On Tradeoffs in Learning-Augmented Algorithms

Ziyad Benomar
CREST, ENSAE, Ecole polytechnique,
Fairplay joint team

Vianney Perchet
CREST, ENSAE, Criteo AI LAB
Fairplay joint team

Abstract

The field of learning-augmented algorithms has gained significant attention in recent years. Using potentially inaccurate predictions, these algorithms must exhibit three key properties: consistency, robustness, and smoothness. In scenarios with stochastic predictions, a strong average-case performance is required. Typically, the design of such algorithms involves a natural tradeoff between consistency and robustness, and previous works aimed to achieve Pareto-optimal tradeoffs for specific problems. However, in some settings, this comes at the expense of smoothness. In this paper, we explore other tradeoffs between all the mentioned criteria and show how they can be balanced.

1 INTRODUCTION

Many decision-making problems under uncertainty are commonly studied using competitive analysis. In this context, the performance of online algorithms, operating under uncertainty, is compared to that of the optimal offline algorithm, which has full knowledge of the problem instance. While competitive analysis provides a rigorous method for evaluating online algorithms, it is often overly pessimistic. In real-world scenarios, decision-makers can have some prior knowledge, though possibly imperfect, about the complete problem instance. For example, predictions of unknown variables might be obtained via machine learning models, or an expert might provide advice on the best course of action. This more realistic setting was formalized by Lykouris and Vassilvtiskii [2018] and Purohit et al. [2018] leading to the development of what is now known as learning-augmented algorithms. In this paradigm, the algorithm receives predictions about the current problem instance, but without any guarantees on their accuracy, and must satisfy three main properties:

- Consistency: perform almost as well as the optimal offline algorithm if the predictions are perfect.
- Robustness: maintain a performance level close to the worst-case scenario without predictions when the predictions are arbitrarily bad.
- Smoothness: the performance should degrade gracefully as the prediction error increases.

Consistency, Robustness, and Brittleness. Consider a minimization problem under uncertainty, and let ALG be an algorithm augmented with a prediction y of an unknown parameter x. The input of the algorithm can contain parameters other than x, but for simplicity, we denote by $\mathsf{ALG}(x,y)$ the value of the objective function achieved by ALG, and $\mathsf{OPT}(x)$ the value of the optimal offline algorithm. The consistency c and robustness r of ALG are defined as

$$c = \sup_x \frac{\mathsf{ALG}(x,x)}{\mathsf{OPT}(x)} \quad \text{and} \quad r = \sup_{x,y} \frac{\mathsf{ALG}(x,y)}{\mathsf{OPT}(x)} \ .$$

Consistency is the worst-case ratio when the prediction is perfectly accurate, i.e. $y = x$, while robustness is the worst-case ratio with adversarial prediction. Most research on learning-augmented algorithms focuses on achieving good tradeoffs between consistency and robustness. Some studies also establish algorithms with Pareto-optimal tradeoffs, i.e. no algorithm can have simultaneously better consistency and better robustness. However, the proposed algorithms in such studies sometimes lack smoothness. Specifically, their worst-case performance can degrade abruptly, moving from the consistency bound when the predictions are perfect to the robustness bound even with an arbitrarily small error in the prediction. Following the terminology of [Angelopoulos et al., 2024a], we say that such an algorithm is brittle.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

Definition 1.1 (Brittleness)**.** *a deterministic algorithm ALG with robustness r is brittle if*

$$\forall \varepsilon > 0 : \sup_{x,y\colon |x-y| \leq \varepsilon x} \frac{\mathsf{ALG}(x,y)}{\mathsf{OPT}(x)} = r \ .$$

Notably, for a brittle algorithm, it is impossible to establish an upper bound on $\mathsf{ALG}(x,y)/\mathsf{OPT}(x)$ as a continuous function of the prediction error $f(|x-y|)$ while ensuring $f(0) < r$. The definitions of consistency, robustness, brittleness, and the above observation extend to randomized algorithms by considering the expected value of the algorithm's output, $\mathbb{E}[\mathsf{ALG}(x,y)]$.

In real-world scenarios, predictions are rarely perfect. As a result, the only reliable guarantee for brittle algorithms is the robustness bound, which is at best equivalent to the worst-case bound without predictions. This greatly limits the practical usefulness of these algorithms. In the case of the *one-way trading* problem, Angelopoulos et al. [2024a] demonstrated in a very recent work that any algorithm achieving a Pareto-optimal tradeoff between consistency and robustness is brittle. This finding implies that, in some problems, achieving smoothness requires deviating from the Pareto-optimal tradeoff between consistency and robustness.

Average-Case Performance. In the context of learning-augmented algorithms, the consistency, robustness, and smoothness of an algorithm represent worst-case guarantees with respect to the prediction error. On the other hand, there are numerous scenarios where the decision-maker might know some information about the distribution of the prediction, which motivates the design of algorithms with good average performance [Dütting et al., 2021, Gupta et al., 2022, Benomar and Perchet, 2023, Henzinger et al., 2023, Cohen-Addad et al., 2024]. Nevertheless, while achieving a balance between worst-case and average-case performance has been widely studied in various fields of algorithm design and machine learning [Szirmay-Kalos and Márton, 1998, Witt, 2005, Peikert and Rosen, 2007, Antunes and Fortnow, 2009, Chuangpishit et al., 2018, Rice et al., 2021, Robey et al., 2022], this aspect has not yet been investigated in the context of learning-augmented algorithms for the prediction error.

1.1 Contributions

In this work, we explore various tradeoffs that arise in the design and analysis of learning-augmented algorithms. While existing literature has primarily focused on the tradeoff between consistency and robustness, our investigation centers on the tradeoffs between consistency and smoothness, as well as the relationships between the standard criteria for learning-augmented algorithms—namely consistency, robustness, and smoothness—and their average performance under stochastic assumptions regarding predictions.

We begin by examining the *line search problem*, revisiting the algorithm proposed by Angelopoulos et al. [2019]. This algorithm achieves a Pareto-optimal tradeoff between consistency and robustness among deterministic algorithms, but we demonstrate that it is inherently brittle. We show that this brittleness can be mitigated by introducing randomness into the predictions used by the algorithm. The variance of this randomization is quantified by a parameter $\rho \geq 0$. Our analysis reveals that tuning this parameter leads to opposing effects on the consistency and smoothness of the algorithm, thus yielding a tradeoff between these two criteria.

Next, we apply a similar approach to the *one-max search problem*. We examine the Pareto-optimal algorithm introduced by Sun et al. [2021], and we show its brittleness. Furthermore, we demonstrate how randomization can be used to guarantee smoothness at the cost of consistency. Once again, the resulting tradeoff is governed by a parameter $\rho \geq 0$.

In both problems, the guarantees we obtain are suboptimal within the class of randomized algorithms. However, our focus is on a practical objective: enhancing the real-world applicability of brittle algorithms.

Finally, we address the *ski-rental problem*, proposing a deterministic algorithm that generalizes that of Purohit et al. [2018]. Through a tight analysis of its performance, we prove that the Pareto-optimal tradeoff between consistency and robustness can be achieved with different levels of smoothness. However, we show that striving for optimal smoothness degrades the average-case performance of the algorithm, assuming that the prediction induces the correct decision (renting or buying at time 0) with a probability $q \in [\frac{1}{2}, 1]$. In this context, a parameter $\rho \in [0, 1]$ can be utilized to tune the levels of smoothness and average-case performance, all while maintaining fixed consistency and robustness.

Additionally, we conduct numerical experiments for the three problems studied in the paper, highlighting the various tradeoffs demonstrated in our analysis.

1.2 Related Work

Learning-Augmented Algorithms. The design of learning-augmented algorithms relies on using machine-learned advice to go beyond worst-case limitations [Lykouris and Vassilvtiskii, 2018, Purohit et al., 2018]. These algorithms operate under the assumption that the decision-maker has access to noisy predictions about certain problem parameters. The goal of learning-augmented algorithms is to improve performance if the predictions are accurate, while also ensuring robustness in the face of incorrect or adversarial predictions. Many fundamental algorithmic problems were studied in this setting, such as ski rental [Gollapudi and Panigrahi, 2019, Diakonikolas et al., 2021, Antoniadis et al., 2021, Shin et al., 2023], caching [Lykouris and Vassilvtiskii, 2018, Chlkedowski et al., 2021, Antoniadis et al., 2023b,a], scheduling [Purohit et al., 2018, Merlis et al., 2023, Lassota et al., 2023, Benomar and Perchet, 2024], and the design of data structures [Kraska et al., 2018, Lin et al., 2022, Zeynali et al., Benomar and Coester, 2024].

Overcoming Brittleness. Pareto-optimal tradeoffs between consistency and robustness were studied in [Angelopoulos et al., 2019, Bamas et al., 2020, Wei and Zhang, 2020, Sun et al., 2021, Angelopoulos, 2023]. However, the proposed algorithms do not always have smoothness guarantees. For example, Angelopoulos et al. [2024a] proved that any Pareto-optimal algorithm for the one-way trading problem is necessarily brittle (Definition 1.1). In this paper, we will show in the line search and in the one-max search problems how a randomized deviation from the Pareto-optimal algorithm allows for overcoming brittleness. A similar approach was used to guarantee smoothness in nonclairvoyant scheduling with limited predictions [Benomar and Perchet, 2024].

Line Search. The line search problem [Beck, 1964], also known as the cow path problem, consists of finding a hidden target on an infinite line starting from an initial position, without any information regarding the direction or distance to the target. The goal is to minimize the total distance traveled before reaching the target. The best deterministic algorithm is based on doubling the search distance in alternating directions, and it ensures a competitive ratio of 9 [Beck and Newman, 1970, Baezayates et al., 1993]. The line search problem has been extensively studied in the learning-augmented framework with different types of predictions [Angelopoulos, 2023, Angelopoulos et al., 2024b].

One-Max Search. In the one-max search problem [El-Yaniv et al., 2001], the decision-maker observes a sequence of adversarially chosen prices $p_1, \dots, p_n \in [L, U]$, with $0 < L < U$. At each step i, the price p_i is revealed to the decision-maker, and the latter can decide to stop the game and have a payoff of p_i, or reject it irrevocably and move to the next observation. The best deterministic algorithm for this problem consists simply in selecting the first price larger than $\sqrt{UL}$, which guarantees a payoff of at least $\sqrt{L/U}\max_{i\in[n]} p_i$. This problem, as well as its randomized version—*online conversion*—were studied in the learning-augmented setting with a prediction of the maximal price [Sun et al., 2021].

Ski-Rental In the ski-rental problem, the decision maker faces a daily choice between renting a ski at a unit cost or buying it for a one-time cost of b, after which skiing becomes free. The length of the ski season, x, is unknown, and the goal is to minimize the total cost of renting and buying. A straightforward algorithm for this problem is renting for the first $b-1$ days and then buying at day b, resulting in a competitive ratio of 2, which is the best achievable by any deterministic algorithm Karlin et al. [1988]. The best competitive ratio with randomized algorithms is $\frac{e}{e-1}$ [Karlin et al., 1994].

2 SMOOTH ALGORITHM FOR LINE SEARCH

In the line search problem, a target is hidden in an unknown position $x \in \mathbb{R}$ on the line, with $|x| \geq 1$, and the searcher, initially at the origin of the line O, must find the target, minimizing the total traveled distance. The optimal offline algorithm only travels a distance of $|x|$ to find the target. On the other hand, the searcher, ignoring if x is to the left or the right side of O, must alternate the search direction multiple times before finding the target. Any deterministic algorithm for this problem can be defined as an iterative strategy, parameterized by an initial search direction $s_0 \in \{-1, 1\}$ and a sequence of turn points $(d_i)_{i\in\mathbb{N}} \in [1,\infty]^{\mathbb{N}}$. At the beginning of any iteration $i \geq 0$, the searcher is located at the origin O, then it travels a distance d_i in the direction $(-1)^i s_0$ and returns to the origin. The algorithm terminates when the position x is reached.

Pareto-Optimal Algorithm. Given a prediction y of x, Angelopoulos [2023] designed an algorithm $\mathsf{A}_b^{\mathrm{LS}}$ that has a consistency of $\frac{b+1}{b-1}$, and robustness of $1 + \frac{2b^2}{b-1}$, where $b \geq 2$ is a hyperparameter of the algorithm. Denoting by $\mathsf{A}_b^{\mathrm{LS}}(x, y)$ the distance traveled by $\mathsf{A}_b^{\mathrm{LS}}$ to find the target x given the prediction y, the consistency and robustness guarantees can be written as

$$\forall x: \quad \frac{\mathsf{A}_b^{\mathrm{LS}}(x,x)}{|x|} \leq \frac{b+1}{b-1}\,,$$
$$\forall x, y: \quad \frac{\mathsf{A}_b^{\mathrm{LS}}(x,y)}{|x|} \leq 1 + \frac{2b^2}{b-1}\,.$$

Moreover, the author proves that these consistency and robustness levels are Pareto-optimal. The proposed algorithm has a simple structure: let $k_y \in \mathbb{N}$

such that $b^{k_y - 2} < |y| \le b^{k_y}$, and $\gamma_y = b^{k_y}/|y| \ge 1$, the algorithm $\mathsf{A}_b^{\mathrm{LS}}$ is defined by the initial search direction $s_0 = (-1)^{k_y}\operatorname{sign}(y)$ and the turn points $d_i = b^i/\gamma_y$ for all $i \ge 0$. The algorithm is defined so that, during the iteration k_y, the searcher travels a distance of $|y|$ in the direction given by $\operatorname{sign}(y)$, i.e. it reaches the position y exactly at the turning point of iteration k_y.

Brittleness of $\mathsf{A}_b^{\mathrm{LS}}$. In the following, we will prove that $\mathsf{A}_b^{\mathrm{LS}}$ is brittle, in the sense of Definition 1.1, then we will demonstrate how a simple randomization idea enables making the algorithm smooth. To better understand the impact of the prediction error on the performance of $\mathsf{A}_b^{\mathrm{LS}}$, we first prove an expression of $\mathsf{A}_b^{\mathrm{LS}}(x,y)/x$ as a function of y and x.

Lemma 2.1. *Let $x \ge 1$, $y > 0$ and $j = \lceil \frac{\ln(x/y)}{2\ln b} \rceil \in \mathbb{Z}$, so that $1 \le \frac{y}{x} b^{2j} < b^2$. It holds that*

$$\frac{A_b^{\mathrm{LS}}(x,y)}{x} = 1 + \frac{2b^2}{b-1} \cdot b^{2(j-1)} \frac{y}{x} - o(1/x) \; .$$

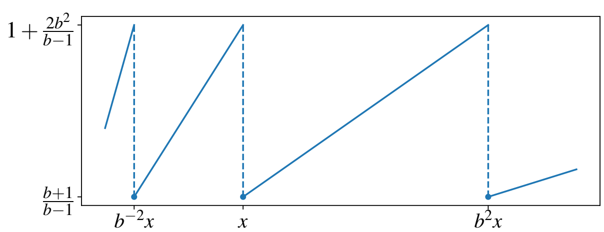

Figure 1: The mapping $y \mapsto \mathsf{A}_b^{\mathrm{LS}}(x,y)/x$ for x arbitrary large and $y \in [\frac{x}{b^2}, b^2 x]$.

.

The expression proved in Lemma 2.1 is illustrated in Figure 1 for x arbitrary large and $y \in [x/b^2, b^2 x]$. It shows that the ratio $\mathsf{A}_b^{\mathrm{LS}}(x,y)/x$ increases smoothly from the consistency to the robustness bound if the prediction y is larger than x, but presents a discontinuity, going immediately from the consistency to robustness bound if $y < x$ and arbitrarily close to x. This proves the brittleness of $\mathsf{A}_b^{\mathrm{LS}}$, which can be formally stated as follows.

Proposition 2.2 ($\mathsf{A}_b^{\mathrm{LS}}$ is brittle)**.** *For any $\varepsilon > 0$, it holds that*

$$\sup_{x,y:|x-y|\le \varepsilon x} \frac{\mathsf{A}_b^{\mathrm{LS}}(x,y)}{|x|} = 1 + \frac{2b^2}{b-1} \; .$$

Smoothness via Randomization. In all the following, we assume without loss of generality that $x > 0$. The worst-case ratio $\mathsf{A}_b^{\mathrm{LS}}(x,y)/x$, given in Proposition 2.2, occurs when $x = y + \varepsilon$ with ε arbitrarily small. To avoid it, we perturb y and run instead the algorithm $\mathsf{A}_b^{\mathrm{LS}}$ with a randomized prediction of the form $\tilde{y} = (1+\rho\xi)y$, where $\rho > 0$ is a hyperparameter and ξ a positive random variable.

Theorem 2.3. *Let $b \ge 2$, $\rho \in [0,1]$, and ξ a random variable with tail distribution $\Pr(\xi \ge t) = \frac{1}{(1+t)^2}$ for all $t \ge 0$. Then for any $x \ge 1$ and $y \in \mathbb{R}$, denoting by $\eta = |x-y|$, we have for $\tilde{y} = (1+\rho\xi)y$ that*

$$\frac{\mathbb{E}_\xi[A_b^{\mathrm{LS}}(x,\tilde{y})]}{x} \le \frac{b+1+2\rho}{b-1} + \begin{cases} \dfrac{2(1+\rho)}{b-1} \cdot \dfrac{\eta}{x} & \textit{if } y \ge x \\[2ex] \dfrac{4(b+1)}{\rho} \cdot \dfrac{\eta}{x} & \textit{if } y < x \end{cases}$$

and we have with probability 1 that

$$\frac{A_b^{\mathrm{LS}}(x,\tilde{y})}{x} \le 1 + \frac{2b^2}{b-1} \; .$$

The first bound in the previous theorem establishes the algorithm's consistency and smoothness, while the second bound characterizes its robustness, which remains unaffected by randomizing the prediction. For $\rho > 0$, the performance of the algorithm is upper bounded by a continuous function of the prediction error η, which shows that our algorithm is not brittle.

Beyond the consistency-robustness tradeoff governed by the parameter b, the algorithm also exhibits a tradeoff between consistency and smoothness, governed by the parameter $\rho \in [0,1]$. For $\rho = 0$, the algorithm is identical to that of Angelopoulos et al. [2019] and has optimal levels of consistency and smoothness. For $\rho > 0$, the smoothness factor for $y < x$ improves, but the algorithm becomes less consistent.

Although the algorithm is well-defined for all values of $\rho > 0$, we limit our analysis to $\rho \in [0,1]$, as this range allows for simpler expressions of the upper bound.

3 SMOOTH ALGORITHM FOR ONE-MAX SEARCH

In the one-max search problem, a decision-maker sequentially observes prices $p_1, \dots, p_n \in [L, U]$, where $0 < L < U$, and upon observing each price p_i they must decide either to select it, halting the process and receiving a payoff of p_i, or to reject it irrevocably and move on to the next price.

Let ALG denote an online algorithm for this problem. We use $\mathsf{ALG}(p)$ to denote the price selected by ALG when given an input instance $p = (p_1, \dots, p_n)$, and we use $p^* = \max_{i\in[n]} p_i$ to denote the highest price in the instance. Since this is a maximization problem The competitive ratio of ALG is defined as:

$$\mathsf{CR}(\mathsf{ALG}) = \inf_p \frac{\mathsf{ALG}(p)}{p^*} \tag{1}$$

where the infimum is taken over all possible price sequences of arbitrary length, with prices in the range $[L, U]$. In some works, the competitive ratio is alternatively defined as the inverse of our definition, i.e., $\mathsf{CR}(\mathsf{ALG}) = \sup_p \frac{p^*}{\mathsf{ALG}(p)}$. However, since our objective is to prove smoothness guarantees, it is more convenient to define the competitive ratio as in (1). Finally, let $\theta = U/L$. Without loss of generality, we can assume that the prices are in the interval $[1, \theta]$.

Pareto-Optimal Algorithm. Given a prediction y of the maximum price, the consistency c and robustness r of any algorithm ALG are defined as $c = \inf_p \frac{\mathsf{ALG}(p,p^*)}{p^*}$ and $r = \inf_{p,y} \frac{\mathsf{ALG}(p,y)}{p^*}$. In this setting, Sun et al. [2021] proved for all $\lambda \in [0,1]$ an algorithm $\mathsf{A}^{\mathrm{OM}}_\lambda$ with consistency $c(\lambda)$ and robustness $r(\lambda)$, defined as the unique solution of the system

$$\frac{1}{c(\lambda)} = \theta \cdot r(\lambda) \quad \text{and} \quad \frac{1}{c(\lambda)} = \frac{\lambda}{r(\lambda)} + 1 - \lambda \; . \qquad (2)$$

The proposed algorithm is a threshold policy, where the first price at least equal to the threshold $\Phi(\lambda, y)$ is selected, with

$$\Phi(\lambda, y) = \begin{cases} \frac{1}{c(\lambda)} & \text{if } y \in [1, \frac{1}{c(\lambda)}) \\ \frac{\lambda}{r(\lambda)} + (1-\lambda)c(\lambda)y & \text{if } y \in [\frac{1}{c(\lambda)}, \frac{1}{r(\lambda)}) \\ \frac{1}{r(\lambda)} & \text{if } y \in [\frac{1}{r(\lambda)}, \theta] \end{cases} .$$

If all the prices are less than $\Phi(\lambda, y)$, then the algorithm selects the last price p_n. Furthermore, the authors prove that the levels of consistency and smoothness of $\mathsf{A}^{\mathrm{OM}}_\lambda$ are Pareto-optimal. However, they do not provide any smoothness guarantees.

Brittleness of $\mathsf{A}^{\mathrm{OM}}_\lambda$. We will prove in the following that this algorithm is brittle, and we will show, similarly to the line search problem, how this brittleness can be overcome via randomization. In the following, we simply write c, r instead of $c(\lambda), r(\lambda)$, and we denote by $\eta := |p^* - y|$ the prediction error. Our first result is that the competitive ratio of A_λ degrades smoothly as a function of the prediction error when $y \in [1, \frac{1}{r})$. Then, we will prove the brittleness of the algorithm in Proposition 3.3 by considering $y \geq \frac{1}{r}$. The following lemma shows smoothness for $y \in [1, \frac{1}{c})$.

Lemma 3.1. *If $y \in [1, \frac{1}{c})$, then*

$$\frac{A^{\mathrm{OM}}_\lambda(p, y)}{p^*} \geq c - \frac{c \cdot \eta}{p^*} \; .$$

The next lemma proves a similar result for $y \in [\frac{1}{c}, \frac{1}{r})$.

Lemma 3.2. *If $y \in [\frac{1}{c}, \frac{1}{r})$, then*

$$\frac{A^{\mathrm{OM}}_\lambda(p, y)}{p^*} \geq c - (1-\lambda)\max(1, \tfrac{c}{\lambda})\frac{c \cdot \eta}{p^*} \; .$$

Finally, we demonstrate the brittleness of $\mathsf{A}^{\mathrm{OM}}_\lambda$ by considering y greater than, but arbitrarily close to, $\frac{1}{r}$.

Proposition 3.3 ($\mathsf{A}^{\mathrm{OM}}_\lambda$ is brittle)**.** *For any $\varepsilon > 0$, it holds that*

$$\inf_{p,y:\, \frac{|p^*-y|}{p^*} \leq \varepsilon} \frac{A^{\mathrm{OM}}_\lambda(p, y)}{p^*} = r(\lambda) \; .$$

Smoothness via Randomization. As we proved in Lemmas 3.1 and 3.2, if $y \in [1, \frac{1}{r})$ then performance of $\mathsf{A}^{\mathrm{OM}}_\lambda$ degrades smoothly with the prediction error. The brittleness of $\mathsf{A}^{\mathrm{OM}}_\lambda$ in Proposition 3.3 arises in the case where $y \in [1/r, \theta]$: the ratio $\mathsf{A}^{\mathrm{OM}}_\lambda(p, y)/p^*$ is larger than c for $p^* \geq 1/r$, but it drops immediately to r for $p^* < 1/r$, even arbitrarily close to $1/r$.

To attenuate this extreme behavior, we randomize the threshold used when $y \in [1/r, \theta]$. Let $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$ the algorithm accepting the first price at least equal to the random threshold $\tilde{\Phi}(\lambda, \rho, y)$ defined by

$$U \sim \mathcal{U}[0,1], \quad \tilde{\Phi}(\lambda, \rho, y) = \begin{cases} \Phi(\lambda, y) & \text{if } y \in [1, \frac{1}{r}) \\ \dfrac{e^{-\rho U}}{r} & \text{if } y \in [\frac{1}{r}, \theta] \end{cases} .$$

If $y \in [1, \frac{1}{r})$, then $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$ is equivalent to $\mathsf{A}^{\mathrm{OM}}_\lambda$, thus $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$ is r-robust in that case, and the consistency and smoothness guarantees from Lemmas 3.1 and 3.2 extend to $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$. Consequently, it suffices to study $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$ when $y \in [\frac{1}{r}, \theta]$, and we obtain the following result.

Theorem 3.4. *Let $\lambda \in [0,1]$, $\rho \geq 0$, and let $c = c(\lambda)$ and $r = r(\lambda)$ as defined in (2). For any sequence of prices $p = (p_1, \ldots, p_n) \in [1, \theta]^n$ and prediction $y \in [1, \theta]$ of $p^* := \max_{i \in [n]} p_i$, it holds that*

$$\frac{\mathbb{E}_U[A^{\mathrm{OM}}_{\lambda,\rho}(p, y)]}{p^*} \geq \left(\frac{1 - e^{-\rho}}{\rho}\right) r \; ,$$

and denoting by $\eta = |p^* - y|$, the ratio $\frac{\mathbb{E}_U[A^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*}$ is at least

$$\begin{cases} c - \frac{c \cdot \eta}{p^*} & \text{if } y \in [1, 1/c) \\ c - (1-\lambda)\max(1, \frac{c}{\lambda})\frac{c \cdot \eta}{p^*} & \text{if } y \in [1/c, 1/r) \\ \left(\frac{1-e^{-\rho}}{\rho}\right)c - \left(\frac{c-r}{\rho}\right)\frac{\eta}{p^*} & \text{if } y \in [1/r, \theta] \end{cases} ,$$

The first lower bound, independent of the prediction error η, is the robustness of the algorithm, while the second bound characterizes its consistency and smoothness. The theorem shows that, in order to guarantee a certain level of smoothness, $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$ degrades both the consistency and robustness of $\mathsf{A}^{\mathrm{OM}}_\lambda$ by a factor of $(1 - e^{-\rho})/\rho$, hence exhibiting a tradeoff between smoothness and both consistency and robustness.

4 AVERAGE-CASE ANALYSIS IN SKI-RENTAL

In this section, we focus on *ski-rental*, which is one of the fundamental problems in competitive analysis. In this problem, the decision-maker must choose each day between renting a ski for a unit cost or buying it for a fixed cost b, allowing them to ski for free for the remainder of the ski season, which has an unknown duration x. The objective is to minimize the total cost incurred from renting and buying. To simplify our presentation, we consider the continuous version of the problem, where the number of skiing days increases continuously, with $x, b > 0$. In this model, the cost of renting for a time period $[t, t+\delta)$ is equal to δ.

The ski-rental problem was one of the first problems studied in the learning-augmented framework. Purohit et al. [2018] proved that, with a prediction y of x, there is a deterministic algorithm with a competitive ratio of at most

$$\min\left(1 + \frac{1}{\lambda}\ ,\ (1+\lambda) + \frac{|x-y|}{(1-\lambda)\min(x,b)}\right) .$$

where $\lambda \in [0,1]$. It was proved later in Wei and Zhang [2020] that the consistency $(1+\lambda)$ and robustness $(1+\frac{1}{\lambda})$ are Pareto-optimal. On the other hand, Benomar and Perchet [2023] analyzed the same algorithm under the assumption that $\Pr(\mathbb{1}_{y \ge b} = \mathbb{1}_{x \ge b}) = q$ for some $q \in [1/2, 1]$, and showed how to optimally choose λ to minimize the expected cost of the algorithm.

In the following, we combine the analysis of average-case performance with the criteria of consistency, robustness, and smoothness. To achieve this, we propose a modified version $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ of the algorithm introduced by Purohit et al. [2018], which is parameterized by two parameters $\lambda, \rho \in [0,1]$.

Algorithm 1: $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y)$

if $y \ge b$ **then** buy at time λb ;
if $y < b$ **then** buy at time $(1 + \rho(\frac{1}{\lambda} - 1))b$;

Note that the algorithm of Purohit et al. [2018] corresponds to $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ with $\rho = 1$, i.e. buying at time b/λ if $y < b$. We start by proving the consistency, robustness, and smoothness of this algorithm.

Theorem 4.1. *For all $x, y > 0$, denoting by $\eta = |x - y|$, it holds that $\frac{\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y)}{\min(x,b)}$ is at most*

$$\min\left(1 + \frac{1}{\lambda}\ ,\ (1+\lambda) + \left(1 + \frac{\lambda}{\rho}\right)\frac{\eta}{\min(x,b)}\right) .$$

The theorem above demonstrates that, for any value of $\rho \in [0,1]$, the algorithm $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ achieves Pareto-optimal consistency and robustness, albeit with varying levels of smoothness. Furthermore, note that our analysis is tighter than that of Purohit et al. [2018]. Specifically, when $\rho = 1$, we proved a smoothness factor of $1+\lambda$ instead of $\frac{1}{1-\lambda}$.

In the subsequent theorem, we assume that the prediction y lies on the same side of b as x with a probability of at least $q \in \left[\frac{1}{2}, 1\right]$, and we establish an upper bound on the expected cost of Algorithm 1. The assumption on y is pertinent for this setting, as the decision made by the algorithm depends only on where y is situated compared to b. The same assumption was considered in Benomar and Perchet [2023].

Theorem 4.2. *For all $x > 0$, if the prediction y is a random variable satisfying $\Pr(\mathbb{1}_{y \ge b} = \mathbb{1}_{x \ge b}) \ge q$ for some $q \in [\frac{1}{2}, 1]$, then $\frac{\mathbb{E}_y[A^{\mathrm{SR}}_{\lambda,\rho}(x,y)]}{\min(x,b)}$ is at most*

$$\max\left(2 + (\tfrac{1}{\lambda} - 1)\big((1-q)\rho - q\lambda\big), 1 + \frac{1-q}{\lambda}\right) .$$

Note that the proven upper bound is non-decreasing with ρ, regardless of the values of q or λ. Hence, $\rho = 0$ is the optimal choice for achieving the best average-case performance of the algorithm. Moreover, if the value of q is known, then λ can also be chosen optimally to minimize the upper bound, as demonstrated in the following corollary.

Corollary 4.3. *Under the same assumptions of Theorem 4.2, it holds for $\rho = 0$ and $\lambda^* = \frac{1}{2}\sqrt{(\frac{1}{q} - 1)(\frac{1}{q} + 3)} - \frac{1}{2}(\frac{1}{q} - 1)$ that*

$$\frac{\mathbb{E}_y[A^{\mathrm{SR}}_{\lambda^*,0}(x,y)]}{\min(x,b)} \le \frac{3-q}{2} + \frac{1}{2}\sqrt{(1-q)(1+3q)} .$$

Under the same assumption on the prediction y, Benomar and Perchet [2023] proved an upper bound of $(1 + 2\sqrt{q(1-q)})\min(x,b)$ on the average cost of $\mathsf{A}^{\mathrm{SR}}_{\lambda,1}$ for a well-chosen value of λ. The bound of Corollary 4.3 is better than the latter as shown in Figure 2.

Moreover, note that for $q = 1/2$, the bound of Corollary 4.3 is better than 2, which is the best competitive ratio achievable by a deterministic algorithm for the ski-rental problem. This is because, in that case, $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ uses the Bernoulli random variable $\mathbb{1}_{y \ge b} \sim \mathcal{B}(1/2)$.

Smoothness and Average-Cost Tradeoff. A natural tradeoff arises between the consistency, robustness, and average cost of the algorithm. Minimizing the average cost necessitates selecting λ optimally to minimize the upper bound established in Theorem 4.2. However, the decision-maker may opt to deviate from this value to achieve a better level of consistency or robustness, depending on the specific use case.

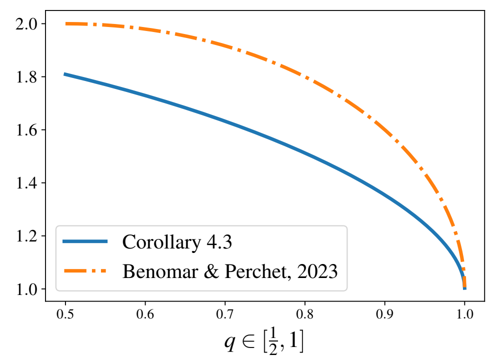

Figure 2: Upper bound on the competitive ratio of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ with λ, ρ as in Corollary 4.3, and with λ, ρ as in Lemma 2.2 of Benomar and Perchet [2023].

Furthermore, Theorems 4.1 and 4.2 imply that, for a fixed $\lambda \in [0, 1]$, the levels of consistency and robustness of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ are constant, while the smoothness and average cost can be further adjusted using ρ. While increasing ρ enhances the smoothness, it degrades the average cost, regardless of the accuracy q of the predictions. This indicates that, in addition to the tradeoff between consistency and robustness governed by λ, the algorithm also exhibits a tradeoff between average cost and smoothness, governed by the parameter ρ.

5 EXPERIMENTS

In this section, we present experimental results to validate our theoretical findings and provide additional insights into the tradeoffs discussed in the paper.

Line Search. As established in Lemma 2.1 and illustrated in Figure 1, given a target position $x \geq 1$ and prediction $y > 0$, the ratio between the distance traveled by Algorithm $\mathsf{A}^{\mathrm{LS}}_b$ and the optimal offline algorithm depends solely on the ratio x/y when x is large. To investigate the impact of the parameter ρ, we fix $x = 100$ and $b = 2.5$, then compare the behavior of the algorithm presented in Section 2 for three different values of $\rho \in \{0.05, 0.5, 5\}$, with $y \in [\frac{x}{b^2}, b^2 x]$. For each point in the experiment, the average and standard deviation are computed over 10^5 independent trials. Figure 3 shows the results of this experiment. It demonstrates that smaller values of ρ lead to better consistency, but make the algorithm highly sensitive to prediction errors, particularly to predictions slightly smaller than the true value x. Note that $\rho = 0.05$, yields a consistency of almost $\frac{b+1}{b-1}$, but the ratio $\mathsf{A}^{\mathrm{LS}}_b/\mathsf{OPT}$ degrades almost immediately to reach a larger value close to $1 + \frac{2b^2}{b-1}$ for a slight prediction error $y < x$. On the other hand, $\rho = 5$ yields a less consistent algorithm, but its performance is not strongly affected by small prediction errors. This highlights the consistency-smoothness tradeoff established in Theorem 2.3.

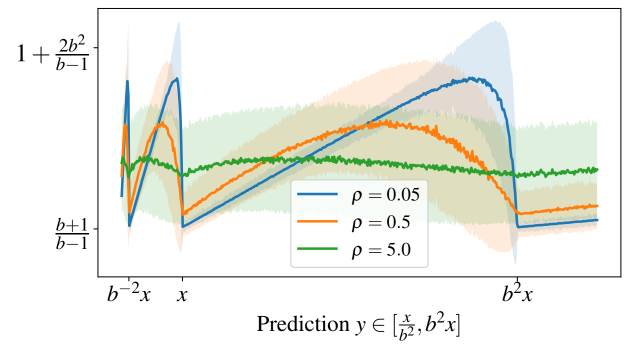

Figure 3: Consistency-smoothness tradeoff of $\mathsf{A}^{\mathrm{LS}}_b$ with a prediction randomized as in Theorem 2.3

One-Max Search. We conduct an analogous experiment for the one-max search problem to demonstrate the consistency-smoothness tradeoff for $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$. Given a sequence with a maximal price p^*, the algorithm is provided with a noisy prediction in the form $y = p^* + \varepsilon$, where $\varepsilon \sim \mathcal{U}[-\sigma, \sigma]$. The threshold set by $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$, denoted as $\tilde{\Phi}(\lambda, \rho, y)$, determines its worst-case payoff: if $p^* \geq \tilde{\Phi}(\lambda, \rho, y)$, the algorithm gains $\tilde{\Phi}(\lambda, \rho, y)$; otherwise, the gain is 1, which is the minimum possible price in the sequence. This scenario is asymptotically achieved by the sequence $p = (p_1, \ldots, p_{n+1})$, where $p_i = 1 + \frac{i-1}{n-1}(p^* - 1)$ for $i \leq n$, and $p_{n+1} = 1$. In the experiment, $\lambda = 0.1$ and $\theta = 5$ are fixed, and for each $\sigma \in [0, \theta]$, the worst-case average ratio $\sup_{p^* \in [1,\theta]} \mathbb{E}[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p, p^* + \varepsilon)]/p^*$ and the corresponding standard deviation are evaluated over 10^5 independent samples. Figure 4 shows that the algorithm suffers from brittleness for $\rho = 0$, as the slightest prediction error substantially degrades its performance. In contrast, as ρ increases and randomization is introduced, the algorithm becomes smoother; at the cost of consistency.

Ski Rental. For the ski-rental problem, two experiments are conducted to investigate, on one hand, the impact of the parameter ρ on the consistency and smoothness of the algorithm $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$, and on the other hand, its impact on the average performance of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$, under the assumption that the prediction lies on the same side of b as the true value of the number of snow days x with probability q. In both experiments, we set $b = 10$ and $\lambda = 0.5$.

In Figure 5, the performance of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ is evaluated against the prediction error $\eta = |y - x|$. Predic-

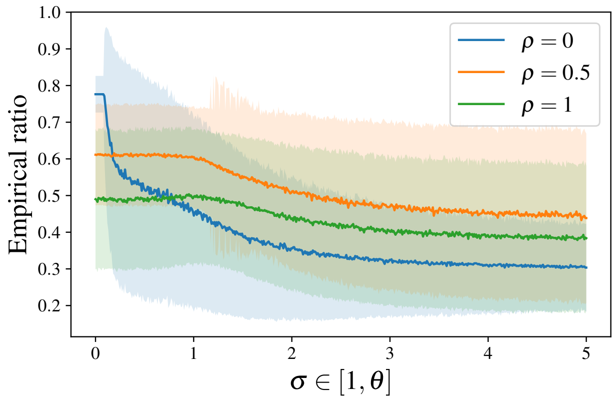

Figure 4: Consistency-smoothness tradeoff of $\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}$

tions are of the form $y = x + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. For each value of σ, the figure shows the worst-case ratio $\sup_{x \in (0,5b]} \mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y)/\min(x, b)$, which compares the cost induced by $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ to that of the optimal offline algorithm at a fixed level of error.

When $\rho = 0$, the algorithm lacks smoothness, resulting in a significant increase in cost from the consistency value of $1 + \lambda = 1.5$ when the error is zero, to ≈ 1.7 for even a minimal positive error. For larger values of ρ, the algorithm exhibits improved smoothness while maintaining the same consistency level, as proved in Theorem 4.1. However, it is noteworthy that smaller values of ρ tend to yield a better average ratio when the prediction error is large.

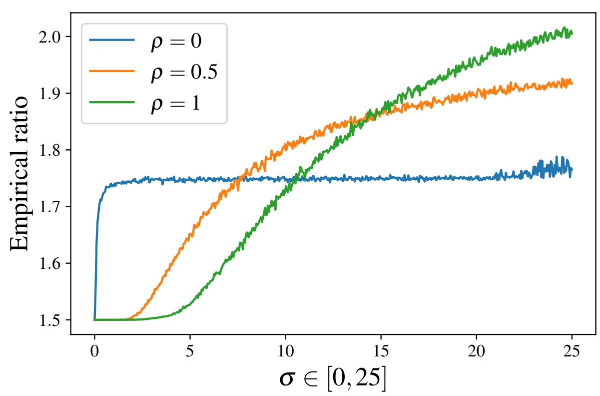

Figure 5: Consistency-smoothness tradeoff of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ with $y \sim x + \mathcal{N}(0, \sigma^2)$

On the other hand, Figure 6 examines the the average cost of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$, assuming that $\mathbb{1}_{y \geq b} = \mathbb{1}_{x \geq b}$ with probability $q \in \left[\frac{1}{2}, 1\right]$. Each data point in the figure is computed over 10^5 independent trials, where $x \sim \mathcal{U}[1, 4b]$, and y is selected arbitrarily on the same side of b as x with probability q, while on the opposite side with probability $1 - q$. The results further corroborate our theoretical findings, indicating that smaller values of ρ yield the best average performance.

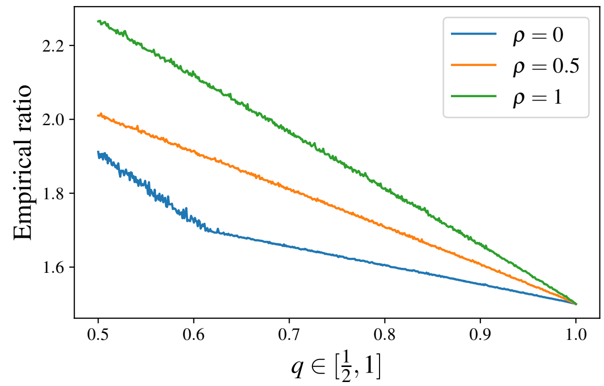

Figure 6: Average-case performance of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$ when $\mathbb{1}_{y \geq b} = \mathbb{1}_{x \geq b}$ with probability q

Combined with the observations from Figure 5, these experiments delineate the tradeoff between the smoothness and average-case performance of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$, which can be tuned using the parameter ρ.

6 CONCLUSION

Achieving Pareto-optimal tradeoffs between consistency and robustness in the design of learning-augmented algorithms is a challenging and highly interesting question. However, the absence of smoothness guarantees strongly limits the applicability of these algorithms in practical scenarios, where predictions are rarely perfectly accurate.

In the cases of the *line search* and *one-max search* problems, we demonstrated that by introducing randomization into brittle algorithms that exhibit Pareto-optimal consistency-robustness tradeoffs, we can establish smoothness guarantees. This approach consequently reveals an additional tradeoff between consistency and smoothness.

On the other hand, in the ski rental problem, we interestingly find that the Pareto-optimal tradeoff between consistency and robustness can be achieved by multiple algorithms, each exhibiting a different level of smoothness. However, improving smoothness results in a degradation of the average-case performance of the algorithm, hence showing another tradeoff between these criteria.

Our work highlights that, although learning-augmented algorithms have been extensively studied in the last few years, many of their aspects might not have been examined in the existing literature. In particular, we showed that the mutual dependencies among the various criteria involve additional tradeoffs and limitations, opening new research questions in the field. Potential avenues for follow-up research include

- establishing Pareto-optimal fronts on the consistency, robustness, and smoothness of deterministic or randomized algorithms,
- further exploration regarding how improvements in these criteria influence average-case performance under reasonable stochastic assumptions on prediction errors.

ACKNOWLEDGMENTS

This research was supported in part by the French National Research Agency (ANR) in the framework of the PEPR IA FOUNDRY project (ANR-23-PEIA-0003) and through the grant DOOM ANR23-CE23-0002. It was also funded by the European Union (ERC, Ocean, 101071601). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Spyros Angelopoulos. Online search with a hint. *Information and Computation*, 295:105091, 2023.

Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. Online computation with untrusted advice. *arXiv preprint arXiv:1905.05655*, 2019.

Spyros Angelopoulos, Christoph Dürr, Alex Elenter, and Yanni Lefki. Overcoming brittleness in pareto-optimal learning-augmented algorithms. *arXiv preprint arXiv:2408.04122*, 2024a.

Spyros Angelopoulos, Thomas Lidbetter, and Konstantinos Panagiotou. Search games with predictions. *arXiv preprint arXiv:2401.01149*, 2024b.

Antonios Antoniadis, Christian Coester, Marek Eliás, Adam Polak, and Bertrand Simon. Learning-augmented dynamic power management with multiple states via new ski rental bounds. *Advances in Neural Information Processing Systems*, 34:16714–16726, 2021.

Antonios Antoniadis, Joan Boyar, Marek Eliás, Lene Monrad Favrholdt, Ruben Hoeksma, Kim S Larsen, Adam Polak, and Bertrand Simon. Paging with succinct predictions. In *International Conference on Machine Learning*, pages 952–968. PMLR, 2023a.

Antonios Antoniadis, Christian Coester, Marek Eliáš, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. *ACM Transactions on Algorithms*, 19(2):1–34, 2023b.

Luis Antunes and Lance Fortnow. Worst-case running times for average-case algorithms. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 298–303. IEEE, 2009.

Ricardo A Baezayates, Joseph C Culberson, and Gregory JE Rawlins. Searching in the plane. *Information and computation*, 106(2):234–252, 1993.

Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. *Advances in Neural Information Processing Systems*, 33:20083–20094, 2020.

Anatole Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.

Anatole Beck and Donald J Newman. Yet more on the linear search problem. *Israel journal of mathematics*, 8(4):419–429, 1970.

Ziyad Benomar and Christian Coester. Learning-augmented priority queues. *arXiv preprint arXiv:2406.04793*, 2024.

Ziyad Benomar and Vianney Perchet. Advice querying under budget constraint for online algorithms. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Ziyad Benomar and Vianney Perchet. Non-clairvoyant scheduling with partial predictions. In *Forty-first International Conference on Machine Learning*, 2024.

Jakub Chlkedowski, Adam Polak, Bartosz Szabucki, and Konrad Tomasz .Zolna. Robust learning-augmented caching: An experimental study. In *International Conference on Machine Learning*, pages 1920–1930. PMLR, 2021.

Huda Chuangpishit, Konstantinos Georgiou, and Preeti Sharma. Average case-worst case tradeoffs for evacuating 2 robots from the disk in the face-to-face model. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 62–82. Springer, 2018.

Vincent Cohen-Addad, Tommaso d'Orsi, Anupam Gupta, Euiwoong Lee, and Debmalya Panigrahi. Max-cut with ε-accurate predictions. *arXiv preprint arXiv:2402.18263*, 2024.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, Ali Vakilian, and Nikos Zarifis. Learning online algorithms with distributional advice. In *International Conference on Machine Learning*, pages 2687–2696. PMLR, 2021.

Paul Dütting, Silvio Lattanzi, Renato Paes Leme, and Sergei Vassilvitskii. Secretaries with advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 409–429, 2021.

Ran El-Yaniv, Amos Fiat, Richard M Karp, and Gordon Turpin. Optimal search and one-way trading online algorithms. *Algorithmica*, 30:101–139, 2001.

Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *International Conference on Machine Learning*, pages 2319–2327. PMLR, 2019.

Anupam Gupta, Debmalya Panigrahi, Bernardo Subercaseaux, and Kevin Sun. Augmenting online algorithms with ε-accurate predictions. *Advances in neural information processing systems*, 35:2115–2127, 2022.

Monika Henzinger, Barna Saha, Martin P Seybold, and Christopher Ye. On the complexity of algorithms with predictions for dynamic graph problems. *arXiv preprint arXiv:2307.16771*, 2023.

Anna R Karlin, Mark S Manasse, Larry Rudolph, and Daniel D Sleator. Competitive snoopy caching. *Algorithmica*, 3:79–119, 1988.

Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11 (6):542–571, 1994.

Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 international conference on management of data*, pages 489–504, 2018.

Alexandra Anna Lassota, Alexander Lindermayr, Nicole Megow, and Jens Schlöter. Minimalistic predictions to schedule jobs with online precedence constraints. In *International Conference on Machine Learning*, pages 18563–18583. PMLR, 2023.

Honghao Lin, Tian Luo, and David Woodruff. Learning augmented binary search trees. In *International Conference on Machine Learning*, pages 13431–13440. PMLR, 2022.

Thodoris Lykouris and Sergei Vassilvtiskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning*, pages 3296–3305. PMLR, 2018.

Nadav Merlis, Hugo Richard, Flore Sentenac, Corentin Odic, Mathieu Molina, and Vianney Perchet. On preemption and learning in stochastic scheduling. In *International Conference on Machine Learning*, pages 24478–24516. PMLR, 2023.

Chris Peikert and Alon Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 478–487, 2007.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. *Advances in Neural Information Processing Systems*, 31, 2018.

Leslie Rice, Anna Bair, Huan Zhang, and J Zico Kolter. Robustness between the worst and average case. *Advances in Neural Information Processing Systems*, 34:27840–27851, 2021.

Alexander Robey, Luiz Chamon, George J Pappas, and Hamed Hassani. Probabilistically robust learning: Balancing average and worst-case performance. In *International Conference on Machine Learning*, pages 18667–18686. PMLR, 2022.

Yongho Shin, Changyeol Lee, Gukryeol Lee, and Hyung-Chan An. Improved learning-augmented algorithms for the multi-option ski rental problem via best-possible competitive analysis. *arXiv preprint arXiv:2302.06832*, 2023.

Bo Sun, Russell Lee, Mohammad Hajiesmaili, Adam Wierman, and Danny Tsang. Pareto-optimal learning-augmented algorithms for online conversion problems. *Advances in Neural Information Processing Systems*, 34:10339–10350, 2021.

László Szirmay-Kalos and Gábor Márton. Worst-case versus average case complexity of ray-shooting. *Computing*, 61(2):103–131, 1998.

Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. *Advances in Neural Information Processing Systems*, 33:8042–8053, 2020.

Carsten Witt. Worst-case and average-case approximations by simple randomized search heuristics. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 44–56. Springer, 2005.

Ali Zeynali, Shahin Kamali, and Mohammad Hajiesmaili. Robust learning-augmented dictionaries. In *Forty-first International Conference on Machine Learning*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No] Not relevant in our experiments.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

On Tradeoffs in Learning-Augmented Algorithms: Proofs

1 SMOOTH ALGORITHM FOR LINE SEARCH

1.1 Proof of Lemma 2.1

Proof. By definition of j, we have $1 \leq b^{2j} y/x < b^2$, and recalling that $y = b^{k_y}/\gamma_y$, we obtain

$$\frac{b^{k_y+2j-2}}{\gamma_y} < x \leq \frac{b^{k_y+2j}}{\gamma_y} .$$

Using these inequalities, we deduce that the algorithm finds the target at iteration $k_y + 2j$. Indeed, as $y > 0$, at iteration $k_y + 2j - 2$, the searcher travels along the positive branch until reaching the position b^{k_y+2j-2}/γ_y and returns to the origin, then it explores the negative branch at iteration $k_y + 2j - 1$. The total distance traveled up to that point is $2\sum_{i=0}^{k_y+2j-1} b^i/\gamma_y$. At iteration $k_y + 2j$, since $x \leq b^{k_y+2j}/\gamma_y$, the algorithm finds the target and terminates. The total traveled distance is therefore

$$\begin{aligned}
\mathsf{A}_b^{\mathrm{LS}}(x, y) &= x + 2\sum_{i=0}^{k_y+2j-1} \frac{b^i}{\gamma_y} \\
&= x + \frac{2}{b-1}\left(\frac{b^{k_y+2j}}{\gamma_y} - \frac{1}{\gamma_y}\right) \\
&= x + \frac{2b^2}{b-1}\left(b^{2(j-1)} y - \frac{1}{\gamma_y b^2}\right) ,
\end{aligned}$$

where we used in the last inequality that $y = b^{k_y}/\gamma_y$. Finally, using that $\gamma_y \geq 1$, we deduce that

$$1 + \frac{2b^2}{b-1}\left(b^{2(j-1)}\frac{y}{x} - \frac{1}{b^2 x}\right) \leq \frac{\mathsf{A}_b^{\mathrm{LS}}(x, y)}{x} \leq 1 + \frac{2b^2}{b-1} \cdot b^{2(j-1)}\frac{y}{x} .$$

□

1.2 Proof of Proposition 2.2

Proof. Let $\varepsilon > 0$, $x \geq 1$, and $y \in [\max\{\frac{x}{b^2}, (1-\varepsilon)x\}, x)$. Since $\frac{x}{b^2} \leq y < x$, the variable j from Lemma 2.1 equals 1, which yields

$$\frac{\mathsf{A}_b^{\mathrm{LS}}(x, y)}{x} \geq 1 + \frac{2b^2}{b-1}\left(\frac{y}{x} - \frac{1}{b^2 x}\right) ,$$

and taking y arbitrarily close to x gives

$$\sup_{y : |x-y| \leq \varepsilon x} \frac{\mathsf{A}_b^{\mathrm{LS}}(x, y)}{x} \geq 1 + \frac{2b^2}{b-1}\left(1 - \frac{1}{b^2 x}\right) = \left(1 + \frac{2b^2}{b-1}\right) - \frac{2}{(b-1)x} .$$

Finally, given that $1 + \frac{2b^2}{b-1}$ is also an upper bound on $\mathsf{A}_b^{\mathrm{LS}}(x, y)$, taking arbitrarily large concludes the proof. □

1.3 Proof of Theorem 2.3

Proof. The online algorithm $\mathsf{A}_b^{\mathrm{LS}}$ has a robustness of $1 + \frac{2b^2}{b-1}$. This guarantee remains true with any arbitrary prediction, in particular with $\tilde{y}$, which gives almost surely that

$$\frac{\mathsf{A}_b^{\mathrm{LS}}(x, \tilde{y})}{x} \leq 1 + \frac{2b^2}{b-1} \ .$$

Regarding the consistency and smoothness of the algorithm, we give separate proofs depending on the position of y relative to x.

If $y \geq x$, then $y = x + \eta$, the perturbed prediction $\tilde{y}$ is larger than x almost surely, and $j = \lceil \frac{\ln(x/\tilde{y})}{2\ln(b)} \rceil \leq 0$. Thus, Lemma 2.1 with x and $\tilde{y}$ yields

$$\begin{aligned}
\frac{\mathsf{A}_b^{\mathrm{LS}}(x, \tilde{y})}{x} &\leq 1 + \frac{2}{b-1} \cdot \frac{\tilde{y}}{x} \\
&= 1 + \frac{2(1+\rho\xi)}{b-1} \cdot \left(1 + \frac{\eta}{x}\right) \\
&= \frac{b - 1 + 2\rho\xi}{b-1} + \frac{2(1+\rho\xi)}{b-1} \cdot \frac{\eta}{x}
\end{aligned}$$

and using that $\mathbb{E}[\xi] = 1$ gives in expectation

$$\frac{\mathbb{E}_\xi[\mathsf{A}_b^{\mathrm{LS}}(x, \tilde{y})]}{x} \leq \frac{b - 1 + 2\rho}{b-1} + \frac{2(1+\rho)}{b-1} \cdot \frac{\eta}{x} \ .$$

On the other hand, if $y < x$, let first prove that the upper bound of the theorem exceeds the robustness bound when $y \leq x/b^2$, hence it is true almost surely in that case. For $y \leq x/b^2$, the absolute prediction error satisfies $\eta/x = 1 - y/x \geq 1 - 1/b^2$, and given that $\rho \leq 1$, we obtain

$$\begin{aligned}
\frac{b+1+2\rho}{b-1} + \frac{4(b+1)}{\rho} \cdot \frac{\eta}{x} &\geq \frac{b+1}{b-1} + 4(b+1)\left(1 - \frac{1}{b^2}\right) \\
&= 1 + \frac{2b^2}{b-1}\left(\frac{1}{b^2} + 2\left(1 - \frac{1}{b^2}\right)^2\right) \\
&= 1 + \frac{2b^2}{b-1}\left(1 + \left(1 - \frac{3}{b^2} + \frac{2}{b^4}\right)\right) \\
&\geq 1 + \frac{2b^2}{b-1} \\
&\geq \frac{\mathsf{A}_b^{\mathrm{LS}}(x, \tilde{y})}{x} \qquad \text{(w.p. 1)} \ .
\end{aligned}$$

Consequently, in the rest of the proof, we focus on showing the claimed result when $y \in (\frac{x}{b^2}, x)$. In that case, the random variable $\tilde{y}$ takes values in $(x/b^2, \infty)$. By Lemma 2.1, using that $\tilde{y} = (1+\rho\xi)y$, we obtain that

$$\begin{aligned}
\frac{\mathsf{A}_b^{\mathrm{LS}}(x, \tilde{y})}{x} &\leq 1 + \frac{2\tilde{y}/x}{b-1} \times \begin{cases} b^2 & \text{if } \tilde{y} \in (x/b^2, x) \\ 1 & \text{if } \tilde{y} \geq x \end{cases} \\
&= 1 + \frac{2y/x}{b-1}\left((1+\rho\xi)\mathbb{1}_{\xi \geq \frac{1}{\rho}(\frac{x}{y}-1)} + b^2(1+\rho\xi)\mathbb{1}_{\xi < \frac{1}{\rho}(\frac{x}{y}-1)}\right) \\
&= 1 + \frac{2y/x}{b-1}\left(1 + \rho\xi + (b^2-1)(1+\rho\xi)\mathbb{1}_{\xi < \frac{1}{\rho}(\frac{x}{y}-1)}\right) \ ,
\end{aligned}$$

which gives in expectation

$$\frac{\mathbb{E}[\mathsf{A}_b^{\mathrm{LS}}(x, \tilde{y})]}{x} \leq 1 + \frac{2y/x}{b-1}\left(1 + \rho + (b^2-1)\big(\Pr(\xi < \tfrac{1}{\rho}(\tfrac{x}{y} - 1)) + \rho\mathbb{E}\big[\xi\mathbb{1}_{\xi < \frac{1}{\rho}(\frac{x}{y}-1)}\big]\big)\right) \ . \tag{3}$$

For all $s \geq 0$, it holds that $\Pr(\xi < s) = 1 - \frac{1}{(1+s)^2} = \frac{s^2+2s}{(1+s)^2}$, and

$$\begin{aligned}
\mathbb{E}[\xi \mathbb{1}_{\xi<s}] &= \int_0^\infty \Pr(\xi \mathbb{1}_{\xi<s} \geq u)\mathrm{d}u \\
&= \int_0^s \Pr(\xi \in [u,s))\mathrm{d}u \\
&= \int_0^s \Pr(\xi \geq u)\mathrm{d}u - \int_0^s \Pr(\xi \geq s)\mathrm{d}u \\
&= \int_0^s \frac{\mathrm{d}u}{(1+u)^2} - \frac{s}{(1+s)^2} \\
&= 1 - \frac{1}{1+s} - \frac{s}{(1+s)^2} \\
&= \frac{s^2}{(1+s)^2} \; .
\end{aligned}$$

Therefore, using that $\rho \leq 1$ then $s > 0$, we obtain

$$\Pr(\xi < s) + \rho\mathbb{E}\big[\xi \mathbb{1}_{\xi<s}\big] = \frac{2s + (1+\rho)s^2}{(1+s)^2} \leq \frac{2s+2s^2}{(1+s)^2} = \frac{2s}{1+s} \leq 2s \; .$$

Substituting into (3) with $s = \frac{1}{\rho}(\frac{x}{y} - 1)$ yields

$$\begin{aligned}
\frac{\mathbb{E}[\mathsf{A}_b^{\mathrm{LS}}(x,\tilde{y})]}{x} &\leq 1 + \frac{2y/x}{b-1}\left(1 + \rho + \frac{2(b^2-1)}{\rho}\Big(\frac{x}{y} - 1\Big)\right) \\
&= \frac{b - 1 + 2(1+\rho)y/x}{b-1} + \frac{4(b+1)}{\rho} \cdot \frac{x-y}{x} \\
&\leq \frac{b+1+2\rho}{b-1} + \frac{4(b+1)}{\rho} \cdot \frac{\eta}{x} \; ,
\end{aligned}$$

which concludes the proof. □

2 SMOOTH ALGORITHM FOR ONE-MAX SEARCH

2.1 Proof of Lemma 3.1

Proof. If $y \in [1, \frac{1}{c})$, then $\Phi(\lambda, y) = 1/c$. If $p^* < 1/c$ then all the observed prices are below the threshold, and the algorithm selects p_n, which is 1 in the worst case, hence

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \geq \frac{1}{p^*} \geq c \; .$$

On the other hand, if $p^* \geq 1/c$, then the value selected by the algorithm is at least $1/c$ and

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \geq \frac{1/c}{p^*} \geq \frac{y}{p^*} = 1 - \frac{p^* - y}{p^*} = 1 - \frac{\eta}{p^*} \; .$$

We deduce from both cases that

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \geq \min\left(c \; , \; 1 - \frac{\eta}{p^*}\right) \geq c - \frac{c \cdot \eta}{p^*} \; ,$$

where the last inequality holds because each term is at most 1. □

2.2 Proof of Lemma 3.2

Proof. Assume that $y \in [1/c, 1/r)$, then the acceptance threshold is $\Phi(\lambda, y) = \lambda/r + (1-\lambda)cy$. If $p^* \geq \Phi(\lambda, y)$ then the price selected by the algorithm is at least $\Phi(\lambda, y)$, which gives that

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \geq \frac{\Phi(\lambda,y)}{p^*} = \frac{\lambda}{rp^*} + (1-\lambda)c\frac{y}{p^*} \; .$$

We have from (2) that $p^* \leq \theta = 1/(cr)$, hence $\frac{\lambda}{rp^*} \geq \lambda c$. Additionally, $y \geq p^* - \eta$, which gives

$$\begin{aligned} \frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} &\geq \lambda c + (1-\lambda)c\Big(1 - \frac{\eta}{p^*}\Big) \\ &= c - (1-\lambda)\frac{c \cdot \eta}{p^*} \; . \end{aligned} \tag{4}$$

On the other hand, if $p^* < \Phi(\lambda, y)$, then the value selected by the algorithm can be as low as 1 in the worst case, thus

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \geq \frac{1}{p^*} = \frac{1}{\Phi(\lambda,y)} \cdot \frac{\Phi(\lambda,y)}{p^*} \; . \tag{5}$$

Using (2), we can write

$$\begin{aligned} \Phi(\lambda,y) &= \frac{\lambda}{r} + (1-\lambda)cy \\ &= \frac{1}{c} - (1-\lambda) + (1-\lambda)cy \\ &= \frac{1}{c} + (1-\lambda)c\Big(y - \frac{1}{c}\Big) \; , \end{aligned} \tag{6}$$

hence

$$y - \Phi(\lambda,y) = \big(1 - (1-\lambda)c\big)\Big(y - \frac{1}{c}\Big) \; ,$$

and it follows from $p^* < \Phi(\lambda, y)$ that

$$y - \frac{1}{c} = \frac{y - \Phi(\lambda,y)}{1-(1-\lambda)c} \leq \frac{y - p^*}{1-(1-\lambda)c} = \frac{\eta}{1-(1-\lambda)c} \; .$$

Substituting into (6) then using (2) yields

$$\begin{aligned} \Phi(\lambda,y) &\leq \frac{1}{c} + \frac{(1-\lambda)c}{1-(1-\lambda)c} \cdot \eta \\ &= \frac{1}{c} + \frac{(1-\lambda)}{\frac{1}{c} - (1-\lambda)} \cdot \eta \\ &= \frac{1}{c} + \Big(\frac{1}{\lambda} - 1\Big) r \cdot \eta \; . \end{aligned}$$

Finally, we use this upper bound on $\Phi(\lambda, y)$ in (5) and obtain that

$$\begin{aligned} \frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} &= \frac{1}{\Phi(\lambda,y)} \cdot \frac{\Phi(\lambda,y)}{p^*} \\ &\geq \frac{1}{\frac{1}{c} + (\frac{1}{\lambda} - 1)r \cdot \eta} \cdot \frac{\Phi(\lambda,y)}{p^*} \\ &= \frac{c}{1 + (\frac{1}{\lambda} - 1)rc \cdot \eta} \cdot \frac{\Phi(\lambda,y)}{p^*} \\ &\geq c \cdot \frac{\Phi(\lambda,y)}{p^*} \big(1 - (\tfrac{1}{\lambda} - 1)rc \cdot \eta\big) \\ &\geq c - \Phi(\lambda,y)(\tfrac{1}{\lambda} - 1)rc^2 \cdot \frac{\eta}{p^*} \; . \end{aligned}$$

The penultimate line follows from the inequality $\frac{1}{1+u} \geq 1-u$, which holds for all $u \geq 0$, and the last line from the assumption $p^* \leq \Phi(\lambda, y)$. Given that $y \leq 1/r$, we have

$$\Phi(\lambda, y) = \frac{\lambda}{r} + (1-\lambda)cy \leq \frac{1}{r}\big(\lambda + (1-\lambda)c\big) \leq \frac{1}{r} ,$$

hence

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \geq c - (\tfrac{1}{\lambda} - 1)c^2 \cdot \frac{\eta}{p^*} = c - (1-\lambda)\frac{c}{\lambda} \cdot \frac{c \cdot \eta}{p^*} . \tag{7}$$

From (4) and (7) we deduce that, if $y \in [1/c, 1/r]$ then

$$\begin{aligned}\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} &\geq \min\left(c - (1-\lambda)\frac{c \cdot \eta}{p^*}\ ,\ c - (1-\lambda)\frac{c}{\lambda} \cdot \frac{c \cdot \eta}{p^*}\right) \\ &= c - (1-\lambda)\max(1, \tfrac{c}{\lambda})\frac{c \cdot \eta}{p^*} .\end{aligned}$$

□

2.3 Proof of Proposition 3.3

Proof. Let $\varepsilon > 0$, and consider the instance $p = (\frac{1}{r} - \delta, 1)$ of size 2, with $\delta \in (0, \min(\frac{1}{r} - 1, \frac{\varepsilon}{r})]$, so that p_1 remains at least 1 and $|p_1 - 1/r| \leq \varepsilon p_1$. Given a prediction $y = 1/r$ of $p^* = p_1$, $\mathsf{A}_\lambda^{\mathrm{OM}}$ sets an acceptance threshold of $1/r > p_1$, hence the algorithm rejects p_1 and accepts $p_2 = 1$ as it is the last price in the sequence p. It follows that

$$\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} = \frac{1}{1/r - \delta} ,$$

hence

$$\inf_{p,y:\frac{|p^*-y|}{p^*} \leq \varepsilon} \frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*} \leq \lim_{\delta \to 0} \frac{1}{1/r - \delta} = r.$$

Since, by definition, r is also a lower bound of $\frac{\mathsf{A}_\lambda^{\mathrm{OM}}(p,y)}{p^*}$, we deduce that the inequality above is equality, which concludes the proof. □

2.4 Proof of Theorem 3.4

The consistency/smoothness bounds for $y \in [1, 1/r)$ are proved in Lemmas 3.1 and 3.2, and the robustness in that case is $r \geq \left(\frac{1-e^{-\rho}}{\rho}\right) r$ because $\mathsf{A}_{\lambda,\rho}^{\mathrm{OM}}$ is identical to $\mathsf{A}_\lambda^{\mathrm{OM}}$. Therefore, it only remains to prove the claimed bounds for $y \in [1/r, \theta)$. We demonstrate in Lemma 2.1 the consistency and smoothness factor of the algorithm, while the robustness is proved in Lemma 2.2.

Lemma 2.1 (Consitency-Smoothness)**.** *For any sequence of prices p, if $y \in [\frac{1}{r}, \theta]$, then*

$$\frac{\mathbb{E}_U[A_{\lambda,\rho}^{\mathrm{OM}}(p,y)]}{p^*} \geq \Big(\frac{1-e^{-\rho}}{\rho}\Big)c - \Big(\frac{c-r}{\rho}\Big)\frac{\eta}{p^*} .$$

Proof. Let $y \in [\frac{1}{r}, \theta]$, hence $\tilde{\Phi}(\lambda, \rho, y) = e^{-\rho U}/r$, where U is a uniform random variable in $[0,1]$. If $p^* \geq \tilde{\Phi}(\lambda, \rho, y)$, then the algorithm has a reward of at least $\tilde{\Phi}(\lambda, \rho, y)$, and by (2) we obtain

$$\frac{\mathsf{A}_{\lambda,\rho}^{\mathrm{OM}}(p,y)}{p^*} \geq \frac{\tilde{\Phi}(\lambda,\rho,y)}{p^*} \geq \frac{e^{-\rho U}}{\theta r} = e^{-\rho U} c ,$$

and if $p^* < \tilde{\Phi}(\lambda, \rho, y)$ then

$$\frac{\mathsf{A}_{\lambda,\rho}^{\mathrm{OM}}(p,y)}{p^*} \geq \frac{1}{\tilde{\Phi}(\lambda,\rho,y)} = e^{\rho U} r .$$

Let us denote by $s = -\ln(rp^*)$. Observing that

$$p^* \geq \tilde{\Phi}(\lambda, \rho, y) \iff p^* \geq e^{-\rho U}/r \iff e^{\rho U} \geq \frac{1}{rp^*} \iff U \geq \frac{-\ln(rp^*)}{\rho} = \frac{s}{\rho} \ ,$$

we deduce that

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq c\mathbb{E}[e^{-\rho U}\mathbb{1}_{U\geq\frac{s}{\rho}}] + r\mathbb{E}[e^{\rho U}\mathbb{1}_{U<\frac{s}{\rho}}] \ . \tag{8}$$

Assume that $p^* \in [\frac{e^{-\rho}}{r}, \frac{1}{r}]$, i.e. $s \in [0, \rho]$. The two terms on the right-hand side above can be computed easily

$$\mathbb{E}[e^{-\rho U}\mathbb{1}_{U\geq\frac{s}{\rho}}] = \int_{s/\rho}^{1} e^{-\rho u}\mathrm{d}u = \left[\frac{-e^{-\rho u}}{\rho}\right]_{s/\rho}^{1} = \frac{e^{-s} - e^{-\rho}}{\rho} \ ,$$
$$\mathbb{E}[e^{\rho U}\mathbb{1}_{U<\frac{s}{\rho}}] = \int_{0}^{s/\rho} e^{\rho u}\mathrm{d}u = \left[\frac{e^{\rho u}}{\rho}\right]_{0}^{s/\rho} = \frac{e^{s} - 1}{\rho} \ ,$$

and we obtain by substituting into (8) that

$$\begin{aligned}\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} &\geq \left(\frac{e^{-s} - e^{-\rho}}{\rho}\right)c + \left(\frac{e^{s} - 1}{\rho}\right)r \qquad (9)\\ &= \left(\frac{1 - e^{-\rho}}{\rho}\right)c + \left(\frac{1 - e^{s}}{\rho}\right)ce^{-s} + \left(\frac{e^{s} - 1}{\rho}\right)r\\ &= \left(\frac{1 - e^{-\rho}}{\rho}\right)c - \frac{ce^{-s} - r}{\rho}(e^{s} - 1)\\ &\geq \left(\frac{1 - e^{-\rho}}{\rho}\right)c - \frac{c - r}{\rho}(e^{s} - 1) \ .\end{aligned}$$

By definition of s, we have

$$e^{s} - 1 = \frac{1}{rp^*} - 1 = \frac{1/r - p^*}{p^*} \leq \frac{y - p^*}{p^*} = \frac{\eta}{p^*} \ ,$$

which yields

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq \left(\frac{1 - e^{-\rho}}{\rho}\right)c - \left(\frac{c - r}{\rho}\right)\frac{\eta}{p^*} \ .$$

which corresponds to the consistency/smoothness bound stated in the theorem. However, we only proved it for $p^* \in [\frac{e^{-\rho}}{r}, \frac{1}{r}]$. We demonstrate in the following that the bound remains true if p^* is outside that interval. If $p^* \geq 1/r$, i.e. $s \leq 0$, then (8) gives

$$\begin{aligned}\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} &\geq c\mathbb{E}[e^{-\rho U}] = \left(\frac{1 - e^{-\rho}}{\rho}\right)c \qquad (10)\\ &\geq \left(\frac{1 - e^{-\rho}}{\rho}\right)c - \left(\frac{c - r}{\rho}\right)\frac{\eta}{p^*} \ .\end{aligned}$$

On the other hand, if $p^* \leq e^{-\rho}/r$, i.e. $s \geq \rho$, then (8) again gives

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq r\mathbb{E}[e^{\rho U}\mathbb{1}_{U<1}] = \left(\frac{e^{\rho} - 1}{\rho}\right)r \ , \tag{11}$$

and it holds that

$$\frac{\eta}{p^*} = \frac{y}{p^*} - 1 \geq \frac{1/r}{e^{-\rho}/r} - 1 = e^{\rho} - 1 \geq 1 - e^{-\rho} \ ,$$

hence

$$\begin{aligned}\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} &\geq \left(\frac{e^\rho - 1}{\rho}\right) r\\ &\geq \left(\frac{1-e^{-\rho}}{\rho}\right) r\\ &= \left(\frac{1-e^{-\rho}}{\rho}\right) c - \left(\frac{c-r}{\rho}\right)(1-e^{-\rho})\\ &\geq \left(\frac{1-e^{-\rho}}{\rho}\right) c - \left(\frac{c-r}{\rho}\right)\frac{\eta}{p^*} .\end{aligned}$$

The claimed lower bound is therefore true for all values of p^*. □

Lemma 2.2 (Robustness)**.** *For any sequence of prices p, if $y \in [\frac{1}{r}, \theta]$, then*

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq \left(\frac{1-e^{-\rho}}{\rho}\right) r .$$

Proof. Consider a sequence of prices p. Using Inequality (10) from the proof of Lemma 2.1, if $p^* \geq 1/r$ then

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq \left(\frac{1-e^{-\rho}}{\rho}\right) c \geq \left(\frac{1-e^{-\rho}}{\rho}\right) r ,$$

and if $p^* \leq e^{-\rho}/r$, then by Inequality (11) we have

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq \left(\frac{1-e^{-\rho}}{\rho}\right) r .$$

It remains to prove the same lower bound when $p^* \in (\frac{e^{-\rho}}{r}$. Assume that that is the case, and let $s = -\ln(rp^*) \in [0,\rho]$. Inequality (9) gives that

$$\begin{aligned}\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} &\geq \left(\frac{e^{-s} - e^{-\rho}}{\rho}\right) c + \left(\frac{e^s - 1}{\rho}\right) r\\ &= \frac{r}{\rho}\left(\frac{c}{r}e^{-s} + e^s - \frac{c}{r}e^{-\rho} - 1\right) .\end{aligned}$$

The function $s \mapsto \frac{c}{r}e^{-s} + e^s$ is minimal on $\mathbb{R}$ for $e^s = \sqrt{\frac{c}{r}}$. Therefore, on the interval $[0,\rho]$, it is minimal for $e^s = \min(e^\rho, \sqrt{\frac{c}{r}})$. If $\sqrt{\frac{c}{r}} \geq e^\rho$ then

$$\begin{aligned}\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} &\geq \frac{r}{\rho}\left(\frac{c}{r}e^{-\rho} + e^\rho - \frac{c}{r}e^{-\rho} - 1\right)\\ &= \left(\frac{e^\rho - 1}{\rho}\right) r\\ &\geq \left(\frac{1-e^{-\rho}}{\rho}\right) r .\end{aligned}$$

On the other hand, if $\sqrt{\frac{c}{r}} < e^\rho$ then

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq \frac{r}{\rho}\left(2\sqrt{\frac{c}{r}} - \frac{c}{r}e^{-\rho} - 1\right) .$$

The function $u \mapsto -e^{-\rho}u^2 + 2u - 1$ is non-decreasing on $[1, e^\rho]$, hence, since $\sqrt{c/r \in [1, e^\rho]}$ then

$$\frac{\mathbb{E}_U[\mathsf{A}^{\mathrm{OM}}_{\lambda,\rho}(p,y)]}{p^*} \geq \frac{r}{\rho}\left(2 - e^{-\rho} - 1\right) = \left(\frac{1-e^{-\rho}}{\rho}\right) r ,$$

which concludes the proof.

□

3 AVERAGE-CASE ANALYSIS IN SKI-RENTAL

3.1 Proof of Theorem 4.1

Proof. For simplicity, let us denote by $\beta = (1 + \rho(\frac{1}{\lambda} - 1))$. Note that $1 \leq \beta \leq \frac{1}{\lambda}$, and recall that $\min(x, b) = \min(x, b)$.

Robustness. We first prove the robustness bound. If $y \geq b$:

- if $x < \lambda b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = x = \min(x, b)$,
- if $\lambda b \leq x < b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = (1 + \lambda)b \leq (1 + \frac{1}{\lambda})x = (1 + \frac{1}{\lambda}) \min(x, b)$,
- if $b \leq x$, then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = (1 + \lambda)b = (1 + \lambda) \min(x, b) \leq (1 + \frac{1}{\lambda}) \min(x, b)$.

On the other hand, if $y < b$:

- if $x < b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = x = \min(x, b)$,
- if $b \leq x < \beta b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = x < \beta b = \beta \min(x, b) \leq (1 + \frac{1}{\lambda}) \min(x, b)$,
- if $x \geq \beta x$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = (1 + \beta)b = (1 + \beta) \min(x, b) \leq (1 + \frac{1}{\lambda}) \min(x, b)$.

In all the cases, it always holds that $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) \leq (1 + \frac{1}{\lambda}) \min(x, b)$.

Consistency/Smoothness. Let us first consider the case of $y \geq b$.

- if $x < \lambda b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = x = \min(x, b)$,
- if $\lambda b \leq x < b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = (1 + \lambda)b \leq (1 + \lambda)y \leq (1 + \lambda) \min(x, b) + (1 + \lambda)\eta$,
- if $b \leq x$, then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = (1 + \lambda)b = (1 + \lambda) \min(x, b)$.

In the case of $y < b$, we obtain that

- if $x < b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) = x = \min(x, b)$,
- if $b \leq x < \beta b$ then

$$\begin{aligned}
\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) &= x \leq y + \eta \\
&\leq b + \eta = (1 + \lambda)b - \lambda b + \eta \\
&\leq (1 + \lambda) \min(x, b) + (1 - \tfrac{\lambda}{\beta})\eta \\
&\leq (1 + \lambda) \min(x, b) + \frac{\beta - \lambda}{\beta - 1}\eta
\end{aligned}$$

  where we used in the penultimate inequality that $\eta = x - y \leq x \leq \beta b$.

- if $x \geq \beta x$ then

$$\begin{aligned}
\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x, y) &= (1 + \beta)b = (1 + \lambda)b + (\beta - \lambda)b \\
&= (1 + \lambda) \min(x, b) + (\beta - \lambda)b \\
&\leq (1 + \lambda) \min(x, b) + \frac{\beta - \lambda}{\beta - 1}\eta \,,
\end{aligned}$$

  where we used in the last inequality that $\eta = x - y \geq (\beta - 1)b$.

All in all, we deduce that

$$\forall x, y : \quad \mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) \le (1+\lambda)\min(x,b) + \max\left(1+\lambda, \frac{\beta-\lambda}{\beta-1}\right)\eta \ ,$$

and by definition of β we have

$$\frac{\beta-\lambda}{\beta-1} = \frac{(\rho+\lambda)(\frac{1}{\lambda}-1)}{\rho(\frac{1}{\lambda}-1)} = 1 + \frac{\lambda}{\rho} \ ,$$

hence

$$\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) \le (1+\lambda)\min(x,b) + \left(1+\frac{\lambda}{\rho}\right)\eta \ ,$$

which concludes the proof. ☐

3.2 Proof of Theorem 4.2

Proof. Let us denote by $\beta = 1 + \rho(\frac{1}{\lambda}-1)$, and assume that $\Pr(\mathbb{1}_{y\ge b} = \mathbb{1}_{x \ge b}) = q$. If $x \ge b$, then

- with probability q: $y \ge b$ and
$$\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) = (1+\lambda)b = (1+\lambda)\min(x,b) \ ,$$
- with probability $1-q$: $y < b$ and
  - if $x < \beta b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) = x \le \beta b = \beta \min(x,b)$,
  - if $x \ge \beta b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) = (1+\beta)b = (1+\beta)\min(x,b)$,

hence we have for $x \ge b$ that

$$\begin{aligned}\frac{\mathbb{E}_y[\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y)]}{\min(x,b)} &\le q(1+\lambda) + (1-q)(1+\beta)\\ &= 1 + \lambda q + \beta(1-q)\\ &= 1 + \lambda q + (1 + (\tfrac{1}{\lambda}-1)\rho)(1-q)\\ &= 2 + (\tfrac{1}{\lambda}-1)((1-q)\rho - q\lambda) \ . \end{aligned} \tag{12}$$

On the other hand, for $x < b$

- with probability q: $y < b$ and $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) = x = \min(x,b)$
- with probability $1-q$: $y > b$ and
  - if $x < \lambda b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) = x = \min(x,b)$,
  - if $x < \lambda b$ then $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y) = (1+\lambda)b \le (1+\frac{1}{\lambda})x = (1+\frac{1}{\lambda})\min(x,b)$,

which gives for $x < b$ that

$$\frac{\mathbb{E}_y[\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y)]}{\min(x,b)} \le q + (1-q)(1+\tfrac{1}{\lambda}) = 1 + \frac{1-q}{\lambda} \ . \tag{13}$$

We deduce from (12) and (13) that

$$\forall x : \quad \frac{\mathbb{E}_y[\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}(x,y)]}{\min(x,b)} \le \max\left(2 + (\tfrac{1}{\lambda}-1)\big((1-q)\rho - q\lambda\big), 1 + \frac{1-q}{\lambda}\right) \ .$$

Finally, observe that the right-hand term is a non-increasing function of q, hence the upper bound holds also if $\Pr(\mathbb{1}_{y\ge b} = \mathbb{1}_{x\ge b}) \ge q$. ☐

3.3 Proof of Corollary 4.3

Proof. For all $q \in [\frac{1}{2}, 1]$ and $\rho \in [0,1]$, the upper bound of Theorem 4.2 is non-decreasing with respect to ρ, hence the optimal choice of ρ is 0. With $\rho = 0$, let us examine for which value of λ the two terms in the maximum of the upper bound in Theorem 4.2 are equal. We have for all $\lambda \in [0,1]$ the equivalences

$$\begin{aligned}
2 - (1-\lambda)q = 1 + \frac{1-q}{\lambda} &\iff 1 - (1-\lambda)q = \frac{1-q}{\lambda} \\
&\iff \tfrac{\lambda}{q} - \lambda(1-\lambda) = \tfrac{1}{q} - 1 \\
&\iff \lambda^2 + \left(\tfrac{1}{q} - 1\right)\lambda - \left(\tfrac{1}{q} - 1\right) = 0 \\
&\iff \lambda = \tfrac{1}{2}\sqrt{\left(\tfrac{1}{q} - 1\right)\left(3 + \tfrac{1}{q}\right)} - \tfrac{1}{2}\left(\tfrac{1}{q} - 1\right) \ .
\end{aligned}$$

Let us denote by λ^* the expression of λ above. It holds that $\lambda^* \in [0,1]$. Indeed,

$$\begin{aligned}
\lambda^* &= \tfrac{1}{2}\sqrt{\left(\tfrac{1}{q} - 1\right)\left(\tfrac{1}{q} + 3\right)} - \tfrac{1}{2}\left(\tfrac{1}{q} - 1\right) \geq \tfrac{1}{2}\sqrt{\left(\tfrac{1}{q} - 1\right)^2} - \tfrac{1}{2}\left(\tfrac{1}{q} - 1\right) = 0 \ , \\
\lambda^* &= \tfrac{1}{2}\sqrt{\tfrac{1}{q^2} + \tfrac{2}{q} - 3} - \tfrac{1}{2}\left(\tfrac{1}{q} - 1\right) \leq \tfrac{1}{2}\left(\tfrac{1}{q} + 1\right) - \tfrac{1}{2}\left(\tfrac{1}{q} - 1\right) = 1 \ .
\end{aligned}$$

Therefore, λ^* is a valid value of λ that can be chosen by the decision-maker, which yields the following upper bound on the average cost of $\mathsf{A}^{\mathrm{SR}}_{\lambda,\rho}$

$$\begin{aligned}
\frac{\mathbb{E}_y[\mathsf{A}^{\mathrm{SR}}_{\lambda^*,0}(x,y)]}{\min(x,b)} &\leq \max\left(2 - (1-\lambda^*)q, 1 + \frac{1-q}{\lambda}\right) \\
&= 2 - (1-\lambda^*)q \\
&= 2 - \frac{1+q}{2} + \frac{q^2}{2}\sqrt{\left(\tfrac{1}{q} - 1\right)\left(3 + \tfrac{1}{q}\right)} \\
&= \frac{3-q}{2} + \frac{1}{2}\sqrt{(1-q)(1+3q)} \ .
\end{aligned}$$

□