Multi-Objective *min-max* Online Convex Optimization

Rahul Vaze
School of Technology and Computer Science
Tata Institute of Fundamental Research, Mumbai
rahul.vaze@gmail.com

Sumiran Mishra
Indian Institute of Science Education and Research, Pune
sumiran.mishra@students.iiserpune.ac.in

November 21, 2025

Abstract

In online convex optimization (OCO), a single loss function sequence is revealed over a time horizon of T, and an online algorithm has to choose its action at time t, before the loss function at time t is revealed. The goal of the online algorithm is to incur minimal penalty (called *regret*) compared to a static optimal action made by an optimal offline algorithm knowing all functions of the sequence in advance.

In this paper, we broaden the horizon of OCO, and consider multi-objective OCO, where there are K distinct loss function sequences, and an algorithm has to choose its action at time t, before the K loss functions at time t are revealed. To capture the tradeoff between tracking the K different sequences, we consider the *min-max* regret, where the benchmark (optimal offline algorithm) takes a static action across all time slots that minimizes the maximum of the total loss (summed across time slots) incurred by each of the K sequences. An online algorithm is allowed to change its action across time slots, and its *min-max* regret is defined as the difference between its *min-max* cost and that of the benchmark. The *min-max* regret is a stringent performance measure and an algorithm with small regret needs to 'track' all loss function sequences closely at all times.

We consider this *min-max* regret in the i.i.d. input setting where all loss functions are i.i.d. generated from an unknown distribution. For the i.i.d. model we propose a simple algorithm that combines the well-known *Hedge* and online gradient descent (OGD) and show via a remarkably simple proof that its expected *min-max* regret is $O(\sqrt{T \log K})$.

1 Introduction

Online convex optimization (OCO) [18] is a basic learning problem that models a wide variety of applications such as spam filtering [18], capacity provisioning in cloud networks [25] etc., where at time t, an online algorithm $\mathcal{A}$ has to choose an action $x_t \in \mathcal{X}$, after which the adversary reveals the convex objective/loss function f_t on which x_t is evaluated. As a benchmark, the offline optimal algorithm is considered, that knows the entire set of f_t, $t = 1, \ldots, T$, however, can only choose a single action across all times. To quantify the performance of online algorithms, the metric of *static regret* R_T^{single} is studied, that is defined as follows, where x_t's are the actions taken by $\mathcal{A}$,

$$R_{\mathcal{A}}^{single}(T) = \sup_{f_t} \left\{ \sum_{t=1}^{T} f_t(x_t) - \min_{x} \sum_{t=1}^{T} f_t(x) \right\}. \tag{1}$$

In a long line of work, comprehensively summarized in [18], tight guarantees (both upper and lower bounds) have been obtained for the OCO problem, and $R^{single}(T) = \inf_{\mathcal{A}} R^{single}_{\mathcal{A}}(T) = \Theta(\sqrt{T})$. In particular, online gradient descent (OGD), mirror descent, and follow-the-regularized-leader (FTRL) [18] all achieve the optimal static regret performance for OCO.

Many modern real-world decision problems, however, involve multiple conflicting performance metrics, motivating the multi-objective OCO framework—an extension of OCO that aims to achieve Pareto-efficient trade-offs among competing objectives. However, Pareto optimality is too challenging to work with in the online setting, both in terms of defining the right benchmark as well as the required analytical treatment. One such attempt was made at this recently in [20], however, it used a *weak* regret metric, which remained independent of the number of objectives. We provide more details of this in the Related Work section.

Thus, in this paper, we consider the robust alternative in the multi-objective case: the $min - max$ OCO, where the objective of an algorithm is to minimize the maximum of the total loss (summed across time slots) incurred by each of the multiple objective functions. A compelling example for considering the *min-max* approach is fair machine learning, where performance is typically evaluated across multiple demographic groups, which naturally induces a multi-objective optimization problem. While Pareto efficiency identifies trade-offs between groups, it may still permit highly unequal solutions, as a classifier can be Pareto optimal even if it performs very poorly on a disadvantaged group. To address this limitation, fairness is often modeled via a min–max objective: maximize the minimum accuracy (or minimize the maximum error) across groups [7, 9]).

The *min-max* approach is also well suited for modern networked systems—such as data centers, wireless edge networks, and shared cloud infrastructures—where online decisions must balance performance across multiple service classes, users, or flows whose objectives (throughput, latency, energy) often conflict. The min–max regret criterion provides a principled way to ensure worst-case fairness by minimizing the largest cumulative loss among competing entities, which is crucial in fair resource allocation [38], fair allocation in clouds [17], fair assignment of reusable resources [26] congestion-controlled transport [22, 3], and dynamic resource provisioning in data centers [42]. Unlike average-regret formulations that can conceal systematic bias, minimizing *min–max* regret explicitly guards against persistent under-allocation to any group or flow—an increasingly important property in multi-tenant and QoS-sensitive networks.

The formal *min-max* OCO problem that we consider is as follows. We consider that there are two convex loss function sequences f_t, g_t, that are revealed after $\mathcal{A}$ chooses action x_t at time t for $t = 1, \ldots, T$. The *min-max* cost of $\mathcal{A}$ is

$$C_{\mathcal{A}} = \max\left\{\sum_{t=1}^{T} f_t(x_t), \sum_{t=1}^{T} g_t(x_t)\right\}. \tag{2}$$

Two loss function sequences are considered for simplicity of exposition since they capture the basic difficulty. Extension to $K > 2$ sequences is straightforward and detailed in Remark 7. As a benchmark, we consider the static offline optimal algorithm OPT that knows f_t, g_t for $t = 1, \ldots, T$ at time $t = 1$ itself but can only choose a single action $x^\star$ across all times such that

$$x^\star = \arg\min_{x \in \mathcal{X}} \max\left\{\sum_{t=1}^{T} f_t(x), \sum_{t=1}^{T} g_t(x)\right\}, \tag{3}$$

and the optimal cost is

$$C_{\mathsf{OPT}} = \max\left\{\sum_{t=1}^{T} f_t(x^\star), \sum_{t=1}^{T} g_t(x^\star)\right\}. \tag{4}$$

Consequently, we define the *min-max* static regret of $\mathcal{A}$ with actions x_t as

$$R_{\mathcal{A}}(T) = \sup_{f_t, g_t} \{C_{\mathcal{A}} - C_{\mathsf{OPT}}\}. \tag{5}$$

We refer to (5) as the *min-max* OCO problem.

Remark 1 (On the regret definition)**.** *The chosen benchmark in* (5) *is deliberately stringent. If we considered a weaker benchmark, e.g., minimizing* $\min_{x\in X}\sum_{t=1}^{T}\max\{f_t(x), g_t(x)\}$*), then applying standard OGD algorithm on the surrogate function* $h_t(x) = \max\{f_t(x), g_t(x)\}$ *would yield regret* $O(\sqrt{T})$ *independent of the number of objectives, underestimating the intrinsic hardness of balancing multiple sequences. In particular:*

- *Our definition forces the algorithm to track both sequences* globally and simultaneously. *This means the online algorithm cannot focus on one objective at the expense of the other, nor can it switch actions adaptively across time as the per-slot benchmark allows.*
- *Our benchmark ensures robustness across objectives: the comparator* $x^\star$ *is the* best fixed action *that minimizes the worst cumulative cost.*
- *This definition aligns with fairness and robustness considerations (e.g., in machine learning across demographic groups), where good performance must be guaranteed for* all *objectives, not merely on average or in expectation.*

Thus, while the regret metric in (5) *is harder to minimize, it is an appropriate notion when robustness across multiple objectives is essential.*

1.1 Related Work

Because of its widespread applications, multi-objective OCO has been an object of immense interest, however, with limited success. For example, [30] wanted to study the multi-objective OCO and stated the difficulty as "*A fundamental difference between single- and multi-objective optimization is that for the latter it is not obvious how to evaluate the optimization quality*". Further, it said "*Since it is impossible to simultaneously minimize multiple loss functions and in order to avoid complications caused by handling more than one objective, we choose one function as the objective and try to bound other objectives by appropriate thresholds*. This work gave rise to the now popular problem of **constrained online convex optimization** (COCO), where the problem is transformed into a constrained optimization problem with one objective and one or more constraint functions, that is described next.

1.1.1 COCO

In COCO, on every round t, an online algorithm first chooses an admissible action $x_t \in \mathcal{X} \subset \mathbb{R}^d$, and then the adversary chooses a convex loss/cost function $f_t : \mathcal{X} \to \mathbb{R}$ and a constraint function of the form $g_t(x) \le 0$, where $g_t : \mathcal{X} \to \mathbb{R}$ is a convex function. Let $\mathcal{X}^\star$ be the feasible set consisting of all admissible actions that satisfy all constraints $g_t(x) \le 0, t \in [T]$. A standard assumption is made that $\mathcal{X}^\star$ is not empty (called the *feasibility assumption*). Since g_t's are revealed after the action x_t is chosen, an online algorithm $\mathcal{A}$ need not take feasible actions on each round, and in addition to the static regret defined as

$$\mathrm{R}_{\mathcal{A}}^{\mathrm{COCO}}(T) \equiv \sup_{\{f_t\}_{t=1}^{T}} \left\{ \sum_{t=1}^{T} f_t(x_t) - \inf_{x\in\mathcal{X}^\star} \sum_{t=1}^{T} f_t(x^\star) \right\}, \tag{6}$$

an additional metric of interest is the total cumulative constraint violation (CCV) defined as

$$\mathrm{CCV}_{\mathcal{A}}(T) \equiv \sum_{t=1}^{T} \max(g_t(x_t), 0).$$

The goal with COCO is to design an online algorithm to simultaneously achieve a small regret (6) with respect to any admissible benchmark $x^\star \in \mathcal{X}^\star$ and a small CCV.

1.1.2 Prior Work on COCO

(A) i.i.d. loss and constraints: COCO with independent and identically distributed (i.i.d.) loss functions f_t's and constraint functions g_t's was studied in [30] while its extension to stationary and ergodic processes can be found in [43], where both derived an algorithm with regret $O(\sqrt{T})$ and CCV $O(T^{3/4})$.

(B) Time-invariant constraints: COCO with time-invariant constraints, *i.e.,* $g_t = g, \forall\, t$ [49, 19, 29, 46] has been considered extensively, where functions g are assumed to be known to the algorithm *a priori*. The best known results in this case give static regret of $O(T^{\max\{c,1-c\}})$ and CCV of $O(T^{(1-c)/2})$ for any $c \in (0,1)$. The algorithm is allowed to take actions that are infeasible at any time to avoid the costly projection step of the vanilla projected OGD algorithm and the main objective was to design an *efficient* algorithm with a small regret and CCV while avoiding the explicit projection step.

(C) Time-varying constraints: The more difficult question is solving COCO problem when the constraint functions, *i.e.*, g_t's, change arbitrarily with time t. One popular algorithm for solving COCO considered a Lagrangian function optimization that is updated using the primal and dual variables [48, 41, 47]. Alternatively, [33] and [24] used the drift-plus-penalty (DPP) framework [32] to solve the COCO, but which needed additional assumption, e.g. the Slater's condition in [33] and with weaker form of the feasibility assumption in [33].

[16] obtained bounds similar to [33] but without assuming Slater's condition. Very recently, the state of the art guarantees on simultaneous bounds on regret $O(\sqrt{T})$ and CCV $O(\sqrt{T}\log T)$ for COCO were derived in [39] with a very simple algorithm that combines the loss function at time t and the CCV accrued till time t in a single loss function, and then executes the online gradient descent (OGD) algorithm on the single loss function with an adaptive step-size. Most recently [40] showed lower CCV is possible with increased regret compared to [39] while [45] obtained instance dependent CCV bounds which can be $O(1)$ for specific cases while achieving regret of $O(\sqrt{T})$, respectively. The COCO problem has been considered in the *dynamic* setting as well [8, 4, 44, 27] where the benchmark $x^\star$ in (6) is replaced by $x_t^\star$ ($x_t^\star = \arg\min_x f_t(x)$) that is also allowed to change its actions over time.

Remark 2. *Even though the min-max OCO and the COCO problem appear related, but unfortunately results for one do not give results for the other in either direction.*

1.2 Global Cost Functions in Experts Setting

A special case of the *min-max* OCO problem (5) was considered in [14], where there are K experts, with expert i's loss at time t, being $\ell_t(i)$. If expert i is selected by an algorithm $\mathcal{A}$ with probability $p_t(i)$ at time t, then the overall expected loss for expert i is $L_i = \sum_{t=1}^{T} p_t(i)\ell_t(i)$ where T is the time horizon, and the cost of $\mathcal{A}$ is the global cost [14]

$$C_{\mathcal{A}} = \max_{i=1,\ldots,K} \{L_i\}. \tag{7}$$

The regret of $\mathcal{A}$ is defined as usual with respect to the static optimal offline algorithm that is required to choose a fixed distribution across all time slots while knowing all losses in advance.

The name global refers to the fact that the objective function (7) captures the end-to-end cost function rather than a summation of the instantaneous losses. For this formulation, which we refer to as GlobalExpertsProblem, [14] proposed an algorithm called MULTI and showed its regret to be $O(\sqrt{T}\log(K))$. GlobalExpertsProblem differs with our *min-max* OCO formulation, in two fundamental ways, i) the objective inside the $\max$ is a linear function of the actions $p_t(i)$, and ii) the action $p_t(i)$ taken at time t affects each expert's loss but only in an average sense since $\sum_{i=1}^{K} p_t(i) = 1$ while with *min-max* OCO the action x_t chosen at time t affects both $\sum_{t=1}^{T} f_t(x_t)$ and $\sum_{t=1}^{T} g_t(x_t)$ in *full*. Even though this appears to only be a small difference, it changes everything. As a consequence the result on GlobalExpertsProblem does not apply for *min-max* OCO even when f_t, g_t are linear. Extensions of [14] with additive global functions can be found in [2, 21] while for non-additive ones in [35].

1.3 Pareto Optimal Multi-Objective OCO

More recently, [20] attempted to study the multi-objective OCO from the Pareto optimality point of view. Towards this end, it used the concept of Pareto-suboptimality-gap (PSG), for $k = 1, \dots, K$ loss functions f_t^k, and defined the regret to be[1]

$$\sup_{x^\star \in \mathcal{P}^\star} \min_{k=1,\dots,K} \sum_{t=1}^{T} (f_t^k(x_t) - f_t^k(x^\star)). \tag{8}$$

where $\mathcal{P}^\star$ is the set of points on the Pareto frontier for the K-dimensional vector $[\sum_{t=1}^{T} f_t^k]_{k=1}^{K}$. [20] proposed a mirror-descent inspired algorithm and showed that its regret (8) is $O(\sqrt{T})$ which notably is **independent** of K. The minimization over the K possible loss functions in (8) made the regret (8) not depend on K, which highlights the weakness of the metric. Thus, even though [20] made a valid attempt at modelling the general multi-objective OCO, the weakness of the metric limited its applicability.

1.4 Per-slot $min - max$ multi-objective optimization

Compared to our 'global' *min-max* objective (4), in [23], a **per-slot** $min - max$ multi-objective OCO was studied, where in particular, at each round $t = 1, \dots, T$: i) an online algorithm chooses $x_t \in \mathcal{X}$ (convex compact set), ii) the adversary chooses $y_t \in \mathcal{Y}$ (convex compact set), and iii) the online algorithm suffers a vector loss:

$$\ell_t(x_t, y_t) = \big(\ell_{t,1}(x_t, y_t), \dots, \ell_{t,d}(x_t, y_t)\big) \in [-1, 1]^d.$$

The goal of an online algorithm is to solve $\min_{\{x_t\}} \max_{j\in[d]} \sum_{t=1}^{T} \ell_{t,j}(x_t, y_t)$, knowing $\ell_t(.,.)$, but not y_t, and its performance is compared against a per-round *local minimax value*

$$w_L^t := \min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \max_{j\in[d]} \ell_{t,j}(x, y),$$

where the cumulative benchmark is:

$$W_L^T := \sum_{t=1}^{T} w_L^t, \quad \text{and regret} \quad \mathrm{R}_{\mathcal{A}}^d(T) := \max_{j\in[d]} \sum_{t=1}^{T} \ell_{t,j}(x_t, y_t) - W_L^T. \tag{9}$$

A *Hedge*-type algorithm [6] was shown in [23] to have $O(\sqrt{T})$ regret.

Defining $\mathcal{Y} = \{e\}$ a singleton, and $d = 2$, such that $\ell_{t,1}(x_t, y_t) = f_t(x_t)$ and $\ell_{t,1}(x_t, y_t) = g_t(x_t)$ for $y_t = e$, we recover our model where two convex functions f_t, g_t are revealed at time t, and for which the benchmark W_L^T simplifies to

$$W_L^T = \sum_{t=1}^{T} \min_{x_t\in\mathcal{X}} \max\{f_t(x_t), g_t(x_t)\}). \tag{10}$$

Thus, benchmark (10) is fundamentally different than our considered benchmark (4) since it is additive across time slots, and allows the benchmark to choose different actions across time slots. Even though (10) allows choosing different actions across time slots, surprisingly, we show in Appendix D that W_L^T (10) can be $3/2$ times more than our benchmark C_{OPT} (4) and hence algorithms with $O(\sqrt{T})$ regret (9) do not yield a $O(\sqrt{T})$ *min-max* static regret (5).

In summary, compared to COCO and GlobalExpertsProblem, our considered problem has a wider scope, while our considered metric (5) is more meaningful and stringent compared to (8) and (10).

[1] We write the equivalent simpler form given on Page 21 [20] compared to the original Definition 4 made on Page 5 of [20] that is mathematically more complicated.

1.5 Applications

Multi-objective OCO has also been studied from an applications perspective, e.g. [28] studied the problem of multi-period multi-objective online ride-matching problem, where the multiple objectives included platform revenue, assigning drivers with higher service quality, minimizing the distance to the customer etc. [28] also used the COCO-type reformulation of multi-objective OCO for deriving theoretical guarantees. Other applications of multi-objective OCO include fair resource allocation [38], fair allocation in clouds [17], fair assignment of reusable resources [26] congestion-controlled transport [22, 3], and dynamic resource provisioning in data centers [42], multi-task learning [36], multiple target tracking [31], portfolio selection [12] and safe reinforcement learning [10], where decision-makers must make sequential decisions under uncertainty, balancing objectives such as cost, delivery time, and reliability without full knowledge of future demand or disruptions [22, 3], where the *min-max* criterion naturally makes sense.

1.6 Our Contributions

1. Compared to the discussed prior work, which has primarily focused on bounding the 'respective' regret under worst-case or adversarial inputs, establishing bounds with the adversarial input on the *min–max* static regret (5) appears to be substantially more challenging. We highlight the difficulty for one instance in Section 3, where even when full functions f_t, g_t are revealed before choosing action x_t, the greedy algorithm still has a linear regret.

2. Thus, we consider the *min-max* static regret problem in the **i.i.d.** setting where both f_t, g_t are chosen independently across t with identical (unknown) distribution $\mathcal{D}$. Note that all the discussed applications and use cases for multi-objective OCO (e.g. fair machine learning, fair resource allocation, multi-task learning, network scheduling, multi-objective online ride-matching problem etc.) are well motivated in the i.i.d. case as well. Moreover, analytically also, the problem remains challenging with the i.i.d. input. From a theoretical advancement point of view similar approach was followed for the COCO problem where the earliest results were derived for the i.i.d. input [30], which were then later generalized to the worst case input.

   For the i.i.d. model, the expected cost of $\mathcal{A}$ is $\bar{C}_{\mathcal{A}}$ that is just the expected value of $C_{\mathcal{A}}$ in (2), while the C_{OPT} simplifies to

$$\bar{C}_{\mathsf{OPT}} = T \cdot \min_{x \in \mathcal{X}} \max_{\theta \in \Delta_2} \theta^\top \mu(x) \quad \text{with} \quad (x^\star, \theta^\star) = \arg\min_{x \in \mathcal{X}} \max_{\theta \in \Delta_2} \theta^\top \mu(x), \tag{11}$$

   where $\mu(x) = \mathbb{E}\{f_t(x), g_t(x)\}$, $\theta = (\theta_1, \theta_2) \in \Delta_2$ and Δ_2 is the $2-D$ simplex. The metric is now the expected *min-max* static regret

$$\bar{R}_{\mathcal{A}}(T) = \sup_{\mathcal{D}} \{\bar{C}_{\mathcal{A}} - \bar{C}_{\mathsf{OPT}}\}. \tag{12}$$

   Algorithm: For the i.i.d. model, the basic idea behind our solution approach for minimizing the expected *min-max* static regret (12) is as follows. From (11), the real goal for an algorithm is to find the optimal pair $(x^\star, \theta^\star)$. Since an online algorithm $\mathcal{A}$ does not know $\mathcal{D}$ ahead of time, finding $(x^\star, \theta^\star)$ is not possible. So we propose a simple algorithm that produces pair (x_t, θ_t) at time t in an online manner that uses *Hedge* algorithm[2] [6] to update θ_t to solve the inner maximization problem in (11) given x_t, and OGD to obtain actions x_t given θ_t to solve the outer minimization (OCO) problem in (11) applied to the surrogate loss function $\theta_t^\top [f_t(x), g_t(x)] = \theta_{1t} f_t(x) + \theta_{2t} g_t(x)$ which is a proxy for $\theta^\top \mu(x)$, in an attempt to keep (x_t, θ_t) to stay close to $(x^\star, \theta^\star)$, so that the regret remains small.

[2]We describe the basic Experts problem and Hedge algorithm in Appendix E together with its regret guarantee for completeness purposes.

Note that (11) is not the usual online $min - max$ optimization problem [5], where an online algorithm and adversary are playing against each other. Instead with (11) both x and θ are algorithm's choices depending on $\mathcal{D}$, that is under an adversarial control as defined in (12). The proposed alternating *min-max* algorithm is intuitive, however, the regret analysis is actually quite delicate, and heavily depends on our non-trivial decomposition of regret in (19) into three terms, which also motivates our algorithm.

It is worth noting that there are several other candidate algorithms (e.g. primal-dual algorithm for COCO [48] and OGD with surrogate cost function for COCO [39]) that appear to solve (11), however, analyzing them is rather challenging because of the outside $\max$ in the cost of an online algorithm in (5).

3. **Guarantee:** Using simple and elegant analysis via an insightful regret decomposition, we show that the expected *min-max* static regret (12) of the proposed algorithm is $O(\sqrt{T})$. This is similar to the short and elegant analysis of OGD algorithm's [18] performance for OCO in the adversarial input setting, that has been very useful for further progress. Even though we considered just two sequences f_t, g_t for simplicity, the result generalizes directly if there are K sequences $f_t^k, k = 1, \ldots, K$ and the expected *min-max* static regret (5) is $O(\sqrt{T \log K})$. See Remark 7 for more details.

4. **Improved guarantee for GlobalExpertsProblem [14] in the i.i.d. input:** As a special case of our result, we derive an improved regret guarantee of $O(\sqrt{T \log K})$ for the GlobalExpertsProblem [14] with the i.i.d. input compared to $O(\sqrt{T} \log K)$ of [14] which was applicable in the worst case/adversarial input setting.

5. **With Bandit Feedback:** We also extend our results to the bandit setting, where only function evaluations at the most recent actions are available without any gradient feedback. We show that the expected *min-max* static regret (12) is $O(T^{3/4}\sqrt{\log K})$ and $O(\sqrt{T \log K})$ for one-point and two-point bandit feedback, respectively, matching the best known bounds on OCO in the respective bandit settings.

1.7 Open Questions

- We have only considered the i.i.d. input setting, and the major open question that remains is whether $o(T)$ *min-max* static regret (5) is achievable in the worst case input or not.

- Lower bound on the expected *min-max* regret: Theoretically, deriving a lower bound on the expected *min-max* static regret matching the derived guarantee of $O(\sqrt{T})$ appears challenging, however, simulations (see Fig. 1) with f_t, g_t as random linear functions suggest that $\Theta(\sqrt{T})$ is the correct scaling for the expected *min-max* static regret.

- Strongly Convex Functions: For OCO, the best-known static regret reduces to $O(\log T)$ when functions f_t's are strongly convex. Because of the use of *Hedge* in our proposed algorithm to update weights θ_t , its expected *min-max* static regret (12) remains $O(\sqrt{T})$ even if f_t, g_t are strongly convex. It is not clear whether this is a limitation of the proposed algorithm or the expected *min-max* static regret is $\Omega(\sqrt{T})$ even when f_t, g_t are strongly convex. Simulations (see Fig. 2) suggest that in fact expected *min-max* static regret is $\Omega(\sqrt{T})$ for the proposed algorithm.

- Constrained Multi-Objective OCO: An immediate open question is how to solve the constrained version of the multi-objective OCO. For example, let f_t, g_t be the loss functions under the $min - max$ criteria while h_t be the constraint function. Similar to COCO, let $\mathcal{X}^\star$ be the feasible set consisting of all admissible actions that satisfy all constraints $h_t(x) \leq 0, t \in [T]$. Then, the optimal expected static benchmark cost is

$$\bar{C}_{\mathsf{OPT}} = T \cdot \min_{x \in \mathcal{X}^\star} \max_{\theta \in \Delta_2} \theta^\top \mu(x) \quad \text{with} \quad (x^\star, \theta^\star) = \arg \min_{x \in \mathcal{X}^\star} \max_{\theta \in \Delta_2} \theta^\top \mu(x), \tag{13}$$

and the corresponding regret and CCV are

$$\bar{R}_{\mathcal{A}}(T) = \sup_{\mathcal{D}}\{\bar{C}_{\mathcal{A}} - \bar{C}_{\mathsf{OPT}}\} \quad \bar{\text{CCV}}_{[1:T]} \equiv \sum_{t=1}^{T} \mathbb{E}\{\max(h_t(x_t), 0)\}, \tag{14}$$

respectively. It turns out that our proposed algorithm cannot be married readily with the prior best known approaches for COCO [39, 45, 40] for deriving simultaneous bounds on regret and CCV. Thus, how to solve the *min-max* OCO with constraints remains an interesting open problem.

2 Assumptions

We start by stating the assumptions that we will use throughout, that are standard in the OCO literature [18].

Assumption 1 (Convexity)**.** *$\mathcal{X} \subset \mathbb{R}^d$ is the admissible set that is closed, convex and has a finite Euclidean diameter D. Cost functions $f_t, g_t : \mathcal{X} \mapsto \mathbb{R}$ are convex for all $t \geq 1$, and are bounded $|f_t(x)| \leq B, |g_t(x)| \leq B,\ \forall\, x \in \mathcal{X}$.*

Assumption 2 (Lipschitzness)**.** *All cost functions $\{f_t, g_t\}_{t\geq 1}$ are G-Lipschitz, i.e., for any $x, y \in \mathcal{X}$, we have $|f_t(x) - f_t(y)| \leq G||x - y||,\ |g_t(x) - g_t(y)| \leq G||x - y||,\ \forall t \geq 1$.*

Assumption 3. Information Structure: *On round t, the online algorithm $\mathcal{A}$ first chooses an admissible action $x_t \in \mathcal{X}$ and then the adversary chooses two convex cost functions $f_t : \mathcal{X} \to \mathbb{R}$ and $g_t : \mathcal{X} \to \mathbb{R}$. Once the action x_t has been chosen, both $f_t(x_t), \nabla f_t(x_t)$ and $g_t(x_t), \nabla g_t(x_t)$ are revealed, as is standard in the literature. We will also consider the bandit information structure in Section 6 where no gradient information is available.*

3 *min-max* OCO Problem (5) in the Adversarial Setting

In this section, we highlight the difficulty in solving the *min-max* OCO Problem (5) in the adversarial input setting, where the input can be chosen adversarially depending on the choice of an online algorithm.

To illustrate the basic difficulty, we even allow the algorithm to know f_t, g_t completely **before** choosing its action x_t at time t. Under this enhanced information setting, consider the greedy (locally optimal) algorithm that chooses

$$x_t = \min_{x\in\mathcal{X}} \max\{f_t(x), g_t(x)\}.$$

Recall that when $g_t = f_t\ \forall\, t$, i.e. in the OCO setting, this algorithm will have non-positive regret $R_{\mathcal{A}}^{single}(T)$ (1) since $f_t(x_t) \leq f_t(x)$ for any x. However, as we show next, the greedy algorithm suffers a linear *min-max* static regret (5).

Let $T = 2N$ and consider the following linear functions on $\mathcal{X} = [0, 1]$, for $j = 1, \ldots, N$

$$\begin{aligned} f_{2j-1}(x) &= 1.2 - 0.2x, \qquad f_{2j}(x) = x, \\ g_{2j-1}(x) &= x, \qquad\ \ g_{2j}(x) = 0.8 + 0.2x. \end{aligned} \tag{15}$$

Clearly,

$$\arg\min_{x\in\mathcal{X}}\{\max(f_{2j-1}(x), g_{2j-1}(x))\} = 1 \quad \text{and} \quad \arg\min_{x\in\mathcal{X}}\{\max(f_{2j}(x), g_{2j}(x))\} = 0.$$

Thus, the greedy algorithm chooses $x_{2j-1} = 1$ and $x_{2j} = 0$, and the cost $C_{\mathcal{A}}$ (2) of the greedy algorithm is

$$\max\left\{\sum_{j=1}^{N}\left(f_{2j-1}(x_{2j-1})+f_{2j}(x_{2j})\right),\sum_{j=1}^{N}\left(g_{2j-1}(x_{2j-1})+g_{2j}(x_{2j})\right)\right\}$$
$$=\max\left\{\sum_{j=1}^{N}\left(f_{2j-1}(1)+f_{2j}(0)\right),\sum_{j=1}^{N}\left(g_{2j-1}(1)+g_{2j}(0)\right)\right\}=\max\left\{\sum_{j=1}^{N}(1+0),\sum_{j=1}^{N}(1+0.8)\right\}=1.8N.$$

However, if an algorithm chooses a single static action $x^\star = 0$ for all times, its cost $C_{\mathcal{A}}$ (5) is $1.2N$, and hence

$$C_{\mathsf{OPT}} \leq 1.2N.$$

Thus, the *min-max* static regret (5) of the greedy algorithm is linear in T. This example highlights the key difficulty in minimizing the *min-max* static regret in the adversarial/worst case input. We note that this example does not preclude more intelligent algorithms that take a global view in trying to balance the two costs, such as primal-dual, OGD+*Hedge* or FTRL to achieve sub-linear *min-max* static regret (5), however, the stringent *min-max* static regret definition makes the analysis highly challenging. We will make additional remarks on the technical difficulty in Appendix F. Thus, from here on we consider the i.i.d. input that is defined next.

4 i.i.d. Input model

We consider the i.i.d. input model, where at time t, loss functions (f_t, g_t) are chosen independently and are identically jointly distributed as $\mathcal{D}$ (unknown) across all t. We assume that Assumption 1 and 2 holds, and consider the same information structure as defined in Assumption 3.

Let $\lambda_t = (f_t, g_t)$ and recall that Δ_2 is the $2-D$ simplex, $\theta = (\theta_1, \theta_2) \in \Delta_2$, and

$$\Lambda_t(x, \theta) := \theta_1 f_t(x) + \theta_2 g_t(x).$$

Following (2), the expected cost of any online algorithm $\mathcal{A}$ with the i.i.d. input which chooses actions x_t is

$$\bar{C}_{\mathcal{A}} = \mathbb{E}\left\{\max\left\{\sum_{t=1}^{T} f_t(x_t), \sum_{t=1}^{T} g_t(x_t)\right\}\right\} = \mathbb{E}\left\{\max_{\theta\in\Delta_2}\sum_{t=1}^{T}\Lambda_t(x_t,\theta)\right\}. \tag{16}$$

while from (11)

$$\bar{C}_{\mathsf{OPT}} = T\cdot\min_{x\in\mathcal{X}}\max_{\theta\in\Delta_2}\theta^\top\mu(x) \quad \text{with} \quad (x^\star,\theta^\star) = \arg\min_{x\in\mathcal{X}}\max_{\theta\in\Delta_2}\theta^\top\mu(x), \tag{17}$$

where $\mu(x) = \mathbb{E}\{f_t(x), g_t(x)\}$.

Consequently, the expected *min-max* static regret $\bar{R}_T(\mathcal{A})$ following (12) is

$$\bar{R}_{\mathcal{A}}(T) = \sup_{\mathcal{D}}\left\{\bar{C}_{\mathcal{A}} - \bar{C}_{\mathsf{OPT}}\right\}. \tag{18}$$

We refer to (18) as the i.i.d. *min-max* OCO problem and for the rest of the paper, consider it with the goal of deriving an algorithm that has $o(T)$ expected *min-max* static regret.

Remark 3. *Note that even i.i.d. min-max OCO problem is theoretically challenging, and no prior results apply directly.*

Remark 4. *We note that only two loss function sequences are considered for simplicity of exposition since it captures the basic difficulty of the problem, and all the results that we derive apply directly to the case when there are K function sequences (see details in Remark 7).*

4.1 Algorithm

In this section, we present an algorithm to solve the i.i.d. *min-max* OCO problem (18). Towards that end, if suppose an online algorithm knew the optimizing $\theta^\star$ in (17), then by executing an OGD algorithm for OCO with a single function $\theta_1^\star f_t + \theta_2^\star g_t$ at time $t+1$, it would achieve a regret of $O(\sqrt{T})$ using standard results [18]. However, the optimizing $\theta^\star$ that depends on $\mathcal{D}$ (unknown) remains unknown, but, this observation motivates our *algorithm* that constructs

i) θ_t at time t as a surrogate for $\theta^\star$ using the *Hedge* algorithm (gains version) treating the two function sequences $\{f_t\}_{t=1}^T$ and $\{g_t\}_{t=1}^T$ as the gains of the two experts (see Section E for basic definition of the Experts Problem and the *Hedge* algorithm), and

ii) for each fixed value of θ_t, **plays the action** chosen by OGD applied to the single surrogate function $\theta_{t1} f_t + \theta_{t2} g_t$ at time $t+1$. Pseudocode is given in Algorithm 1.

Algorithm 1 Minimization of the Maximum of The Sum of Two Convex Loss Sequences

Input: step sizes $\eta_{x,t} > 0$, $\eta_{\theta,t} > 0$, feasible set $\mathcal{X}$
Initialize: $w_1 = (1,1)$, $\theta_1 = w_1/\|w_1\|_1$, $f_0 \equiv 0, g_0 \equiv 0$ and play arbitrary $x_0 \in \mathcal{X}$
for $t = 1, \ldots, T$ **do**
    Observe $(f_{t-1}(x_{t-1}), g_{t-1}(x_{t-1}))$ and $(\nabla f_{t-1}(x_{t-1}), \nabla g_{t-1}(x_{t-1}))$
    Using current θ_t, form convex function $x \mapsto \tilde{\Lambda}_t(x, \theta_t) := \theta_{t,1} f_{t-1}(x) + \theta_{t,2} g_{t-1}(x)$
    OGD: Gradient step: $y_t = x_{t-1} - \eta_{x,t} \nabla \tilde{\Lambda}_t(x_{t-1}, \theta_t)$ and find $x_t = \text{Proj}_{\mathcal{X}}(y_t)$
    Play action x_t
    Evaluate $\lambda_t = (f_t(x_t), g_t(x_t))$
    Hedge: Update weights: $w_{t+1,i} = w_{t,i} \exp(\eta_{\theta,t} \lambda_{t,i})$ for $i = 1, 2$
    Normalize: $\theta_{t+1} = w_{t+1}/\|w_{t+1}\|_1$
end for

4.2 Guarantee

Theorem 4. *Under Assumptions 1 and 2, for* $\eta_{\theta,t} \leq \sqrt{\frac{2 \ln 2}{B^2 t}}, \eta_{x,t} = \frac{D}{G\sqrt{t}}$*, the expected min-max static regret* (18) *of the proposed algorithm (Algorithm 1) is*

$$\bar{R}_{\mathcal{A}}(T) = O(\sqrt{T}).$$

Proof. The main ingredient of the proof as well as the inspiration for Algorithm 1 is the following decomposition (19) of the expected *min-max* static regret (18). Recall the definition of

$$\Lambda_t(x, \theta) := \theta_1 f_t(x) + \theta_2 g_t(x).$$

where $\theta = (\theta_1, \theta_2) \in \Delta_2$.

By adding and subtracting two terms, $\sum_{t=1}^T \Lambda_t(x_t, \theta_t)$ and $\min_{x \in \mathcal{X}} \sum_{t=1}^T \Lambda_t(x, \theta_t)$ inside the expectation in the RHS of (18), we write the expected *min-max* static regret (18) as

$$\bar{R}_T(\mathcal{A}) = \sup_{\mathcal{D}} \{\mathbb{E}\{R_1\} + \mathbb{E}\{R_2\} + \mathbb{E}\{R_3\}\} \leq \sup_{\mathcal{D}} \{\mathbb{E}\{R_1\}\} + \sup_{\mathcal{D}} \{\mathbb{E}\{R_2\}\} + \sup_{\mathcal{D}} \{\mathbb{E}\{R_3\}\}, \tag{19}$$

where

$$R_1 = \max_{\theta \in \Delta_2} \sum_{t=1}^T \Lambda_t(x_t, \theta) - \sum_{t=1}^T \Lambda_t(x_t, \theta_t),$$

$$R_2 = \sum_{t=1}^T \Lambda_t(x_t, \theta_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^T \Lambda_t(x, \theta_t),$$

$$R_3 = \min_{x\in\mathcal{X}} \sum_{t=1}^{T} \Lambda_t(x,\theta_t) - \bar{C}_{\mathsf{OPT}} = \min_{x\in\mathcal{X}} \sum_{t=1}^{T} \Lambda_t(x,\theta_t) - T \cdot \min_{x\in\mathcal{X}} \max_{\theta\in\Delta_2} \theta^\top \mu(x).$$

The interpretation of R_1 is that it is the regret for the gains version of 2-Experts Problem [6], where the two function sequences f_t, g_t are the gains of two experts, with respect to an optimal $\theta^\star \in \Delta_2$ for fixed actions x_t across time slots. This motivates the use of *Hedge* algorithm [6] (gains version) in our proposed algorithm that is tailor-made for updating θ_t for fixed actions x_t.

Similarly, R_2 is the static regret for an OCO minimization problem relative to static optimal action $x^\star$ for a single surrogate objective function sequence of $\Lambda_t(x,\theta_t) = \theta_{t1} f_t(x) + \theta_{t2} g_t(x)$ for a fixed θ_t. Hence using an OGD algorithm is natural for this purpose as done in the algorithm for each time t, with a fixed weight θ_t.

The third term R_3 is more nuanced and captures the difference between the cost of OPT for a fixed sequence of $\theta_t, t = 1,\ldots,T$ and the optimal benchmark cost $\bar{C}_{\mathsf{OPT}}$ (17), respectively.

We will bound R_1 and R_2 sample path wise for each realization of f_t, g_t, while for R_3 we will bound $\mathbb{E}\{R_3\}$. In particular, we will show the following that is proven in the Appendix.

Lemma 5. *For $\eta_{\theta,t} \le \sqrt{2\ln 2/(B^2 t)}$, $\sup_{\mathcal{D}}\{R_1\} \le O(\sqrt{T})$.*

Lemma 6. *Choosing $\eta_{x,t} = \frac{D}{G\sqrt{t}}$, $\sup_{\mathcal{D}}\{R_2\} \le O(\sqrt{T})$.*

Lemma 7. $\mathbb{E}\{R_3\} \le 0$.

Because of the connection described above, the proof for R_1 in Lemma 5 and for R_2 in Lemma 6 follows standard analysis [18] for *Hedge* and OGD, respectively. Bounding R_3 needs more care and we prove it in Appendix C where we critically exploit the i.i.d. input model.

Combining Lemma 5, 6 and 7, and the regret decomposition (19), we get that the expected *min-max* static regret (12) of Algorithm 1 with

$$\eta_{\theta,t} \le \sqrt{\frac{2\ln 2}{B^2 t}}, \qquad \text{and} \qquad \eta_{x,t} = \frac{D}{G\sqrt{t}} \tag{20}$$

is

$$\bar{R}_{\mathcal{A}}(T) = O(\sqrt{T}), \tag{21}$$

proving the result. □

Following remarks are in order.

Remark 5. *The regret decomposition* (19) *does multiple things at the same time i) identifies the critical difficulty of the problem, ii) motivates a natural algorithm, and ii) keeps the proof modular, where the regret bound for each module is simple and elegant, almost first principles based.*

Remark 6. *Although the proof of Theorem 4 is remarkably simple and elegant, it hinges on the critical regret decomposition* (19) *– which is not obvious a priori. This decomposition eliminates the need for sophisticated technical machinery needed in related prior work [30] that also considers the i.i.d. input and highlights the underlying simplicity of the result. We would like to emphasize that the elegance of the argument should not obscure its importance.*

Remark 7. *When there are K loss function sequences $f_t^k, k = 1,\ldots,K$ that are chosen i.i.d. from $\mathcal{D}$, define $\theta \in \Delta_K$, (where Δ_K is the K-dimensional simplex) and $\Lambda_t(x,\theta) = \sum_{k=1}^K \theta_k f_t^k$. Then the expected min-max static regret of $\mathcal{A}$ is*

$$\bar{R}_{\mathcal{A}}(T) = \sup_{\mathcal{D}} \mathbb{E}\left\{ \max_{\theta\in\Delta_K} \sum_{t=1}^{T} \Lambda_t(x_t,\theta) - T \cdot \min_{x\in\mathcal{X}} \max_{\theta\in\Delta_K} \theta^\top \mu(x) \right\} \tag{22}$$

where $\mu(x) = \mathbb{E}\{f_t^1(x), \ldots, f_t^K(x)\}$.

With this new definition of θ *and* Λ_t*, Algorithm 1 remains as is with Hedge choosing* $\theta_t \in \Delta_K$ *while* x_t *the single action chosen by OGD. Assuming all* $f_t^k, k = 1, \ldots, K$ *satisfy Assumption 1 and 2, the only change in the guarantee with* $K > 2$ *compared to Theorem 4 with* $K = 2$ *is because of Lemma 5 that is now* $O(\sqrt{T \log K})$ *[18] since there are* K *experts corresponding to the* K *loss function sequences* $f_t^k, k = 1, \ldots, K$ *. The OGD guarantee of Lemma 6 stays the same since* $\|\nabla \Lambda_t(x_t, \theta)\| \leq G$ *as before. Moreover, Lemma 7 holds as is. Thus, the overall expected min-max static regret of Algorithm 1 with* K *loss function sequences is*

$$O(\sqrt{T \log K}). \tag{23}$$

Remark 8. *We note that Algorithm 1 is well motivated even while considering the worst case/adversarial input. The technical problem in its analysis with the worst case/adversarial input is that Lemma 7 that is the critical result fails when considering the adversarial input. We highlight this difficulty in Appendix F and show that the 'best possible' upper bound on* R_3 *is* $R_3 = O(T^{3/2})$ *with the worst case/adversarial input for Algorithm 1 by exploiting just the properties of the Hedge algorithm which is insufficient for achieving sub-linear regret.*

Remark 9. *[Extension to Strongly Convex* f_t*'s and* g_t*'s] One limitation of our proposed algorithm and the proof is that because of the use of the Hedge algorithm for updating* θ_t*, the regret* $\sup_{\mathcal{D}}\{R_1\}$ *can at best be* $O(\sqrt{T})$*. Thus, unlike OCO, where OGD algorithm achieves logarithmic regret in* T *when* f_t*'s are strongly convex, our algorithm's regret is only* $O(\sqrt{T})$ *even when both* f_t, g_t *are strongly convex. Is this a limitation of the algorithm or of that of the problem itself remains an interesting unresolved question.*

5 Improving the result on GlobalExpertsProblem with the i.i.d. input [14]

Let $\mathcal{X} = \Delta_K$, the K-dimensional simplex, and writing $f_t^k(x) = x_k \ell_k(t)$, where x_k is the k-th coordinate of $x \in \mathcal{X}$, $\ell_k(t)$ is the loss corresponding to the k-th expert, and $k = 1, 2 \ldots, K$, we get the GlobalExpertsProblem [14] with K experts in the i.i.d. input case is a special case of the i.i.d. *min-max* OCO problem (18).

Thus, from (23), we get that the expected *min-max* static regret of Algorithm 1 is $O(\sqrt{T \log K})$ for the GlobalExpertsProblem with i.i.d. input. In comparison, [14] derived a *min-max* static regret (5) guarantee of $O(\sqrt{T} \log K)$ with adversarial input for the following algorithm called MULTI:

For $K = 2$,

$$x_1(t+1) = x_1(t) + \frac{x_2(t)\ell_2(t) - x_1(t)\ell_1(t)}{\sqrt{T}},$$

and $x_2(t) = 1 - x_1(t)$. For general K, the $K = 2$ algorithm was used recursively. Compared to our Algorithm 1, the MULTI algorithm [14] is very compute intensive since it recursively uses an algorithm for $K = 2$ and requires the knowledge of time horizon T.

6 Extension to Bandit Feedback

We now analyze the expected *min-max* OCO problem (18) under the bandit feedback, where at time t, once the online algorithm $\mathcal{A}$ chooses its action x_t, it only observes the scalar values $\{f_{t-1}^k(.)\}_{k=1}^K$ at one (one-point model) or two points (two-point model) and does not receive any gradient information.

Algorithm 2 Bandit-MinMax (unified one-point/two-point variant)

Require: Feasible Set $\mathcal{X} \subset \mathbb{R}^d$, smoothing parameters $\{\delta_t\}$, step-sizes $\{\eta_{x,t}\}, \{\eta_{\theta,t}\}$, feedback type mode $\in$ $\{\text{one-point}, \text{two-point}\}$

Initialize $w_1 = \mathbf{1}_K$, $\theta_1 = w_1/\|w_1\|_1$, choose $x_0 \in X$

Define $\Lambda_t(x, \theta_t) = \sum_k \theta_{t,k} f_t^k(x)$

for $t = 1, \ldots, T$ **do**

  Sample u_t uniformly from unit sphere $\mathbb{S}^{d-1}$

  if mode $=$ one-point **then**

    Play $\tilde{x}_t = \text{Proj}_{\mathcal{X}}(x_{t-1} + \delta_t u_t)$ and observe $\{f_t^k(\tilde{x}_t)\}_{k=1}^K$

    Gradient estimator:

$$\widehat{\nabla}_t = \frac{d}{\delta_t} \Lambda_t(\tilde{x}_t, \theta_t)\, u_t$$

  else if mode $=$ two-point **then**

    Query $x_{t-1} + \delta_t u_t$ and $x_{t-1} - \delta_t u_t$, observe $\{f_t^k(x_{t-1} \pm \delta_t u_t)\}_{k=1}^K$

    Gradient estimator:

$$\widehat{\nabla}_t = \frac{d}{2\delta_t} \Big(\Lambda_t(x_{t-1} + \delta_t u_t, \theta_t) - \Lambda_t(x_{t-1} - \delta_t u_t, \theta_t) \Big) u_t$$

  end if

  Gradient step: $y_t = x_{t-1} - \eta_{x,t} \widehat{\nabla}_t, \quad x_t = \text{Proj}_{\mathcal{X}}(y_t)$

  Hedge update: $w_{t+1,k} = w_{t,k} \exp\big(\eta_{\theta,t} f_{t,k}(x_t)\big), \quad \theta_{t+1} = w_{t+1}/\|w_{t+1}\|_1$

end for

6.1 Algorithm

For this bandit model, we consider a simple adaption of Algorithm 1 by replacing its OGD part with the bandit information counterpart, while keeping the *Hedge* part as it is, given by Algorithm 2. Moreover, the analysis also follows the same script by replacing the guarantees by their respective bandit counterparts.

6.2 Guarantee

Theorem 8. *Under Assumption 1 and 2,*

1. *(**One-point feedback**) With $\delta_t = t^{-1/4}$, $\eta_{x,t} = D/(Gd^{1/2}t^{3/4})$, and $\eta_{\theta,t} = \sqrt{\log K}/(B\sqrt{t})$, the expected min-max static regret satisfies*

$$\bar{R}_{\mathcal{A}}(T) \le c_1\, d^{1/2} GD\, T^{3/4} \;+\; c_2 B \sqrt{T \log K},$$

   for universal constants $c_1, c_2 > 0$.

2. *(**Two-point feedback**) With $\delta_t = t^{-1/2}$, $\eta_{x,t} = D/(Gd^{1/2}\sqrt{t})$, and $\eta_{\theta,t} = \sqrt{\log K}/(B\sqrt{t})$, the expected min-max static regret satisfies*

$$\bar{R}_{\mathcal{A}}(T) \le c_3\, d^{1/2} GD\, \sqrt{T} \;+\; c_4 B \sqrt{T \log K},$$

   for universal constants $c_3, c_4 > 0$.

6.3 Proof of Theorem 8

Recall the decomposition (19)

$$\bar{R}_{\mathcal{A}}(T) = \sup_{\mathcal{D}}\{\mathbb{E}\{R_1\} + \mathbb{E}\{R_2\} + \mathbb{E}\{R_3\}\} \leq \sup_{\mathcal{D}}\{\mathbb{E}\{R_1\}\} + \sup_{\mathcal{D}}\{\mathbb{E}\{R_2\}\} + \sup_{\mathcal{D}}\{\mathbb{E}\{R_3\}\}.$$

Bounding R_1 and R_3. For both one-point or two-point bandit feedback, the *Hedge* guarantee remains the same as in Lemma 5. In particular, with $\eta_{\theta,t} = \sqrt{\log K}/(B\sqrt{t})$, sample path wise,

$$R_1 \leq B\sqrt{2T\log K},$$

and similar to Lemma 7,

$$\mathbb{E}\{R_3\} \leq 0.$$

The only change is in bounding R_2.

Bounding R_2 (one-point feedback). The one-point estimator

$$\widehat{\nabla}_t = \frac{d}{\delta_t}\Lambda_t(\tilde{x}_t, \theta_t)\, u_t$$

is unbiased for the gradient of the δ_t-smoothed loss, with variance at most $O(dB^2/\delta_t^2)$. Using $\delta_t = t^{-1/4}$ and $\eta_{x,t} = D/(Gd^{1/2}t^{3/4})$, the bandit-OGD analysis [15] yields

$$R_2 \leq c_1\, d^{1/2} GD\, T^{3/4},$$

for some universal constant c_1.

Combining the three bounds we get

$$\bar{R}_{\mathcal{A}}(T) \leq c_1\, d^{1/2} GD\, T^{3/4} \;+\; 2B\sqrt{2T\log K}.$$

Bounding R_2 (two-point feedback). With two function evaluations per round, the estimator

$$\widehat{\nabla}_t = \tfrac{d}{2\delta_t}\Big(\Lambda_t(x_{t-1} + \delta_t u_t, \theta_t) - \Lambda_t(x_{t-1} - \delta_t u_t, \theta_t)\Big)u_t$$

is unbiased with variance $O(dB^2/\delta_t^2)$. Choosing $\delta_t = t^{-1/2}$ and $\eta_{x,t} = D/(Gd^{1/2}\sqrt{t})$, the two-point bandit-OGD analysis [1, 13, 37] gives

$$R_2 \leq c_3\, d^{1/2} GD\, \sqrt{T},$$

for some universal constant c_3. Thus, with two-point feedback,

$$\bar{R}_{\mathcal{A}}(T) \leq c_3\, d^{1/2} GD\, \sqrt{T} \;+\; 2B\sqrt{2T\log K}.$$

□

Discussion: Compared to the limited information setting considered in Section 2, where gradient information is available, with bandit feedback the regret guarantees suffer for the expected *min-max* OCO problem, however, the degraded guarantees match the best known guarantees for OGD in the bandit setting. The proof of the result is simple because of the modularity of our algorithm and the regret decomposition that decomposes the total regret into regret resulting from *Hedge* and OGD, and the fact that only regret for OGD changes with bandit feedback.

7 Experimental Results

In this section, we present numerical results on the expected *min-max* static regret of Algorithm 1 under relevant settings, that are well motivated from both theoretical as well applications point of view. In all the figures, we plot the expected *min-max* static regret as a function of time horizon T.

7.1 Basic Results

We begin our experimental results by considering the input model when all functions f_t^k for $k = 1, \ldots, K, t = 1, \ldots, T$ are random linear functions in $d = 10$ dimensions, where each coefficient is distributed uniformly between $[0, 1]$. Random linear functions are important since they were sufficient to find lower bounds on the regret of any online algorithm for OCO [18]. In Fig. 1, we plot the expected *min-max* static regret of Algorithm 1 as a function of K and T for random linear functions, where as expected, we observe that the regret scales as $O(\sqrt{T \log K})$.

Next, we consider f_t^k's as strongly convex functions, where each f_t^k is independently and identically selected from a set of $\{(x - a)^2, a = [-9, -8, \ldots, 0, 1, \ldots, 9]\}$. Recall that for OCO with strongly convex functions, the step size η needed for OGD to get a regret guarantee of $O(\log T)$ scales as $\eta = 1/t$ and not $\eta = 1/\sqrt{t}$ [18]. Thus, for this simulation of Algorithm 1[3] we use $\eta_{\theta,t} = O(1/\sqrt{t})$ and $\eta_{x,t} = O(1/t)$. [4] As pointed out earlier, we are unable to improve the expected *min-max* static regret guarantee for Algorithm 1 when all functions f_t^k's are strongly convex. We see that in fact that is true for Algorithm 1 even in simulations as shown in Fig. 2, where for $K = 1$, which is the same as OCO (with $\eta_{x,t} = O(1/t)$) the regret scales as $\log(T)$, while for larger values of K, the expected *min-max* static regret scales as $O(\sqrt{T})$ when $\eta_{\theta,t} = O(1/\sqrt{t})$ and $\eta_{x,t} = O(1/t)$. Thus, if indeed expected *min-max* static regret of $O(\log T)$ is achievable then Algorithm 1 needs to be conceptually modified.

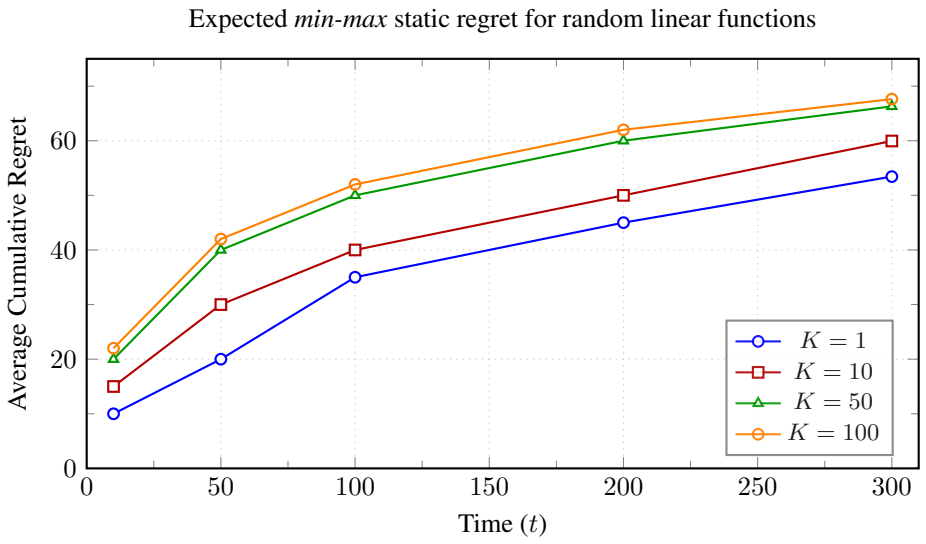

Figure 1: Expected *min-max* static regret of Algorithm 1 when f_t^k are random linear functions for different values of K.

7.2 GlobalExpertsProblem

In this section, we compare the performance of Algorithm 1 and MULTI [14] for the GlobalExpertsProblem with the i.i.d. input. For that purpose, we consider that the loss of each expert at time t is drawn independently from a uniform distribution:

$$\ell_{t,k} \overset{\text{i.i.d.}}{\sim} \mathsf{Uniform}(a, b), \quad t = 1, 2, \ldots, T, \ k = 1, 2, \ldots, K,$$

[3] In contrast to $\eta_{x,t} = O(1/\sqrt{t})$ as needed to prove Theorem 4.

[4] We have checked that other combinations only increase the regret.

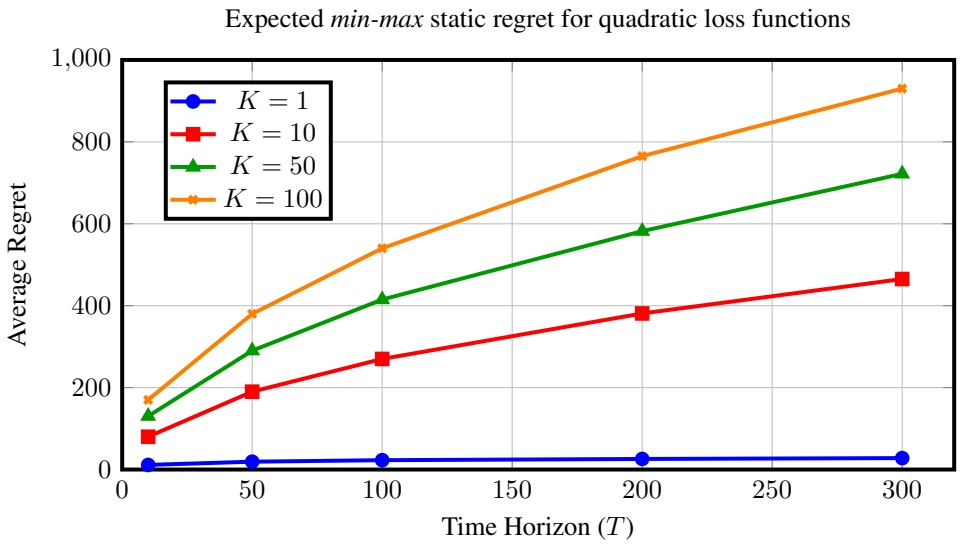

Figure 2: Expected *min-max* static regret of Algorithm 1 when f_t^k are random quadratic functions for different values of K.

where $a, b \in [0, 1]$ are fixed parameters defining the range of losses. In our experiments, we set $a = 0.2$ and $b = 0.8$ to ensure non-trivial loss variation while avoiding extreme loss values.

For this model, we plot the expected *min-max* static regret performance of Algorithm 1 and MULTI in Fig. 3, and see that both perform similarly, even though the regret guarantee we obtained for Algorithm 1 in i.i.d. case is slightly better than MULTI (which is valid even with the adversarial input). It appears that MULTI has better performance than its guarantee [14]. However, MULTI is computationally far more intensive than Algorithm 1 since it uses the $K = 2$ primitive recursively, and moreover, it needs to know the time horizon T in contrast to Algorithm 1.

7.3 Application of *min-max* formulation to Fair Classification Setting

Finally, in this subsection, we simulate the i.i.d. *min-max* OCO problem for a fair classification setting.

Data generation. We consider K groups, each corresponding to a convex loss function sequence. For each group $k \in [K]$, we sample a hidden parameter $w_k^\star \sim \mathcal{N}(0, I_d)$. At each round $t = 1, \ldots, T$, for each group k we draw m feature vectors

$$z_{t,k,i} \sim \mathcal{N}(\mu_k, \sigma^2 I_d), \quad i = 1, \ldots, m,$$

with μ_k drawn once per group. Labels are generated as

$$y_{t,k,i} \sim \text{Bernoulli}(\sigma(\langle w_k^\star, z_{t,k,i} \rangle)),$$

where $\sigma(u) = 1/(1 + e^{-u})$ is the sigmoid function.

Loss functions. The per-group convex loss at round t is the averaged logistic loss with L_2 regularization:

$$f_t^k(x) \;=\; \frac{1}{m} \sum_{i=1}^{m} \log\Big(1 + \exp(-y_{t,k,i} \, \langle x, z_{t,k,i} \rangle)\Big) \;+\; \kappa \|x\|_2^2.$$

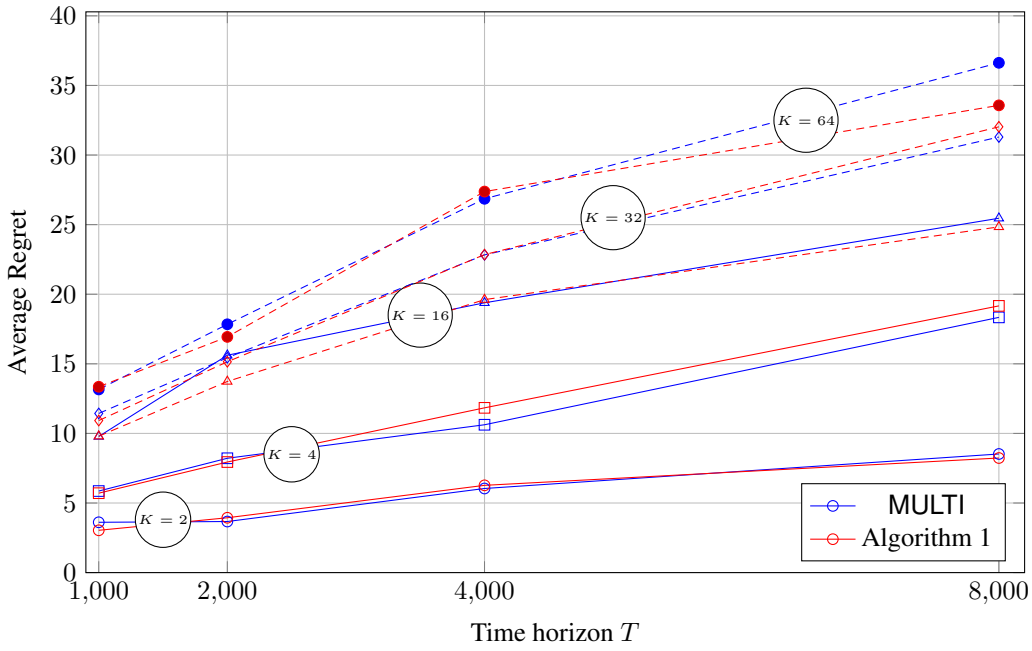

Figure 3: Expected *min-max* static regret of MULTI [14] and Algorithm 1 for different numbers of experts K and time horizons T for the fair classification example.

For this application, we compare the performance of following algorithms.

- **Algorithm 1 (OGD+*Hedge*)**
- **OGD on average loss**: projected OGD on $\frac{1}{K}\sum_{k=1}^{K} f_t^k(x)$.
- **Follow-the-Regularized-Leader (FTRL)**: $x_t = \arg\min_{x\in\mathcal{X}} \left\{\max_k \sum_{s=1}^{t-1} f_s^k(x) + \mathcal{R}(x)\right\}$.

Parameters. We use $d = 20$, $K = 10$, mini-batch size $m = 50$, diameter $D = 5$, and regularization $\kappa = 10^{-3}$. Step sizes are set as $\eta_{x,t} = D/(G\sqrt{t})$ for OGD (with G an upper bound on gradient norms) and $\eta_{\theta,t} = \sqrt{\ln K}/(B\sqrt{t})$ for *Hedge*, with B an upper bound on per-round losses.

In Fig. 4, we plot the expected *min-max* static regret $\bar{R}_T$ for all the three algorithms. We see that all algorithms have sub-linear regret, and Algorithm 1 has larger regret than FTRL but better than average OGD. However, recall that neither FTRL and average OGD have any theoretical guarantees on their expected *min-max* static regret. Next, in Fig. 5, we will consider an input where the index of the most challenging sequence changes over time and show that Algorithm 1 outperforms FTRL in that case.

Non-stationary input

Next, we change the input described in Section 7.3 to model a **highly non-stationary environment** specifically designed to challenge static optimization strategies such as FTRL and highlight the adaptability of Algorithm 1 that uses OGD+*Hedge*. The detailed changes with respect to Section 7.3 are as follows:

- We consider $K = 3$ separate objectives, each with its own logistic loss function defined by fixed but distinct parameter vectors $w_{\text{star}}[k]$ and mean feature shifts $\mu[k]$.

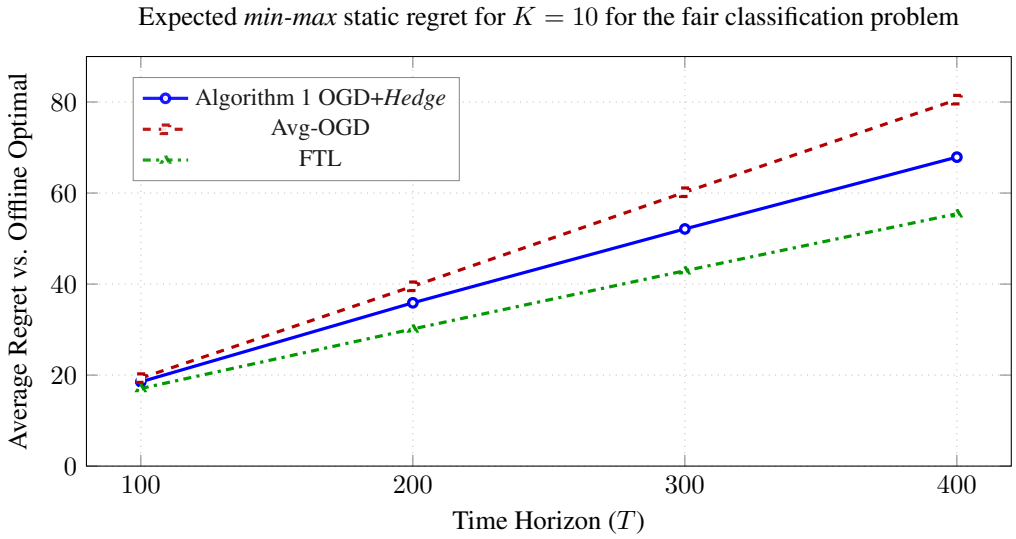

Figure 4: Expected *min-max* static regret comparison for $K = 10$.

- Every `switch_interval` steps (here set to 100), one objective becomes significantly harder by adding a **large mean shift** (`shift_magnitude` $= 5.0$) to its feature distribution. This creates abrupt changes in which objective dominates the cumulative loss objective function.
- **Data generation:** For each round t, and for each objective k, a feature matrix Z of size $m \times d$ is generated from a Gaussian distribution centered at the shifted mean. Labels y are generated from a logistic model using $w_{\text{star}}[k]$ (is a hidden "true" parameter vector for objective k. It defines how that objective behaves.), with additional noise introduced via random sampling.

We clearly see in Fig. 5 that Algorithm 1 (OGD+*Hedge*) adapts better to changing conditions than FTRL, since worst-case objective changes over time, making past loss information misleading for FTRL.

8 Conclusions

In this paper, we have made a basic advance in the area of multi-objective OCO by formulating the *min-max* static regret minimization problem, that requires an algorithm to closely track multiple loss function sequences simultaneously and at all times. Because of analytical challenges with the adversarial input, we considered the i.i.d. input setting, for which we proposed a simple algorithm that used a combination of the well known *Hedge* and OGD algorithm. Using an insightful regret decomposition, which made the analysis modular, we derived a $O(\sqrt{T \log K})$ regret bound on the proposed algorithm, which is shown to be best possible via simulations. We also extended the results to the bandit information setting. Certain questions that remain open are: i) what is the best regret possible with the adversarial input, ii) how to handle the case when all loss functions are strongly convex, and iii) handling constraints like COCO.

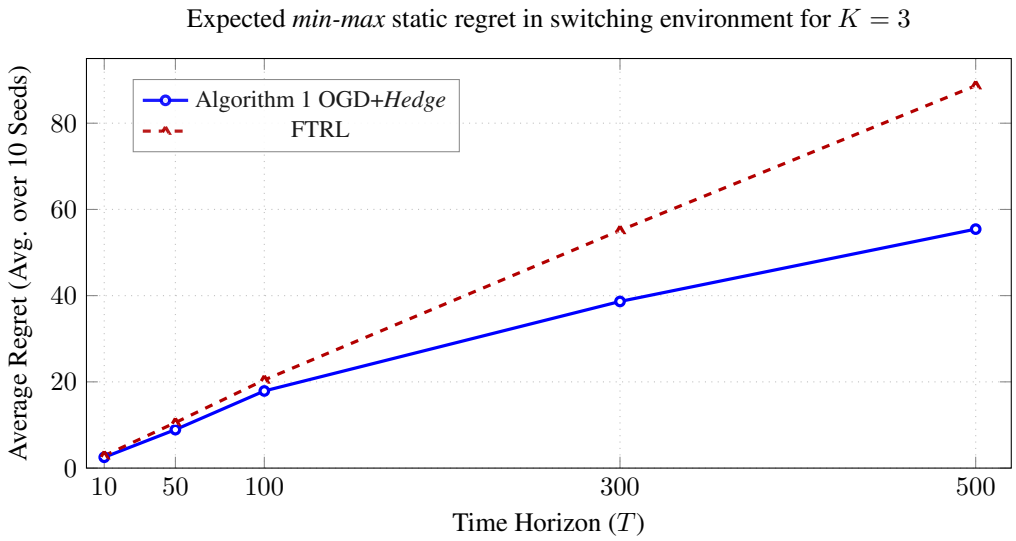

Figure 5: Expected *min-max* static regret for Algorithm 1 vs. FTRL in a piecewise-stationary environment.

References

[1] Alekh Agarwal, Elad Hazan, Satyen Kale, and Robert E. Schapire. 2010. Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*. Omnipress, 28–40.

[2] Yossi Azar, Uriel Felge, Michal Feldman, and Moshe Tennenholtz. 2014. Sequential decision making with vector outcomes. In *Proceedings of the 5th conference on Innovations in theoretical computer science*. 195–206.

[3] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. 2011. Theory and applications of robust optimization. *SIAM review* 53, 3 (2011), 464–501.

[4] Xuanyu Cao and KJ Ray Liu. 2018. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on automatic control* 64, 7 (2018), 2665–2680.

[5] Adrian Rivera Cardoso, Jacob Abernethy, He Wang, and Huan Xu. 2019. Competing Against Nash Equilibria in Adversarially Changing Zero-Sum Games. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 921–930. `https://proceedings.mlr.press/v97/cardoso19a.html`

[6] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

[7] Lijie Chen, Hongyang Zhang, and Tsui-Wei Weng. 2020. Practical Minimax Fair Classification. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[8] Tianyi Chen and Georgios B Giannakis. 2018. Bandit convex optimization for scalable and dynamic IoT management. *IEEE Internet of Things Journal* 6, 1 (2018), 1276–1286.

[9] Alexandra Chouldechova and Aaron Roth. 2020. The Frontiers of Fairness in Machine Learning. *Commun. ACM* 63, 5 (2020), 82–89.

[10] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2018. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 31.

[11] Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

[12] Sanmay Das and Arindam Banerjee. 2011. Meta-optimization and multi-objective online portfolio selection. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*. 1100–1105.

[13] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. 2015. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory* 61, 5 (2015), 2788–2806.

[14] Eyal Even-Dar, Robert Kleinberg, Shie Mannor, and Yishay Mansour. 2009. Online Learning with Global Cost Functions. In *22nd Annual Conference on Learning Theory, COLT*. `http://www.cs.mcgill.ca/~colt2009/papers/005.pdf#page=1`

[15] Abraham D. Flaxman, Adam Kalai, and H. Brendan McMahan. 2005. Online Convex Optimization in the Bandit Setting. In *Proceedings of the 16th Annual Conference on Learning Theory (COLT)*. Springer, 206–211.

[16] Hengquan Guo, Xin Liu, Honghao Wei, and Lei Ying. 2022. Online convex optimization with hard constraints: Towards the best of two worlds and beyond. *Advances in Neural Information Processing Systems* 35 (2022), 36426–36439.

[17] Fang Hao, Murali Kodialam, TV Lakshman, and Sarit Mukherjee. 2016. Online allocation of virtual machines in a distributed cloud. *IEEE/ACM Transactions on Networking* 25, 1 (2016), 238–249.

[18] Elad Hazan. 2019. Introduction to Online Convex Optimization. *CoRR* abs/1909.05207 (2019). arXiv:1909.05207 `http://arxiv.org/abs/1909.05207`

[19] Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. 2016. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*. PMLR, 402–411.

[20] Jiyan Jiang, Wenpeng Zhang, Shiji Zhou, Lihong Gu, Xiaodong Zeng, and Wenwu Zhu. 2023. Multi-Objective Online Learning. In *The Eleventh International Conference on Learning Representations*. `https://openreview.net/forum?id=dKkMnCWfVmm`

[21] Thomas Kesselheim and Sahil Singla. 2020. Online learning with vector costs and bandits with knapsacks. In *Conference on Learning Theory*. PMLR, 2286–2305.

[22] Panos Kouvelis and Gang Yu. 1997. *Robust Discrete Optimization and Its Applications*. Springer.

[23] Daniel Lee, Georgy Noarov, Mallesh Pai, and Aaron Roth. 2022. Online minimax multiobjective optimization: Multicalibeating and other applications. *Advances in Neural Information Processing Systems* 35 (2022), 29051–29063.

[24] Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. 2019. Cautious regret minimization: Online optimization with long-term budget constraints. In *International Conference on Machine Learning*. PMLR, 3944–3952.

[25] Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. 2012. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking* 21, 5 (2012), 1378–1391.

[26] Qingsong Liu and Mohammad Hajiesmaili. 2025. Online Fair Allocation of Reusable Resources. *Proc. ACM Meas. Anal. Comput. Syst.* 9, 2, Article 29 (June 2025), 46 pages. doi:10.1145/3727121

[27] Qingsong Liu, Wenfei Wu, Longbo Huang, and Zhixuan Fang. 2022. Simultaneously achieving sublinear regret and constraint violations for online convex optimization with time-varying constraints. *ACM SIGMETRICS Performance Evaluation Review* 49, 3 (2022), 4–5.

[28] Guodong Lyu, Wang Chi Cheung, Chung-Piaw Teo, and Hai Wang. 2019. Multi-objective online ride-matching. *Available at SSRN* 3356823 (2019), 1–41.

[29] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. 2012. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research* 13, 1 (2012), 2503–2528.

[30] Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. 2013. Stochastic convex optimization with multiple objectives. *Advances in neural information processing systems* 26 (2013).

[31] Anton Milan, S Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. 2017. Online multi-target tracking using recurrent neural networks. In *Proceedings of the AAAI conference on Artificial Intelligence*, Vol. 31.

[32] Michael J Neely. 2010. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks* 3, 1 (2010), 1–211.

[33] Michael J Neely and Hao Yu. 2017. Online convex optimization with time-varying constraints. *arXiv preprint arXiv:1702.04783* (2017).

[34] Alexander Rakhlin and Karthik Sridharan. 2013. Optimization, Learning, and Games with Predictable Sequences. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*. 1–27. http://proceedings.mlr.press/v30/rakhlin13.pdf

[35] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. 2011. Online Learning: Beyond Regret. In *Proceedings of the 24th Annual Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 19)*, Sham M. Kakade and Ulrike von Luxburg (Eds.). PMLR, Budapest, Hungary, 559–594. https://proceedings.mlr.press/v19/rakhlin11a.html

[36] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. 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. https://proceedings.neurips.cc/paper_files/paper/2018/file/432aca3a1e345e339f35a30c8f65edce-Paper.pdf

[37] Ohad Shamir. 2017. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research* 18, 52 (2017), 1–11.

[38] Abhishek Sinha, Ativ Joshi, Rajarshi Bhattacharjee, Cameron Musco, and Mohammad Hajiesmaili. 2023. No-regret Algorithms for Fair Resource Allocation. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 48083–48109. https://proceedings.neurips.cc/paper_files/paper/2023/file/96842011407c2691ab4eefff48fc864d-Paper-Conference.pdf

[39] Abhishek Sinha and Rahul Vaze. 2024. Optimal Algorithms for Online Convex Optimization with Adversarial Constraints. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=TxffvJMnBy

[40] Abhishek Sinha and Rahul Vaze. 2025. Beyond $\tilde{O}(\sqrt{T})$ Constraint Violation for Online Convex Optimization with Adversarial Constraints. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems, NeurIPS 2025*. https://openreview.net/forum?id=yK4Xu7DDd6

[41] Wen Sun, Debadeepta Dey, and Ashish Kapoor. 2017. Safety-aware algorithms for adversarial contextual bandit. In *International Conference on Machine Learning*. PMLR, 3280–3288.

[42] Rahul Urgaonkar, Ulas C. Kozat, Ken Igarashi, and Michael J. Neely. 2010. Dynamic resource allocation and power management in virtualized data centers. In *2010 IEEE Network Operations and Management Symposium - NOMS 2010*. 479–486. doi:10.1109/NOMS.2010.5488484

[43] Guy Uziel and Ran El-Yaniv. 2017. Multi-objective non-parametric sequential prediction. *Advances in Neural Information Processing Systems* 30 (2017).

[44] Rahul Vaze. 2022. On Dynamic Regret and Constraint Violations in Constrained Online Convex Optimization. In *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. 9–16. doi:10.23919/WiOpt56218.2022.9930613

[45] Rahul Vaze and Abhishek Sinha. 2025. $O(\sqrt{T})$ Static Regret and Instance Dependent Constraint Violation for Constrained Online Convex Optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems, NeurIPS 2025*. https://openreview.net/forum?id=YmbQ0qnQ76

[46] Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Tianyou Chai, and Karl Johansson. 2021. Regret and cumulative constraint violation analysis for online convex optimization with long term constraints. In *International Conference on Machine Learning*. PMLR, 11998–12008.

[47] Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Yiguang Hong, Tianyou Chai, and Karl H Johansson. 2023. Distributed Online Convex Optimization with Adversarial Constraints: Reduced Cumulative Constraint Violation Bounds under Slater's Condition. *arXiv preprint arXiv:2306.00149* (2023).

[48] Hao Yu, Michael Neely, and Xiaohan Wei. 2017. Online convex optimization with stochastic constraints. *Advances in Neural Information Processing Systems* 30 (2017).

[49] Jianjun Yuan and Andrew Lamperski. 2018. Online convex optimization for cumulative constraints. *Advances in Neural Information Processing Systems* 31 (2018).

A Proof of Lemma 5

Let $S_1 := \sum_{t=1}^T f_t(x_t)$ and $S_2 := \sum_{t=1}^T g_t(x_t)$, and recall that $\lambda_t = (f_t(x_t), g_t(x_t))$ and $\sum_t \Lambda_t(x_t, \theta_t) = \theta_{t1} f_t + \theta_{t2} g_t = \sum_t \langle \theta_t, \lambda_t \rangle$. Then

$$\max_\theta \sum_t \Lambda_t(x_t, \theta) = \max\{S_1, S_2\}.$$

Standard *Hedge* (gains version) [6] analysis implies that for any expert $i = 1, 2$.

$$\sum_{t=1}^T \langle \theta_t, \lambda_t \rangle \geq S_i - \frac{\ln 2}{\eta_{\theta,T}} - \sum_{t=1}^T \frac{\eta_{\theta,t}}{8} (\lambda_{t,1} - \lambda_{t,2})^2.$$

Maximizing over i and using $(\lambda_{t,1} - \lambda_{t,2})^2 \leq (2B)^2 = 4B^2$ (since values are bounded by B from Assumption 1) yields

$$\sum_{t=1}^T \langle \theta_t, \lambda_t \rangle \geq \max\{S_1, S_2\} - \frac{\ln 2}{\eta_{\theta,T}} - \sum_{t=1}^T \frac{\eta_{\theta,t} 4B^2}{8}.$$

Rearranging gives

$$R_1 = \max_{\theta \in \Delta_2} \sum_{t=1}^T \Lambda_t(x_t, \theta) - \sum_{t=1}^T \Lambda_t(x_t, \theta_t) = \max\{S_1, S_2\} - \sum_{t=1}^T \langle \theta_t, \lambda_t \rangle \leq \frac{\ln 2}{\eta_{\theta,T}} + \sum_{t=1}^T \frac{\eta_{\theta,t} B^2}{2}.$$

B Proof of Lemma 6

Define $\tilde{f}_t(x) := \Lambda_t(x, \theta_t)$, which is convex for fixed θ_t. Thus, the standard OGD (projected) guarantee from [18] implies, for any $x \in \mathcal{X}$,

$$\sum_{t=1}^T \tilde{f}_t(x_t) - \sum_{t=1}^T \tilde{f}_t(x) \leq \frac{\|x - x_0\|^2}{2\eta_{x,T}} + \sum_{t=1}^T \frac{\eta_{x,t}}{2} \|\nabla \tilde{f}_t(x_{t-1})\|^2.$$

Using the finite diameter $\|x - x_0\| \leq D$ and the bounded gradient $\|\nabla \tilde{f}_t(x_{t-1})\| \leq G$ (since both $\|\nabla f_t(x_{t-1})\| \leq G$ and $\|\nabla g_t(x_{t-1})\| \leq G$ from Assumption 1) yields

$$R_2 \leq \frac{D^2}{2\eta_{x,T}} + \sum_{t=1}^T \frac{\eta_{x,t} G^2}{2}.$$

With the choice $\eta_{x,t} = D/(G\sqrt{t})$

$$R_2 \leq DG\sqrt{T}.$$

C Proof of Lemma 7

Recall that

$$\lambda_t(x) := (f_t(x), g_t(x)), \qquad \text{and let} \quad \mu(x) := \mathbb{E}\{\lambda_t(x)\}.$$

where $\mu(x)$ is independent of index t on account of the i.i.d. assumption on functions f_t, g_t. Let $\mathcal{F}_{t-1}$ denote the filtration generated by

$$\mathcal{F}_{t-1} := \sigma\big(x_1, \ldots, x_{t-1}, \theta_1, \ldots, \theta_{t-1}, \lambda_1(x_1), \ldots, \lambda_{t-1}(x_{t-1})\big),$$

which determines θ_t at round t.

Recall the definition of optimal pair $x^\star$ and $\theta^\star$ that defines the static offline optimal solution of the expected min-max problem (17):

$$x^\star := \arg\min_{x\in\mathcal{X}} \max_{\theta\in\Delta_2} \sum_{t=1}^{T} \theta^\top \mu(x), \quad \text{and} \quad \theta^\star := \arg\max_{\theta\in\Delta_2} \sum_{t=1}^{T} \theta^\top \mu(x^\star). \tag{24}$$

By the definition of R_3,

$$\begin{aligned} R_3 &= \min_{x\in\mathcal{X}} \sum_{t=1}^{T} \theta_t^\top \lambda_t(x) - \min_{x\in\mathcal{X}} \max_{\theta} \sum_{t=1}^{T} \theta^\top \lambda_t(x) \\ &\overset{(a)}{=} \min_{x\in\mathcal{X}} \sum_{t=1}^{T} \theta_t^\top \lambda_t(x) - \sum_{t=1}^{T} \theta^{\star\top} \lambda_t(x^\star), \\ &\overset{(b)}{\leq} \sum_{t=1}^{T} \theta_t^\top \lambda_t(x^\star) - \sum_{t=1}^{T} \theta^{\star\top} \lambda_t(x^\star), \\ &= \sum_{t=1}^{T} (\theta_t - \theta^\star)^\top \lambda_t(x^\star), \end{aligned} \tag{25}$$

where (a) follows from the definition of $x^\star$ and (b) follows since $x^\star \in \mathcal{X}$ is feasible. Moreover, we have

$$\mathbb{E}\left\{ (\theta_t - \theta^\star)^\top \lambda_t(x^\star) \,\middle|\, \mathcal{F}_{t-1} \right\} = (\theta_t - \theta^\star)^\top \mathbb{E}\{\lambda_t(x^\star)|\mathcal{F}_{t-1}\} = (\theta_t - \theta^\star)^\top \mu(x^\star), \tag{26}$$

where the first equality follows since θ_t is completely determined by $\mathcal{F}_{t-1}$, while the second equality follows because $\mathbb{E}\{\lambda_t(x^\star)|\mathcal{F}_{t-1}\} = \mu(x^\star)$ from the i.i.d. assumption.

We also have

$$(\theta_t - \theta^\star)^\top \mu(x^\star) \leq 0$$

for each $t = 1, \ldots, T$ using the optimality of $\theta^\star$ (24) since

$$\theta^\star \in \arg\max_{\theta\in\Delta_2} \theta^\top \mu(x^\star)$$

depends *only* on the expected vector $\mu(x^\star)$ and not on any particular realization of $\lambda_t(x^\star)$.

Thus, we get from (26)

$$\mathbb{E}\left\{ (\theta_t - \theta^\star)^\top \lambda_t(x^\star) \,\middle|\, \mathcal{F}_{t-1} \right\} \leq 0, \tag{27}$$

Using (27) in (25) and taking the expectation (using the tower property of conditional expectation),

$$\mathbb{E}\{R_3\} \leq \sum_{t=1}^{T} (\theta_t - \theta^\star)^\top \mu(x^\star) \leq 0. \tag{28}$$

D Per-slot $min - max$ multi-objective optimization [23]

In this section, we show that benchmark W_L^T(10) can be $3/2$ times more than our benchmark C_{OPT} (4).

Input Let $T = 2N$ and $\mathcal{X} = [0, 1]$. For $j = 1, \ldots, N$ let

$$f_{2j-1}(x) = 1.2 - 0.2x, \quad f_{2j}(x) = x, \qquad g_{2j-1}(x) = x, \quad g_{2j}(x) = 0.8 + 0.2x.$$

Odd time slots: $\min_{x\in[0,1]} \max\{1.2-0.2x, x\} = 1$ (attained at $x = 1$). Even time slots rounds: $\min_{x\in[0,1]} \max\{x, 0.8+0.2x\} = 0.8$ (attained at $x = 0$). Thus the benchmark $W_L^T(10)$ is

$$\sum_{t=1}^{T} \min_{x\in\mathcal{X}} \max\{f_t(x), g_t(x)\} = N(1 + 0.8) = 1.8N.$$

For any fixed x, for a pair of time slots $2j - 1, 2j$, the sequence f_t and g_t's cost is

$$\sum_{t=2j-1}^{2j} f_t(x) = 1.2 + 0.8x, \qquad \sum_{t=2j-1}^{2j} g_t(x) = 0.8 + 1.2x.$$

Thus, C_{OPT} (4) is

$$C_{\mathsf{OPT}} = \min_{x\in\mathcal{X}} \max\left\{\sum_{j=1}^{N}\sum_{t=2j-1}^{2j} 1.2 + 0.8x, \sum_{j=1}^{N}\sum_{t=2j-1}^{2j} 0.8 + 1.2x\right\} = 1.2N \quad (\text{at } x^\star = 0).$$

Hence,

$$\frac{W_L^T}{C_{\mathsf{OPT}}} = \frac{1.8N}{1.2N} = 1.5,$$

as claimed. Therefore, even if an online algorithm $\mathcal{A}$ achieves sub-linear regret with respect to the benchmark $W_L^T(10)$, it does not necessarily have sub-linear regret for our benchmark C_{OPT} (4).

E Experts Problem (Gains Version)

We are given N experts. The interaction proceeds for T rounds as follows:

- On each round $t = 1, 2, \ldots, T$:
  1. Each expert $i \in \{1, \ldots, N\}$ receives a gain $g_{t,i} \in [0, 1]$.
  2. The learner chooses a probability distribution
  $$p_t = (p_{t,1}, \ldots, p_{t,N}),$$
  where $p_{t,i} \geq 0$ and $\sum_{i=1}^{N} p_{t,i} = 1$.
  3. The learner's gain is the expected value
  $$G_t = \sum_{i=1}^{N} p_{t,i}\, g_{t,i}.$$

The learner's goal is to compete with the best fixed expert in hindsight by minimizing the regret

$$R_T = \max_i \sum_{t=1}^{T} g_{t,i} \; - \; \sum_{t=1}^{T} G_t.$$

Hedge Algorithm (Gains Version)

Algorithm 3 Hedge (Gains Version)

Require: Number of experts N, learning rate $\eta > 0$

1: Initialize weights: $w_{1,i} \leftarrow 1$ for all $i \in \{1, \ldots, N\}$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Form probabilities:

$$p_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}} \quad \forall i$$

4: Play according to distribution $p_t = (p_{t,1}, \ldots, p_{t,N})$
5: Observe gains $g_{t,i} \in [0, 1]$ for each expert i
6: Update weights:

$$w_{t+1,i} \leftarrow w_{t,i}\, e^{\eta g_{t,i}} \quad \forall i$$

7: **end for**

Guarantee

Theorem 9. *[6] With $\eta = \sqrt{\frac{2 \ln N}{T}}$, Hedge achieves*

$$\max_i \sum_{t=1}^{T} g_{t,i} - \sum_{t=1}^{T} G_t \;\; \leq \;\; \sqrt{2T \ln N}.$$

F Bounding R_3 with the adversarial input

In this section, we show the best bound one can obtain for R_3 on a sample path basis for Algorithm 1 with the adversarial input is $O(\eta_\theta T^2)$.

Theorem 10. *Under Assumption 1, i.e., $\|\lambda_t(x)\|_\infty \leq B$, $R_3 \leq 2B^2 T \sum_{t=1}^{T} \eta_{\theta,t}$.*

Recall the definitions $\lambda_t(x) = (f_t(x), g_t(x))$,

$$R_3 \;=\; \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \langle \theta_t, \lambda_t(x) \rangle - C_{\mathsf{OPT}}, \qquad \text{and} \qquad C_{\mathsf{OPT}} = \min_{x \in \mathcal{X}} \max_{\theta \in \Delta_2} \sum_{t=1}^{T} \langle \theta, \lambda_t(x) \rangle.$$

Thus, an almost natural way (or really the only way for Algorithm 1 that combines OGD+*Hedge*) to bound R_3 is via exploiting the stability $\sum_t \|\theta_{t+1} - \theta_t\|_1$ of the *Hedge* algorithm. Following this path, we next derive Lemma 11, Lemma 12 and Lemma 13 which when combined imply Theorem 10.

However, it is not sufficient since by substituting $\eta_{\theta,t} = \frac{1}{\sqrt{T}}$ for all t, the usual *Hedge* step size, we get $R_3 = \Omega(T^{3/2})$.

Lemma 11. *Define the time-averaged vector $\bar{\theta} := \frac{1}{T} \sum_{t=1}^{T} \theta_t \in \Delta_2$. Then,*

$$R_3 \leq B \sum_{t=1}^{T} \|\theta_t - \bar{\theta}\|_1.$$

Proof. Since $\bar{\theta}$ is a particular (constant) element of Δ_2, for every $x \in \mathcal{X}$

$$\sum_{t=1}^{T} \langle \bar{\theta}, \lambda_t(x) \rangle \leq \max_{\theta \in \Delta_2} \sum_{t=1}^{T} \langle \theta, \lambda_t(x) \rangle,$$

and therefore taking the minimum over x yields

$$\min_{x \in \mathcal{X}} \sum_{t=1}^{T} \langle \bar{\theta}, \lambda_t(x) \rangle \leq C_{\mathsf{OPT}}.$$

Thus,

$$\begin{aligned} R_3 &= \min_x \sum_{t=1}^{T} \langle \theta_t, \lambda_t(x) \rangle - C_{\mathsf{OPT}} \\ &\leq \min_x \sum_{t=1}^{T} \langle \theta_t, \lambda_t(x) \rangle - \min_x \sum_{t=1}^{T} \langle \bar{\theta}, \lambda_t(x) \rangle \\ &\leq \max_x \sum_{t=1}^{T} \big\langle \theta_t - \bar{\theta}, \lambda_t(x) \big\rangle, \end{aligned}$$

where the last inequality follows as below.

Let $x^\star$ satisfy $u(x^\star) = \min_x u(x)$. Then

$$\min_x u(x) - \min_x v(x) = u(x^\star) - \min_x v(x) \leq u(x^\star) - v(x^\star) \leq \max_x (u(x) - v(x)),$$

which proves the claim.

This proves

$$R_3 \leq \max_{x \in \mathcal{X}} \sum_{t=1}^{T} \langle \theta_t - \bar{\theta}, \lambda_t(x) \rangle. \tag{29}$$

Since we are in the bounded setting, for all rounds t and all $x \in \mathcal{X}$ we have $\|\lambda_t(x)\|_\infty \leq B$. Thus, appyling the Hölder's inequality (using duality of $||.||_1$ and $||.||_\infty$ norms) on (29),

$$\sum_{t=1}^{T} \langle \theta_t - \bar{\theta}, \lambda_t(x) \rangle \leq \sum_{t=1}^{T} \|\theta_t - \bar{\theta}\|_1 \, \|\lambda_t(x)\|_\infty \leq B \sum_{t=1}^{T} \|\theta_t - \bar{\theta}\|_1.$$

Thus,

$$R_3 \leq B \sum_{t=1}^{T} \|\theta_t - \bar{\theta}\|_1.$$

This is the desired inequality. □

Next, we state an elementary inequality without proof.

Lemma 12 (Deviation from average via consecutive differences)**.** *Let $\theta_1, \ldots, \theta_T \in \mathbb{R}^d$ be vectors, and define the average*

$$\bar{\theta} = \frac{1}{T} \sum_{t=1}^{T} \theta_t.$$

Then

$$\sum_{t=1}^{T} \|\theta_t - \bar{\theta}\|_1 \;\leq\; 2T \sum_{t=1}^{T-1} \|\theta_{t+1} - \theta_t\|_1.$$

Lemma 13 (One-step TV bound for Hedge)**.** *Let $\{\theta_t\}$ be the Hedge distributions updated by*

$$\theta_{t+1,i} \;=\; \frac{\theta_{t,i}\, e^{\eta_{\theta,t}\lambda_{t,i}}}{\sum_j \theta_{t,j}\, e^{\eta_{\theta,t}\lambda_{t,j}}}, \qquad i = 1, \dots, n,$$

with learning rate $\eta_{\theta,t} > 0$. Assume the losses satisfy $\lambda_{t,i} \in [0, B]$ for all i. Then

$$\|\theta_{t+1} - \theta_t\|_1 \le \eta_{\theta,t} B.$$

Proof. Let

$$Z_t := \sum_j \theta_{t,j} e^{\eta_{\theta,t}\lambda_{t,j}} \;=\; \mathbb{E}_{i\sim\theta_t}\big[e^{\eta_{\theta,t}\lambda_{t,i}}\big].$$

By the definition of Kullback–Leibler divergence,

$$\begin{aligned} D_{\mathrm{KL}}(\theta_t\|\theta_{t+1}) &= \sum_i \theta_{t,i} \ln \frac{\theta_{t,i}}{\theta_{t+1,i}} = \sum_i \theta_{t,i} \ln \frac{\theta_{t,i}}{\theta_{t,i} e^{\eta_{\theta,t}\lambda_{t,i}}/Z_t} \\ &= \sum_i \theta_{t,i} (\ln Z_t - \eta_{\theta,t}\lambda_{t,i}) = \ln Z_t - \eta_{\theta,t}\, \mathbb{E}_{i\sim\theta_t}[\lambda_{t,i}]. \end{aligned}$$

Let $\mu := \mathbb{E}_{i\sim\theta_t}[\lambda_{t,i}]$. Then $\ln Z_t = \ln \mathbb{E}_{\theta_t}\big[e^{\eta_{\theta,t}\lambda_{t,i}}\big] = \eta_{\theta,t}\mu + \ln \mathbb{E}_{\theta_t}\big[e^{\eta_{\theta,t}(\lambda_{t,i}-\mu)}\big]$, so (1) is equivalently

$$D_{\mathrm{KL}}(\theta_t\|\theta_{t+1}) = \ln \mathbb{E}_{\theta_t}\big[e^{\eta_{\theta,t}(\lambda_{t,i}-\mu)}\big]. \tag{30}$$

Thus to upper bound the one-step KL it suffices to bound the centered MGF $\ln \mathbb{E}[e^{\eta_{\theta,t}(\lambda_{t,i}-\mu)}]$.

Proposition 14 (Hoeffding's MGF bound)**.** *For a random variable X that satisfies $X \in [a, b]$ almost surely and $\mathbb{E}[X] = \mu$, then for any $s \in \mathbb{R}$*

$$\ln \mathbb{E}[e^{s(X-\mu)}] \le \frac{s^2(b-a)^2}{8}. \tag{31}$$

Since $\lambda_{t,i} \in [0, B]$, using (31) in (30) with $X = \lambda_{t,i}$, $[a, b] = [-B, B]$, $s = \eta_{\theta,t}$, and mean μ, we get

$$D_{\mathrm{KL}}(\theta_t\|\theta_{t+1}) \le \frac{\eta_{\theta,t}^2 4B^2}{8}. \tag{32}$$

Proposition 15 (Pinsker's inequality [11])**.** *For any two probability vectors p, q,*

$$\|p - q\|_1 \le \sqrt{2\, D_{\mathrm{KL}}(p\|q)}. \tag{33}$$

Since both θ_t and θ_{t+1} are distributions, applying (33) on (32), we get

$$\|\theta_{t+1} - \theta_t\|_1 \le \sqrt{2 \cdot \frac{\eta_{\theta,t}^2 B^2}{2}} = \eta_{\theta,t} B,$$

the claimed bound. □