) dt_i \\ & = \ln \left(\frac{1}{h^{-1}(v^*)} \right) \cdot h^{-1}(v^*) \cdot v^* \\ & \quad + \int_{h^{-1}(v^*)}^1 h(t_j) \cdot \left(\int_{t_j}^1 \frac{1}{t_i} dt_i \right) dt_j \\ & = \ln \left(\frac{1}{h^{-1}(v^*)} \right) \cdot h^{-1}(v^*) \cdot v^* \\ & \quad + \int_{h^{-1}(v^*)}^1 \ln \left(\frac{1}{t_j} \right) \cdot h(t_j) dt_j. \end{aligned}$$

By definition, Γ is the worst ratio among all choices of $v^* \in [1, \Delta]$, which completes the proof. \square

4.2. Optimization

In this subsection, we give a numerical method to optimize the choice of the time-requirement function h that maximize the competitive ratio. General speaking, we restrict h to be

a step function with N pieces, and prove that the minimum value of Γ must be achieved at the $N - 1$ discontinuous points or at Δ . Hence, we can form a non-linear optimization problem with $O(N)$ variables and constraints to characterize the competitive ratio Γ and solve it by computer. We leave the formal definition and proofs in the appendix file in the Supplementary Material and only list results here.

Numerical Experiments. We solve the optimization on a standard PC with the KNITRO solver. We compare the result of competitive ratios for different accuracies Δ of the N -step threshold algorithm to the algorithm in adversary setting (Refer to Alg. 1). See the following table for the competitive ratios for various values of N and Δ .

	N=2	N=3	N=10	N= 20	N=100	Alg. 1
$\Delta=1$	1	1	1	1	1	1
$\Delta=1.5$	0.6832	0.7394	0.7718	0.7758	0.7786	0.7115
$\Delta=3$	0.4820	0.5486	0.6152	0.6241	0.6304	0.4765
$\Delta=10$	0.3962	0.4340	0.5028	0.5153	0.5242	0.3027
$\Delta=100$	0.3705	0.3808	0.4248	0.4380	0.4482	0.1784

Table 1. Competitive ratios of N -step functions for different accuracies Δ .

Observe that the threshold algorithm achieves competitive ratio at least $1/e$ for all Δ since the optimal $1/e$ -competitive algorithm can be characterized by a 2-step function according to previous discussions. On the other hand, the competitive ratio of Alg 1 approaches 0 when Δ goes to infinity. See Figure 1a for this observation. When Δ goes to infinity, the competitive ratio of our algorithm converges to $1/e$, but the convergence speed is relatively slow. Focusing on small Δ , the threshold algorithm achieves significant improvement on the competitive ratio over $\frac{1}{e} \approx 0.367$ for sufficiently large N , and it performs strictly better than Alg 1 when $N \geq 3$. See Figure 1b for this observation.

Finally, we briefly discuss the optimized time-requirement function h , which should be approached when N grows to infinity. By our experiment, we observe that when the prediction accuracy is small, the optimized time-requirement function is close to linear. When the accuracy grows, the time-requirement function consists of two pieces, where the first piece is concave and the second piece is convex. In Figure 2, we plot three concrete results, for fixed $N = 100$, with respect to three different accuracies $\Delta = 3, 10, 100$.

5. Online Edge-weighted Bipartite Matching with Predictions

In this section, we extend our results to the online edge-weighted bipartite matching setting with predictions. We focus on the setting with adversarial arrival order. As observed by (Antoniadis et al., 2020b), when the prediction

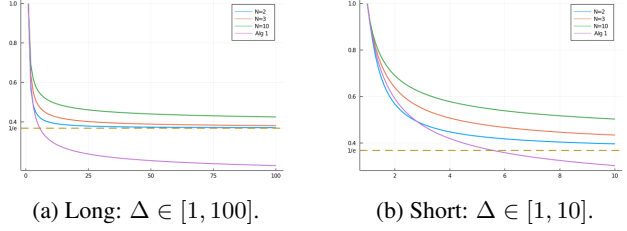


Figure 1. The competitive ratios of Alg 1 and the N -step threshold algorithm as functions of Δ .

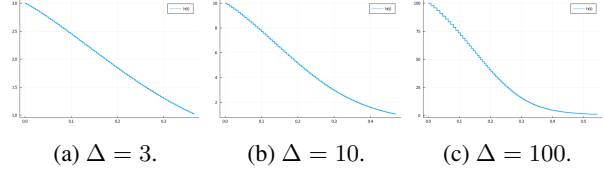


Figure 2. The optimized 100-step time-requirement function h for $\Delta = 3, 10, 100$.

is exact (i.e. $\Delta = 1$), the edge-weighted problem is degenerated to the vertex-weighted case. We first design a fractional algorithm that generalizes the water-filling algorithm from the vertex-weighted online bipartite matching setting. Then, we provide a Ranking-like rounding of the fractional algorithm that preserves the competitive ratio.

Fractional algorithm. Consider the following economic interpretation of our algorithm. Fix a non-decreasing function $g : [0, 1] \mapsto [0, \Delta]$. For each offline vertex v , let x_v be its matched portion. At any time, set the price of vertex v to be $p_v g(x_v)$. Upon the arrival of an online vertex u , u could pay the price of v in order to match v and to receive the edge weight w_{uv} as a reward. u would then continuously matches to the neighbor that offers the largest utility, i.e. $w_{uv} - p_v g(x_v)$, until u is fully matched or none of its neighbors offers positive utility. Refer to Algorithm 2 for a formal definition. We use variables r_v/r_u to keep track of the revenue/utility of offline/online vertices respectively. Note that these variable are only for the analysis.

Theorem 5.1. *There exists a function g so that Algorithm 2 is Γ -competitive, where $\Gamma = \ell^{-1}(\Delta)$ and $\ell(y) = e^{\frac{1+\ln(1-y)}{y}}$.*

Proof. Let M^* be the optimal matching. Fix an arbitrary pair $(v, u) \in M^*$, where $v \in L$ and $u \in R$. According to the setting, we have $w_{uv} \in [p_v, \Delta \cdot p_v]$. Consider the moment right after u 's arrival and let x_v be the matched portion of v . Then we have $r_v = \int_0^{x_v} p_v g(x) dx$. If $x_v < 1$ and $p_v g(x) < w_{uv}$, we must have that u is fully matched (i.e. $x_u = 1$) after its arrival, as otherwise, u would continue matching with v . Furthermore, we know that $r_u \geq w_{uv} - p_v g(x_v)$, since the prices of all offline vertices only increase

Algorithm 2 Online edge-weighted bipartite matching with prediction (fractional)

- 1: For each $v \in L$, let $x_v = 0, r_v = 0$. $\{x_v, r_v$ denote v 's matched portion and revenue}
- 2: Upon the arrival of an online vertex u :
- 3: Let $x_u = 0, r_u = 0$. $\{x_u, r_u$ denote u 's matched portion and utility}
- 4: Let $N(u)$ be the set of neighbors of v .
- 5: **While** $x_u < 1$ **and** $\max_{v \in N(u): x_v < 1} \{w_{uv} - p_v g(x_v)\} > 0$ **do**
- 6: Let $v^* = \arg \max_{v \in N(u): x_v < 1} \{w_{uv} - p_v g(x_v)\}$.
- 7: u matches with v^* for a small fraction dx .
- 8: Increase x_u and x_v by dx .
- 9: Increase r_v by $p_v g(x_v) dx$ and r_u by $(w_{uv} - p_v g(x_v)) dx$.

during the process.

To sum up, the total utility of v is at least $\int_0^{x_v} p_v g(x) dx$, and the total utility of u is at least $w_{uv} - p_v g(x_v)$ if $x_v < 1$. Note that $r_u \geq 0$. Considering whether $x_v = 1$ or $x_v < 1$, we have

$$r_u + r_v \geq \min \left\{ \begin{array}{l} \int_0^1 p_v g(y) dy, \\ \int_0^{x_v} p_v g(x) dx + \max(w_{uv} - p_v g(x_v), 0) \end{array} \right\} \quad (2)$$

Finally, we conclude the proof with the mathematical fact below:

Lemma 5.1. *There exists a non-decreasing function g that $\forall w_{uv} \in [p_v, \Delta \cdot p_v]$,*

$$\min \left\{ \begin{array}{l} \int_0^1 p_v g(y) dy, \int_0^{x_v} p_v g(x) dx \\ \max(w_{uv} - p_v g(x_v), 0) \end{array} \right\} \geq \Gamma \cdot w_{uv}.$$

where $\Gamma = \ell^{-1}(C)$ and $\ell(\Gamma) = e^{\frac{1+\ln(1-\Gamma)}{\Gamma}}$.

Therefore, by the definition of r , we have

$$\begin{aligned} \text{ALG} &= \sum_{u \in R} r_u + \sum_{v \in L} r_v \geq \sum_{(u,v) \in M^*} (r_u + r_v) \\ &\geq \sum_{(u,v) \in M^*} \Gamma \cdot w_{uv} = \Gamma \cdot \text{OPT}. \end{aligned}$$

□

We leave the mathematical calculation of proving Lemma 5.1 in the appendix file in the Supplementary Material.

Remark 5.1. *When $\Delta = 1$, the function g can be simplified as $g(x) = e^{x-1}$ and our algorithm becomes the $1 - 1/e$ -competitive optimal algorithm of the online vertex-weighted bipartite matching problem by (Aggarwal et al., 2011; Devanur et al., 2013). When Δ grows, our competitive ratio quickly converges to the optimal competitive ratio of the single item setting from Section 3. In other words, when the predictions are inaccurate (i.e., Δ is large), the difficulty of the problem caused by the matching structure is negligible. See Figure 3 for this observation.*

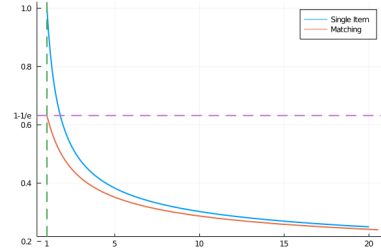


Figure 3. The competitive ratio

Ranking-like Rounding. Next, we give a lossless randomized rounding of the fractional Algorithm 2. Roughly speaking, the fractional algorithm continuously increases the price of each offline vertex according to its matched portion, while the integral algorithm sets a fixed randomized price at the beginning. This is a natural generalization of the classical Ranking algorithm (Karp et al., 1990) for the online vertex-weighted bipartite matching problem. In particular, we use the same non-decreasing function $g(x)$ as previously:

$$g(x) = \begin{cases} e^{x-(1-\Gamma \ln \Delta)}, & x < 1 - \Gamma \ln \Delta; \\ \Delta \cdot e^{(x-1)/\Gamma}, & x \geq 1 - \Gamma \ln \Delta, \end{cases} \quad (3)$$

where $\Gamma = \ell^{-1}(\Delta)$ and $\ell(y) = e^{\frac{1+\ln(1-y)}{y}}$. Then, we draw a rank y_v independently for each offline vertex v uniformly from $[0, 1]$, and set v 's price as $p_v g(y_v)$. Consider at the time when an online vertex u chooses v and matches $e = (u, v)$, the total reward that create should be w_e . As the price required, the reward w_e should be enough to cover the price $p_v g(y_v)$ as v 's utility r_v . As a result, u 's utility r_u should be equal to the remaining reward $w_e - p_v g(y_v)$. Assume u is rational, when u comes, it should always choose the unmatched neighbor who can maximize its utility $w_e - p_v g(y_v)$. See Algorithm 3 for a formalized version of the integral matching algorithm.

Theorem 5.2. *Algorithm 3 with g defined above in Eqn (4) is Γ -competitive.*

The analysis is similar to the analysis of Ranking by (Karp et al., 1990). Thus, we can have that the same gain as that in

Algorithm 3 Online edge-weighted bipartite matching with prediction (integral)

- 1: For each $v \in B$, draw y_v uniformly and independently from $[0, 1]$. $\{y_v$ denotes v 's rank}
- 2: When an online vertex u is reached:
- 3: Let $\tilde{N}(u)$ be the set of unmatched neighbors of v .
- 4: **if** $\max_{v \in \tilde{N}(u)} \{w_{uv} - p_v g(y_v)\} > 0$ **do**
- 5: Let $v^* = \arg \max_{v \in \tilde{N}(u)} \{w_{uv} - p_v g(y_v)\}$.
- 6: u matches with v^* .
- 7: $r_v = p_v g(y_v)$, $r_u = w_{uv} - p_v g(y_v)$.

Eqn 2. For completeness, we include a proof in the appendix file in the Supplementary Material.

6. Conclusion and Future Directions

In this work, we study several online selection problems under the constraint adversary model. The principle of our model is to design competitive algorithms when the prediction is within a reasonable range. Our framework fits in the theme of beyond worst-case algorithm design. First, the constrained adversary model allows us to study online problems that do not admit non-trivial solutions in the worst-case scenario. Our results show that we can achieve constant competitive ratios with any constant prediction accuracy in the secretary and edge-weighted bipartite matching problem with *adversarial arrival*. Observe that without the predictions, both problems have unbounded competitive ratios.

Moreover, by studying structured/constrained adversary, we are able to design algorithms with better competitive ratios, compared to the corresponding unconstrained setting. Our results show that we can strictly improve the classical tight competitive ratio $1/e$ in the secretary problem, for any fixed prediction accuracy. Obtaining tight competitive ratio for any fixed accuracy Δ remains an interesting open question.

Finally, the constrained adversary model is naturally applicable to other online problems. We briefly discuss a few interesting directions to explore in the future. 1) Combining the random arrival model with the online edge-weighted bipartite matching problem studied in this paper, what would be a natural extension of the algorithm of Huang et al. (Huang et al., 2019b)? 2) The online makespan minimization problem is another well-studied online problem. Constant competitive ratio is achieved in the worst-case scenario. Assume that we are given the prediction of the optimal makespan in advance, can we achieve an improved competitive ratio against a constrained adversary?

Acknowledgments

This work is supported by Science and Technology Innovation 2030 — “New Generation of Artificial Intelligence” Major Project No.(2018AAA0100903), Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities. Zhihao Gavin Tang is supported by NSFC grant 61902233.

References

- Aggarwal, G., Goel, G., Karande, C., and Mehta, A. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *SODA*, pp. 1253–1264. SIAM, 2011.
- Anand, K., Ge, R., and Panigrahi, D. Customizing ml predictions for online algorithms. In *International Conference on Machine Learning*, pp. 303–313. PMLR, 2020.
- Antoniadis, A., Coester, C., Elias, M., Polak, A., and Simon, B. Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning*, pp. 345–355. PMLR, 2020a.
- Antoniadis, A., Gouleakis, T., Kleer, P., and Kolev, P. Secretary and online matching problems with machine learned advice. In *NeurIPS*, 2020b.
- Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. Matroid secretary problems. *J. ACM*, 65(6):35:1–35:26, 2018. doi: 10.1145/3212512. URL <https://doi.org/10.1145/3212512>.
- Bamas, É., Maggiori, A., Rohwedder, L., and Svensson, O. Learning augmented energy minimization via speed scaling. In *NeurIPS*, 2020a.
- Bamas, É., Maggiori, A., and Svensson, O. The primal-dual method for learning augmented algorithms. In *NeurIPS*, 2020b.
- Berezovskiy, B. and Gnedin, A. The best choice problem. *Akad. Nauk, USSR*, 1984.
- Bhaskara, A., Cutkosky, A., Kumar, R., and Purohit, M. Online learning with imperfect hints. In *International Conference on Machine Learning*, pp. 822–831. PMLR, 2020.
- Correa, J., Dütting, P., Fischer, F., and Schewior, K. Prophet inequalities for iid random variables from an unknown distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 3–17, 2019.
- Devanur, N. R., Jain, K., and Kleinberg, R. D. Randomized primal-dual analysis of RANKING for online bipartite matching. In *SODA*, pp. 101–107. SIAM, 2013.

- Dynkin, E. B. The optimum choice of the instant for stopping a markov process. *Soviet Mathematics*, 4:627–629, 1963.
- Fahrbach, M., Huang, Z., Tao, R., and Zadimoghaddam, M. Edge-weighted online bipartite matching. *arXiv preprint arXiv:2005.01929*, 2020.
- Ferguson, T. S. et al. Who solved the secretary problem? *Statistical science*, 4(3):282–289, 1989.
- Fiat, A., Rabani, Y., and Ravid, Y. Competitive k-server algorithms. *Journal of Computer and System Sciences*, 48(3):410–428, 1994.
- Gilbert, J. P. and Mosteller, F. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, pp. 35–73, 1966.
- Gnedin, A. V. On the full information best-choice problem. *Journal of applied probability*, pp. 678–687, 1996.
- Gollapudi, S. and Panigrahi, D. Online algorithms for rent-or-buy with expert advice. In *International Conference on Machine Learning*, pp. 2319–2327. PMLR, 2019.
- Huang, Z., Kang, N., Tang, Z. G., Wu, X., Zhang, Y., and Zhu, X. How to match when all vertices arrive online. In Diakonikolas, I., Kempe, D., and Henzinger, M. (eds.), *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pp. 17–29. ACM, 2018. doi: 10.1145/3188745.3188858. URL <https://doi.org/10.1145/3188745.3188858>.
- Huang, Z., Peng, B., Tang, Z. G., Tao, R., Wu, X., and Zhang, Y. Tight competitive ratios of classic matching algorithms in the fully online model. In Chan, T. M. (ed.), *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 2875–2886. SIAM, 2019a. doi: 10.1137/1.9781611975482.178. URL <https://doi.org/10.1137/1.9781611975482.178>.
- Huang, Z., Tang, Z. G., Wu, X., and Zhang, Y. Online vertex-weighted bipartite matching: Beating $1-1/e$ with random arrivals. *ACM Transactions on Algorithms (TALG)*, 15(3):1–15, 2019b.
- Huang, Z., Kang, N., Tang, Z. G., Wu, X., Zhang, Y., and Zhu, X. Fully online matching. *J. ACM*, 67(3):17:1–17:25, 2020a. doi: 10.1145/3390890. URL <https://doi.org/10.1145/3390890>.
- Huang, Z., Tang, Z. G., Wu, X., and Zhang, Y. Fully online matching II: beating ranking and water-filling. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pp. 1380–1391. IEEE, 2020b. doi: 10.1109/FOCS46700.2020.00130. URL <https://doi.org/10.1109/FOCS46700.2020.00130>.
- Jiang, Z., Panigrahi, D., and Sun, K. Online algorithms for weighted paging with predictions. In *ICALP*, volume 168 of *LIPICs*, pp. 69:1–69:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- Karande, C., Mehta, A., and Tripathi, P. Online bipartite matching with unknown distributions. In *STOC*, pp. 587–596, 2011.
- Karp, R. M., Vazirani, U. V., and Vazirani, V. V. An optimal algorithm for on-line bipartite matching. In *STOC*, pp. 352–358, 1990.
- Lattanzi, S., Lavastida, T., Moseley, B., and Vassilvitskii, S. Online scheduling via learned weights. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1859–1877. SIAM, 2020.
- Lykouris, T. and Vassilvitskii, S. Competitive caching with machine learned advice. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3302–3311. PMLR, 2018.
- Mahdian, M. and Yan, Q. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs. In *STOC*, pp. 597–606, 2011.
- Mitzenmacher, M. Scheduling with predictions and the price of misprediction. In *ITCS*, 2020.
- Munoz, A. and Vassilvitskii, S. Revenue optimization with approximate bid predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 1858–1866. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/884d79963bd8bc0ae9b13a1aa71add73-Paper.pdf>.
- Purohit, M. Ski-rental with predictions: Three models. In *TTIC workshop slides*, 2019.
- Purohit, M., Svitkina, Z., and Kumar, R. Improving online algorithms via ML predictions. In *NeurIPS*, pp. 9684–9693, 2018.
- Rohatgi, D. Near-optimal bounds for online caching with machine learned advice. In *SODA*, pp. 1834–1845. SIAM, 2020.
- Roughgarden, T. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021. ISBN 9781108494311. URL <https://books.google.com.hk/books?id=FF8JEAAAQBAJ>.

Samuels, S. Exact solutions for the full information best choice problem. *Purdue Univ. Stat. Mimeo Series*, pp. 81–87, 1982.

Sun, X., Zhang, J., and Zhang, J. Near optimal algorithms for online weighted bipartite matching in adversary model. *J. Comb. Optim.*, 34(3):689–705, 2017. doi: 10.1007/s10878-016-0100-2. URL <https://doi.org/10.1007/s10878-016-0100-2>.

Wei, A. and Zhang, F. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. *arXiv preprint arXiv:2010.11443*, 2020.