Trading Off Resource Budgets For Improved Regret Bounds

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider a variant of adversarial online learning where in each round one picks B arms and incurs cost equal to the minimum of the costs of each arm chosen. We study the trade-off between the budget B and the regret R_T^* relative to the *single* best fixed arm in hindsight. By adapting the techniques of Kalai and Vempala [2005], we show that in the full feedback setting Following the Top B perturbed Leaders (FTL) achieves expected regret $\mathcal{O}(T^{\frac{1}{B+1}} \ln(N)^{\frac{B}{B+1}})$ with N arms over time horizon T, and allows one to trade off an arm budget for improved regret bounds. We observe that algorithms which use standard regret minimizers as subroutines can sometimes be adapted by replacing these subroutines with FTL. We use this observation to generalize existing algorithms for Online Submodular Function Maximization [Streeter and Golovin, 2008] in both the full feedback and semi-bandit feedback settings to allow trade-offs between resource budgets and regret bounds, and we empirically evaluate these new algorithms on an online hyperparameter optimization problem. Likewise, we show how FTL can lead to new algorithms for Linear Programming which require stronger oracles at the benefit of fewer oracle calls.

1 Introduction

Adversarial online learning is a well-studied framework for sequential decision making with numerous applications. In round $t = 1, \ldots, T$, an adversary chooses a hidden cost function $c_t : \mathcal{A} \to [0, 1]$ from a set $\mathcal{A}$ to costs in $[0, 1]$. An algorithm must then choose an arm $a_t \in \mathcal{A}$, and incurs cost $c_t(a_t)$. In the *full feedback* setting (Online Learning with Experts (OLwE)), the algorithm then observes the cost function c_t. In the *partial feedback* setting (Multi-Armed Bandits (MAB)) the algorithm only observes the incurred cost $c_t(a_t)$. The objective is to give algorithms which minimize the difference between the algorithm's cumulative cost and the cumulative cost of the single best arm in hindsight. In this paper we consider a search-like variant of these problems where in each round one can pick a subset of arms $S_t \subset \mathcal{A}$ with $|S_t| = B \geq 1$, and keep the arm with the lowest cost. Three examples include:

1. Online algorithm portfolios [Gomes and Selman, 2001]: In each round t, one receives a problem instance x_t, and can pick a subset $S_t \subset \mathcal{A}$ of algorithms to run in parallel to solve x_t. For example, x_t could be a boolean satisfiability (SAT) problem, and $\mathcal{A}$ could be a collection of different SAT solving algorithms. We let $c_t(a) = 0$ if a solves x_t and $c_t(a) = 1$ otherwise. Then if any $a \in S_t$ finds a solution to x_t we incur 0 cost in this round. Another example is online hyperparameter optimization (see Section 4).
2. Online bidding [Chen et al., 2016a]: In each round t, an auctioneer sets up a first-price auction for bidders $S_t \subset \mathcal{A}$. Each bidder $a \in \mathcal{A}$ has a price $1 - c_t(a)$ they are willing to pay,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

and the auctioneer receives $\max_{a \in S_t} 1 - c_t(a) = 1 - \min_{a \in S_t} c_t(a)$, and so maximizing revenue is equivalent to minimizing costs.

3. Adaptive network routing [Awerbuch and Kleinberg, 2008]: In each round t, a network router receives a data packet x_t and can pick a selection of network routes $S_t \subset \mathcal{A}$ to send it to its destination via in parallel. Let the cost $c_t(a)$ of a route a be the total time taken for x_t to reach its destination via a; the router receives cost $\min_{a \in S_t} c_t(a)$ equal to the smallest delay.

In many natural applications the budget B is a restricted resource (e.g. compute time) we would like to keep small. This paper studies how one can trade off budget resources for guarantees on the standard regret objective. Formally, for any randomized algorithm **ALG** which chooses subset $S_t \subset \mathcal{A}$ in round t, and where $c_t(S_t) := \min_{a \in S_t} c_t(a)$,

$$R_T^*(\textbf{ALG}) := \max_{c_1, \ldots, c_T} \mathbb{E}\left[\sum_{t=1}^{T} c_t(S_t) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(a^*)\right]$$

is the worst-case expected regret of **ALG** relative to the single best arm in hindsight, where the expectation is with respect to the randomness of **ALG**.[1] What guarantees can we give on R_T^* as a function of our budget B? In the full feedback setting when $B = 1$, this is the standard OLwE problem where it is known that $R_T^* = \Omega(\sqrt{T})$ and there are algorithms which achieve $R_T^* \leq 2\sqrt{T \ln(N)}$ [Lattimore and Szepesvári, 2020] where $N = |\mathcal{A}|$. When $B = N$ the algorithm which chooses $S_t = \mathcal{A}$ in each round achieves $R_T^* = 0$. But what bounds on R_T^* can be achieved in the intermediate regime when $1 < B < N$? To the best of our knowledge this question has not been directly answered by any prior work.

Theoretical results: We present a new algorithm, *Follow the Top perturbed Leaders* (**FTL**), and show that in the full feedback setting $R_T^*(\textbf{FTL}) \leq \mathcal{O}(T^{\frac{1}{B+1}} \ln(N)^{\frac{B}{B+1}})$. This allows for a direct trade-off between the budget B and regret bounds, and recovers the familiar $\mathcal{O}(\sqrt{T \ln(N)})$ bound when $B = 1$. We show this can be converted to $R_T^*(\textbf{FTL}) \leq \mathcal{O}(T^{\frac{1}{B+1}} (K \ln(N))^{\frac{B}{B+1}})$ in the semi-bandit feedback setting if one has unbiased cost estimators bounded in $[0, K]$.

Online Submodular Function Maximization (OSFM) is a problem which has applications in algorithm portfolio design, database query processing and sensor placement. When given a budget of B per round, Streeter and Golovin [2008] give an algorithm **OG** which minimizes regret with respect to the best fixed schedule of length B hindsight. We present $\textbf{OG}_{\textbf{hybrid}}$, a generalization of **OG** which uses **FTL** as a subroutine and is capable of trading off a budget constraint B for regret bounds against the best length $B' \leq B$ schedule in hindsight (See Section 3 for details).

Techniques: Minimizing R_T^* is an important subroutine for a large variety of applications including Linear Programming, Boosting, and solving zero sum games [Arora et al., 2012]. Traditionally an experts algorithm such as **Hedge** [Littlestone and Warmuth, 1994], which pulls a single arm per round, will be used as a subroutine to minimize R_T^*. We highlight how in the cases of OSFM and Linear Programming, one can simply replace a single arm R_T^*-minimizing subroutine with **FTL** and get performance bounds with little or no alteration to the original proofs. The resulting algorithms have improved bounds (due to improved bounds on R_T^* when $B > 1$) at the cost of qualitatively changing the algorithm (e.g. requiring a larger budget or more powerful oracle). This is significant because it highlights the potential of how bounds on R_T^* when $B > 1$ can lead to new results in other application areas. In Section 2.1 we highlight how the proof techniques of Kalai and Vempala [2005] for bounding R_T^* in the traditional experts setting can naturally be generalized to the case when $B > 1$, which may be of independent interest.

Experimental results: We benchmark both **FTL** and $\textbf{OG}_{\textbf{hybrid}}$ in a setting similar to the NeurIPS BBO challenge [Turner et al., 2021] for the problem of online black-box hyperparameter optimization. We find that both these algorithms outperform **OG** for various compute budgets. We are able to explain why this happens for this specific dataset, and discuss the scenarios under which each algorithm would perform better.

[1] Here we consider an oblivious adversary model for simplicity, but we believe the results of this paper carry through to adaptive adversaries as well.

1.1 Relation to prior work

One can alternatively formulate the problem we consider as receiving the maximum *reward* $r_t(a) = 1 - c_t(a)$ of each arm chosen instead of the minimum cost. This maximum of rewards formulation fits within the framework of Online Submodular Function Maximization (OSFM), and has also been separately studied as the K-MAX problem (here $K = B$) Chen et al. [2016a]. In the OSFM setting, Streeter and Golovin [2008] give an online greedy approximation algorithm which guarantees $\mathbb{E}[(1 - e^{-1})\text{OPT}(B) - \text{Reward}_T] \leq \mathcal{O}(\sqrt{TB\ln(N)})$ in the full feedback adversarial setting, where $\text{OPT}(B)$ is the cumulative reward of the best fixed subset of size B in hindsight, and Reward_T is the cumulative reward of the algorithm. A similar bound of $\mathcal{O}(B\sqrt{TN\ln(N)})$ can be given in a semi-feedback setting. Conversely in the full feedback setting, Streeter and Golovin [2007] shows that any algorithm has worst-case regret $\mathbb{E}[\text{OPT}(B) - \text{Reward}_T] \geq \Omega(\sqrt{TB\ln(N/B)})$ when one receives the maximum of rewards in each round. Chen et al. [2016a] study the K-MAX problem and other non-linear reward functions in the stochastic combinatorial multi-armed bandit setting. Assuming the rewards satisfy certain distributional assumptions, they give an algorithm which achieves distribution-independent regret bounds of $\mathbb{E}[(1 - \varepsilon)\text{OPT}(B) - \text{Reward}_T] \leq \mathcal{O}(\sqrt{TBN\ln(T)})$ for $\varepsilon > 0$ with semi-bandit feedback. Note however that we consider the adversarial setting in this paper.

More broadly, these problems fall within the combinatorial online learning setting where an algorithm may pull a subset of arms in each round. Much prior work has focused on combinatorial bandits where the reward is linear in the subset of arms chosen, which can model applications including online advertising and online shortest paths [Cesa-Bianchi and Lugosi, 2012, Audibert et al., 2014, Combes et al., 2015]. The case of non-linear reward is comparatively less studied, but having non-linear rewards (such as max) allows one to model a wider variety of problems including online expected utility maximization [Li and Deshpande, 2011, Chen et al., 2016a]. As some examples of prior work in the stochastic setting, [Gopalan et al., 2014] uses Thompson Sampling to deal with non-linear rewards of functions of subsets of arms (including the max function), but requires the rewards to come from a known parametric distribution. Chen et al. [2016b] considers a model where the subset of arms pulled is randomized based on pulling a 'super-arm', and the reward is a non-linear function of the values of the arms pulled. In the adversarial setting, Han et al. [2021] study the combinatorial MAB problem when rewards can be expressed as a d-degree polynomial.

In contrast to prior work which focuses on giving algorithms which compete against benchmarks which have *the same* budget as the algorithm, this work is concerned with the trade-off between regret bounds and budget size. We focus on giving regret bounds against $\text{OPT}(1)$, and we use this result in Section 3 to get regret bounds against $\text{OPT}(B')$ for $B' < B$ in OSFM. Decoupling the regret benchmark $\text{OPT}(B')$ from the algorithm budget B can be useful when one would like to control the strength of a regret bound against a specific target $\text{OPT}(B')$ for theoretical or applied reasons. For example Arora et al. [2012] survey a wide variety of applications which rely on bounding R_T^*, but bounds such as $\mathbb{E}[(1 - e^{-1})\text{OPT}(B) - \text{Reward}_T] \leq \mathcal{O}(\sqrt{TB\ln(N)})$ do not immediately imply useful bounds on $\text{OPT}(1) - \text{Reward}_T$.

2 Follow the Top Perturbed Leaders

We begin by considering the full feedback setting. We first check that allowing the algorithm to choose $B > 1$ arms per round, while only competing against the best *single* fixed arm in hindsight, does not make the problem trivial. We do this by showing that any deterministic algorithm with budget $B < N$ still achieves linear regret in the number of rounds. This is achieved by setting $c_t(a) = 1$ if $a \in S_t$, $c_t(a) = 0$ otherwise.

Proposition 1. *In the full feedback setting, any deterministic algorithm with arm budget $B \leq N$ per round has worst-case regret $R_T^* \geq \left(1 - \frac{B}{N}\right) T$.*

Likewise, it can be shown that the algorithm which chooses a uniformly random subset of B arms in each round has worst-case expected regret at least $T(1 - \frac{B}{N})^B$ (achieved by having one arm have cost 0 across all rounds and every other arm having cost 1). These two observations show that any solution for achieving sub-linear regret in T requires randomization which depends in some non-trivial way on the prior observed costs even when $B > 1$.

2.1 Generalizing Follow the Perturbed Leader

Choosing the current lowest perturbed-cost arm in each round, *Follow the Perturbed Leader* (**FPL**) [Kalai and Vempala, 2005], is a well-known regret minimization technique which achieves optimal worst-case regret against adaptive adversaries in the OLwE setting. In this section we generalize the **FPL** algorithm to *Follow the Top Perturbed Leaders* (**FTL**). In each round, **FTL** perturbs the cumulative costs of each arm by adding noise, and then picks the B arms with lowest cumulative perturbed cost. This is precisely **FPL** when $B = 1$. We show how one can extend the proof techniques of Kalai and Vempala [2005] in a natural way to prove that **FTL** achieves worst-case regret $R_T^*(\textbf{FTL}) \leq 2T^{\frac{1}{B+1}}(1+\ln(N))^{\frac{B}{B+1}}$.

Algorithm 1 FTL(B,ε)

Require: $N \geq B \geq 1, \varepsilon > 0$.
Initialize the cumulative cost $C_0(a) \leftarrow 0$ for each arm $a \in \mathcal{A}$.
for round $t = 1, \ldots, T$ **do**
1. For each arm $a \in \mathcal{A}$, draw a noise perturbation $p_t(a) \sim \frac{1}{\varepsilon}$ Exp.
2. Calculate the perturbed cumulative costs for round $t-1$, $\tilde{C}_{t-1}(a) \leftarrow C_{t-1}(a) - p_t(a)$.
3. Pull the B arms with the lowest perturbed cumulative costs according to $\tilde{C}_{t-1}$. Break ties arbitrarily.
4. Update the cumulative costs for each arm, $C_t(a) \leftarrow C_{t-1}(a) + c_t(a)$.
end for

Theorem 2. *In the full feedback setting, where $S_t \subset \mathcal{A}$ is the subset of arms chosen by **FTL** in round t, we have:*

$$\max_{c_1,\ldots,c_T} \mathbb{E}\left[(1-\varepsilon^B)\sum_{t=1}^{T} c_t(S_t) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(a^*)\right] \leq \frac{(1+\ln(N))}{\varepsilon}.$$

In particular, for $\varepsilon = ((\ln(N)+1)/T)^{1/(B+1)}$, we have

$$R_T^*(\textbf{FTL}) \leq 2T^{\frac{1}{B+1}}(1+\ln(N))^{\frac{B}{B+1}}.$$

The proof follows the same three high level steps which appear in Kalai and Vempala [2005] for **FPL**, but we extend these ideas to the case where $B > 1$. We first observe that the algorithm which picks the B lowest cumulative cost arms in each round only incurs regret when the best arm in round t is not one of the best B arms in round $t-1$.

Lemma 3. *Consider a fixed sequence of cost functions $c_1, \ldots, c_T$. Let $a_t^{*,j}$ be the jth lowest cumulative cost arm in hindsight after the first t rounds, breaking ties arbitrarily. Let $S_t^* := \{a_{t-1}^{*,j} \mid j \in [B]\}$ be the set of the B lowest cost arms at the end of round $t-1$. Then for each $i \in [T]$,*

$$R_i := \sum_{t=1}^{i} c_t(S_t^*) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{i} c_t(a^*) \leq \sum_{t=1}^{i} \mathbb{1}[a_t^{*,1} \notin S_t^*]$$

and $R_i - R_{i-1} \leq \mathbb{1}[a_i^{*,1} \notin S_i^*]$.

This is a generalization of the familiar result that when $B = 1$, following the leader has regret bounded by the number of times the leader is overtaken [Kalai and Vempala, 2005].

The second step is to argue that if the cumulative costs are perturbed slightly, it becomes unlikely that the event $\{a_t^{*,1} \notin S_t^*\}$ will occur. One way to see this is as follows: fix a round t, and let $C_{t-1}(a)$ be the cumulative cost of a at the end of round $t-1$. Let $M := C_{t-1}(a^{*,B+1})$. Then every $a \in S_t$ has $C_{t-1}(a) \leq M$. If it is also true that $C_{t-1}(a) < M - c_t(a)$ for any $a \in S_t$ then the event $\{a_t^{*,1} \notin S_t^*\}$ cannot occur. This is because $C_t(a) < (M - c_t(a)) + c_t(a) < M$ but any $a' \in \mathcal{A} - S_t$ has $C_t(a') \geq M$, so $a' \neq a_t^{*,1}$. If we had initially perturbed each $C_{t-1}(a)$ by subtracting independent exponential noise $p(a) \sim \frac{1}{\varepsilon}$Exp, then conditional on M the event $\{C_{t-1}(a) < M - c_t(a)\}$ is jointly independent for each $a \in S_t$. Moreover the probability of this inequality not holding is equal to $\mathbb{P}[p(a) < v + c_t(a) | p(a) \geq v]$ for v equal to the unperturbed cost of a at round $t-1$ minus M, which is bounded by $\varepsilon c_t(a)$ (due to the memorylessness property of the exponential distribution).

Lemma 4. *Fix a sequence of cost functions $c_1, \ldots, c_T$. Let $C_i(a) = \sum_{t=1}^{i} c_t(a)$ and $\tilde{C}_i(a) = C_i(a) - p(a)$ be the perturbed cumulative cost of arm a at the end of round i, where $p(a) \sim \frac{1}{\varepsilon}$Exp. Let $\tilde{a}_t^{*,j}$ be the jth lowest cumulative cost arm in hindsight after the first t rounds using these perturbed costs, and let $\tilde{S}_t^* = \{\tilde{a}_{t-1}^{*,j} \mid j \in [B]\}$. Then*

$$\mathbb{E}\left[\mathbb{1}\left[\tilde{a}_t^{*,1} \notin \tilde{S}_t^*\right]\right] \leq \mathbb{E}\left[\varepsilon^B c_t(\tilde{S}_t^*)\right].$$

Again, when $B = 1$ this argument and bound coincides with the argument given by Kalai and Vempala [2005].

The final step is to combine Lemmas 3 and 4 to argue that **FTL** achieves expected regret at most $\mathbb{E}\left[\varepsilon^B \sum_{t=1}^{T} c_t(\tilde{S}_t^*)\right]$ with respect to the perturbed cumulative cost $\tilde{C}_T$. Since $\max_{a \in \mathcal{A}} \mathbb{E}[p(a)] \leq \frac{1+\ln(N)}{\varepsilon}$ we can argue we also achieve low expected regret with respect to the unperturbed cost C_T. In the setting of this paper, drawing new random perturbations $p_t(a)$ in each round is not strictly necessary (we can take $p_t(a) = p_1(a)$ for $t > 1$), but it is necessary to achieve regret bounds when cost functions can depend on prior arm choices of the algorithm (the *adaptive* adversarial setting). In the setting of this paper where the costs are fixed, the expected regret in either case is the same.

Probabilistic guarantees: One advantage of this proof technique is that the regret is bounded using the positive random variable $\sum_{t=1}^{T} \mathbb{1}\left[\tilde{a}_t^{*,1} \notin \tilde{S}_t^*\right]$. This means that one can apply methods like Markov inequality to give a probabilistic guarantee of small regret, which is substantially stronger than saying the regret is small in expectation.

Comments on settings of parameters: When $B = 1$ we recover the standard $\mathcal{O}(\sqrt{T \ln(N)})$ regret bound for the OLwE problem. For $B > 1$, the regret growth rate as a function of the number of rounds is $T^{\frac{1}{B+1}}$. In particular, when $B = \Omega(\ln(T))$ grows slowly with the number of rounds, the expected regret becomes $\mathcal{O}(\ln(N))$ and does not grow with the number of rounds T. If we use a tighter inequality in the proof of Lemma 4, it is possible to get constant expected regret when $B = \ln(T)\ln(A)$ grows slowly with the number of arms and rounds.

Lower bounds: A standard technique for constructing lower bounds in the online experts setting with $B = 1$ is to consider costs which are i.i.d. Bernoulli(p) [Lattimore and Szepesvári, 2020]. Unfortunately this technique fails when $B > 1$ because the expected cost of the minimum of B i.i.d. Bernoulli random variables is generally smaller than the expected cost of the best arm in hindsight unless p is very close to 1. We are able to show very weak lower bounds in the full feedback setting of $R_T^* = \Omega(\ln(N))$ for constant B and $T = \Omega(\ln(N))$, but there is a substantial gap between this and the upper bound. We think constructing stronger lower bounds is an interesting problem for future work which may require new analysis techniques.

Partial feedback: The *semi-bandit feedback* setting is a form of partial feedback where the algorithm only observes the individual costs of the arms it pulls. It can be shown that passing unbiased cost function estimates to **FTL** results in a similar regret bound in the semi-bandit feedback setting; the result is specific to **FTL** and using unbiased cost functions does not generally work for any R_T^*-minimizing algorithm when $B > 1$ because of the non-linearity of the cost function. In the case of **FTL** this is not an issue because the same bounding technique using Lemma 3 holds in expectation when using unbiased cost estimators. A naïve way to generate unbiased cost estimates in this setting is to use an additional arm to uniformly sample costs; in Section 4 we explore geometric sampling [Neu and Bartók, 2013] for getting unbiased cost estimates which is effective in practice.

Proposition 5. *Define the algorithm **FTL-partial** which simulates **FTL**, passing it unbiased cost estimates $\hat{c}_t \in [0, K]$ at round $t \in [T]$,[2] and copies the arm choices of **FTL**. Then we have R_T^*(**FTL-partial**) $\leq \ln(N)/\varepsilon + T(1 - \mathrm{e}^{-K\varepsilon})^B$. For $\varepsilon = \left(\frac{\ln(N)}{TK^B}\right)^{\frac{1}{B+1}}$, R_T^*(**FTL-partial**) $\leq \mathcal{O}(T^{\frac{1}{B+1}}(K\ln(N))^{\frac{B}{B+1}})$.*

[2]More formally: $\mathbb{E}[\hat{c}_t(\cdot) \mid \mathcal{F}_{t-1}] = c_t(\cdot)$ where $\mathcal{F}_{t-1}$ is the σ-algebra generated by all the randomness up to and including round $t-1$.

3 Generalized regret bounds for Online Submodular Function Maximization

Streeter and Golovin [2008] considered the related problem of Online Submodular Function Maximization which captures a number of previously studied problems as special cases. For the sake of emphasizing the key ideas, we restrict attention to the full feedback setting where each action has unit duration.

Definition 1. *Define a* schedule *to be a finite sequence of actions*[3] $a \in \mathcal{A}$*, and let* $\mathcal{S}$ *be the set of all schedules; the* length $\ell(S)$ *of a schedule* $S \in \mathcal{S}$ *is the number of actions it contains. Define a* job *to be a function* $f : \mathcal{S} \to [0, 1]$ *such that for any schedules* $S_1, S_2 \in \mathcal{S}$ *and any action* $a \in \mathcal{A}$*:*

1. $f(S_1) \leq f(S_1 \oplus S_2)$ *and* $f(S_2) \leq f(S_1 \oplus S_2)$ ***(monotonicity)****;*
2. $f(S_1 \oplus S_2 \oplus \langle a \rangle) - f(S_1 \oplus S_2) \leq f(S_1 \oplus \langle a \rangle) - f(S_1)$ ***(submodularity)****.*

Definition 2 (Online Submodular Function Maximization)**.** *The problem consists of a game with* T *rounds. We are given some fixed* $B > 0$ *and at each round* $t \in [T]$ *we must choose a schedule* $S_t \in \mathcal{S}$ *with* $\ell(S_t) \leq B$ *to be evaluated by a job* f_t *which is only revealed after our choice. The goal is to maximize the cumulative output* $\text{Reward}_T := \sum_{t=1}^{T} f_t(S_t)$*.*

Streeter and Golovin [2008] propose an online greedy algorithm **OG** which achieves the guarantee $(1 - e^{-1})\text{OPT}(B) - \text{Reward}_T \leq \mathcal{O}(\sqrt{TB \ln(N)})$ in expectation, where $\text{OPT}(B)$ is the cumulative reward of the best fixed schedule of length B in hindsight. In this section we explain how to use **FTL** to extend their algorithm to allow a trade-off between budget resources and regret bounds.

The algorithm **OG** has two key ideas. Suppose that we start with an empty schedule $S_0 := \emptyset$. The first idea is that if we knew $f_1, \ldots, f_T$ in advance, we could greedily construct $S_i := S_{i-1} \oplus \langle \arg\max_{a_i \in \mathcal{A}} \sum_{t=1}^{T} f_t(S_{i-1} \oplus \langle a_i \rangle) - f_t(S_{i-1}) \rangle$ for $i = 1, \ldots, B$, where a_i is the best greedy arm in hindsight for greedy round i. It can be shown that submodularity then implies $(1 - e^{-1})\text{OPT}(B) \leq \sum_{t=1}^{T} f(S_B)$. Since we don't know $f_1, \ldots, f_T$ in advance, the second idea is to run B copies of a R_T^* regret minimizer, where the ith copy tries to compete with achieving the same *improvement* of cumulative reward as the best fixed greedy action a_i in hindsight. The regret bound on the ith copy with respect to the best greedy arm in hindsight is $\mathcal{O}(\sqrt{T \ln N})$; across the B copies one can show the net regret compared to the offline greedy solution is bounded by $\mathcal{O}(\sqrt{TB \ln N})$, which is where the final bound comes from. In summary, **OG** works as follows: for each round $t \in [T]$, run B greedy rounds. In greedy round $i = 1, \ldots, B$, pull the arm a_i^t proposed by the ith black box R_T^*-minimizer, set $S_{t,i} := S_{t,i-1} \oplus \langle a_i^t \rangle$, and feed back the greedy rewards $r_{t,i}(a) = f_t(S_{t,i-1} \oplus \langle a \rangle) - f_t(S_{t,i-1})$ to the ith black box.

We propose a hybrid version of **OG**, named $\textbf{OG}_{\textbf{hybrid}}$, which allows us to compete against $\text{OPT}(B')$ for $B' \leq B$ based on the following two changes: (1) Instead of having B' greedy rounds, we have $B' \ln(T)$ greedy rounds. One can show that the extra factor of $\ln(T)$ allows one to drop the $(1 - e^{-1})$ term in the regret bound. (2) Instead of running a one-arm-pulling R_T^* minimizer in each greedy round, we run **FTL** which pulls $\tilde{B}$ arms. This allows us to improve the regret bound for each of the $B' \ln(T)$ R_T^*-minimizers and directly translates to a tighter overall regret bound. Besides these two changes, the algorithm is identical to that in Streeter and Golovin [2008]. More generally, we let $\textbf{OG}_{\textbf{hybrid}}(B, \tilde{B})$ denote the algorithm where each **FTL** box has a budget of $\tilde{B}$ and there are $B/\tilde{B}$ greedy rounds, so that the total number of arms pulled in each round is B. This algorithm is a hybrid of **OG** and **FTL** in the sense that $\textbf{OG}_{\textbf{hybrid}}(B, 1)$ is **OG** with *Follow the Perturbed Leader* as the R_T^*-minimizing subroutine. On the other hand, $\textbf{OG}_{\textbf{hybrid}}(B, B)$ is **FTL**. Varying $\tilde{B}$ allows us to interpolate between these two algorithms by varying the budget we give to the **FTL** subroutines. On a technical note, because we are pulling $\tilde{B} \geq 1$ arms for each greedy choice, we require a slight strengthening on the monotonicity condition which is common to many practical applications including the experiments we consider in the next section:

Assumption 1. *In addition to monotonicity and submodularity, each job* $f : \mathcal{S} \to [0, 1]$ *also satisfies* $f(S_1 \oplus S_2 \oplus S_3) \geq f(S_1 \oplus S_3)$ *for any schedules* $S_1, S_2, S_3 \in \mathcal{S}$*.*

[3] In their original problem definition actions may each have a different associated *duration*.

We can then give the following bounds:

Theorem 6. *Under Assumption 1 and for* $B = \lceil B'\widetilde{B}\ln(T)\rceil$*, algorithm* $\boldsymbol{OG_{hybrid}}$ *experiences expected regret*

$$\mathcal{O}\left(B'\ln(T)T^{1/(\widetilde{B}+1)}\ln(N)^{\widetilde{B}/(\widetilde{B}+1)}\right)$$

relative to the best-in-hindsight fixed schedule of length B'*. In particular, if* $\widetilde{B} = \lceil\ln(T)\rceil$ *(and so* $B = B'\lceil\ln(T)\rceil^2$*) the regret is bounded by* $\mathcal{O}(B'\ln(T)\ln(N))$*.*

This bound allows us the flexibility of trading off a schedule budget B for how tightly we would like to compete with the best fixed schedule of length $B' \leq B$ in hindsight.

Partial feedback: Streeter and Golovin [2008] extend their algorithm to handle partial feedback by replacing the R_T^* regret minimizers with the bandit algorithm **Exp3** [Auer et al., 2002] which only requires feedback on the arms which are pulled. Likewise, one can replace **FTL** with **FTL-partial** in $\mathbf{OG_{hybrid}}$ to get an algorithm which gives regret bounds in a semi-bandit feedback setting (where we can observe $f_t(S)$ for any schedule S consisting of actions which were pulled in round t). We empirically compare the partial feedback versions of **OG** and $\mathbf{OG_{hybrid}}$ in the next section.

4 Experiments: online hyperparameter optimization

The problem of *black-box optimization*—where a hidden function is to be minimized using as few evaluations as possible—has recently generated increased interest in the context of hyperparameter selection in deep learning [Snoek et al., 2012, Liu et al., 2020b, Bouthillier and Varoquaux, 2020]. As a result, the 2020 NeurIPS BBO Challenge [Turner et al., 2021] invited participants' optimizers to compete to find the best possible configurations of several ML models on a number of common datasets, given a limited budget of training cycles for each. One of the key findings was that sophisticated new algorithms are normally outperformed on average by techniques that ensemble existing methods [Liu et al., 2020a], i.e. optimizers tend to have varying strengths and weaknesses that are suited to different task types. In a scenario where many hyperparameter selection problems are to be processed (e.g. in a data center) and limited computing resources are available, it may thus be desirable to learn over time how best to choose B optimizers to apply independently to each problem (e.g. to run in parallel on B available CPU cores). This is a natural partial feedback application of bandit algorithms that pull multiple arms per round and receive the best score found across the optimizers which are run.

4.1 Experimental setup

We follow a similar approach to the NeurIPS BBO Challenge; the optimization problem at each round $t \in [T]$ is to choose the hyperparameters of either a multi-layer perceptron (MLP) or a lasso classifier for one of 184 classification tasks from the Pembroke Machine Learning Benchmark [Olson et al., 2017] (so $T = 368$). At each round the bandit algorithm must select B Bayesian black-box optimizers from a choice of 9 to run in parallel on the current problem; the received *reward*[4] for each optimizer was calculated as a $[0, 1]$-normalized measure of where the best training loss attained sits between **(a)** the expected best loss from a random hyperparameter search and **(b)** an estimate of the best possible loss attainable (the same approach used by the Bayesmark package [Uber, 2020]). Rewards are only observed for the optimizers which are run (semi-bandit feedback). For each budget level $B = 1, \ldots, 6$, we ran a benchmarking study comparing the performance of the following bandit algorithms in this setting: (1) **FTL-partial**(B) from Section 2.1, using geometric sampling [Neu and Bartók, 2013] to construct unbiased cost estimates (see appendix for details). (2) $\mathbf{OG_{hybrid}}(B, \tilde{B})$ from Section 3, with **FTL-partial** as a subroutine. We do this for varying values of $\tilde{B}$ to see how performance changes. (3) **OG**(B), the partial feedback version of the original online greedy algorithm from Streeter and Golovin [2008]. We benchmark performance against **BIH**(B), the score of the Best fixed subset of size B In Hindsight. In all cases the ε parameter for the bandit subroutines **FTL-partial** and **Exp3** was set to their theoretically optimal values given B, N and T without any fine-tuning (for **FTL-partial** we use Proposition 5). We run each algorithm setting 100 times to estimate the mean and standard deviation of the performance.

[4] Here we use rewards to be consistent with the setting of Online Submodular Function Maximization.

Table 1: Experimental results for $B = 6$ and $B = 4$.

$B = 6$		
Algorithm	Mean Reward	StD
BIH(6)	0.901	NA
FTL-partial(6)	0.888	0.0072
$\mathbf{OG_{hybrid}}(6,3)$	0.836	0.0111
$\mathbf{OG_{hybrid}}(6,2)$	0.814	0.0143
$\mathbf{OG_{hybrid}}(6,1)$	0.785	0.0137
OG	0.767	0.0157

$B = 4$		
Algorithm	Mean Reward	StD
BIH(4)	0.836	NA
FTL-partial(4)	0.813	0.0108
$\mathbf{OG_{hybrid}}(4,2)$	0.756	0.0149
$\mathbf{OG_{hybrid}}(4,1)$	0.716	0.0178
OG(4)	0.689	0.0151

4.2 Results and discussion

As expected, $\mathbf{OG_{hybrid}}(B,1)$ has very similar performance to $\mathbf{OG}(B)$ in all cases because they implement essentially the same algorithm (the difference being due to different choices of one-arm pulling R_T^* minimizers, **FTL-partial**(1) and **Exp3**). However, we notice that in every instance, allocating more budget to the **FTL-partial** subroutine in $\mathbf{OG_{hybrid}}$ (increasing $\tilde{B}$) while keeping the overall budget constant improved the average score performance. This means that $\mathbf{OG_{hybrid}}$ was always at least as good as **OG** for all parameter settings, and that **FTL-partial** outperformed both of these algorithms in all cases. This is perhaps surprising, because **OG** and $\mathbf{OG_{hybrid}}$ are designed to achieve low regret against the stronger benchmark of the best subset of $B' \geq 1$ arms in hindsight, while **FTL-partial** is only designed to achieve low regret with respect to the single best optimizer in hindsight. Towards explaining this observation, we notice that for this particular dataset, the best subset of B arms in hindsight happens to be very similar to the set of the individual best performing B arms in hindsight, and **FTL** achieves low regret with with respect to the latter (see Proposition 7 below). This raises the interesting question of when certain regret objectives might be better than others in practice.

Proposition 7. *In the full feedback setting, the expected regret of **FTL** with B arms relative to the set of the individual best performing B arms in hindsight is at most* $[1 - B^{-1} + \ln(|\mathcal{A}|/B)]/\varepsilon + T(1 - \exp(-\varepsilon B))$.

Synthetic tasks: We also evaluated these partial feedback algorithms in a number of synthetic environments, exploring examples where **(a)** the optimal subset of B arms, **(b)** the subset of B arms chosen greedily, **(c)** the individual best performing B arms, perform in various ways relative to each other; **OG** approximates **(b)** and **FTL-partial** approximates **(c)**, so the closeness of either of these to **(a)** determines each algorithm's performance. We find that problem instances exist where **OG** outperforms **FTL-partial** and vice versa. See the appendix for details.

5 Connections to other problems

Arora et al. [2012] surveys a wide variety of algorithmic problems and shows how they can all be solved using a R_T^* minimizing subroutine in the standard OLwE setting with $B = 1$. The purpose of this section is to highlight the relative ease with which algorithms like **FTL** can sometimes be *plugged in* to existing algorithms which use an R_T^* minimizer as black box subroutine, with little or no alteration to their proofs of correctness. We already saw an example of this in Section 3, where the regret minimizing subroutine was replaced with **FTL**. The resulting algorithm gave improved regret guarantees at the cost of higher budget requirements, without changing the structure of the original proof given in Streeter and Golovin [2008]. In this section we take Linear Programming as an example, and illustrate what its plug-in algorithm looks like. We are unsure whether the resulting algorithm is necessarily useful because it requires a more powerful oracle than the one supposed in Arora et al. [2012]; but we do think that exploring the algorithms which result from this plug-in technique more broadly may be an area of interest for future work.

5.1 Linear Programming

We consider the Linear Programming (LP) problem from Arora et al. [2012]: Given a convex set P, an $n \times m$ matrix A with entries in $\mathbb{R}$, and a vector $b \in \mathbb{R}^n$, the task is to find an $x \in P$ such that $Ax \geq b$, or determine that no such x exists. We assume that we have an oracle which solves the following easier problem (where A_i denotes the ith row of A):

Definition 3. *A (ρ, B)-bounded oracle for $\rho \geq 0$ is an algorithm, which when given a joint distribution d over $[n]^B$, finds an $x \in P$ such that*

$$\mathop{\mathbb{E}}_{(i_1,\ldots,i_B)\sim d}\left[\min_{i\in\{i_1,\ldots,i_B\}} A_i x - b_i\right] \geq 0$$

or determines that no such x exists. If an x is found, then $\forall i \in [n], |A_i x - b_i| \leq \rho$.

When $B = 1$ this is a simplified version of the oracle defined in Arora et al. [2012], and it is equivalent to an oracle which can find $x \in P$ which satisfies a single linear constraint $d^\top Ax \geq d^\top b$ (when viewing d as a vector in $\mathbb{R}^n$). When $B = n$, solving this problem can be as hard as solving the original LP problem by jointly choosing i_j to have probability mass 1 on the jth linear constraint. $1 < B < n$ represents an intermediate regime where the oracle needs to find an x which 'fools' the joint distribution d. For simplicity, we assume that we have access to an oracle which takes as input bounded cost functions $c_1, \ldots, c_{t-1}$, and outputs the joint distribution d_t over arms of **FTL** in round t after observing cost functions $c_1, \ldots, c_{t-1}$. In practice such an oracle could be achieved by e.g. sampling arm draws from **FTL** to approximate d_t.

Proposition 8. *Let $\varepsilon > 0$. Suppose there exists a (ρ, B)-bounded oracle for the feasibility problem $?\exists x \in P$ s.t. $Ax \geq b$. Then there is an algorithm which either finds an $x \in P$ s.t. $\forall i \in [n], A_i x \geq b_i - \varepsilon$, or correctly concludes that the problem is infeasible. The algorithm makes at most $T = \left(\frac{1}{\varepsilon}\right)^{\frac{B+1}{B}} (4\rho)^{\frac{B+1}{B}} (1 + \ln(n))$ calls to the (ρ, B)-bounded oracle and **FTL** oracle, with a total runtime of $\mathcal{O}(T)$.*

The proof technique for general $B \geq 1$ is essentially identical to the proof given in Arora et al. [2012] for $B = 1$ except that we are able to use a stronger bound on R_T^* (see appendix for details).

Comment on bound: When $B = 1$, we require $\Omega((\frac{1}{\varepsilon})^2)$ steps in order to find an x which is ε-close to satisfying the constraint. The $(\frac{1}{\varepsilon})^2$ term comes from the fact that when $B = 1$, the average regret for OLwE is $\Omega(\sqrt{T}/T) = T^{-\frac{1}{2}}$, so it takes $T \geq (\frac{1}{\varepsilon})^2$ steps for the average regret to be $\leq \varepsilon$. The quadratic dependence on an accuracy parameter is therefore common in many applications which use OLwE with $B = 1$ as a subroutine (including Boosting [Schapire, 1990] and solving zero sum games [Freund and Schapire, 1999]). For general $B \geq 1$, we only require $\Omega((\frac{1}{\varepsilon})^{\frac{B+1}{B}})$ steps for the average regret to be $\leq \varepsilon$. In the case of Linear Programming, this is at the expense of requiring a stronger oracle for the problem.

6 Conclusion

This paper presented a new algorithm, Follow the Perturbed Top Leaders, which allows one to directly trade off budget constraints for bounds on regret. We showed how **FTL** can be used as a subroutine to generate new algorithms for Online Submodular Function Optimization and Linear Programming which trade off resources and oracle power for improved performance guarantees.

We also highlight a number of areas for future work: (a) **Lower bounds:** Can we reduce the gap between the upper and lower bounds of this problem? Improving lower bounds may require new techniques than are traditionally used in the OLwE setting. (b) **Plug-in algorithms:** Are there other cases where using **FTL** in existing algorithms can lead to new theoretical results? And when are these new algorithms practically useful? (c) **Which regret benchmarks are useful in practice:** The experiments of Section 4 showed that algorithms designed to minimize regret with respect to a single arm can sometimes in practice outperform algorithms designed to minimize regret with respect to the stronger benchmark of the best subset of arms. Are certain regret objectives better than others in different practical applications?

References

Jason Ansel, Shoaib Kamil, Kalyan Veeramachaneni, Jonathan Ragan-Kelley, Jeffrey Bosboom, Una-May O'Reilly, and Saman Amarasinghe. Opentuner: An extensible framework for program autotuning. In *Proceedings of the 23rd international conference on Parallel architectures and compilation*, pages 303–316, 2014.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8(1):121–164, 2012. doi: 10.4086/toc.2012.v008a006. URL `https://doi.org/10.4086/toc.2012.v008a006`.

Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Math. Oper. Res.*, 39(1):31–45, 2014. doi: 10.1287/moor.2013.0598. URL `https://doi.org/10.1287/moor.2013.0598`.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.

Xavier Bouthillier and Gaël Varoquaux. *Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020*. PhD thesis, Inria Saclay Ile de France, 2020.

Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *J. Comput. Syst. Sci.*, 78(5):1404–1422, 2012. doi: 10.1016/j.jcss.2012.01.001. URL `https://doi.org/10.1016/j.jcss.2012.01.001`.

Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1651–1659, 2016a. URL `https://proceedings.neurips.cc/paper/2016/hash/aa169b49b583a2b5af89203c2b78c67c-Abstract.html`.

Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *J. Mach. Learn. Res.*, 17:50:1–50:33, 2016b. URL `http://jmlr.org/papers/v17/14-298.html`.

Richard Combes, Mohammad Sadegh Talebi, Alexandre Proutière, and Marc Lelarge. Combinatorial bandits revisited. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2116–2124, 2015. URL `https://proceedings.neurips.cc/paper/2015/hash/0ce2ffd21fc958d9ef0ee9ba5336e357-Abstract.html`.

David Eriksson, David Bindel, and Christine A Shoemaker. pysot and poap: An event-driven asynchronous framework for surrogate optimization. *arXiv preprint arXiv:1908.00420*, 2019.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Carla P. Gomes and Bart Selman. Algorithm portfolios. *Artif. Intell.*, 126(1-2):43–62, 2001. doi: 10.1016/S0004-3702(00)00081-3. URL `https://doi.org/10.1016/S0004-3702(00)00081-3`.

Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 100–108. JMLR.org, 2014. URL `http://proceedings.mlr.press/v32/gopalan14.html`.

Yanjun Han, Yining Wang, and Xi Chen. Adversarial combinatorial bandits with general non-linear reward functions. In *International Conference on Machine Learning*, pages 4030–4039. PMLR, 2021.

Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Zé Vinícius, cmmalone, Christopher Schröder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan, rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak, SoManyUsernamesTaken, Fred Callaway, Loïc Estève, Lilian Besson, Mehdi Cherti, Karlson Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander Fabisch. scikit-optimize/scikit-optimize: v0.5.2, March 2018. URL `https://doi.org/10.5281/zenodo.1207017`.

Adam Tauman Kalai and Santosh S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005. doi: 10.1016/j.jcss.2004.10.016. URL `https://doi.org/10.1016/j.jcss.2004.10.016`.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Jian Li and Amol Deshpande. Maximizing expected utility for stochastic combinatorial optimization problems. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 797–806. IEEE Computer Society, 2011. doi: 10.1109/FOCS.2011.33. URL `https://doi.org/10.1109/FOCS.2011.33`.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2): 212–261, 1994. doi: 10.1006/inco.1994.1009. URL `https://doi.org/10.1006/inco.1994.1009`.

Jiwei Liu, Bojan Tunguz, and Gilberto Titericz. Gpu accelerated exhaustive search for optimal ensemble of black-box optimization algorithms. *arXiv preprint arXiv:2012.04201*, 2020a.

Zhengying Liu, Zhen Xu, Shangeth Rajaa, Meysam Madadi, Julio CS Jacques Junior, Sergio Escalera, Adrien Pavao, Sebastien Treguer, Wei-Wei Tu, and Isabelle Guyon. Towards automated deep learning: Analysis of the autodl challenge series 2019. In *NeurIPS 2019 Competition and Demonstration Track*, pages 242–252. PMLR, 2020b.

Gergely Neu and Gábor Bartók. An efficient algorithm for learning with semi-bandit feedback. In *International Conference on Algorithmic Learning Theory*, pages 234–248. Springer, 2013.

Randal S Olson, William La Cava, Patryk Orzechowski, Ryan J Urbanowicz, and Jason H Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10(1):1–13, 2017.

Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5:197–227, 1990. doi: 10.1007/BF00116037. URL `https://doi.org/10.1007/BF00116037`.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Matthew J. Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Technical Report CMU-CS-07-171*. Carnegie Mellon University, 2007.

Matthew J. Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1577–1584. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper/2008/hash/5751ec3e9a4feab575962e78e006250d-Abstract.html`.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 3–26. PMLR, 2021.

Uber. Bayesmark documentation, 2020. URL `https://bayesmark.readthedocs.io/en/latest/index.html`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section 4 discusses when the new algorithms perform better or worse relative to existing work. Section 5 acknowledges that plug-in reductions with FTL do not necessarily result in new algorithms which are useful in practice.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] The contributions are primarily theoretical/related to optimization in general.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The setting and assumptions are fully defined in the main body of the paper.
   (b) Did you include complete proofs of all theoretical results? [Yes] Every proposition, lemma and theorem in the main body of the paper has a proof in the appendix. The main body of the paper is primarily used for communicating the high level intuition behind the proofs.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Included in supplementary material. We will also provide a github link after review.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All choices of algorithm parameters are explained in the main body of the paper. Further details can be found in the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Each algorithm variant in Section 4 was run multiple times and standard deviations of results are included.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] This paper is about benchmarking algorithm performance on idealized standardized budgets. Actual compute time is application specific and not central to the contributions of the paper.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] All prior datasets and algorithms are cited.
   (b) Did you mention the license of the assets? [Yes] A permissive license is included with the code in the supplementary material, and will be made available publicly after review.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Yes, in the supplemental material we provide code for replicating the experiments in Section 4.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Consent was not required and the data used is publicly available for use as a benchmarking dataset by the academic community.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] This is not applicable to the dataset considered.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Section 2: Follow the the top leaders

Proof of Proposition 1

Proof. Construct a cost sequence inductively for $t = 1, 2, \ldots, T$: Let $S_t \subset \mathcal{A}$ be the deterministic choice of arms the bandit algorithm chooses for round t given the previously chosen cost functions $c_1, \ldots, c_{t-1}$. Now choose $c_t(a) = 1$ if $a \in S_t$, and $c_t(a) = 0$ otherwise. Then the bandit algorithm achieves cost T. The total cost summed over all arms is $\sum_{t=1}^{T} |S_t| \leq BT$, so there must exist at least one $a' \in \mathcal{A}$ such that $\sum_{t=1}^{T} c_t(a') \leq \frac{BT}{N}$. Thus $R_T^* \geq T - \frac{BT}{N} = (1 - \frac{B}{N})T$. □

Proof of Lemma 3

Proof. Let $R_i = \sum_{t=1}^{i} c_t(S_t^*) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{i} c_t(a^*)$ be the regret at the end of round i. Then the increase in regret in round i is

$$
\begin{aligned}
r_i &:= R_i - R_{i-1} \\
&= \sum_{t=1}^{i} \Big(c_t(S_t^*) - c_t(a_i^{*,1}) \Big) - \sum_{t=1}^{i-1} \Big(c_t(S_t^*) - c_t(a_{i-1}^{*,1}) \Big) \\
&= c_i(S_i^*) - c_i(a_i^{*,1}) + \left(\sum_{t=1}^{i-1} c_t(a_{i-1}^{*,1}) - \sum_{t=1}^{i-1} c_t(a_i^{*,1}) \right) \\
&\leq c_i(S_i^*) - c_i(a_i^{*,1}) \\
&\leq \mathbb{1}[a_i^{*,1} \notin S_i^*]
\end{aligned}
$$

and the result follows by evaluating $\sum_{t=1}^{T} r_t$. □

Lemma 9. *(Proof of independence for Lemma 4) Let $X_1, \ldots, X_K$ be jointly independent continuous random variables. Let $i_1, \ldots, i_k$ and $v_{i_1}, \ldots, v_{i_k}$ be the indices and values of the largest $k < K$ random variables, and let $X := \{X_i | i \notin \{i_1, \ldots, i_k\}$ be the smallest $K - k$ random variables. Then conditional on $(i_j, v_{i_j})_{j \in [k]}$, the values of each $X_i \in X$ are jointly independent. Moreover, the marginal distribution $X_i | (i_j, v_{i_j})_{j \in [k]}$ for $i \notin \{i_1, \ldots, i_k\}$ is $X_i | X_i \leq \min_{j \in [k]} v_{i_j}$.*

Proof. Let $M := \min_{j \in [k]} v_{i_j}$. The conditional joint density function is

$$
\begin{aligned}
f(X_1, \ldots, X_K | (i_j, v_{i_j})_{j \in [k]}) &\propto f(X \wedge (i_j, v_{i_j})_{j \in [k]}) \\
&= \prod_{j \in [K] - \{i_1, \ldots, i_k\}} f(X_j) \prod_{j \in \{i_1, \ldots, i_k\}} f(X_j = v_j) \prod_{j \in [K] - \{i_1, \ldots, i_k\}} \mathbb{1}[X_j \leq M] \\
&\propto \prod_{j \in [K] - \{i_1, \ldots, i_k\}} f(X_j) \mathbb{1}[X_j \leq M]
\end{aligned}
$$

i.e. the joint density factorizes for each X_j (which implies joint independence), and marginally the density for $X_j \in X$ is $\propto f(X_j)\mathbb{1}[X_j \leq M]$ which gives the required result. □

Proof of Lemma 4

Proof. Fix a round t. Consider the jointly independent random variables $X_a = \tilde{C}_{t-1}(a)$ for $a \in \mathcal{A}$. Condition on the values and identities of the $N - B$ largest of these random variables, i.e. condition on $E = \{(X_{a_{t-1}^{*,j}}, a_{t-1}^{*,j})\}_{j=B+1}^{N}$, and let $M = X_{a_{t-1}^{*,B+1}}$ be the minimum perturbed cost among these non-leading arms. Impose an ordering on $\mathcal{A}$ and let $l_1, \ldots, l_B \in \mathcal{A} - \{a^{*,j}\}_{j=B+1}^{N}$ be the remaining arms (the top B leaders) ordered lexicographically (i.e. not necessarily in order of cumulative perturbed

cost). Then it can be shown that the distribution of the random variables $X_{l_1}, \dots, X_{l_B}$ conditioned on E is jointly independent, and the marginal distribution of X_{l_j} given E is $X_{l_j}|(X_{l_j} \leq M)$ (see Lemma 9). Now observe that if $X_{l_j} < M - c_t(l_j)$ for any $j \in [B]$, then the event $(\tilde{a}_t^{*,1} \notin \tilde{S}_t^*)$ is impossible. This is because $l_j \in \tilde{S}_t^*$, but for any $a \notin S_t^*$, $\tilde{C}_t(a) \geq M$ but $\tilde{C}_t(l_j) = X_{l_j} + c_t(l_j) < M$ (i.e. l_j cannot be overtaken by any non-top-B-leader in round t). Therefore we have

$$
\begin{aligned}
\mathbb{E}\left[\mathbb{1}[\tilde{a}_t^{*,1} \notin \tilde{S}_t^*]|E\right] &\leq P\left[\bigwedge_{j=1}^{B} \neg(X_{l_j} < M - c_t(l_j)) \,\middle|\, E\right] && (1)\\
&= \prod_{j=1}^{B} P\left[\neg(X_{l_j} < M - c_t(l_j))|X_{l_j} \leq M\right] && (2)\\
&= \prod_{j=1}^{B} \left(1 - P\left[p(l_j) > C_{t-1}(l_j) + c_t(l_j) - M | p(l_j) \geq C_{t-1}(l_j) - M\right]\right) && (3)\\
&\leq \prod_{j=1}^{B} \left(1 - P\left[p(l_j) > c_t(l_j)\right]\right) && (4)\\
&= \prod_{j=1}^{B} \left(1 - \int_{c_t(l_j)}^{\infty} \varepsilon e^{-\varepsilon x} dx\right) = \prod_{j=1}^{B} (1 - e^{-\varepsilon c_t(l_j)}) && (5)\\
&\leq \prod_{j=1}^{B} (\varepsilon c_t(l_j)) \leq \varepsilon^B c_t(\tilde{S}_t^*) && (6)
\end{aligned}
$$

(2) follows by independence, (4) is due to the memorylessness property of the exponential distribution (with equality unless $C_{t-1}(l_j) - M < 0$), and (6) follows because $1 - e^{-x} \leq x$ for $x \geq 0$ and $c_t(S_t^*) \geq \prod_{a \in S_t^*} c_t(a) = \prod_{j=1}^{B} c_t(l_j)$. The final claim follows by taking the expectation over the conditioned event E. □

Proof of Theorem 2

Proof. Consider a modified version of FTL where $p_t(a) = p_1(a) = p(a)$ for all $t > 1$ (i.e. we keep the random perturbation fixed across rounds). Then this version of FTL picks the set $\tilde{S}_t^*$ in round t, and the regret can be bounded as

$$
\begin{aligned}
\sum_{t=1}^{T} c_t(\tilde{S}_t^*) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(a^*) &= \left(\sum_{t=1}^{T} c_t(\tilde{S}_t^*) - \min_{\tilde{a}^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(\tilde{a}^*) - p(\tilde{a}^*)\right)\\
&\quad + \left(\min_{\tilde{a}^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(\tilde{a}^*) - p(\tilde{a}^*) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(a^*)\right)
\end{aligned}
$$

The second term is ≤ 0. The first term can be interpreted as the regret of a modified version of MAB with a 0th round with cost function $-p$, where we are only allowed to pull arms from round $t = 1$. The regret increase incurred in the 0th round is at most $\max_{a \in \mathcal{A}} p(a)$. For the remaining rounds, we use Lemma 3 followed by Lemma 4 to get

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} c_t(S_t^*) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{T} c_t(a^*)\right] &\leq \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}[\tilde{a}_t^{*,1} \notin \tilde{S}_t^*]\right] + \mathbb{E}\left[\max_{a \in \mathcal{A}} p(\tilde{a})\right]\\
&\leq \varepsilon^T \, \mathbb{E}\left[\sum_{t=1}^{T} c_t(S_t^*)\right] + \frac{(1 + \ln(N))}{\varepsilon}
\end{aligned}
$$

Where the inequality on $\mathbb{E}\left[\max_{a\in\mathcal{A}} p(\tilde{a})\right]$ comes from Kalai and Vempala [2005]. The final step is to argue that the unmodified version of FTL which chooses independent noise $p_t(a)$ in each round also achieves this bound. This is immediate because both versions of the algorithm incur the same expected cost in each round, and $\mathbb{E}\left[\sum_{t=1}^{T} c_t(\textbf{FTL})\right] = \sum_{t=1}^{T} \mathbb{E}\left[c_t(\textbf{FTL})\right]$. Having new random perturbations in each round is not necessary against oblivious adversaries, but is necessary to achieve the regret bound against adaptive adversaries. □

Lower bound:

Proposition 10. *(Lower bounds) In the full feedback setting, any randomized algorithm has* $R_T^* \geq \Omega\left(\left(\frac{1}{4}\right)^B (\log_2(N) - \log_2(B))\right)$ *for* $T \geq \Omega(\log_2(N) - \log_2(B))$.

Proof. First suppose $N = 2^k$ for some $k \in \mathbb{N}$. Let $\mathcal{A}_0 = \mathcal{A}$. In round $t = 1, \ldots, T$, we let $\mathcal{A}_t$ be a uniformally randomly chosen subset of $\mathcal{A}_{t-1}$ of size $\max(2^{k-t}, 1)$, and we let $c_t(a) = 0$ if $a \in \mathcal{A}_t$, 1 otherwise. Suppose an algorithm chooses arms $S_t \subset \mathcal{A}$ in round t. Then

$$
\begin{aligned}
P[S_t \cap A_t = \emptyset] &\geq \prod_{i=0}^{B-1}\left(1 - \frac{|\mathcal{A}_t|}{|\mathcal{A}_{t-1}| - i}\right) \\
&\geq \left(1 - \frac{|\mathcal{A}_t|}{|\mathcal{A}_{t-1}| - B}\right)^B \\
&= \left(1 - \frac{1}{2 - B/|\mathcal{A}_t}\right)^B \\
&\geq \left(1 - \frac{3}{4}\right)^B \\
&= \left(\frac{1}{4}\right)^B
\end{aligned}
$$

provided that $|\mathcal{A}_t| \geq \frac{3}{2}B$ and $t \leq k$, which holds when $t \leq k - \log_2(B) - \log_2\left(\frac{3}{2}\right)$. If we set $T = \lfloor k - \log_2(B) - \log_2\left(\frac{3}{2}\right)\rfloor$, then the expected cost of any fixed algorithm ALG is $\geq \left(\frac{1}{4}\right)^B T = \Omega\left(\left(\frac{1}{4}\right)^B (\log_2(N) - \log_2(B))\right)$. By construction, the cost of the best expert in hindsight is 0. Since the expected regret is 0, there exists fixed cost functions $c_1, \ldots, c_T$ such that the expected regret of ALG on this on this sequence is $\geq \Omega\left(\left(\frac{1}{4}\right)^B (\log_2(N) - \log_2(B))\right)$. If N is not a power of 2, we can just let $\mathcal{A}_0 \subset \mathcal{A}$ be any subset of size $2^{\lfloor \log_2(N)\rfloor}$ and the asymptotic bounds remain the same. □

Proof of Proposition 5

Proof. Fix deterministic cost functions $c_1, \ldots, c_T$. Writing $\mathcal{F}_t$ for the σ-algebra generated by all actions and observations (as well as any other randomness) up to and including round t, for each $t \in [T]$ let $\hat{c}_t$ be a $\mathcal{F}_t$-measurable random function $\mathcal{A} \to [0, K]$ such that $\mathbb{E}[\hat{c}_t(a) \mid \mathcal{F}_{t-1}] = c_t(a)$ for each a. Assume an oblivious adversary and that w.l.o.g. instead of perturbations there is a 'round zero' with costs $(-p(a))_{a\in\mathcal{A}}$ where $p(a) \sim \mathrm{Exp}(\varepsilon)$ independently for each a; define $\mathcal{F}_0 := \sigma((p(a))_{a\in\mathcal{A}}$ to be the σ-algebra generated by these and include it in each $(\mathcal{F}_t)_{t\geq 1}$.

Writing $\hat{C}_i(\cdot) := \sum_{t=1}^{i} \hat{c}_t(\cdot)$ for cumulative estimated reward and $\hat{C}_i^*(\cdot) := \sum_{t=0}^{i} \hat{c}_t(\cdot) = \hat{C}_i(\cdot) - p(\cdot)$ for the same but including the 'round zero' random initializations, define

$$
R_i' := \sum_{t=1}^{i} c_t(S_t) - \min_{a\in\mathcal{A}} \hat{C}_i^*(a)
$$

for each $i \in [T]$. Let S_i be the set of arms chosen by the algorithm at round i and a_i^* be the best of these by perturbed estimated cost. We follow the argument from Lemma 3:

$$R'_i - R'_{i-1} = c_i(S_i) - \hat{c}_i(a_i^*) + \left(\sum_{t=1}^{i-1} \hat{c}_t(a_{i-1}^*) - \sum_{t=1}^{i-1} \hat{c}_t(a_i^*) \right) \leq c_i(S_i) - \hat{c}_i(a_i^*)$$

and so

$$\mathbb{E}[R'_i - R'_{i-1} \mid \mathcal{F}_{i-1}] \leq c_i(S_i) - \mathbb{E}[\hat{c}_i(a_i^*) \mid \mathcal{F}_{i-1}] = c_i(S_i) - c_i(a_i^*) \leq \mathbb{1}[a_i^* \notin S_i].$$

Hence (using the tower law)

$$\begin{aligned} \mathbb{E}[R'_T \mid \mathcal{F}_0] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}[R'_t - R'_{t-1} \mid \mathcal{F}_{t-1}] + R'_0 \right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[a_t^* \notin S_t] + R_0 \mid \mathcal{F}_0 \right] \\ &= \mathbb{E}[|\mathcal{I}| \mid \mathcal{F}_0] + \max_{a \in \mathcal{A}} p(a), \end{aligned}$$

where $\mathcal{I} := \{t \in [T] : a_t^* \notin S_t\}$. Noting that by Jensen's inequality

$$\begin{aligned} \mathbb{E}[R'_T \mid \mathcal{F}_0] \geq \sum_{t=1}^{T} c_t(S_t) - \min_{a \in \mathcal{A}} \mathbb{E}[\hat{C}_T(a) \mid \mathcal{F}_0] &= \sum_{t=1}^{T} c_t(S_t) - \min_{a \in \mathcal{A}} (C_T(a) - p(a)) \\ &\geq \sum_{t=1}^{T} c_t(S_t) - C_T(a^*) + p(a^*) \end{aligned}$$

(where a^* is the best-in-hindsight arm) hence gives that the algorithm regret satisfies

$$\mathbb{E}[R_T^* \mid \mathcal{F}_0] \leq \mathbb{E}[|\mathcal{I}| \mid \mathcal{F}_0] + \max_{a \in \mathcal{A}} p(a) - p(a^*).$$

Since $\mathbb{E}[p(a^*)] = 1/\varepsilon$ as for any fixed action, and $\max_{a \in \mathcal{A}} p(a)$ is the maximum of $|\mathcal{A}|$ i.i.d. $\mathrm{Exp}(\varepsilon)$ random variables, so has expectation at most $(1 + \ln |\mathcal{A}|)/\varepsilon$ as argued in Kalai and Vempala [2005], taking expectations gives

$$\mathbb{E}[R_T^*] \leq \frac{\ln |\mathcal{A}|}{\varepsilon} + \mathbb{E}[|\mathcal{I}|].$$

It remains to upper-bound $\mathbb{E}[|\mathcal{I}|]$. This will bound the expected regret of the modified version of **FPL** which uses the same random perturbations in each round (i.e. $p_t(a) = p_1(a)$ for $t > 1$). Since the expected cost incurred by the modified version of **FPL** doesn't change if we draw new random perturbations in each round (for any fixed round t, the distribution of arms picked for both versions of **FPL** is identical), the regret bound holds for the unmodified version of **FTL** as well.

Fix $t \in [T]$ and let $V := \min_{a \in \mathcal{A} - S_t} \hat{C}_{t-1}^*(a)$. So for any a, $a \in S_t$ iff $\hat{C}_{t-1}^*(a) < V$. Define $E_a := \{\hat{C}_{t-1}^*(a) < V - K\}$; if this holds then a must have been ahead of every action $a' \notin S_t$ by at least K and therefore *cannot* be overtaken by any such action, since the estimated costs are all upper-bounded by K. So

$$\{a \text{ overtaken by some } a' \notin S_t\} \subset E_a^c.$$

Note that

$$\begin{aligned} \{a_t^* \notin S_t\} &= \{\exists a' \in \mathcal{A} - S_t : \forall a \in S_t,\ a' \text{ overtakes } a \text{ at round } t\} \\ &= \bigcup_{a' \in \mathcal{A} - S_t} \bigcap_{a \in S_t} \{a' \text{ overtakes } a \text{ at round } t\} \\ &\subset \bigcap_{a \in S_t} \bigcup_{a' \in \mathcal{A} - S_t} \{a' \text{ overtakes } a \text{ at round } t\} = \bigcap_{a \in S_t} \{a \text{ overtaken by some } a' \notin S_t\}. \end{aligned}$$

Let $\mathcal{G}_t \coloneqq \sigma(S_t, (\hat{C}^*_{t-1}(a))_{a \notin S_t})$ be the σ-algebra generated by the random set S_t and the current perturbed estimated cumulative costs of the actions not in it. So we have

$$\begin{aligned}
\mathbb{P}\left(a^*_t \notin S_t \mid \mathcal{G}_t\right) &\leq \mathbb{P}\left(\bigcap_{a \in S_t} \{a \text{ overtaken by some } a' \notin S_t\} \mid \mathcal{G}_t\right) \\
&\leq \mathbb{P}\left(\bigcap_{a \in C} E^c_a \mid \mathcal{G}_t\right) \\
&= \mathbb{P}\left(\bigcap_{a \in S_t} \{\hat{C}^*_{t-1}(a) < V - K\} \mid \mathcal{G}_t\right).
\end{aligned}$$

But, since $V = \min_{a \in \mathcal{A} - S_t} \hat{C}^*_{t-1}(a)$, applying Lemma 9 gives us that

$$\mathbb{P}\left(\bigcap_{a \in S_t} \{\hat{C}^*_{t-1}(a) < V - K\} \mid \mathcal{G}_t\right) = \prod_{a \in S_t} \mathbb{P}\left(\hat{C}^*_{t-1}(a) < V - K \mid \hat{C}^*_{t-1}(a) \leq V\right).$$

By the memoryless property of the exponential distribution, each term here just becomes

$$1 - \mathbb{P}\left(p(a) \geq \hat{C}_{t-1}(a) - V + K \mid p(a) \geq \hat{C}_{t-1}(a) - V\right) \leq 1 - \mathbb{P}(p(a) \geq K) = 1 - \mathrm{e}^{-K\varepsilon}.$$

Thus $\mathbb{P}(a^*_t \notin S_t \mid \mathcal{G}_t) \leq (1 - \mathrm{e}^{-K\varepsilon})^B$. Since this expression is deterministic and so trivially independent from the σ-algebra $\mathcal{G}_t$, this immediately implies that $\mathbb{P}(a^*_t \notin S_t) \leq (1 - \mathrm{e}^{-K\varepsilon})^B$.

The result then follows, since $\mathbb{E}[|\mathcal{I}|] = \sum_{t=1}^T \mathbb{P}(a^*_t \notin S_t) \leq T(1 - \mathrm{e}^{-K\varepsilon})^B$. □

B Section 3: Generalized regret bounds for Online Submodular Function Maximization

Algorithm 2 $\mathrm{OG}_{\text{hybrid}}(B, \widetilde{B})$

Require: $B \geq \widetilde{B} \geq 1$. Assume for simplicity that $\widetilde{B} \mid B$; define $L \coloneqq B/\widetilde{B}$.
Let $\mathcal{B}_1, \ldots, \mathcal{B}_L$ be instances of FTL, each with budget $\widetilde{B}$.
for rounds $t = 1, \ldots, T$ **do**
  Let $S_{t,0} = \langle\rangle$ be the empty schedule.
  for $i = 1, \ldots, L$ **do**
    Use $\mathcal{B}_i$ to choose $\widetilde{B}$ actions $a^t_{(i-1)\widetilde{B}+1}, \ldots, a^t_{i\widetilde{B}}$.
    Set $S_{t,i} \coloneqq S_{t,i-1} \oplus \langle a^t_{(i-1)\widetilde{B}+1}, \ldots a^t_{i\widetilde{B}}\rangle$.
  end for
  Set $S_t \coloneqq S_{t,L}$; receive the job f_t.
  for $i = 1, \ldots, L$ **do**
    For each action $a \in \mathcal{A}$ feed back the cost $c^{(i)}_t(a) \coloneqq 1 - (f_t(\langle a^*_{1,t}, \ldots, a^*_{i-1,t}, a\rangle) - f_t(\langle a^*_{1,t}, \ldots, a^*_{i-1,t}\rangle))$ to FTL instance $\mathcal{B}_i$.
    Define $a^*_{i,t} \coloneqq \arg\min_{j \in [\widetilde{B}]} c^{(i)}_t(a_{(i-1)\widetilde{B}+j})$.
  end for
end for

Proof of Theorem 6

Before proving the theorem, we give a modification to the original result from Streeter and Golovin [2008]. The problem setting they considered was slightly more general:

Definition 4. *Let an action now be an activity-duration pair $a = (\nu, \tau) \in \mathcal{V} \times (0, \infty) = \mathcal{A}$ for some fixed finited set of* activities $\mathcal{V}$.[5] *The* length $\ell(S)$ *of a schedule $S \in \mathcal{S}$ is now the sum of the durations of all the actions in S. Write $S_{\langle i \rangle}$ for the prefix of length i of schedule S.*

[5] We will enforce integer durations so that there are only finitely many possible actions to choose from given a duration constraint.

The algorithm OG they introduced, which takes a budget B and experts algorithm $\mathcal{E}$, is given in Algorithm 3 using our notation for ease of reference.

Algorithm 3 OG($B, \mathcal{E}$)

Require: $B \geq 1$.
Let $\mathcal{E}_1, \ldots, \mathcal{E}_B$ be instances of experts algorithm $\mathcal{E}$ (e.g. Hedge).
for rounds $t = 1, \ldots, T$ **do**
  Let $S_{t,0} = \langle\rangle$ be the empty schedule.
  for $i = 1, \ldots, B$ **do**
    Use $\mathcal{E}_i$ to choose an action $a_i^t = (\nu, \tau) \in \mathcal{A}$.
    With probability $1/\tau$ set $S_{t,i} \coloneqq S_{t,i-1} \oplus \langle a_i^t \rangle$, otherwise set $S_{t,i} \coloneqq S_{t,i-1}$.
  end for
  Set $S_t \coloneqq S_{t,B}$; receive the job f_t.
  For each $i \in [B]$ and each action $a = (\nu, \tau) \in \mathcal{A}$ feed back the cost $c_t^{(i)}(a) \coloneqq (f_t(S_{t,i} \oplus \langle a \rangle) - f_t(S_{t,i}))/\tau$ to experts algorithm $\mathcal{E}_i$.
end for

We first prove a lemma generalizing Theorem 6 in Streeter and Golovin [2008]:

Lemma 11. *Let f be any job and let $\bar{G} = \langle \bar{g}_1, \bar{g}_2, \ldots \rangle$ be an infinite 'greedy' schedule satisfying*

$$\frac{f(\bar{G}_j \oplus \bar{g}_j) - f(\bar{G}_j)}{\bar{\tau}_j} \geq \max_{(\nu,\tau) \in \mathcal{V} \times (0,\infty)} \left(\frac{f(\bar{G}_j \oplus \langle (\nu, \tau) \rangle) - f(\bar{G}_j)}{\tau} \right) - \varepsilon_j, \qquad j \geq 1$$

for additive errors $\varepsilon_1, \varepsilon_2, \ldots \geq 0$, where $\bar{g}_j = (\bar{v}_j, \bar{\tau}_j)$ and $\bar{G}_j = \langle \bar{g}_1, \ldots, \bar{g}_{j-1} \rangle$ for each $j \geq 1$.

Then for any $L, B_0 \in \mathbb{N}$ and for $B_1 \coloneqq \sum_{j=1}^{L} \bar{\tau}_j$,

$$f(\bar{G}_{\langle B_1 \rangle}) > \left(1 - \mathrm{e}^{-B_1/B_0}\right) f(S^*_{B_0}) - \sum_{j=1}^{L} \varepsilon_j \bar{\tau}_j$$

where $S^*_{B_0} \coloneqq \arg\max_{S \in \mathcal{S}: \ell(S) = B_0} f(S)$ is the best schedule of length B_0 for f.

Proof. For each $j \in \mathbb{N}$ write $\Delta_j \coloneqq f(S^*_{B_0}) - f(\bar{G}_j)$. By Fact 2 from Streeter and Golovin [2008], for any $j \in \mathbb{N}$, $b > 0$ and $S \in \mathcal{S}$ with $\ell(S) \leq b$,

$$f(S) \leq f(\bar{G}_j) + b \cdot (s_j + \varepsilon_j),$$

where

$$s_j \coloneqq \max_{(v,\tau) \in \mathcal{V} \times (0,\infty)} \frac{f(\bar{G}_j \oplus \langle (n, \tau) \rangle) - f(\bar{G}_j)}{\tau} = \frac{f(\bar{G}_j \oplus \bar{g}_j) - f(\bar{G}_j)}{\bar{\tau}_j} = \frac{f(\bar{G}_{j+1}) - f(\bar{G}_j)}{\bar{\tau}_j},$$

so in particular for any j

$$f(S^*_{B_0}) = \max_{S \in \mathcal{S}: \ell(S) = B_0} f(S) \leq f(\hat{G}_j) + B_0 \cdot (s_j + \varepsilon_j) \tag{7}$$

$$= f(\hat{G}_j) + B_0 \left(\frac{f(\bar{G}_{j+1}) - f(\bar{G}_j)}{\bar{\tau}_j} + \varepsilon_j \right) \tag{8}$$

$$= f(\hat{G}_j) + B_0 \left(\frac{\Delta_j - \Delta_{j+1}}{\bar{\tau}_j} + \varepsilon_j \right), \tag{9}$$

giving $\Delta_j \leq B_0 \left(\frac{\Delta_j - \Delta_{j+1}}{\bar{\tau}_j} + \varepsilon_j \right)$.

Rearranging gives $\Delta_{j+1} \leq \Delta_j \left(1 - \frac{\bar{\tau}_j}{B_0}\right) + \bar{\tau}_j \varepsilon_j$ for each j, and unrolling this inequality and using that $1 - \frac{\bar{\tau}_j}{B_0} < 1 \; \forall j$ as in Streeter and Golovin [2008] gives us

$$\Delta_{L+1} \leq \Delta_1 \left(\prod_{j=1}^{L} 1 - \frac{\bar{\tau}_j}{B_0} \right) + \sum_{j=1}^{L} \bar{\tau}_j \varepsilon_j.$$

By definition $B_1 = \sum_{j=1}^{L} \bar{\tau}_j \varepsilon_j$, and maximizing the product above subject to this constraint results in $\bar{\tau}_j = \frac{B_1}{L}$ for all j. Thus

$$\prod_{j=1}^{L} 1 - \frac{\bar{\tau}_j}{B_0} \leq \prod_{j=1}^{L} 1 - \frac{B_1/L}{B_0} = \left(1 + \frac{(-B_1/B_0)}{L}\right)^L < \mathrm{e}^{-B_1/B_0}$$

and so

$$f(S^*_{B_0}) - f(\bar{G}_{L+1}) = \Delta_{L+1} < \Delta_1 \mathrm{e}^{-T_1/T_0} + \sum_{j=1}^{L} \bar{\tau}_j \varepsilon_j \leq f(S^*_{B_0})\mathrm{e}^{-B_1/B_0} + \sum_{j=1}^{L} \bar{\tau}_j \varepsilon_j,$$

giving $f(\bar{G}_{\langle T_1 \rangle}) = f(\bar{G}_{L+1}) > (1 - \mathrm{e}^{-B_1/B_0}) f(S^*_{B_0}) - \sum_{j=1}^{L} \bar{\tau}_j \varepsilon_j$ as required. □

Next we prove a generalized regret bound for the original OG algorithm:

Lemma 12. *For $B \geq B' \log T$ the algorithm OG, run using Hedge as the subroutine experts algorithm, produces a sequence of schedules $S_1, \ldots, S_B$ with regret*

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(S^*_{B'}) - \sum_{t=1}^{T} f_t(S_t)\right] = \mathcal{O}\left(\mathbb{E}\left[\sum_{j=1}^{B} R_{T,1}(\mathcal{E}_j)\right]\right)$$

relative to $S^*_{B'} := \arg\max_{S \in \mathcal{S}: \ell(S) = B'} \sum_{t=1}^{T} f_t(S)$, the best-in-hindsight fixed schedule of length B', where $R_{T,1}(\mathcal{E}_j)$ is the 1-regret incurred by the jth experts algorithm.

In particular, when run with Hedge as the subroutine experts algorithm, this is $\mathcal{O}\left(\sqrt{BT \log N}\right)$.

Proof. Consider the quantity $\rho_{B,B'} := \left(1 - \mathrm{e}^{-B/B'}\right) \sum_{t=1}^{T} f_t(S^*_{B'}) - \sum_{t=1}^{T} f_t(S_t)$. As argued in Streeter and Golovin [2008], we may view the sequence of actions $a_i^1, \ldots, a_i^T$ selected by each experts algorithm $\mathcal{E}_i$ as a single 'meta-action' $\tilde{a}_i \in \mathcal{A}^T$; so the schedules $S_1, \ldots, S_T$ output by **OG** can be viewed as a single 'meta-schedule' $\tilde{S} = \langle \tilde{a}_1, \ldots, \tilde{a}_B \rangle$ over $\mathcal{A}^T$ which is a version of the greedy schedule $\bar{G}_{B+1}$ for the job $f = \frac{1}{T} \sum_{t=1}^{T} f_t$, and it may be assumed that each meta-action $\tilde{a}_t$ takes unit time per job. Thus we may write

$$\rho_{B,B'} = T\left[\left(1 - \mathrm{e}^{-B/B'}\right) f(S^*_{B'}) - f(\tilde{S})\right]$$

(after extending the domain of f appropriately). Applying Lemma 11 with $L = B$, $B_0 = B'$, $B_1 = \sum_{j=1}^{B} \bar{\tau}_j = B$ (by the unit-time assumption) then immediately gives

$$\rho_{B,B'} < T \sum_{j=1}^{B} \bar{\tau}_j \varepsilon_j = T \sum_{j=1}^{B} \varepsilon_j.$$

Taking expectations,

$$\mathbb{E}[\rho_{B,B'}] \leq T \sum_{j=1}^{B} \mathbb{E}[\varepsilon_j] = T \sum_{j=1}^{B} \mathbb{E}\left[\frac{R_{T,1}(\mathcal{E}_j)}{T}\right]$$

where $R_{T,1}(\mathcal{E}_j)$ is the 1-regret incurred by the jth experts algorithm; here we used that $\mathbb{E}[\varepsilon_j] = \mathbb{E}[R_{T,1}(\mathcal{E}_j)/n]$ as argued in Streeter and Golovin [2008]. So $\mathbb{E}[\rho_{B,B'}] \leq \mathbb{E}\left[\sum_{j=1}^{B} R_{T,1}(\mathcal{E}_j)\right]$.

The result then follows quickly: since $B \geq B' \log T$, so $\mathrm{e}^{-B/B'} \leq \mathrm{e}^{-\ln T} = T^{-1}$. Thus

$$\rho_{B,B'} \geq \left(1 - T^{-1}\right) \sum_{t=1}^{T} f_t(S^*_{B'}) - \sum_{t=1}^{T} f_t(S_t) = \mathcal{R}_{B'} - T^{-1} \sum_{t=1}^{T} f_T(S^*_{B'})$$

where $\mathcal{R}_{B'} := \sum_{t=1}^{T} f_t(S^*_{B'}) - \sum_{t=1}^{T} f_t(S_t)$ is the regret of interest. Consequently,

$$\mathcal{R}_{B'} \leq \rho_{B,B'} + T^{-1} \sum_{t=1}^{T} f_t(S^*_{B'}) \leq \rho_{B,B'} + T^{-1} \cdot T = \rho_{B,B'} + 1.$$

and the result follows.

The bound $\mathbb{E}\left[\sum_{j=1}^{B} R_{T,1}(\mathcal{E}_j)\right] = \mathcal{O}(\sqrt{BT\log N})$ when using Hedge was shown in Streeter and Golovin [2008]. □

Finally we prove the theorem on $\text{OG}_{\text{hybrid}}$:

Proof of Theorem 6. Note first that under Assumption 1, any job f, any schedule $S \in \mathcal{S}$ and any sub-schedule S' of S (i.e. the actions of S' appear in order in S) satisfy

$$f(S) \geq f(S');$$

this is immediate using monotonicity and induction.

Suppose for each $i \in [L]$ there is a fictional experts algorithm (classical full feedback multi-armed bandit algorithm) $\mathcal{E}_i$ which picks $a^*_{i,t}$ at each round t, and consider a hypothetical instance of the standard algorithm **OG** run with time allowance L and these fictional experts algorithms $\mathcal{E}_1, \ldots, \mathcal{E}_L$ as subroutines.

Since $L \geq B' \log T$ (by our assumption that $B \geq B'\widetilde{B}\log T$), by Lemma 12 the B'-regret of our **OG** instance is upper-bounded in expectation by $\sum_{i=1}^{L} \mathcal{R}_1(\mathcal{E}_i)$, where $\mathcal{R}_1(\mathcal{E}_i)$ is the total 1-regret experienced by $\mathcal{E}_i$.

But the payoff received by this **OG** instance at each round t is $f(\langle a^*_{1,t}, \ldots, a^*_{L,t}\rangle)$, which by Appendix B is upper-bounded by $f(S_t)$, the payoff of $\mathbf{OG_{hybrid}}$, since the actions $a^*_{1,t}, \ldots, a^*_{L,t}$ appear in order in S_t. So the B'-regret $\mathcal{R}_{B'}$ of $\mathbf{OG_{hybrid}}$ must be at most that of our fictional **OG** instance, giving the upper bound

$$\mathbb{E}[\mathcal{R}_{B'}] \leq \sum_{i=1}^{L} \mathbb{E}[\mathcal{R}_1(\mathcal{E}_i)].$$

It remains to argue how large each of the regret of each of these 'fictional' experts algorithms $\mathcal{E}_i$ is. Writing $a^{**}_i = \arg\min_{a\in\mathcal{A}} \sum_{t=1}^{T} c^{(i)}_t(a)$ for the best-in-hindsight fixed action under the costs passed to these subroutines, the regret incurred by $\mathcal{E}_i$ is therefore

$$\mathcal{R}_1(\mathcal{E}_i) = \sum_{t=1}^{T} c^{(i)}_t(a^*_{i,t}) - \sum_{t=1}^{T} c^{(i)}_t(a^{**}_i) \tag{10}$$

$$= \sum_{t=1}^{T} \max_{j\in[\widetilde{B}]} c^{(i)}_t(a^t_{(i-1)\widetilde{B}+j}) - \sum_{t=1}^{T} c^{(i)}_t(a^{**}_i) = \mathcal{R}_1(\mathcal{B}_i). \tag{11}$$

where $\mathcal{R}_1(\mathcal{B}_i)$ is the 1-regret incurred by multitasking bandit algorithm $\mathcal{B}_i$. So by Appendix B

$$\mathbb{E}[\mathcal{R}_{B'}] \leq \sum_{i=1}^{L} \mathbb{E}[\mathcal{R}_1(\mathcal{B}_i)] = L\mathbb{E}[\mathcal{R}_1(\mathcal{B})] = \frac{B\mathbb{E}[\mathcal{R}_1(\mathcal{B})]}{L},$$

where $\mathbb{E}[\mathcal{R}_1(\mathcal{B})]$ is the expected 1-regret of any of the instances $\mathcal{B}_1, \ldots, \mathcal{B}_L$ of $\mathcal{B}$. □

Proof of Proposition 7

Sketch proof. This is a special case of the more general result that the expected regret relative to the best-in-hindsight fixed set of size B' is at most

$$\frac{1 - B'^{-1} + \ln(N/B')}{\varepsilon} + T\sum_{j=0}^{B'-1} \binom{B}{j} \mathrm{e}^{-j\varepsilon}(1-\mathrm{e}^{-\varepsilon})^{B-j} + \mathtt{err}_{B'}$$

where $\mathtt{err}_{B'}$ is the difference in cumulative cost between the best-in-hindsight set of B' actions and the set of the top B' actions in hindsight on the given problem instance.

The proof of this is a simple adaptation of the 1-regret argument, using "an action not in the top B enters the best B'-set" as the event of interest; use the harmonic series form of the expectation of the max of exponential random variables to get a lower bound, and use a binomial counting argument to bound the probability of the event. □

Table 2: Sample means and standard deviations of normalized validation scores of FTL, $\text{OG}_{\text{hybrid}}$ and OG over black-box optimizers.

(a) $B = 1$

	Mean	StD
Best in hindsight	0.574	0
FTL	0.426	0.0202
Exp3	0.351	0.0194

(b) $B = 2$

	Mean	StD
Best in hindsight	0.710	0
FTL	0.652	0.0194
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (1, 2))$	0.577	0.0187
OG	0.519	0.0179

(c) $B = 3$

	Mean	StD
Best in hindsight	0.779	0
FTL	0.751	0.0151
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (1, 3))$	0.657	0.0191
OG	0.617	0.0166

(d) $B = 4$

	Mean	StD
Best in hindsight	0.836	0
FTL	0.813	0.0108
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (2, 2))$	0.756	0.0149
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (1, 4))$	0.716	0.0178
OG	0.689	0.0151

(e) $B = 5$

	Mean	StD
Best in hindsight	0.874	0
FTL	0.855	0.0094
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (1, 5))$	0.756	0.0150
OG	0.734	0.0140

(f) $B = 6$

	Mean	StD
Best in hindsight	0.901	0
FTL	0.888	0.0072
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (3, 2))$	0.836	0.0111
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (2, 3))$	0.814	0.0143
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = (1, 6))$	0.785	0.0137
OG	0.767	0.0157

C Experiments

C.1 Full comparison of $\text{OG}_{\text{hybrid}}$

In Table 2 we give a more detailed comparison of FTL and OG with various instantiations of $\text{OG}_{\text{hybrid}}$ on the hyperparameter-selection task from Section 4. Specifically, we include for each B and each possible pair (B_1, B_2) s.t. $B_1 B_2 = B$ a version of $\text{OG}_{\text{hybrid}}$ with B_2 internal boxes and arm budget B_1 per box. As can be seen, in all cases decreasing the greediness and adding more arms per box is beneficial in this application.

C.2 Synthetic tasks

In this section we evaluate our algorithms on three synthetic tasks. In all cases,

- let S^* be the best-in-hindsight set of B arms;
- let S_{greedy} be the greedy choice of B arms in hindsight;
- let S_{top} be the top B arms in hindsight.

Task 1: The first environment is one where $S^* = S_{\text{greedy}}$ and this set does better than S_{top}; greediness is better than picking the top B arms. There are $|\mathcal{A}| = 15$ available arms and two types of round, A and B, which occur with equal probability; costs are distributed within each round according to Table 3. So the best fixed arm set of any size up to 10 will be split evenly across arms $\{1, 2, 3, 4, 5\}$ and arms $\{11, 12, 13, 14, 15\}$—and will be the greedy choice—but for $B \leq 5$ the top B arms will always be in $\{1, 2, 3, 4, 5\}$. We see in Fig. 1 that FTL does not outperform the greedy algorithms on this task.

Task 2: The second environment is one where (approximately) $S^* = S_{\text{greedy}} = S_{\text{top}}$; greediness is good but no better than picking the top B arms. There are $|\mathcal{A}| = 10$ available arms and costs are distributed according to Table 4; because there are no groups of anticorrelated actions, the

Table 3: Cost distributions for round types A and B in the first synthetic environment; Beta distributions are parameterized by mean and variance, not shape.

Arm	A-rounds	B-rounds	Resulting mean
Actions 1 to 5	$\text{Beta}(0.4, 0.01)$	Always 1	0.7
Actions 6 to 10	$\text{Beta}(0.6, 0.01)$	Always 1	0.8
Actions 11 to 15	Always 1	$\text{Beta}(0.8, 0.01)$	0.9

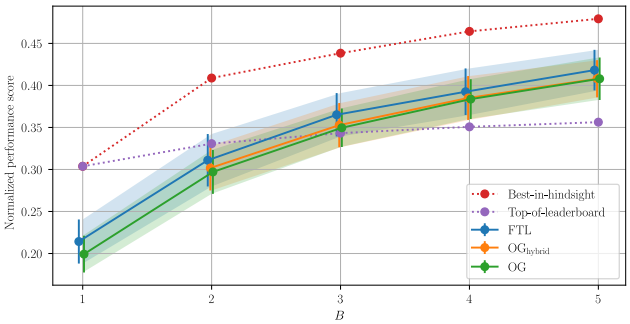

Figure 1: Performances (1-cost) on the first synthetic task. OG$_{\text{hybrid}}$ is run with FTL boxes each with arm budget 1.

performance gap between the best set and the top B arms is trivially small. The results in Fig. 2 show that FTL outperforms the greedy algorithms on this task.

Task 3: The third environment is one where $S^* = S_{\text{top}}$ and this set does better better than S_{greedy}; greediness is worse than just picking the top B arms. Suppose there are $|\mathcal{A}| = 4$ available arms and a budget of $B = 3$. Costs are deterministic and listed in Table 5 for some parameter δ which we set to 0.01. The top 3 arms are $S_{\text{top}} = \{1, 3, 4\}$ and this is also the best-in-hindsight set S^*, incurring minimum cost 0 at each round. A quick calculation shows that the greedy choice S_{greedy} is either $\{1, 2, 3\}$ or $\{1, 2, 4\}$, though, and either of these sets incur an average minimum cost of $1/8 - \delta/4$, substantially higher. Our empirical results in Table 6 show this gap in practice.

Table 4: Cost distributions in the second synthetic environment.

Arm	Distribution
1	$\text{Beta}(0.4, 0.01)$
2	$\text{Beta}(0.45, 0.01)$
3	$\text{Beta}(0.5, 0.01)$
4	$\text{Beta}(0.55, 0.01)$
5	$\text{Beta}(0.6, 0.01)$
6	$\text{Beta}(0.65, 0.01)$
7	$\text{Beta}(0.7, 0.01)$
8	$\text{Beta}(0.75, 0.01)$
9	$\text{Beta}(0.8, 0.01)$
10	$\text{Beta}(0.85, 0.01)$

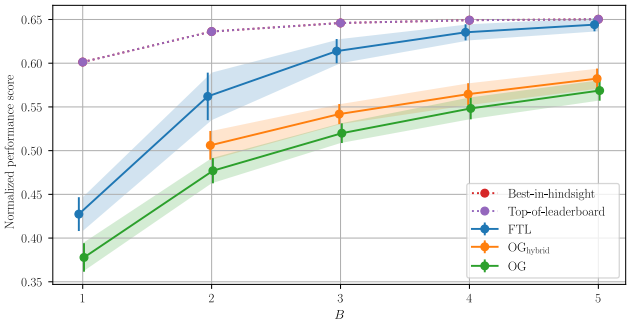

Figure 2: Performances (1-cost) on the second synthetic task. $\text{OG}_{\text{hybrid}}$ is run with FTL boxes each with arm budget 1.

Table 5: Costs in the third synthetic environment, for some parameter $\delta \in (0, 1/2)$.

Arm	**Reward at rounds** $i \equiv k \mod 4$ **for...** $k=1$	$k=2$	$k=3$	$k=4$	**Average cost**
1	$1-\delta$	$1-\delta$	0	0	$1/2-\delta/2$
2	$1/2-\delta$	$1/2-\delta$	1	1	$3/4-\delta/2$
3	0	1	0	1	$1/2$
4	1	0	1	0	$1/2$

C.3 Geometric resampling

The *geometric resampling* technique used in the second and third partial feedback versions of FTL in the experiments is adapted from Neu and Bartók [2013]. At each round cost estimates

$$\hat{c}_t(a) := \begin{cases} \frac{c_t(a)}{\hat{q}_{t,a}} & \text{if } a \text{ was pulled,} \\ 0 & \text{otherwise} \end{cases}$$

are made, where $\hat{q}_{t,a}$ is an estimate of the probability $q_{t,a} := \mathbb{P}(\text{arm } a \text{ pulled at round } t)$. These estimates are made by sampling $\frac{1}{\hat{q}_{t,a}} \sim \text{Geom}(q_{t,a})$, which is done by repeating the algorithm's execution at this round and counting how many trials are needed until a is pulled again. In practice, the number of repetitions must be capped and this introduces some bias to the estimates, but this is not problematic in practice. In fact, there is a bias variance trade-off, because $K = \max_{a \in \mathcal{A}} |\hat{c}_t(a)|$ is bounded by the number of samples we take. Therefore more samples lead to lower bias but higher variance. Using bounds similar to those of Proposition 5 as a guide (the bounds of Proposition 5 were subsequently refined after the experiments were concluded), we picked the number of samples to be $\left(N\left(\frac{TN}{\ln(N)}\right)^{\tilde{B}}\right)^{1/(2\tilde{B}+1)}$, so $K = \left(N\left(\frac{TN}{\ln(N)}\right)^{\tilde{B}}\right)^{1/(2\tilde{B}+1)}$ and $\varepsilon = \left(\frac{\ln(N)}{T}\left(\frac{\ln(N)}{TN}\right)^{\tilde{B}}\right)^{1/(2\tilde{B}+1)}$, where $\tilde{B}$ is the budget of each **FTL-partial** box.

Table 6: Means and standard deviations over 50 trials of performances (1-cost) for various combinations of FTL and online greedy algorithms in the third synthetic environment, with $\delta = 0.01$.

Algorithm	**Mean**	**StD**
Best-in-hindsight	1.000	0
Top-of-leaderboard	1.000	0
FTL	0.964	0.0145
$\textbf{OG}_{\textbf{hybrid}}$ $((B_1, B_2) = 1, 3))$	0.823	0.0200
OG	0.799	0.0202

These estimators make complete use of the information received at each round, unlike the simple one-arm uniform sampling, B arms exploiting version of FTL with partial feedback mentioned in Section 2.1. Moreover, the construction of cost estimates means no explicit exploration is necessary; an arm that hasn't been pulled for several rounds will be overtaken in estimated cumulative cost by ones that have, and so will eventually be pulled again, thus inducing a self-stabilizing property that would not occur if we used the same technique to estimate rewards $r_t(a) \coloneqq 1 - c_t(a)$ instead.

C.4 Methods

Reward definitions: For the black-box optimization experiments in Section 4, the *reward* ($1-$cost) for each black-box optimizer on each machine learning task (i.e. round) was defined as follows. This approach was inspired heavily by the Bayesmark package used in the 2020 NeurIPS BBO Challenge and which we based our implementation on [Uber, 2020].

Fix a round t and an optimizer a. Let $\texttt{opt}_t$ be an estimate of the global minimum classification/regression loss achievable (at validation, not test) on the task corresponding to round t. Define $\overline{\texttt{rand}}_t$ to be the mean performance of a random hyperparameter search on this task (i.e. the smallest loss achieved using any hyperparameter in the random search, averaged over trials).[6] Finally, let $\overline{\texttt{loss}}_t(a)$ be the actual averaged minimum loss of the optimizer a on this problem.

The reward is then defined as

$$r_i(a) \coloneqq \frac{\overline{\texttt{loss}}_t(a) - \texttt{opt}_t}{\overline{\texttt{rand}}_t(a) - \texttt{opt}_t}.$$

Conceptually, the reward is 0 when optimizer a performs as badly as a random search, and 1 when it performs as well as is possible on this task.

As per usual, the reward for a bandit algorithm selecting multiple optimizers at each round is then calculated as the maximum of the rewards of each optimizer (equivalent to the minimum of costs).

Bayesian optimizers used: The nine black-box optimization algorithms we ran the experiments in Section 4 over were as follows:

1. Hyperopt [Bergstra et al., 2015]
2. The `AUCBanditMetaTechniqueA` technique from OpenTuner [Ansel et al., 2014]
3. The `PSO_GA_Bandit` technique from OpenTuner [Ansel et al., 2014]
4. The `PSO_GA_DE` technique from OpenTuner [Ansel et al., 2014]
5. PySOT [Eriksson et al., 2019]
6. Scikit-Optimize [Head et al., 2018] using base estimator `GBRT` and acquisition objective `gp_hedge`
7. Scikit-Optimize [Head et al., 2018] using base estimator `GP` and acquisition objective `gp_hedge`
8. Scikit-Optimize [Head et al., 2018] using base estimator `GP` and acquisition objective `LCB`
9. Random search

The default settings of each package were used.

D Section 5: Linear Programming

Proof of Proposition 8 (Note: this proof was given for $B = 1$ in Arora et al. [2012] with slightly tighter bounds, and essentially remains unchanged for $B \geq 1$).

Proof. We run the FTL oracle with budget B, $N = n$ arms, and $\varepsilon = ((\ln(N) + 1)/T)^{1/(B+1)}$. In round $t \in [T]$ we do the following: Let d_t be the joint distribution over N arms returned the FTL oracle in this round. We pass d_t to the (ρ, B)-bounded oracle, and receive either a vector $x_t \in P$ or

[6] In reality this is estimated using a more statistically efficient technique than actually performing the random search, as in the Bayesmark package.

that no x_t exists which satisfies the oracle problem. Let us first suppose that we always receive an x_t for each round. Then define the cost function $c_t(i) := A_i x_t - b_i \in [-\rho, \rho]$ and pass this to FTL. After T rounds, and by scaling and translating the cost functions to lie in $[0, 1]$, Theorem 2 implies that $\forall j \in [N]$

$$\frac{\sum_{t=1}^{T} \mathbb{E}_{(i_1,\ldots,i_B)\sim d_t} \left[\min_{i\in\{i_1,\ldots,i_B\}} A_i x - b_i\right]}{T} \leq \frac{4\rho T^{\frac{1}{B+1}} (1 + \ln(N))^{\frac{B}{B+1}}}{T} + \frac{\sum_{t=1}^{T} A_j x_t - b_j}{T}$$

By assumption of the (ρ, B)-bounded oracle, the left hand side is ≥ 0. When $T \geq \left(\frac{1}{\varepsilon}\right)^{\frac{B+1}{B}} (4\rho)^{\frac{B+1}{B}} (1 + \ln(N))$, it follows that $x := \frac{\sum_{t=1}^{T} x_t}{T}$ satisfies $\forall j \in [N], A_j x \geq b_j - \varepsilon$. Since P is convex, $x \in P$ and we are done. Now suppose that in some round t we were told the oracle problem was not solvable. We claim that we can conclude that the problem is not feasible and we are done. This is because if $\exists x \in P$ s.t. $Ax \geq b$, then $\mathbb{E}_{(i_1,\ldots,i_B)\sim d} \left[\min_{i\in\{i_1,\ldots,i_B\}} A_i x - b_i\right] \geq \mathbb{E}_{(i_1,\ldots,i_B)\sim d} \left[\min_{i\in\{i_1,\ldots,i_B\}} 0\right] = 0$ and so the oracle problem would be solvable. $\square$