Fair Online Knapsack with Value Density Predictions

Adam Lechowicz
alechowicz@cs.umass.edu
University of Massachusetts Amherst

Rik Sengupta
hajiesmaili@cs.umass.edu
University of Massachusetts Amherst

Bo Sun
hajiesmaili@cs.umass.edu
The Chinese University of Hong Kong

Shahin Kamali
hajiesmaili@cs.umass.edu
York University

Mohammad Hajiesmaili
hajiesmaili@cs.umass.edu
University of Massachusetts Amherst

ABSTRACT

The online knapsack problem is a classic online resource allocation problem. Its basic version studies how to pack online arriving items of different weights and values into a capacity-limited knapsack, so as to maximize the total value of the admitted items. Prior work has shown optimal deterministic and randomized algorithms for this problem – such algorithms have provably good competitive ratios, but they may treat individual items unfairly or inequitably in several different ways. In this work, we identify a natural notion of *arrival time fairness* previously studied for similar online problems and motivate its applicability to resource allocation. We extend this notion of fairness to the context of online knapsack, and show that existing algorithms perform poorly under this metric.

We propose a parameterized deterministic algorithm where the parameter precisely captures the Pareto-optimal trade-off between fairness and competitiveness. We further show that predictions approximating the lowest value density of an item accepted by an optimal solution can improve both fairness and competitiveness. In particular, we derive an algorithm which leverages such predictions for better (competitive) performance, while simultaneously providing better fairness guarantees when the predictions are trustworthy. Our work shows that even relatively simple predictions can be utilized remarkably effectively to simultaneously improve the fairness and performance of online algorithms.

1 INTRODUCTION

Achieving basic benchmarks of fairness and equity in algorithmic approaches to classical problems has become the focus of much recent research in computer science. To illustrate the importance of these considerations in the context of the widely-studied *online knapsack problem*, it is perhaps best to start with an example.

Example 1.1. Consider a cloud computing resource accepting heterogeneous jobs online from clients sequentially. Each job includes a bid that the client is willing to pay, and an amount of resources required to execute it. The cloud computing resource is limited; there are not enough resources to service all of the incoming requests. We define the *quality* of a job as the ratio of the bid price paid by the client to the quantity of resources required for it.

How do we algorithmically solve the problem posed in Example 1.1? Note that the limit on the resource implies that the problem of accepting and rejecting items reduces precisely to the online knapsack problem. If we only cared about the overall quality of accepted jobs, we would intuitively be solving the unconstrained online knapsack problem. However, at the same time, it might be desirable for an algorithm to apply the same *quality criteria* to each job that arrives. As we will see formally in Section 2, existing optimal algorithms for online knapsack *do not* fulfill this second requirement. In particular, although two jobs may have *a priori* identical quality, the optimal algorithm discriminates between them based on their *arrival time* in the online queue: a typical job, therefore, may have a much higher chance of being accepted if it happens to arrive earlier rather than later. Preventing these kinds of discriminating choices while still maintaining competitive standards of overall quality will form the focus of this work.

2 PRELIMINARIES & FAIRNESS

Problem Formulation. In the online knapsack problem (OKP), we have a knapsack (resource) with capacity B and items arriving online from an unknown sequence. We denote an instance $\mathcal{I}$ of OKP as a sequence of n items, where each item has a *value* v_j and a *weight* w_j. Formally, $\mathcal{I} = \left[(v_j, w_j)\right]_{j=1}^{n}$. In general, the arrival sequence of the items in $\mathcal{I}$ will correspond to some permutation $\pi \in \mathcal{S}_n$. For simplicity, we will keep π implicit, and assume that the jth item to arrive has value v_j and weight w_j. The objective in OKP is to accept items into the knapsack maximizing the sum of values, while not violating the weight capacity limit of B. As is standard in the online setting, at each time step, the algorithm is presented with an item, and must immediately and irrevocably decide whether to accept it into the knapsack or reject it.

OKP has been extensively studied under the *competitive analysis* framework, where the goal is to design an online algorithm that maintains a small *competitive ratio* [2], i.e., performs nearly as well as the offline optimal solution.

Assumptions and additional notation. We make no assumptions on the underlying distribution of items other than the assumption that each online item's *value density* $\{v_j/w_j\}_{j\in[n]}$ has bounded support, i.e., $(v_j/w_j) \in [L, U]$ $\forall j \in [n]$. We assume L and U are known. These are standard assumptions for many online problems, including OKP, one-way trading, and online search; without them the competitive ratio of any online algorithm is unbounded [3, 4]. For the rest of this paper, we will assume WLOG that $B = 1$ (we can scale everything down by a factor of B otherwise).

Existing results. Prior work on OKP has resulted in an optimal deterministic algorithm for the problem setting described above, shown by Zhou et al. [4]. This seminal algorithm made use of a framework known as online threshold-based algorithm design. The algorithm in [4] (henceforth referred to as the ZCL algorithm) is a deterministic algorithm which achieves a competitive ratio of $\ln(U/L) + 1$; [4] also shows that this is the optimal competitive ratio for any deterministic *or* randomized OKP procedure. The algorithm admits items based on a "growing" threshold function $\Phi(z) = (Ue/L)^z (L/e)$, where $z \in [0,1]$ is the current *knapsack utilization* (i.e. fraction that is filled). The jth item in the online sequence is admitted iff $v_j/w_j \geq \Phi(z_j)$, where z_j is the knapsack's utilization at the time of the item's arrival.

Prediction model. Consider receiving a single prediction $\hat{v}$ which represents a *critical value density* for the upcoming OKP sequence. In the rest of this paper we will refer to a naïve algorithm ORACLE, which simply admits any items with value density $\geq \hat{v}$. If this prediction is accurate, we say that it represents the *lowest value density* of any item which would be accepted by OPT, and we would expect ORACLE to obtain a small competitive ratio against OPT.

3 MAIN RESULTS

Recall that throughout this section, we assume $B = 1$, i.e., the knapsack has unit capacity.

3.1 Impossibility results for fairness

The introduction had an example of a specific type of *time fairness* constraint that was infringed. This concept was explored in the context of prophet inequalities in [1]. In Example 1.1, it is reasonable to ask that the probability of an item's admission into the knapsack should depend solely on its value density x, and not on its arrival time j. We begin by generalizing the definition of Time-Independent Fairness proposed in [1] to OKP.

Definition 3.1 (Time-Independent Fairness (TIF) for OKP). *An OKP algorithm ALG is said to satisfy TIF if there exists a function $p : [L, U] \to [0,1]$ such that:*

$$\Pr\left[\text{ALG accepts } j\text{th item in } \mathcal{I} \mid v_j/w_j = x\right] = p(x),$$
$$\forall \mathcal{I} \in \Omega, j \in [|\mathcal{I}|], x \in [L, U],$$

where Ω is the set of all feasible instances.

Stated differently, the probability of admitting an item of value density x depends only on x, and not on its arrival time. We start by noting that the ZCL algorithm does not satisfy TIF.

Observation 3.2. *The ZCL algorithm [4] is not TIF, because the threshold for an item's admittance changes* over time.

In fact, we show that Definition 3.1 is *too restrictive* for OKP.

Theorem 3.3. *Other than the trivial OKP algorithm which rejects all items, there is no algorithm (deterministic or randomized) for OKP which guarantees TIF.*

Motivated by these results, in Definition 3.4, we present a slightly revised notion, which relaxes this constraint and narrows the scope of fairness to consider items which arrive while the knapsack's utilization is in some subinterval of the knapsack's capacity.

Definition 3.4 (α-Conditional Time-Independent Fairness (α-CTIF) for OKP). *For $\alpha \in [0,1]$, an OKP algorithm ALG is said to satisfy α-CTIF if there exists a subinterval $\mathcal{A} = [a,b] \subseteq [0,1]$ where $b - a = \alpha$, and a function $p : [L,U] \to [0,1]$ such that:*

$$\Pr\left[\text{ALG accepts } j\text{th item in } \mathcal{I} \mid \left(\frac{v_j}{w_j} = x\right) \wedge (z_j + w_j \in \mathcal{A})\right] = p(x),$$
$$\forall \mathcal{I} \in \Omega, j \in [|\mathcal{I}|], x \in [L, U].$$

In particular, if $\alpha = 1$, then $\mathcal{A} = [0,1]$, and any item that arrives while the knapsack still has the capacity to admit it is considered.

3.2 Algorithmic guarantees and trade-offs

Using Definition 3.4, in this section we present algorithms which satisfy CTIF constraints while remaining competitive and leveraging predictions for better performance. We start with a result that captures the essence of the trade-offs inherent to this problem.

Theorem 3.5. *Any constant threshold-based algorithm for OKP satisfies 1-CTIF. Furthermore, any constant threshold-based deterministic algorithm for OKP cannot be better than (U/L)-competitive.*

We can now extend these results to general values of α.

Extended Constant Threshold (ECT). We define a threshold function $\Phi^\alpha(z)$ on the interval $z \in [0,1]$, where z_j is the knapsack utilization when the jth item arrives, and $\alpha \in [1/(\ln(U/L)+1), 1]$ is the *fairness parameter.* Φ^α is defined as follows:

$$\Psi^\alpha(z) = \begin{cases} L & z \in [0,\alpha], \\ Ue^{\beta(z-1)} & z \in (\alpha, 1], \end{cases} \tag{1}$$

where $\beta = \frac{W\left(\frac{U(1-\alpha)}{L\alpha}\right)}{1-\alpha}$. Let us call this algorithm . The following result shows the achieved trade-off between fairness and competitiveness in .

Theorem 3.6. ECT$[\alpha]$ *satisfies α-CTIF. Furthermore, for any instance $\mathcal{I} \in \Omega$, we have*

$$\text{OPT}(\mathcal{I}) \leq \text{ECT}[\alpha](\mathcal{I}) \cdot \frac{U[\ln(U/L)+1]}{L\alpha[\ln(U/L)+1] + (U-L)(1-\ell)}.$$

Thus, ECT$[\alpha]$ *is $\frac{U[\ln(U/L)+1]}{L\alpha[\ln(U/L)+1]+(U-L)(1-\ell)}$-competitive.* ECT$[\alpha]$*, in fact,* ***exactly*** *achieves the Pareto-optimal competitiveness trade-off.*

We also explore how simple predictions improve both competitiveness and fairness. We propose LA-ECT, which integrates such predictions.

Prediction Model. Consider an offline approximation algorithm APX for OKP, which sorts items by non-increasing value density and packs them in this order. Let $x \in [L,U]$ denote the smallest value density of any packed item, and V is the total value obtained by APX. Then, if the total value of items with value density x in the knapsack is $\geq V/2$, define $d^\star := x$. Otherwise, define $d^\star := x^+$, where x^+ is the next highest value density in $\mathcal{I}$. We assume that our algorithm receives a single prediction $\hat{d} \in [L,U]$ for each instance, where the prediction is perfect if $\hat{d} = d^\star$.

Learning-Augmented Extended Constant Threshold (LA-ECT). Fix a *trust parameter* $\gamma \in [0,1]$. We define the threshold function $\Psi^{\gamma,\hat{d}}(z)$:

$$\Psi^{\gamma,\hat{d}}(z) = \begin{cases} (Ue/L)^{\frac{z}{1-\gamma}} (L/e) & z \in [0,\kappa], \\ \hat{d} & z \in (\kappa, \kappa+\gamma), \\ (Ue/L)^{\frac{z-\gamma}{1-\gamma}} (L/e) & z \in [\kappa+\gamma, 1], \end{cases} \tag{2}$$

where κ is the point where $(Ue/L)^{(z/1-\gamma)}(L/e) = \hat{d}$. Call the resulting threshold algorithm LA-ECT$[\gamma]$. The following theorem characterizes the fairness as well as the trade-off between consistency and robustness for this algorithm.

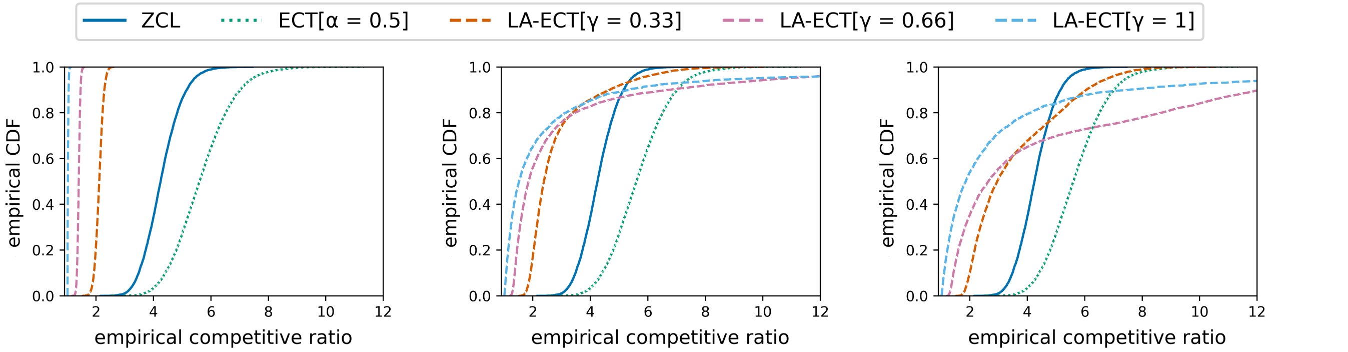

Figure 1(a). $\sigma = 0$ Figure 1(b) $\sigma = 0.5$ Figure 1(c) $\sigma = 1$

Theorem 3.7. LA-ECT$[\gamma]$ *satisfies* γ*-CTIF. Also, for any* $\mathcal{I} \in \Omega$,

- *For any accurate prediction* $\hat{v} \in [L, U]$, *we have* ORACLE$(\mathcal{I}) \leq$ LA-ECT$[\gamma](\mathcal{I}) \cdot \frac{\varrho+2}{\gamma}$.
- *For any prediction* $\hat{v} \in [L, U]$, *we have*
  OPT$(\mathcal{I}) \leq$ LA-ECT$[\gamma](\mathcal{I}) \cdot (1/1-\gamma)\ln(U/L) + 1$.

Thus, LA-ECT$[\gamma]$ *is* $\left(\frac{1}{1-\gamma}\ln(U/L)+1\right)$*-robust. For most instances,* $\varrho = O(1)$*, and so* LA-ECT$[\gamma]$ *is* $O(1/\gamma)$*-consistent.*

Experiments. We extract real-world item sequences from Google cluster traces and compare the empirical competitive ratios of the ZCL algorithm, ECT, and LA-ECT. We test three different regimes for **prediction error**, parameterized by σ; when $\sigma = 0$, predictions are perfect.

4 CONCLUSION

We showed impossibility results for the online knapsack problem under arrival time-based fairness constraints, which led to a natural definition of time-independent fairness. We then described a deterministic algorithm achieving the optimal trade-off between fairness and competitiveness. We parameterized this trade-off in a more fine-grained analysis. Finally, we showed that even a simple prediction model can simultaneously improve fairness and performance.

REFERENCES

[1] M. Arsenis and R. Kleinberg. 2022. Individual Fairness in Prophet Inequalities. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC '22)*. Association for Computing Machinery, New York, NY, USA.

[2] A. Borodin, N. Linial, and M. E. Saks. 1992. An Optimal On-Line Algorithm for Metrical Task System. *J. ACM* 39, 4 (Oct 1992).

[3] A. Marchetti-Spaccamela and C. Vercellis. 1995. Stochastic on-line knapsack problems. *Mathematical Programming* 68 (Jan. 1995).

[4] Y. Zhou, D. Chakrabarty, and R. Lukose. 2008. Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. Association for Computing Machinery, New York, NY, USA.