Trade-off Analysis in Learning-augmented Algorithms with Societal Design Criteria

Mohammad H. Hajiesmaili

University of Massachusetts Amherst
hajiesmaili@cs.umass.edu

ABSTRACT

Traditionally, computer systems are designed to optimize classic notions of performance such as flow completion time, cost, etc. The system performance is then typically evaluated by characterizing theoretical bounds in worst-case settings over a single performance metric. In the next generation of computer systems, societal design criteria, such as carbon awareness and fairness, becomes a first-class design goal. However, the classic performance metrics may conflict with societal criteria. Foundational understanding and performance evaluations of systems with these inherent trade-offs lead to novel research questions that could be considered new educational components for performance analysis courses. The classic techniques, e.g., worst-case analysis, for systems with conflicting objectives may lead to the impossibility of results. However, a foundational understanding of the impossibility of results calls for new techniques and tools. In traditional performance evaluation, to understand the foundational limits, typically, it is sufficient to derive lower-bound arguments in worst-case settings. In the new era of system design, lower bounds are inherently about trade-offs between different objectives. Characterizing these trade-offs in settings with multiple design criteria is closer to the notion of Pareto-optimality, which is drastically different from classic lower bounds. With the impossibility of results using classic paradigms, one possible direction is to (re)design systems following the emerging direction of learning-augmented algorithms. With this approach, it might be possible to remove/mitigate the foundational conflict between classic vs. societal metrics using the right predictions. However, the performance evaluation of learning-augmented algorithms calls for a new set of technical questions, which we highlight in this paper.

1 Introduction

The traditional approach for algorithm design targets classic objective functions that model some notions of *efficiency*, such as performance or cost. The performance of the proposed algorithms is then typically analyzed by characterizing theoretical bounds, e.g., approximation ratio, competitive ratio, regret, etc., in worst-case settings. With the wide deployment of algorithmic ideas in society, it is essential to systematically add *societal* criteria, such as fairness, carbon awareness, safety, privacy, etc., into the system design.

However, the classic efficiency metrics may conflict with societal criteria in several scenarios. We outline two examples of such conflicts in the context of fairness and carbon awareness in computing systems.

Example 1: The trade-off between fairness and competitiveness in the online knapsack problem. The online knapsack problem (OKP) [1, 2, 3] (formally introduced in Section 2.2) is well-studied in the literature on online algorithms. In its basic version of OKP, one provider allocates a limited resource (i.e., the knapsack's *capacity*) to users arriving sequentially to maximize the total value of admitted users. In OKP, as in many other online decision problems, there is a trade-off between *efficiency*, i.e., maximizing the value of the packed items, and *fairness*, i.e., ensuring equitable "treatment" for different items across some desirable criteria [4]. To illustrate the importance of these considerations in the context of OKP, it is perhaps best to start with an example.

Consider a cloud computing resource accepting heterogeneous jobs online from clients sequentially. Each job includes a bid the user is willing to pay and the resource requirement. The cloud resource is limited – there are not enough resources to service all incoming requests. Consider the *quality* of a job as the ratio of the bid price paid by the client to the quantity of resources required for it. 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 cared only about the overall quality of accepted jobs, we would intuitively be solving the unconstrained online knapsack problem. However, simultaneously, it might be desirable for an algorithm to apply a fair *quality criteria* to each job that arrives. But, adding fair criteria will come at the expense of degrading competitiveness. In Section 4, we formally explore the trade-off between fairness and competitiveness in this context.

Example 2: The trade-off between energy efficiency and carbon efficiency. While the first example was concretely about a specific problem, in the second example, we focus on a more high-level trade-off concept. Motivating by the goal of reducing the carbon footprint of computing systems, recently, there has been attention to elevate the importance of *carbon efficiency*—the ability to do more work when and where low-carbon and clean energy is available—relative to *energy efficiency*—the ability to do the most work for the least amount of energy. While optimizing energy efficiency has been a focus of research in sustainable computing for decades, optimizing carbon efficiency is new and largely under-explored. Technically speaking, optimizing carbon

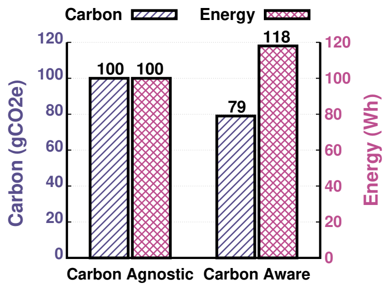

Figure 1: ***Trade-off between energy- vs. carbon-efficiency.***

efficiency is more closely related to the concept of energy flexibility—the degree to which a workload can be shifted temporally or spatially—than energy efficiency. The relationship between energy flexibility and energy efficiency may conflict, i.e., increasing energy flexibility can decrease energy efficiency [5]. A representative example in Figure 1 demonstrates that by while a carbon-aware workload scheduling decreases the carbon footprint by 11%, it increases the energy consumption by 18% (more details on setup and traces in [5]). As another example, data centers are most energy-efficient at high utilization, so leveraging their energy flexibility to reduce carbon emissions by periodically reducing their utilization and power usage makes them less energy-efficient. Another example on theoretical understanding of the trade-off between carbon and energy efficiency is studied in [6].

More broadly, the addition of societal design objectives could lead to multiple other conflicts, such as between safety and regret in online learning [7, 8, 9], and other notions of fairness and learning performance [10, 11].

Learning-augmented algorithms. Besides the emerging topic of societal algorithms design, recently, there has been extensive work on the systematic integration of algorithm design with advice from machine learning. The main motivation is that the classic algorithms (particularly online algorithms, which are the focus of examples in this paper) designed purely with guarantees of the worst-case performance tend to ignore predictions outright. Thus they often have poor performance in common average-case scenarios. In practice, however, for most application scenarios, abundant historical data could be leveraged by machine learning (ML) tools for generating some predictions of the unknown future input, e.g., item values in the online knapsack problem. Then, the possibility of leveraging ML predictions in algorithmic design has led to the recent development of learning-augmented algorithms [12, 13], where the goal is to leverage predictions to improve the performance when predictions are accurate and preserve the robust worst-case guarantees when facing erroneous ML predictions. This high-level idea has led researchers to revisit a wide range of online problems, including but not limited to caching [12], rent-or-buy problems [13, 14, 15], facility location [16, 17], secretary matching [18], metrical task systems [19], bin packing [20], and beyond.

Consistency-robustness Trade-off in learning-augmented algorithms. In the framework of learning-augmented algorithms, there is a natural trade-off between consistency and robustness [12], where consistency represents the competitive ratio (formally defined in (1)) when the prediction is accurate, and robustness is the competitive ratio regardless of the prediction error. The ultimate design goal is to develop an algorithm that can achieve the Pareto-optimal trade-off between consistency and robustness, i.e., no other learning-augmented algorithms can simultaneously achieve better consistency and robustness than the proposed algorithm. A few examples regarding the Pareto-optimality are for ski-rental problem [14], online conversion problem [21], and online matching problem [22].

1.1 Paper Organization

The goal of the remaining sections of this paper is to provide more concrete examples of the trade-off analysis in both societal algorithm design and learning-augmented algorithms to further motivate the need for different notions of trade-off analysis in the broader context of performance evaluation courses. Towards this, in Section 2, we provide a brief background of online algorithms and introduce the online knapsack problem as the running example used in the rest of the paper. Second, in Section 3, we present the consistency-robustness trade-off analysis for learning-augmented online knapsack algorithms. Third, in Section 4, we provide a trade-off analysis between fairness and competitiveness for a notion of time fairness in the knapsack problems. We provide concluding remarks in Section 5.

2 Background

In this section, we provide a brief background on competitive online algorithms and then introduce the classic online knapsack problem.

2.1 Online Algorithms

Decision-making under uncertainty is one of the most challenging issues that real-world computational problems face. In the context of sustainable computing problems, for example, questions like "*will it be enough solar in the next few hours to run the workload later?*" to "*which location will have the lowest carbon intensity to move the workload?*", cannot be answered reliably due to unpredictability of the inputs. Competitive design [23, 24] is a remarkably successful framework for tackling decision making under uncertainty scenarios by developing worst-case optimized algorithms. This framework assumes no stochastic modeling of the input, and its ultimate goal is to devise algorithms with the best possible competitive ratio. Competitive ratio refers to the maximum ratio of the cost incurred by an online algorithm A and the optimal cost incurred by solving the problem in an offline manner under any feasible input instance ω, i.e.,

$$\texttt{CR}(A) = \max_{\omega\in\Omega} \frac{\texttt{cost}(A(\omega))}{\texttt{cost}(\texttt{opt}(\omega))}, \qquad (1)$$

where Ω is the set of all feasible instances, and $\texttt{cost}(A(\omega))$ and $\texttt{cost}(\texttt{opt}(\omega))$ are the cost of A and the offline optimal cost under input ω. This framework has been successfully applied to numerous systems and networking applications such as TCP acknowledgement [25], renting cloud servers [26], dynamic capacity provisioning of data centers [27, 28], energy optimization problems [29, 30, 15], scheduling [31, 32, 33, 34], to name a few.

In the next section, we introduce the online knapsack problem and review the existing competitive algorithms with optimal competitive ratios for this problem.

2.2 The Online Knapsack Problem

The online knapsack problem (OKP) is a classic problem that has been studies extensively in the context of competitive online algorithms. In the basic version of OKP, the goal is to pack items that are arriving online into a knapsack with unit capacity such that the aggregate value of admitted items is maximized. In each round, item $i \in [n] = \{1, \dots, n\}$, with value v_i and weight w_i, arrives, and an online algorithm must decide whether to admit or reject i with the objective of maximizing the total value of selected items while respecting the capacity. More formally, given items' values and weights $\{v_i, w_i\}_{i\in[n]}$, OKP can be formulated as

$$
\begin{aligned}
\text{[OKP]} \quad \max \quad & \textstyle\sum_{i\in[n]} v_i x_i, \\
\text{s.t.,} \quad & \textstyle\sum_{i\in[n]} w_i x_i \le 1, \\
\text{vars.,} \quad & x_i \in \{0, 1\}, \quad i \in [n],
\end{aligned}
$$

where the binary variable $x_i = 1$ denotes the admission of item i and $x_i = 0$ represents a decline. In an online setting, the admission decision x_i for item i must be made only based on the information of current and past items. It is straightforward to show that without any assumptions on the item value and weights, it is impossible to design online algorithms with a bounded competitive ratio for the above formulation of OKP [1]. Hence, in the literature [26, 1, 2, 3], the following two standard assumptions are made to design online algorithms with bounded competitive ratios.

ASSUMPTION 1. *The weight of each individual item is much smaller than the unit capacity of the knapsack, i.e.,* $w_i \ll 1, \forall i \in [n]$.

ASSUMPTION 2. *The value-to-weight ratio (or value density) of each item is lower and upper bounded between* L *and* U*, i.e.,* $L \le v_i/w_i \le U, \forall i \in [n]$.

Assumption 1 naturally holds in large-scale systems where the capacity of the entire system is way larger than individual requests. Assumption 2 is to eliminate the potential for rare items that have extremely high or low-value densities and again is reasonable from practical perspective. This version of OKP has been used in numerous applications, including online cloud resource allocation [35, 36], budget-constrained bidding in keyword auction [1], online routing [37], and electric vehicle charging scheduling [38, 34, 39].

Prior work on OKP has resulted in an optimal deterministic algorithm for the problem described above, shown by [1] in a seminal work using the framework of online threshold-based algorithms (OTA). In OTA, a carefully designed *threshold function* is used to facilitate the decisions made at each time step. This threshold is specifically designed so that greedily accepting inputs whose values meet or exceed the threshold at each step provides a competitive guarantee. This algorithmic framework has seen success in the related online search and one-way trading problems [38, 40, 41] as well as OKP [1, 2, 3].

The ZCL algorithm: Prior literature [1] proposed a deterministic threshold-based algorithm that achieves a competitive ratio of $\ln(U/L) + 1$. The authors also show that this is the optimal competitive ratio for any deterministic or randomized algorithm. We henceforth refer to this algorithm as the ZCL algorithm. In the ZCL algorithm, items are admitted based on the monotonically increasing threshold function $\Phi(z) = (Ue/L)^z (L/e)$, where $z \in [0, 1]$ is the current utilization. The jth item in the sequence is accepted iff it satisfies $v_j/w_j \ge \Phi(z_j)$, where z_j is the utilization at the time of the item's arrival. This algorithm is optimally competitive [1, Theorems. 3.2, 3.3].

In what follows, we provide two examples of trade-off analysis for learning-augmented algorithms (in Section 3) and societal algorithms (in Section 4) for the online knapsack problem.

3 Learning-augmented Algorithms for the Online Knapsack Problem

In this section, we overview a recent consistency-robustness trade-off results for the 1-max search problem [42], which is a simplified version of the online knapsack problem. A 1-max search problem considers how to convert one asset (e.g., dollars) to another (e.g., yens) over a trading period $[N] := \{1, \dots, N\}$. At the beginning of step $n \in [N]$, an exchange rate (or price), v_n, is announced, and a decision maker must immediately determine the amount of dollars, x_n, to convert and obtains $v_n x_n$ yens. The trading horizon N is unknown to the decision maker, and if there are any remaining dollars after $N - 1$ trading steps, all of them will be compulsorily converted to yens at the last price v_N. The 1-max search problem is a special case of OKP in the sense of setting item sizes equal to the capacity of the knapsack and the goal of picking the top most valuable item. If the asset is allowed to convert fraction-by-fraction over multiple transactions, the decision $x_n \in [0, 1]$ is a continuous variable, and this fractional version is referred to as *one-way trading* [43]. Similar to that of OKP, we assume the prices $\{v_n\}_{n\in[N]}$ are bounded, i.e., $v_n \in [L, U], \forall n \in [N]$, where L and U are known parameters, and define $\theta = U/L$ as the price fluctuation.

The optimal algorithm for 1-max-search. There is a simple threshold-based algorithm, which determines a threshold function as a constant $\Phi = \sqrt{UL}$, where Φ is also called a reservation price. Then the algorithm selects the first price that is at least Φ. In [43], it has been shown that this algorithm achieves the optimal competitive ratio $\sqrt{\theta}$.

3.1 1-max search with prediction

In this section, we review an existing algorithm with a Pareto-optimal trade-off between consistency-robustness for the 1-max search problem. We refer to [21] for the full explanation of the results. First, we assume that a prediction of the maximum price P is given to the learning-augmented algorithm. The goal is to design the reservation price Φ_P given a prediction P. We denote η as the consistency and γ as the robustness. set $\eta := \eta(\lambda)$ and $\gamma := \gamma(\lambda)$ as

$$\gamma(\lambda) = [\sqrt{(1-\lambda)^2 + 4\lambda\theta} - (1-\lambda)]/(2\lambda), \text{ and } \eta(\lambda) = \theta/\gamma(\lambda), \tag{3}$$

where $\lambda \in [0, 1]$ is the robustness parameter. In other words, parameter λ determines the level of trust on prediction P, where $\lambda = 0$ means full trust; and $\lambda = 1$ means no trust at all. and η and γ are predetermined parameters for designing Φ_P that represent the consistency and robustness that we target to achieve. In particular, η and γ are designed as the solution of

$$\eta(\lambda) = \theta/\gamma(\lambda), \text{ and } \eta(\lambda) = \lambda\gamma(\lambda) + 1 - \lambda. \quad (4)$$

The first equation is the desired trade-off between robustness and consistency and thus represents a Pareto-optimal trade-off. The second equation sets η as a linear combination of 1 and γ. In this way, as λ increases from 0 to 1 , η increases from the best possible ratio 1 to the optimal competitive ratio $\sqrt{\theta}$, and γ decreases from the worst possible ratio θ to $\sqrt{\theta}$. Taking η and γ as inputs, we design the reservation price Φ_P as follows:

$$\text{when } P \in [L, L\eta), \ \Phi_P = L\eta; \quad (5a)$$
$$\text{when } P \in [L\eta, L\gamma), \ \Phi_P = \lambda L\gamma + (1-\lambda)P/\eta; \quad (5b)$$
$$\text{when } P \in [L\gamma, U], \ \Phi_P = L\gamma. \quad (5c)$$

The following theorem provides robustness and consistency bounds for this algorithm.

THEOREM 1. *Given $\lambda \in [0,1]$,* `OTA` *with the reservation price in Equation* (5) *for* 1*-max-search is $\gamma(\lambda)$-robust and $\eta(\lambda)$-consistent, where $\gamma(\lambda)$ and $\eta(\lambda)$ are given in Equation* (3).

THEOREM 2. *Any γ-robust learning-augmented online algorithms for 1-max-search must have consistency $\eta \geq \theta/\gamma$. Thus, the algorithm proposed with the reservation price* (5) *is Pareto-optimal.*

For additional insights on the algorithm design and Pareto-optimal trade-off, we refer to [21]. Putting together the results in the above two theorems, we conclude that the consistency-robustness trade-off of the above algorithm is Pareto-optimal. As a concluding remark for this section, we further note that the Pareto-optimal trade-off analysis in the context of learning-augmented algorithms is an emerging topic, and to the best of our knowledge, finding a Pareto-optimal learning-augmented algorithm for the general online knapsack is still an open problem.

It is worth noting that the main purpose of presenting the Pareto-optimality trade-off results was to highlight the contrast with respect to classic competitive analysis where the optimality of results reduces to only showing a lower bound on a single criterion instead of two (or multiple) criteria as in learning-augmented algorithm design.

4 Trade-offs in Social Algorithm Design

To show the potential trade-offs between societal vs. classic design criteria in performance analysis of the algorithms, we demonstrate the results in [4] as a running example. The purpose of this example is just to provide an example of a trade-off between fairness (as a societal criterion) and competitiveness (as an efficiency metric). The high-level concept could be applicable to other societal concerns such as carbon awareness as we mentioned in the introduction.

4.1 Fairness in Online Knapsack Problems

In this section, we briefly explore this trade-off in the context of the online knapsack problem. We refer to [4] for a comprehensive statement of the results.

4.1.1 Trade-off Results

Fairness definition. The example in the introduction presented a specific type of time fairness that was explored in the context of similar problems such as prophet inequalities [44]. 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 [44] to OKP. Motivated by these results, in Definition 3, we present a slightly revised notion, which relaxes this constraint and narrows the scope of fairness to consider items that arrive while the knapsack's utilization is in some subinterval of the knapsack's capacity. In the following, we formally define the notion of α-Conditional Time-Independent Fairness (α-CTIF) for OKP.

DEFINITION 3. *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:*

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

In particular, if $\alpha = 1$, then $\mathcal{A} = [0,1]$, and any item that arrives while the knapsack still can admit it is considered. Using Definition 3, in this section we present algorithms that 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 4. *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:

$$\Psi^\alpha(z) = \begin{cases} L & z \in [0,\alpha], \\ Ue^{\beta(z-1)} & z \in (\alpha, 1], \end{cases} \quad (6)$$

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

THEOREM 5. 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.*

While the above results provide a tight trade-off between fairness and competitiveness, in the following, we show how to improve the trade-off by using simple predictions.

4.1.2 Learning-augmented Design Helps

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 $\Psi^{\gamma,\hat{d}}(z)$:

$$\Psi^{\gamma,\hat{d}}(z) = \begin{cases} (Ue/L)^{\frac{z}{1-\gamma}}\,(L/e) & z \in [0, \kappa], \\ \hat{d} & z \in (\kappa, \kappa+\gamma), \\ (Ue/L)^{\frac{z-\gamma}{1-\gamma}}\,(L/e) & z \in [\kappa+\gamma, 1], \end{cases} \tag{7}$$

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

THEOREM 6. $\mathsf{LA\text{-}ECT}[\gamma]$ *satisfies γ-CTIF. Also, for any $\mathcal{I} \in \Omega$,*

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

$$\mathsf{OPT}(\mathcal{I}) \leq \mathsf{LA\text{-}ECT}[\gamma](\mathcal{I}) \cdot (1/1-\gamma)\ln(U/L) + 1.$$

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

The proposed learning-augmented algorithm substantially improves the performance in practice, as shown in the experiments in [4]. Hence, interestingly the learning-augmented algorithm design paradigm is an appropriate tool to improve the conflicting trade-offs between classic and societal design criteria.

5 Concluding Remarks

In this paper, we highlighted two notions of trade-off analysis in the context of (1) learning-augmented algorithms design, where the trade-off is between consistency and robustness; and (2) algorithms design with societal criteria, e.g., fairness and carbon awareness, where the trade-off is between classic performance notions, e.g., competitive ratio, and societal metrics. Interestingly, leveraging learning-augmented design could be considered as a potential tool to improve the trade-offs in societal algorithm design. Lastly, these trade-offs are inherent to both emerging topics and could be considered as new teaching elements to classic performance evaluations and algorithm design and analysis courses.

Acknowledgment

The author acknowledges that the technical parts of this paper are from collaborative research projects with Shahin Kamali, Adam Lechowicz, Russell Lee, Rik Sengupta, Bo Sun, and Adam Wierman.

6 References

[1] Y. Zhou, D. Chakrabarty, and R. Lukose, "Budget constrained bidding in keyword auctions and online knapsack problems," in *Proc. of WINE*, pp. 566–576, 2008.

[2] B. Sun, L. Yang, M. Hajiesmaili, A. Wierman, J. C. Lui, D. Towsley, and D. H. Tsang, "The online knapsack problem with departures," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, pp. 1–32, 2022.

[3] L. Yang, A. Zeynali, M. H. Hajiesmaili, R. K. Sitaraman, and D. Towsley, "Competitive algorithms for online multidimensional knapsack problems," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 3, pp. 1–30, 2021.

[4] A. Lechowicz, R. Sengupta, B. Sun, S. Kamali, and M. Hajiesmaili, "Time fairness in online knapsack problems," *arXiv preprint arXiv:2305.13293*, 2023.

[5] W. A. Hanafy, R. Bostandoost, N. Bashir, D. Irwin, M. Hajiesmaili, and P. Shenoy, "The war of the efficiencies: Understanding the tension between carbon and energy optimization," in *The 2nd Workshop on Sustainable Computer Systems (HotCarbon'23)*, 2023.

[6] A. Lechowicz, N. Christianson, J. Zuo, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "The online pause and resume problem: Optimal algorithms and an application to carbon-aware load shifting," *arXiv preprint arXiv:2303.17551*, 2023.

[7] L. Yang, M. Hajiesmaili, M. S. Talebi, J. Lui, W. S. Wong, *et al.*, "Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19943–19952, 2020.

[8] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 114–122, 2018.

[9] A. Gupta, T. Koren, and K. Talwar, "Better algorithms for stochastic bandits with adversarial corruptions," in *Conference on Learning Theory*, pp. 1562–1578, PMLR, 2019.

[10] D. Freund, T. Lykouris, E. Paulson, B. Sturt, and W. Weng, "Group fairness in dynamic refugee assignment," *arXiv preprint arXiv:2301.10642*, 2023.

[11] S. R. Sinclair, S. Banerjee, and C. L. Yu, "Sequential fair allocation: Achieving the optimal envy-efficiency tradeoff curve," *ACM SIGMETRICS Performance Evaluation Review*, vol. 50, no. 1, pp. 95–96, 2022.

[12] T. Lykouris and S. Vassilvtiskii, "Competitive caching with machine learned advice," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 3296–3305, PMLR, 10–15 Jul 2018.

[13] M. Purohit, Z. Svitkina, and R. Kumar, "Improving online algorithms via ml predictions," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31,

Curran Associates, Inc., 2018.

[14] A. Wei and F. Zhang, "Optimal robustness-consistency trade-offs for learning-augmented online algorithms," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8042–8053, 2020.

[15] R. Lee, J. Maghakian, M. Hajiesmaili, J. Li, R. Sitaraman, and Z. Liu, "Online peak-aware energy scheduling with untrusted advice," in *Proc. of ACM -e-Energy*, 2021.

[16] D. Fotakis, E. Gergatsouli, T. Gouleakis, and N. Patris, "Learning augmented online facility location," *arXiv preprint arXiv:2107.08277*, 2021.

[17] S. H.-C. Jiang, E. Liu, Y. Lyu, Z. G. Tang, and Y. Zhang, "Online facility location with predictions," in *International Conference on Learning Representations*, 2021.

[18] A. Antoniadis, T. Gouleakis, P. Kleer, and P. Kolev, "Secretary and online matching problems with machine learned advice," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7933–7944, 2020.

[19] A. Antoniadis, C. Coester, M. Elias, A. Polak, and B. Simon, "Online metric algorithms with untrusted predictions," in *International Conference on Machine Learning*, pp. 345–355, PMLR, 2020.

[20] S. Angelopoulos, S. Kamali, and K. Shadkami, "Online bin packing with predictions," in *Proc. of IJCAI*, 2022.

[21] B. Sun, R. Lee, M. Hajiesmaili, A. Wierman, and D. Tsang, "Pareto-optimal learning-augmented algorithms for online conversion problems," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[22] B. Jin and W. Ma, "Online bipartite matching with advice: Tight robustness-consistency tradeoffs for the two-stage model," *arXiv preprint arXiv:2206.11397*, 2022.

[23] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator, "Competitive snoopy caching," in *Proc. of IEEE FOCS*, pp. 244–254, 1986.

[24] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. cambridge university press, 2005.

[25] A. R. Karlin, C. Kenyon, and D. Randall, "Dynamic TCP acknowledgement and other stories about e/(e-1)," in *Proc. of ACM STOC*, pp. 502–509, 2001.

[26] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," in *Proc. of ACM SIGMETRICS*.

[27] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, 2012.

[28] L. Yang, M. H. Hajiesmaili, R. Sitaraman, A. Wierman, E. Mallada, and W. S. Wong, "Online linear optimization with inventory management constraints," *Proc. of ACM SIGMETRICS*, 2020.

[29] L. Yang, M. H. Hajiesmaili, H. Yi, and M. Chen, "Hour-ahead offering strategies in electricity market for power producers with storage and intermittent supply," in *Proc of. ACM SIGMETRICS*, 2017.

[30] Y. Zhang, M. H. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, "Peak-aware online economic dispatching for microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 323–335, 2018.

[31] B. Alinai, M. S. Talebi, M. H. Hajiesmaili, A. Yekkehkhany, and N. Crespi, "Competitive online scheduling algorithms with applications in deadline-constrained EV charging," in *Proc. of IEEE/ACM IWQoS*, 2018.

[32] B. Alinia, M. H. Hajiesmaili, Z. J. Lee, N. Crespi, and E. Mallada, "Online EV scheduling algorithms for adaptive charging networks with global peak constraints," *IEEE Transactions on Sustainable Computing*, 2020.

[33] L. Yang, W. S. Wong, and M. H. Hajiesmaili, "An optimal randomized online algorithm for QoS buffer management," in *Proc. ACM SIGMETRICS*, 2018.

[34] R. Bostandoost, B. Sun, C. Joe-Wong, and M. Hajiesmaili, "Near-optimal online algorithms for joint pricing and scheduling in ev charging networks," in *Proc. of ACM e-Energy*, 2023.

[35] S. R. M. Amarante, F. M. Roberto, A. R. Cardoso, and J. Celestino, "Using the multiple knapsack problem to model the problem of virtual machine allocation in cloud computing," in *Proc. of IEEE CSE*, 2013.

[36] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. Lau, "Online auctions in IaaS clouds: Welfare and profit maximization with server costs," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1034–1047, 2017.

[37] N. Buchbinder and J. S. Naor, "The design of competitive online algorithms via a primal–dual approach," *Foundations and Trends® in Theoretical Computer Science*, vol. 3, no. 2–3, pp. 93–263, 2009.

[38] B. Sun, A. Zeynali, T. Li, M. Hajiesmaili, A. Wierman, and D. H. Tsang, "Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging," *Proc. of ACM SIGMETRICS*, 2021.

[39] A. Zeynali, B. Sun, M. Hajiesmaili, and A. Wierman, "Data-driven competitive algorithms for online knapsack and set cover," in *Proc. AAAI*, 2021.

[40] J. Lorenz, K. Panagiotou, and A. Steger, "Optimal algorithms for k-search with application in option pricing," *Algorithmica*, vol. 55, no. 2, pp. 311–328, 2009.

[41] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin, "Optimal search and one-way trading online algorithms," *Algorithmica*, vol. 30, no. 1, pp. 101–139, 2001.

[42] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin, "Optimal search and one-way trading online algorithms," *Algorithmica*, vol. 30, pp. 101–139, May 2001.

[43] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin, "Optimal search and one-way trading online algorithms," *Algorithmica*, vol. 30, no. 1, pp. 101–139, 2001.

[44] M. Arsenis and R. Kleinberg, "Individual fairness in prophet inequalities," in *Proc. of ACM EC*, 2022.