

Primal-Dual Algorithms with Predictions for Online Bounded Allocation and Ad-Auctions Problems

Enikő Kevi and Nguyễn Kim Thắng

LIG, Grenoble INP, University Grenoble-Alpes, France

Abstract

Matching problems in general and Ad-Auctions problems in particular have been widely studied with many applications from network design to advertising. With the advance of machine learning, a natural question is to study whether one can benefit from machine learning to improve the performance of algorithms for such problems, given that an improvement of a small percentage would result in significant increases of the network throughput or of the revenue in Ad-Auctions. In this paper, we consider the design of algorithms with learning predictions for the Online Bounded Allocation and the Online Ad-Auctions problems. We present primal-dual algorithms that achieve competitive performance depending on the quality of predictions while always maintaining the best performance guarantees in the worst-case analysis when the predictions are misleading. We additionally provide supporting experiments for our theoretical findings.

1 Introduction

The matching problem is fundamental in combinatorial optimization and operations research with wide applications from student admission to colleges through kidney exchange to ad-auctions. Motivated by different markets, for example, advertising and labor markets, online bipartite matching has been intensively studied. Online bipartite matching revolves around matching a set of items (impressions) to a set of agents (advertisers). Online items arrive over time, and at the arrival of an item, one needs to make an irrevocable decision on the assignment of the item to an agent. For unweighted bipartite graphs, Karp et al. [14] gave an elegant algorithm - the RANKING algorithm - which always outputs a matching of size at least $(1 - 1/e)$ times that of the optimal solution [14, 5]. The ratio of $(1 - 1/e)$ is the best achievable *competitive ratio* in the worst-case paradigm. Besides, for online matchings in edge-weighted bipartite graphs, no online algorithm is competitive¹ to maximize the total (edge-) weight of the output matching, even though this problem is well-motivated by advertising and AdWords markets (see, for example, [10]).

To circumvent the limitations of the worst-case paradigm, several models have been proposed [27, 28]. One of the models is motivated by spectacular advances in machine learning (ML). In particular, the capability of ML methods in predicting patterns of future requests could provide valuable information that online algorithms can exploit. Lykouris and Vassilvtiskii [19] formally introduces a general framework to incorporate ML predictions into the algorithm design to improve the worst-case performance. Other researchers followed their work, studying online algorithms with predictions [25] in a large spectrum of problems, such as scheduling [17, 24], caching (paging)

¹except in relaxed models such as disposal-free, etc.

[19, 26, 3] and ski rental [11, 16, 2]. In this paper, we study the design of algorithms with predictions for variants of online edge-weighted matching.

1.1 Model and Problems

We consider a model (formally stated in [4]) in which, whenever a request is released, an algorithm receives some predictions and can use these predictions to make decisions. The designed algorithms aim to outperform the best-known algorithms when the predictions are correct, but maintain close worst-case guarantees when the predictions are incorrect. Specifically, we study the following problems in this model.

Online Bounded Allocation. In this problem, there are n buyers, each buyer $1 \leq i \leq n$ has a budget B_i . Items arrive over time; upon the arrival of item j , the price b_j of the item as well as the set of buyers $S_j \subseteq \{1, 2, \dots, n\}$ who are interested in purchasing j are revealed. In the classic online setting, one needs to *irrevocably* sell item j to some buyer in S_j (or sell it to no one) while always respecting the budget constraints of all buyers. In a learning-enhanced model with predictions, each item has a predicted buyer to whom the item should be sold. This additional information may come from a learning procedure based on the data analysis of buyers and items. We consider the learning procedure as a black box and treat the predictions as outputs of an oracle. Concretely, at the arrival of item j , the oracle provides a predicted buyer $\text{pred}(j)$ to whom the item j should be sold (conventionally, $\text{pred}(j) = 0$ if the item is predictably not sold). The objective is to maximize the revenue, which is the total price of sold items.

Online Ad-auction. This problem is the generalization of Online Bounded Allocation. In this setting, items do not have a fixed price; instead, at the arrival of item j , each buyer i proposes a price (or bid) b_{ij} for each item j . (Buyers can decline to buy an item by bidding 0 value.) We assume that the bids are significantly smaller than the buyers' budgets, so $b_{ij} \ll B_i \forall i, j$. Similarly, at the arrival of item j , one receives the prediction $\text{pred}(j)$ and decides to whom to sell the item. The objective is again to maximize the revenue, which is the total price of sold items.

Learning augmented algorithms. We aim to design algorithms that incorporate predictions to achieve performance beyond the worst-case bounds. The learning augmentation depends on the algorithm's confidence in the prediction oracle. We represent the algorithm's doubt in the prediction by a parameter $\eta \in [0, 1]$ and assume that η is fixed during the algorithm's execution. Large η values represent high doubt, while small η values show good confidence.

Let $\mathcal{A}(I)$ be the objective value of the solution produced by algorithm \mathcal{A} on an instance I . Similarly, let $\mathcal{P}(I)$ and $\mathcal{O}(I)$ be the objective values of the prediction oracle and the optimal solution, respectively. When the prediction oracle provides an infeasible solution, $\mathcal{P}(I) = 0$. Given a confidence parameter $\eta \in [0, 1]$, we say that an algorithm \mathcal{A} is $c(\eta)$ -consistent and $r(\eta)$ -robust if for every instance I ,

$$\mathcal{A}(I) \geq \max\{c(\eta) \cdot \mathcal{P}(I), r(\eta) \cdot \mathcal{O}(I)\}.$$

Ideally, we would like $c(\eta)$ to tend to 1 when η approaches 0, meaning that with high confidence, the algorithm performs at least as well as the prediction. Additionally, we would like $r(\eta)$ to tend to the best guarantee as in the standard online setting (without predictions) when η approaches 1.

1.2 Our Approach and Contributions

Given an algorithm A that blindly follows the predictions and another algorithm B , for example, the best-known algorithm without predictions, a natural question is whether we can derive a new and more efficient algorithm C by taking the linear combination of the two original algorithms. Due to the linear combination, algorithm C can only maintain a consistency of $O(1 - \eta)$ for Online Bounded Allocation (or Ad-Auction), if the robustness is $\Omega(\eta)$ times the worst-case guarantee of algorithm B . In this paper, we aim for more substantial and non-trivial guarantees.

We rely on the primal-dual approach to design learning augmented algorithms for both studied problems. The primal-dual method is an elegant and powerful algorithm design technique [30], especially for online algorithms [6]. To unify previous ad-hoc approaches, Bamas et al. [4] presented a primal-dual method to design online algorithms with predictions for covering problems with linear objective functions. However, their approach is not applicable for problems with packing constraints, particularly matching problems and their variants. In this paper, we present learning augmented online algorithms with predictions for the Online Bounded Allocation and the Online Ad-Auction problems, answering some open questions raised in [4].

Specifically, in Section 2, we provide an algorithm which is $(1 - \eta)$ -consistent and $(\frac{e-1}{e} \cdot \frac{1}{1+(1-\eta)(1-e^{\eta-1})})$ -robust for the Online Bounded Allocation problem. In Section 3, we give a $(1 - \eta)$ -consistent and $(1 - e^{-\eta})$ -robust algorithm for the Online Ad-Auctions problem. Note that by varying η between 0 and 1, one can observe the tradeoff between the consistency and the robustness (therefore the smoothness) of the algorithms. Similarly to [4], we provide algorithms that produce fractional solutions for the studied problems. When the algorithms have high confidence in the prediction (meaning that η is closed to 0), the algorithms achieve a similar objective value as the prediction; and when the confidence is low (η is close to 1), the algorithms guarantee the best worst-case bound, $(e - 1)/e$. This robustness guarantee holds by our algorithms even if the predictions give an infeasible solution (realized only during the execution).

1.3 Related work

The domain of algorithms with predictions [25], or learning augmented algorithms, has recently emerged and rapidly grown at the intersection of (discrete) algorithm design and machine learning (ML). Its main concept is to incorporate learning predictions and ML techniques in the algorithm design to achieve performance guarantees beyond the worst-case analysis and provide specifically adapted solutions to different problems. Recent studies showed interesting results over a large spectrum of problems, such as scheduling [17, 24], caching (paging) [19, 26, 3], ski rental [11, 16, 2], counting sketches [12] and bloom filters [15, 23]. Bamas et al. [4] recently proposed a primal-dual approach to design online algorithms with predictions for linear problems with covering constraints.

The online matching and ad-auction problems have been widely studied (see [21] and references therein). Mehta et al. [22] introduced the Online Ad-Auction problem and gave an optimal $(1 - 1/e)$ competitive ratio when $R_{\max} = \max_{i,j}\{b_{ij}/B_i\}$ is small. Buchbinder et al. [8] simplified the analysis by a primal-dual analysis. In the same paper, the authors gave an algorithm with a refined competitive ratio for the Online Bounded Allocation problem. Aggarwal et al. [1] studied another particular case in which for each i , the bids b_{ij} 's are the same for every j . They obtained the optimal $(1 - 1/e)$ competitive ratio with the generalization of the RANKING algorithm [14]. In this problem (without any assumptions), the existence of an $(1 - 1/e)$ -competitive algorithm has been conjectured, but remained an open problem. Huang et al. [13] recently presented a 0.5016-competitive algorithm for this problem.

Motivated by internet advertising applications, several works considered the ad-auction problem

$$\begin{array}{ll}
\max \sum_{j=1}^m \sum_{i \in S_j} b_j x_{ij} & \min \sum_{i=1}^n B_i y_i + \sum_{j=1}^m z_j \\
\sum_{j: i \in S_j} b_j x_{ij} \leq B_i \quad \forall i & (y_i) \quad b_j y_i + z_j \geq b_j \quad \forall j, i \in S_j \quad (x_{ij}) \\
\sum_{i \in S_j} x_{ij} \leq 1 \quad \forall j & (z_j) \quad y_i \geq 0 \quad \forall i \\
x_{ij} \geq 0 \quad \forall i, j & z_j \geq 0 \quad \forall j
\end{array}$$

Figure 1: Formulation of the Online Bounded Allocation problem

in various settings where forecasts or predictions are available or learnable. Esfandiari et al. [9] proposed a model in which the input is stochastic, and the model gets a forecast for future items. Intuitively, we can measure the forecast accuracy by the optimal solution’s fraction one can obtain from the stochastic input. They provide algorithms with provable bounds in this setting. Schild et al. [29] introduced a semi-online model in which the unknown future has a predicted and an adversarial part. They gave algorithms with competitive ratios depending on the fraction of the adversarial part in the input. Closely related to our work is the model by Mahdian et al. [20] in which, given two algorithms, one needs to design a (new) algorithm that is robust to both given algorithms. They derived an algorithm for the ad-auction problem that achieves a fraction of the maximum revenue of the given algorithms. The main difference compared to our model is that their algorithm is *not* robust if one of the given algorithms provides infeasible solutions (which could happen with predictions), whereas our algorithm is.

2 An Algorithm with Predictions for Online Bounded Allocation

Recall that in this problem, there are n buyers, each buyer $1 \leq i \leq n$ has a budget B_i . Upon the arrival of item j , we discover its fixed price b_j , the subset of buyers S_j interested in purchasing, and also gets a prediction about buyer $\text{pred}(j)$ to whom the item j should be sold ($\text{pred}(j) = 0$ if the item is predictably not sold). As we are interested in fractional solutions as in [4], before the arrival of the next item, we need to make an *irrevocable* decision and sell the current item in some fractions to some buyers. (In other words, the items are splittable.) The objective is to gain maximum revenue from selling items to buyers.

Formulation. Let x_{ij} be the fraction of item j sold to buyer i . The problem can be cast as the following primal linear program in Figure 1 (in which we also show its dual program).

Algorithm. Buchbinder et al. [8] proposes an algorithm for the Online Bounded Allocation problem without predictions. They define *buyer levels* based on the fraction of the spent budget. Their algorithm splits each arriving item j equally among its interested buyers $i \in S_j$ who are on the lowest level. Intuitively, this algorithm corresponds to water-filling the buyer levels.

We propose an algorithm that allocates items using the water-filling strategy of [8] and subtly incorporating the predictions. Recall that the parameter η represents the confidence in the predictions. When η is close to 1, our algorithm resembles the water-filling algorithm, while with η close to 0, the prediction has a stronger impact on the algorithm’s decision. To enable water-filling, our

algorithm uses buyer levels as well. At any moment during the execution, a buyer i 's level is ℓ , if the fraction of buyer i 's spent budget is within the range of $[\frac{\ell}{d}B_i, \frac{\ell+1}{d}B_i)$. We formally describe the algorithm as follows. (One can also find the pseudo-code of the algorithm in Appendix A.)

Upon the arrival of a new item j , let i^* be the predicted buyer $\text{pred}(j)$ suggested by the oracle. Do the following.

Stage 1: As long as some interested buyers in S_j spend less than η fraction of their budget, the algorithm allocates item j equally to buyers in S_j who are on the lowest level.

Stage 2: Once all interested buyers spend at least η fraction of their budgets, the algorithm assigns the remaining fraction of item j to the predicted buyer i^* . The prediction assignment ends when one of the two following conditions occurs:

- (a) the algorithm assigned to the predicted buyer either $(1 - \eta)$ fraction or the remaining fraction of j , whichever is smaller;
- (b) buyer i^* exhausted its budget.

Stage 3: If the algorithm does not assign item j completely during Stage 1 or 2, and there exists a buyer in S_j that does not exhaust its budget, then, similarly to the first step, the algorithm allocates the remaining fraction of item j equally to buyers in S_j who are on the lowest level.

Analysis. We first prove the consistency and then the robustness of the algorithm.

Lemma 1 *Algorithm 2 is $(1 - \eta)$ -consistent with the prediction.*

Proof Let V be the set of buyers who exhaust their budget at some point during the algorithm's execution. Let U be the set of items whose buyers did not exhaust their budgets, formally, $U = \{j : \text{pred}(j) \notin V\}$. We can formulate the total gain of a feasible prediction as

$$\sum_{j:\text{pred}(j)\neq\emptyset} b_j = \sum_{j:\text{pred}(j)\in V} b_j + \sum_{j:\text{pred}(j)\notin V} b_j \leq \sum_{i\in V} B_i + \sum_{j\in U} b_j \quad (1)$$

where the inequality holds since any feasible allocation - including the prediction - can allocate items of value at most B_i (the total budget) to buyer i .

Let $j \in U$ be an arbitrary item. By the construction of the algorithm and the fact that the predicted buyer for item j did not exhaust its budget, the following can occur. The algorithm assigned item j entirely during the limit assignment of Stage 1 (to satisfy the condition of η fraction spent by each interested buyer), or otherwise allocated the remaining fraction of item j to its predicted buyer i up to $(1 - \eta)$ fraction. We note that the algorithm can always assign the remaining fraction, since buyer i did not exhaust its budget by the end of the execution. In any case, we sell item j in at least $(1 - \eta)$ fraction, therefore the gain of the algorithm on j is at least $(1 - \eta) b_j$.

Before the prediction assignment of Stage 2, the total value of items in U sold to buyers in V is at most $\sum_{i\in V} \eta B_i$ due to the limit assignment rule of Stage 1. As a consequence, each predicted buyer i^* where $i^* \notin V$ gets at least $(1 - \eta) - \frac{\eta \sum_{i\in V} B_i}{b_j}$ fraction of item j . Therefore, the total gain of the algorithm is at least

$$\sum_{i\in V} B_i + \sum_{j\in U} (1 - \eta) b_j - \eta \sum_{i\in V} B_i \quad (2)$$

where the first term is the gain of the algorithm restricted to buyers in V , the second one is the gain of the algorithm restricted to items in U , and the last term is the upper bound of the total value of items in U sold to buyers in V before the prediction assignment of Stage 2. By (1) and (2), the gain of the algorithm is at least $(1 - \eta)$ that of the prediction. \square

We are now establishing the robustness of the algorithm. Let $d = \max\{|S_j|\}$ the *bound* on the number of interested buyers. We will characterize the robustness as the function of d .

Define a piece-wise linear function $f_d : [0, 1] \rightarrow \mathbb{R}_{\geq 0}$ such that $f_d(1) = 1$ and for $0 \leq u < 1$, if $\frac{\ell-1}{d} \leq u < \frac{\ell}{d}$ for some $1 \leq \ell \leq d$ then

$$\begin{aligned} f_d(u) &= da_\ell \left(u - \frac{\ell-1}{d} \right) + da_{\ell-1} \frac{1}{d} + da_{\ell-2} \frac{1}{d} + \dots + da_1 \frac{1}{d} \\ &= da_\ell \left(u - \frac{\ell-1}{d} \right) + a_{\ell-1} + a_{\ell-2} + \dots + da_1 \end{aligned}$$

where

$$a_1 = \frac{1}{d(1 + \frac{1}{d-1})^{d-1} - (d-1)} \quad \text{and} \quad a_\ell = a_1 \left(1 + \frac{1}{d-1} \right)^{\ell-1} \quad \forall 2 \leq \ell \leq d.$$

Informally, f_d is linear with coefficient da_ℓ on every interval $[\frac{\ell-1}{d}, \frac{\ell}{d})$ for $1 \leq \ell \leq d$. Note that the maximum derivative of f_d is $1/C(d)$, where

$$C(d) = \frac{d \cdot (1 + \frac{1}{d-1})^{d-1} - (d-1)}{d \cdot (1 + \frac{1}{d-1})^{d-1}} = 1 - \frac{d-1}{d \cdot (1 + \frac{1}{d-1})^{d-1}} \xrightarrow{d \rightarrow \infty} \frac{e-1}{e}.$$

and $C(d)$ is always larger than $(e-1)/e$.

Lemma 2 *The robustness of Algorithm 2 is $1/(C(d) + (1-\eta)(1-f_d(\eta)))$. When d is large enough, $f_d(\eta) \approx 1 + \frac{e(e^{\eta-1}-1)}{e-1}$ and so the robustness is approximately $\frac{e-1}{e} \cdot \frac{1}{1+(1-\eta)(1-e^{\eta-1})}$.*

Proof We fix an arbitrary item j and bound the ratio of increase of the primal and the dual objective values caused by the arrival of item j . We set the dual variables as

$$y_i = f_d \left(\frac{\sum_j b_j x_{ij}}{B_i} \right) \quad \forall i, \quad z_j = (1 - f_d(\min_{i \in S_j} \{\ell_i\})) b_j$$

where ℓ_i is the level of buyer i at the end of the algorithm's execution.

By the definition of the dual variables, $y_i \geq f_d(\min_{i' \in S_j} \{\ell_{i'}\})$ for every $i \in S_j$. Therefore, $b_j y_i + z_j \geq b_j$ for every $i \in S_j$, which indicates that the dual variables are feasible.

Let us assume that item j is not entirely sold during the algorithm's execution. This means that all interested buyers in S_j exhausted their budget. Hence, by the definition of the dual variables, $\forall i \in S_j : y_i = 1$ and $z_j = 0$. The rate of change of the dual objective value related only to the y -variables is at most:

$$B_i \frac{\partial f_d}{\partial x_{ij}} \leq B_i \cdot \frac{b_j}{B_i} \cdot f'_d \left(\frac{\sum_j b_j x_{ij}}{B_i} \right) \leq \frac{b_j}{C(d)}$$

where $1/C(d)$ is the maximum derivative of f_d . Besides, the increasing rate of the primal objective value is b_j . We obtain that the primal change is at least $1/C(d)$ that of the dual.

In the remaining of the proof, we assume that the algorithm sold all items completely. Note that z_j can be written as $z_j = \int_0^1 (1 - f_d(\min_{i \in S_j} \{\ell_i\})) b_j dy$. In other words, one can imagine that during the allocation of item j , z_j is increasing at a rate of $(1 - f_d(\min_{i \in S_j} \{\ell_i\})) b_j$.

There are three stages in the algorithm. In Stages 1 and 3, the algorithm always allocates the fractions of item j equally to buyers in S_j on the lowest level. Buchbinder et al. [8] showed that during these allocations, the increasing rate of the dual is at most $1/C(d)$ of the primal. We provide the proof in Lemma 3 for completeness, which is similar to the proof of [8].

We are now interested in the allocation during Stage 2 of the algorithm. Denote $i^* = \text{pred}(j)$. In this stage, the algorithm allocates a part of item j only to the predicted buyer i^* . The increasing rate of the dual objective value related only to y_{i^*} is

$$B_{i^*} \frac{\partial f_d}{\partial x_{i^*j}} \leq B_{i^*} \cdot \frac{b_j}{B_{i^*}} \cdot f'_d \left(\frac{\sum_j b_j x_{i^*j}}{B_{i^*}} \right) \leq \frac{b_j}{C(d)}$$

since $f'_d(u) \leq 1/C(d)$ for all $0 \leq u \leq 1$. We note that every buyer $i \in S_j$ has spent at least η fraction of its budget before this stage, so $z_j \leq (1 - f_d(\eta)) b_j$. Therefore, the total increasing rate of the dual in this step is at most

$$B_{i^*} \frac{\partial f_d}{\partial x_{i^*j}} + \frac{\partial z_j}{\partial x_{i^*j}} \leq \frac{b_j}{C(d)} + (1 - f_d(\eta)) b_j.$$

Combining all the cases, the increasing rate of the dual is at most

$$\begin{cases} \frac{b_j}{C(d)} + (1 - f_d(\eta)) b_j & \text{during Stage 2,} \\ \frac{b_j}{C(d)} & \text{during Stage 1 and 3.} \end{cases}$$

We highlight that the algorithm allocates at most $(1-\eta)$ fraction of item j during Stage 2. Therefore, the total increase of the dual due to the arrival of j is at most:

$$\int_0^{(1-\eta)} \left(\frac{b_j}{C(d)} + (1 - f_d(\eta)) b_j \right) dy + \int_0^\eta \left(\frac{b_j}{C(d)} \right) dy = \left(\frac{1}{C(d)} + (1 - \eta)(1 - f_d(\eta)) \right) b_j.$$

Besides, the increase of the primal is b_j , so we can deduce that the robustness of the algorithm is $1 / \left(\frac{1}{C(d)} + (1 - \eta)(1 - f_d(\eta)) \right)$.

We compute and estimate the value of $f_d(\eta)$ for large value of d . We have

$$\begin{aligned} f_d(\eta) &= f_d \left(\frac{\lfloor \eta \cdot d \rfloor}{d} \right) = a_1 \left(d \left(1 + \frac{1}{d-1} \right)^{\lfloor \eta \cdot d \rfloor - 1} - (d-1) \right) \\ &= \frac{d \left(1 + \frac{1}{d-1} \right)^{\lfloor \eta \cdot d \rfloor - 1} - (d-1)}{d \left(1 + \frac{1}{d-1} \right)^{d-1} - (d-1)} \\ &= 1 + \frac{d \left(1 + \frac{1}{d-1} \right)^{\lfloor \eta \cdot d \rfloor - 1} - d \left(1 + \frac{1}{d-1} \right)^{d-1}}{d \left(1 + \frac{1}{d-1} \right)^{d-1} - (d-1)} \\ &= 1 + \left(\frac{d \left(1 + \frac{1}{d-1} \right)^{\lfloor \eta \cdot d \rfloor - 1}}{d \left(1 + \frac{1}{d-1} \right)^{d-1}} - 1 \right) \cdot \frac{1}{1 - \frac{(d-1)}{d \left(1 + \frac{1}{d-1} \right)^{d-1}}} \\ &= 1 + \frac{1}{C(d)} \left(\left(1 + \frac{1}{d-1} \right)^{\lfloor \eta \cdot d \rfloor - d} - 1 \right) \\ &\xrightarrow{d \rightarrow \infty} 1 + \frac{e(e^{\eta-1} - 1)}{e-1} \end{aligned}$$

□

Lemma 3 *We assume that j is completely sold by the algorithm. Then, at all times during the allocations of Stages 1 and 3 of Algorithm 2, the increasing rate of the dual objective value is at most $1/C(d)$ that of the primal.*

Proof The proof follows the proof of Buchbinder and Naor [7, Theorem 13.1]. We recall that item j was sold entirely. We consider two cases.

Case 1: The highest level item j was sold at is ℓ , and at the end of the algorithm, all buyers in S_j spent at least $\frac{\ell+1}{d}$ fraction of their budget.

By the definition of the dual variables $z_j \leq (1 - f_d(\frac{\ell+1}{d})) b_j$. Since item j is sold entirely ($\sum_i x_{ij} = 1$), the increasing rate of z_j at any time during the allocation of item j is at most $(1 - f_d(\frac{\ell+1}{d})) b_j$. Besides, as the highest level on which item j was sold at is ℓ , the rate of change of the dual value due to the change in y_i is:

$$B_i \frac{\partial f_d}{\partial x_{ij}} \leq B_i \cdot \frac{b_j}{B_i} \cdot d \cdot a_{\ell+1} \leq b_j d a_{\ell+1}$$

Therefore, the total change of the dual is at most

$$\begin{aligned} & b_j d a_{\ell+1} + b_j \left(1 - f_d\left(\frac{\ell+1}{d}\right)\right) \\ &= b_j d a_1 \left(1 + \frac{1}{d-1}\right)^\ell + b_j \left(1 - a_1 \left(d \left(1 + \frac{1}{d-1}\right)^\ell - (d-1)\right)\right) \\ &= b_j(1 + a_1(d-1)) = \frac{b_j}{C(d)} \end{aligned} \tag{3}$$

Case 2: The highest level on which item j was sold at is ℓ and at the end of the execution at least one buyer in S_j spent less than $\frac{\ell+1}{d}$ fraction of its budget.

In this case we have to set $z_j = (1 - f_d(\ell/d)) b_j$. During any (short) period of time Δt , the increase of $\sum_i B_i y_i$ over i in the highest level is at most:

$$b_j \cdot a_{\ell+1} \cdot d \cdot \frac{d-1}{d} \cdot \Delta t = b_j (d-1) a_{\ell+1} \Delta t$$

where the fraction $(d-1)/d$ comes from the fact that there are at most $(d-1)$ buyers in the highest level. We recall that d is the bound on the size of all S_j and at least one buyer did not reach the highest level as of the statement of this case. The increase of $\sum_i B_i y_i$ over i in the lower levels is at most $b_j d a_\ell (1 - \Delta t)$. By definition $a_{\ell+1} = a_\ell(1 + 1/(d-1))$ and $a_\ell = ((d-1)/d)a_{\ell+1}$. Therefore, by adding the terms together we get that the increasing rate of $\sum_i B_i y_i$ is at most $b_j (d-1) a_{\ell+1}$. Hence, the increasing rate of the dual is at most:

$$\begin{aligned} & b_j (d-1) a_{\ell+1} + b_j \left(1 - f_d\left(\frac{\ell}{d}\right)\right) \\ &= b_j (d-1) a_{\ell+1} + b_j \left(1 - f_d\left(\frac{\ell+1}{d}\right) + a_{\ell+1}\right) \\ &= b_j d a_{\ell+1} + b_j \left(1 - f_d\left(\frac{\ell+1}{d}\right)\right) = \frac{b_j}{C(d)} \end{aligned}$$

where the first equality holds since $f_d(\frac{\ell+1}{d}) = f_d(\frac{\ell}{d}) + a_{\ell+1}$ and the last equality follows Equation (3).

In both cases the increasing rate of the dual is at most $b_j/C(d)$, while for the primal it is b_j . The lemma follows from this conclusion. \square

Theorem 1 *Algorithm 2 is $(1 - \eta)$ -consistent and $1/(\frac{1}{C(d)} + (1 - \eta)(1 - f_d(\eta)))$ -robust. When d is large enough, $f_d(\eta) \approx 1 + \frac{e(e^{\eta-1}-1)}{e-1}$ and so the robustness is approximately $\frac{e-1}{e} \cdot \frac{1}{1+(1-\eta)(1-e^{\eta-1})}$.*

Proof Follows from Lemma 1 and Lemma 2. □

3 An Algorithm with Predictions for Ad-Auction

Recall that in this problem, there are n buyers, each buyer $1 \leq i \leq n$ has a budget B_i . Upon the arrival of item j , we discover the bid² $b_{ij} \geq 0$ a buyer i needs to pay to purchase item j for all $1 \leq i \leq n$, and also gets a prediction about buyer $\text{pred}(j)$ to whom the item j should be sold ($\text{pred}(j) = 0$ if the item is predictably not sold). As we are interested in fractional solutions as in [4], before the arrival of the next item, we need to make an *irrevocable* decision and sell the current item in some fractions to some buyers. (In other words, the items are splittable.) The objective is to gain maximum revenue from selling items to buyers. In this problem, we assume that the $b_{ij} \ll B_i \forall i, j$.

Formulation. The formulation of the Online Ad-auction problem follows that of the Online Bounded Allocation problem. Both the primal and the dual linear programs on Figure 2 now use the buyer-dependent prices, the bids.

$$\begin{array}{ll}
 \max \sum_{i=1}^n \sum_{j=1}^m b_{ij} x_{ij} & \min \sum_{i=1}^n B_i y_i + \sum_{j=1}^m z_j \\
 \sum_{j=1}^m b_{ij} x_{ij} \leq B_i \quad \forall i & (y_i) \quad b_{ij} y_i + z_j \geq b_{ij} \quad \forall i, j \quad (x_{ij}) \\
 \sum_{i=1}^n x_{ij} \leq 1 \quad \forall j & (z_j) \quad y_i \geq 0 \quad \forall i \\
 x_{ij} \geq 0 \quad \forall i, j & z_j \geq 0 \quad \forall j
 \end{array}$$

Figure 2: Formulation of the Online Ad-auction problem

Algorithm. We introduce a fictitious buyer with identity 0, such that $b_{0j} = 0$ for all item j . When the algorithm does not sell an item, it assigns it to the fictitious buyer 0. The purpose of buyer 0 is to simplify the description of the algorithm. We use a constant C in our algorithm that we define as

$$C = (1 + R_{\max})^{\frac{\eta}{R_{\max}}} \quad \text{where } R_{\max} = \max_{i,j} \left\{ \frac{b_{ij}}{B_i} \right\}$$

We present the pseudo-code of our algorithm in Algorithm 1. For each arriving item j , we consider two buyers: the predicted buyer $i^* = \text{pred}(j)$ and buyer i who maximizes $b_{ij} (1 - y_j)$, where y_j is an indicator to know how exhausted the buyer's budget is. We make use of the confidence parameter in the predictions. Our algorithm reserves $(1 - \eta)$ fraction of each buyer's budget for the prediction

²we call now bids instead of prices due to the motivations which come from the auctions setting.

assignment. Whenever the bid of buyer i^* is greater than the bid of buyer i , the algorithm assigns $(1 - \eta)$ fraction of item j to i^* and η fraction to i .

Intuitively, our proposed algorithm attempts to reserve some fraction of each buyer's budget for future purchases. However, in the case of the predicted buyer, it allows assignments even if the budget is close to saturation if the predicted buyer's bid is high enough.

Algorithm 1: Learning Augmented Algorithm for the Online Ad-auction Problem.

All primal and dual variables are initially set to 0.

We maintain two sets $N(i), M(i)$ for each buyer i for the purpose of analysis only.

for each new item j do

let i^* be the predicted buyer of item j , formally, $\text{pred}(j) = i^*$

if the prediction is not feasible **then** $i^* = 0$; // the fictitious buyer

$i \leftarrow \arg \max_{i'} \{b_{i'j} (1 - y_{i'})\}$; // weight bids with remaining budget

if $b_{ij} (1 - y_i) \leq 0$ **then** $i = 0$

$z_j \leftarrow \max\{0, b_{ij}(1 - y_i)\}$

if $b_{ij} < b_{i^*j}$ **then**

$x_{ij} \leftarrow \eta$ and $x_{i^*j} \leftarrow (1 - \eta)$

$N(i^*) \leftarrow N(i^*) \cup \{j\}$

else

$x_{ij} \leftarrow 1$; // includes the case when $\text{pred}(j)$ is infeasible

$M(i) = M(i) \cup \{j\}$

$y_i = y_i \left(1 + \frac{b_{ij}}{B_i}\right) + \frac{b_{ij}}{B_i} \cdot \frac{1}{C-1}$

Lemma 4 *During the execution of Algorithm 1, for every i the following always holds*

$$y_i \geq \frac{1}{C-1} \left(C^{\frac{\sum_{j \in M(i)} b_{ij}}{\eta B_i}} - 1 \right)$$

Proof We adapt the proof of [7] with a slight modification. We prove the dual inequality by induction on the number of processed items. Initially, when no items arrived yet, the inequality is trivially true. Assume that the inequality holds right before the arrival of an item j . The inequality remains unchanged for all buyers, except for buyer i , who maximizes $b_{ij} (1 - y_i)$. We denote with y_i the value before the update triggered by the arrival of item j and y'_i its value after the update. We have

$$\begin{aligned} y'_i &= y_i \left(1 + \frac{b_{ij}}{B_i}\right) + \frac{b_{ij}}{B_i} \cdot \frac{1}{C-1} \geq \frac{1}{C-1} \left(C^{\frac{\sum_{j' \in M(i) \setminus \{j\}} b_{ij'}}{\eta B_i}} - 1 \right) \cdot \left(1 + \frac{b_{ij}}{B_i}\right) + \frac{b_{ij}}{B_i} \cdot \frac{1}{C-1} \\ &= \frac{1}{C-1} \left(C^{\frac{\sum_{j' \in M(i) \setminus \{j\}} b_{ij'}}{\eta B_i}} \cdot \left(1 + \frac{b_{ij}}{B_i}\right) - 1 \right) \\ &\geq \frac{1}{C-1} \left(C^{\frac{\sum_{j' \in M(i) \setminus \{j\}} b_{ij'}}{\eta B_i}} \cdot C^{\frac{b_{ij}}{\eta B_i}} - 1 \right) = \frac{1}{C-1} \left(C^{\frac{\sum_{j \in M(i)} b_{ij}}{\eta B_i}} - 1 \right) \end{aligned}$$

The first inequality holds due to the induction hypothesis. The second inequality holds by the following sequence of transformations. For any y and z where $0 < y \leq z \leq 1$ we have

$$\frac{\ln(1+y)}{y} \geq \frac{\ln(1+z)}{z} \quad \Leftrightarrow \quad \ln(1+y) \geq \ln(1+z) \cdot \frac{y}{z} \quad \Leftrightarrow \quad 1+y \geq (1+z)^{y/z}$$

We apply the above transformation with $y = b_{ij}/B_i$ and $z = R_{\max}$ at the second inequality. By the definition of C we obtain

$$\left(1 + R_{\max}\right)^{\frac{1}{R_{\max}} \cdot \frac{b_{ij}}{B_i}} = C^{\frac{b_{ij}}{\eta B_i}}$$

We recall that $R_{\max} = \max_{i,j} \left\{ \frac{b_{ij}}{B_i} \right\}$. The induction step is complete, and the lemma holds. \square

Lemma 5 *The primal solution is feasible up to a factor of $(1 + R_{\max})$.*

Proof The first primal constraint requires $\sum_{j=1}^m b_{ij} x_{ij} \leq B_i$ for every i . By Lemma 4, we know that for every i the following holds.

$$y_i \geq \frac{1}{C-1} \left(C^{\frac{\sum_{j \in M(i)} b_{ij}}{\eta B_i}} - 1 \right)$$

Therefore, whenever $\sum_{j \in M(i)} b_{ij} \geq \eta B_i$, we have $y_i \geq 1$ and the algorithm stops allocating items to buyer i . The set $M(i)$ and the value of y_i are updated after the assignments. Therefore buyer i can receive at most one additional item fraction once its budget is already saturated. We obtain $\sum_{j \in M(i)} b_{ij} < \eta B_i + \max_j \{b_{ij}\}$ and the following formula.

$$\sum_{j=1}^m b_{ij} x_{ij} = \sum_{j \in M(i)} b_{ij} x_{ij} + \sum_{j \in N(i)} b_{ij} x_{ij} < B_i + \max_j \{b_{ij}\}$$

The inequality holds due to the feasibility of $N(i)$, the set of items assigned by the prediction to buyer i . We can bound the prediction assignments as follows.

$$\sum_{j \in N(i)} b_{ij} x_{ij} \leq (1 - \eta) \sum_{j \mid \text{pred}(j)=i} b_{ij} x_{ij} \leq (1 - \eta) B_i$$

Therefore, $\sum_{j=1}^m b_{ij} x_{ij} \leq B_i (1 + R_{\max})$, which means that the first primal constraint is feasible up to a factor of $(1 + R_{\max})$. The second primal constraint requires $\sum_i x_{ij} \leq 1$ to hold. During the allocations of Algorithm 1 the values of x_{ij} and x_{i^*j} do not exceed 1 by the design of the algorithm. The lemma follows. \square

Theorem 2 *Algorithm 1 is $(1 - \eta)$ -consistent and $\frac{1-1/C}{1+R_{\max}}$ -robust. The robustness tends to $1 - e^{-\eta}$ when R_{\max} tends to 0.*

Proof First, we establish robustness. Upon the arrival of item j , the increase in the primal is

$$\begin{cases} (1 - \eta) b_{i^*j} + \eta b_{ij} & \text{if } b_{ij} < b_{i^*j}, \\ b_{ij} & \text{if } b_{ij} \geq b_{i^*j} \end{cases}$$

which is always larger than or equal to b_{ij} . Besides, the increase in the dual is

$$B_i \Delta y_i + z_j = b_{ij} y_i + \frac{b_{ij}}{C-1} + b_{ij} (1 - y_i) = \left(1 + \frac{1}{C-1}\right) b_{ij} = \frac{C}{C-1} b_{ij}$$

Hence, by Lemma 5, the robustness is $\frac{C-1}{C} \cdot \frac{1}{1+R_{\max}}$.

Secondly, we straightforwardly establish consistency. Every time the prediction solution gets a profit of b_{i^*j} , the algorithm achieves a profit of at least $(1 - \eta) b_{i^*j}$. The theorem follows. \square

4 Experiments

In the following experiments, we evaluate the algorithms’ performance with the competitive ratio metric. We calculate the competitive ratio by dividing the objective value of the algorithm, $\text{ALGO}(I)$, with the objective value of the optimal fractional offline solution, $\text{OPT}(I)$. The competitive ratio is visible on the y -axis of the figures. The x -axis corresponds to the ratio with which the algorithms consider the prediction. We indicate with $\eta = 0$ no doubt and $\eta = 1$ complete doubt in the prediction. Therefore, the left-hand side of the figures corresponds to a prediction dominated algorithm, while the right-hand side is closer to a classical online algorithm. The different colors on the figures correspond to different prediction error rates. The lines represent the average over several executions, while the colored areas show the 95% confidence intervals.

4.1 Online Bounded Allocation

We present here four experiments. Table 1 below summarizes the configurations of the instances. Figure 3 displays the competitive ratio of the algorithm on different instances.

Predictions. We create predictions for the experimental instances by introducing perturbations to the optimal offline integral solution. The perturbation is skipped when $|S_j| = 1$. When $|S_j \setminus \{i^*\}| > 1$ we choose uniformly at random a remaining buyer to replace the optimal buyer.

Name	Type	Buyers	Items	$d = \max\{ S_j \}$	Budget range	Item price range
Instance 1	manual	5	5	5	100 – 100	100 – 100
Instance 2	random	100	1.000	5	10 – 100	0.1 – 8
Instance 3	random	100	10.000	3	10 – 1.000	1 – 10
Instance 4	random	80	80	40	10 – 100	10 – 100

Table 1: Properties of the experiment instances

Manual Instance. Instance 1 corresponds to one of the pathological inputs for the water-filling strategy of [7]. We created this instance manually to observe the behavior of Algorithm 2 when this strategy performs poorly. In Instance 1’s scenario, each buyer i is interested in each item j when $i \geq j$. The optimal solution is to sell each item j to buyer i , where $i = j$, while the water-filling strategy attempts to allocate each item equally.

Randomized Instances. Instances 2-4 are randomly generated based on their corresponding configuration visible on Table 1. We executed each instance 20 times. The lines on Figures 4.1-4.1 correspond to the average of these executions, and the colored areas correspond to the 95% confidence intervals. Instance 2 and Instance 3 mimic real-life instances, where the general expectation is to have a small bound d on the interested buyers. The average item price over the average budget value across the executions was 6.36% for Instance 2 and 0.98% for Instance 3. Finally, Instance 4 shows an example where the integral solution (and therefore the prediction) is not optimal. While Instance 2 and 3 have no integrality gap, Instance 4 has an average of 17.99% observed integrality gap over the 20 executions. Instance 4 has a large bound on the number of interested buyers, and the items’ prices vary greatly.

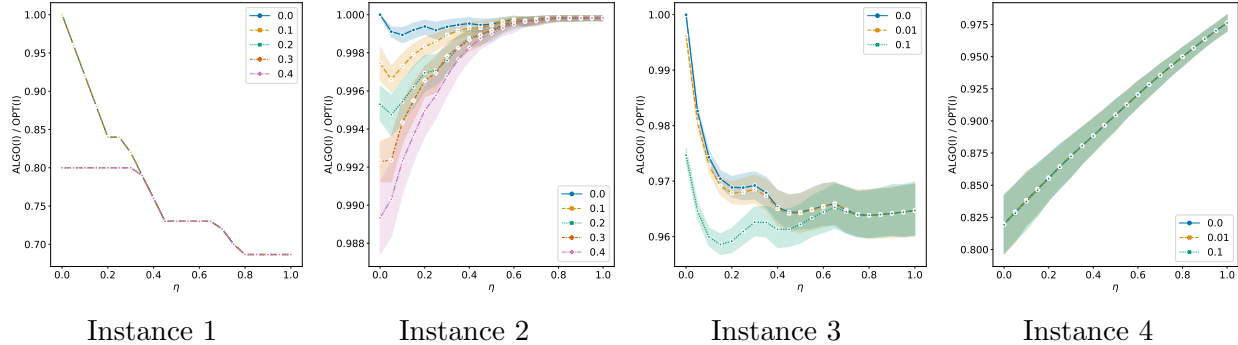


Figure 3: Observed competitive ratios

Evaluation. The experiments confirm the water-filling algorithm’s benefit from the prediction information. We showed that Algorithm 2 has improved performance on the pathological input of the water-filling strategy even with high prediction error rates. Besides, the algorithm demonstrated firm robustness against elevated prediction error rates in the second instance, where the algorithm performed close to optimality without predictions ($\eta = 1$). Furthermore, we observed improved performance on the third instance, which represents a close to real-life use case. In the fourth instance, we can remark the drawback of the predictions. When the integral solution of the linear program is not optimal or even far from optimal, the prediction can significantly misguide our algorithm.

4.2 Online Ad-auction

We present an experiment for the Online Ad-auction problem. Figure 4 displays the competitive ratio of the algorithm.

Predictions. The predictions rely on the optimal offline integral solution, which is a partial mapping from items to buyers. We perturbed the solution as suggested by [4]. We mapped every item with the probability of the given error rate during the execution to a uniformly chosen random buyer among the buyers who placed a non-zero bid on the item. However, the perturbation is only possible if the solution remains feasible.

Instance. The instance is a randomized one with 100 buyers and 10,000 items, adapting the model described in [18]. In this instance, for every item j , there are exactly 6 random buyers proposing a bid. The values of the bids follow a log-normal distribution with mean and deviation set to 0.5. By choosing the budget of the bidders, we can tune the hardness of the instance under the constraint that R_{\max} remains reasonably small. We set the budget to 0.1 fraction of the total bids, leading to a value of $R_{\max} \approx 0.1$.

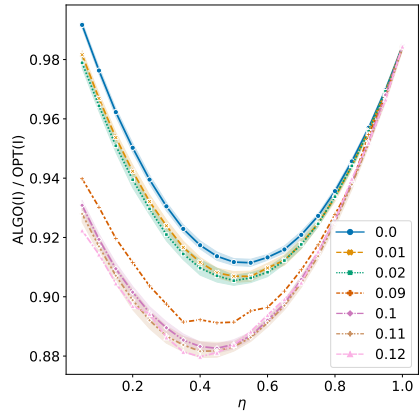


Figure 4: Observed competitive ratio in the given instance

Evaluation. We can observe the algorithm’s benefit of the prediction. We note that in the algorithm η can not be 0 since, in this case, the update of y_i would require division by 0. The closer η gets to 0, the closer the performance of the algorithm gets to 1. As expected, the performance degrades with increased prediction perturbation. This performance degradation for η around 0.5 is due to the simplistic mixture between the primal-dual and the predicted solution. In this case ($\eta \approx 0.5$), one does not know which direction, either the primal-dual or the predicted solution, to follow.

5 Conclusion

In the paper, we have presented algorithms that incorporate prediction information and achieve competitive consistency and robustness. An open question is to study whether the tradeoff between the consistency and the robustness is tight. Besides, it is interesting to consider the direction in which the confidence parameter varies over time due to the quality of predictions so far. More generally, one might design more complete systems that evolve through the interactions between the algorithms and the learning oracles. Specifically, learning oracles provide useful information (predictions) to improve the algorithm’s performance (as studied in this paper), and inversely, the algorithm would give feedbacks that enhance the learning predictions.

References

- [1] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proc. 22nd Symposium on Discrete Algorithms*, pages 1253–1264, 2011.
- [2] Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. Online Computation with Untrusted Advice. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151, pages 52:1–52:15, 2020.
- [3] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning*, pages 345–355, 2020.

- [4] Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. In *Proc. 34th Conference on Neural Information Processing Systems*, 2020.
- [5] Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *Acm Sigact News*, 39(1):80–87, 2008.
- [6] Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.
- [7] Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2009.
- [8] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264, 2007.
- [9] Hossein Esfandiari, Nitish Korula, and Vahab Mirrokni. Allocation with traffic spikes: Mixing adversarial and stochastic models. *ACM Trans. Econ. Comput.*, 6(3–4), 2018.
- [10] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. In *Proc. 61st Symposium on Foundations of Computer Science*, 2020.
- [11] Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *International Conference on Machine Learning*, pages 2319–2327, 2019.
- [12] Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *Proc. Conference on Learning Representations*, 2019.
- [13] Zhiyi Huang, Qiankun Zhang, and Yuhao Zhang. Adwords in a panorama. In *Proc. 61st Symposium on Foundations of Computer Science*, 2020.
- [14] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proc. 22nd ACM symposium on Theory of computing*, pages 352–358, 1990.
- [15] Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proc. Conference on Management of Data*, pages 489–504, 2018.
- [16] Ravi Kumar, Manish Purohit, and Zoya Svitkina. Improving online algorithms via ML predictions. In *Proc. 32nd Conference on Neural Information Processing Systems*, pages 9684–9693, 2018.
- [17] Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In *Proc. Symposium on Discrete Algorithms*, pages 1859–1877, 2020.
- [18] Thomas Lavastida, Benjamin Moseley, R Ravi, and Chenyang Xu. Learnable and instance-robust predictions for online matching, flows and load balancing. Technical Report arXiv:2011.11743, arXiv, 2020.
- [19] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning*, pages 3296–3305, 2018.
- [20] Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Online optimization with uncertain information. *ACM Trans. Algorithms*, 8(1), 2012.

- [21] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.
- [22] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):22–es, 2007.
- [23] Michael Mitzenmacher. A model for learned bloom filters, and optimizing by sandwiching. In *Proc. Conference on Neural Information Processing Systems*, pages 464–473, 2018.
- [24] Michael Mitzenmacher. Scheduling with predictions and the price of misprediction. In *Proc. 11th Innovations in Theoretical Computer Science Conference*, 2020.
- [25] Michael Mitzenmacher and Sergei Vassilvitskii. *Beyond the Worst-Case Analysis of Algorithms*, chapter Algorithms with Predictions. Cambridge University Press, 2020.
- [26] Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proc. Symposium on Discrete Algorithms*, pages 1834–1845, 2020.
- [27] Tim Roughgarden. Beyond worst-case analysis. *Communications of the ACM*, 62(3):88–96, 2019.
- [28] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2020.
- [29] Aaron Schild, Erik Vee, Manish Purohit, Ravi Kumar Ravikumar, and Zoya Svitkina. Semi-online bipartite matching. In *Proc. 10th Innovations in Theoretical Computer Science Conference*, 2019.
- [30] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

A Online Bounded Allocation

The pseudo code of the learning augmented algorithm for the Online Bounded Allocation problem:

Algorithm 2: Learning Augmented Algorithm for the Online Bounded Allocation Problem.

Initialize the variables and the level sets

for *each new item j* **do**

 Let i^* be the predicted buyer of item j , formally, $pred(j) = i^*$

while *there exists $i \in S_j$ who spends less than η fraction of its budget* **do**

 └ Continuously allocate fractions of j to buyers on the lowest levels // Stage 1

repeat

 └ Continuously allocate fractions of j to i^* // Stage 2

until *either:*

- $(1 - \eta)$ fraction has been allocated to i^*
- or j is completely sold
- or i^* exhausted its budget

while *there exists $i \in S_j$ with non-exhausted budget and j is not completely sold* **do**

 └ Continuously allocate fractions of j to buyers on the lowest levels // Stage 3
