2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)

Adversarial Bandits with Knapsacks

Nicole Immorlica
Microsoft Research, New York, NY.
nicimm@microsoft.com.

Karthik Abinav Sankararaman
Facebook Research.
karthikabinavs@gmail.com.

Robert Schapire
Microsoft Research, New York, NY.
schapire@microsoft.com.

Aleksandrs Slivkins
Microsoft Research, New York, NY.
slivkins@microsoft.com.

***Abstract*—We consider *Bandits with Knapsacks* (henceforth, *BwK*), a general model for multi-armed bandits under supply/budget constraints. In particular, a bandit algorithm needs to solve a well-known *knapsack problem*: find an optimal packing of items into a limited-size knapsack. The BwK problem is a common generalization of numerous motivating examples, which range from dynamic pricing to repeated auctions to dynamic ad allocation to network routing and scheduling. While the prior work on BwK focused on the stochastic version, we pioneer the other extreme in which the outcomes can be chosen adversarially. This is a considerably harder problem, compared to both the stochastic version and the "classic" adversarial bandits, in that regret minimization is no longer feasible. Instead, the objective is to minimize the *competitive ratio*: the ratio of the benchmark reward to algorithm's reward.**

We design an algorithm with competitive ratio $O(\log T)$ relative to the best fixed distribution over actions, where T is the time horizon; we also prove a matching lower bound. The key conceptual contribution is a new perspective on the stochastic version of the problem. We suggest a new algorithm for the stochastic version, which builds on the framework of regret minimization in repeated games and admits a substantially simpler analysis compared to prior work. We then analyze this algorithm for the adversarial version, and use it as a subroutine to solve the latter.

Our algorithm is the first "black-box reduction" from bandits to BwK: it takes an arbitrary bandit algorithm and uses it as a subroutine. We use this reduction to derive several extensions.

***Keywords*-Multi-armed bandits; Online Packing; Adversarial Online Learning;**

I. Introduction

Multi-armed bandits is a simple abstraction for the trade-off between *exploration* and *exploitation*, *i.e.,* between making potentially suboptimal decisions for the sake of acquiring new information and using this information for making better decisions. Studied over many decades, multi-armed bandits is a very active research area spanning computer science, operations research, and economics [30, 22, 46, 25].

In this paper, we focus on bandit problems which feature supply or budget constraints, as is the case in many realistic applications. For example, a seller who experiments with prices may have a limited inventory, and a website optimizing ad placement may be constrained by the advertisers' budgets. This general problem is called *Bandits with Knapsacks* (*BwK*) since, in this model, a bandit algorithm needs effectively to solve a *knapsack problem* (find an optimal packing of items into a limited-size knapsack) or generalization thereof. The BwK model was introduced in [18] as a common generalization of numerous motivating examples, ranging from dynamic pricing to ad allocation to repeated auctions to network routing/scheduling. Various special cases with budget/supply constraints were studied previously, *e.g.,* [23, 16, 17, 79, 35].

In BwK, the algorithm is endowed with $d \geq 1$ limited resources that are consumed by the algorithm. In each round, the algorithm chooses an action (*arm*) from a fixed set of K actions. The outcome consists of a reward and consumption of each resource; all lie in $[0, 1]$. The algorithm observes *bandit feedback*, *i.e.,* only the outcome of the chosen arm. The algorithm stops at time horizon T, or when the total consumption of some resource exceeds its budget. The goal is to maximize the total reward, denoted `REW`.

For a concrete example, consider *dynamic pricing*.[1] The algorithm is a seller with limited supply of some product. In each round, a new customer arrives, the algorithm chooses a price, and the customer either buys one item at this price or leaves. A sale at price p implies reward of p and consumption of 1. This example easily extends to $d > 1$ products/resources. Now the algorithm chooses the per-unit price for each resource, and the customer decides how much of each resource to buy at this price.

Prior work on BwK focused on the stochastic version of the problem, called *Stochastic BwK*, where the outcome of each action is drawn from a fixed distribution. This problem

Full version of this paper is available at `arxiv.org` [53].
Throughout this research project, K.A. Sankararaman has been a student at University of Maryland, College Park supported in part by NSF Awards CNS 1010789, CCF 1422569, CCF-1749864 and research awards from Adobe, Amazon, and Google. Most of the results were obtained in the course of his internship at Microsoft Research NYC.

[1] See [18] for a more detailed discussion of the motivating examples.

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00022

has been solved optimally using three different techniques [18, 4], and extended in various directions in subsequent work [4, 19, 6, 5].

We go beyond the stochastic version, and instead study the most "pessimistic", adversarial version where the rewards and resource consumptions can be arbitrary. We call it *adversarial bandits with knapsacks* (*Adversarial BwK*), as it extends the classic model of "adversarial bandits" [12]. Bandits aside, this problem subsumes online packing problems [64, 28], where algorithm observes *full feedback* (the outcomes of all possible actions) in each round, and observes it *before* choosing an action.

Hardness of the problem. Adversarial BwK is a much harder problem compared to Stochastic BwK. The new challenge is that the algorithm needs to decide how much budget to save for the future, without being able to predict it. (It is also the essential challenge in online packing problems, and it drives our lower bounds.) This challenge compounds the ones already present in Stochastic BwK: that exploitation may be severely limited by the resource consumption during exploration, that optimal per-round reward no longer guarantees optimal total reward, and that the best fixed distribution over arms may perform much better than the best fixed arm. Jointly, these challenges amount to the following. An algorithm for Adversarial BwK must compete, during any given time segment $[1, \tau]$, with a distribution over arms that maximizes the total reward on this time segment. However, this distribution may behave very differently, in terms of expected per-round outcomes, compared to the optimal distribution for some other time segment $[1, \tau']$.

In more concrete terms, let $\texttt{OPT}_{\texttt{FD}}$ be the total expected reward of the *best fixed distribution* over arms. In Stochastic BwK (as well as in adversarial bandits) an algorithm can achieve sublinear regret: $\texttt{OPT}_{\texttt{FD}} - \mathbb{E}[\texttt{REW}] = o(T)$.[2] In contrast, in Adversarial BwK regret minimization is no longer possible, and we therefore are primarily interested in the *competitive ratio* $\texttt{OPT}_{\texttt{FD}} / \mathbb{E}[\texttt{REW}]$.

It is instructive to consider a simple example in which the competitive ratio is at least $\frac{5}{4} - o(1)$ for any algorithm. There are two arms and one resource with budget $\frac{T}{2}$. Arm 1 has zero rewards and zero consumption. Arm 2 has consumption 1 in each round, and offers reward $\frac{1}{2}$ in each round of the first half-time ($\frac{T}{2}$ rounds). In the second half-time, it offers either reward 1 in all rounds, or reward 0 in all rounds. Thus, there are two problem instances that coincide for the first half-time and differ in the second half-time. The algorithm needs to choose how much budget to invest in the first half-time, without knowing what comes in the second. Any choice leads to competitive ratio at least $\frac{5}{4}$ on one of the instances.

[2]More specifically, one can achieve regret $\tilde{O}(\sqrt{KT})$ for adversarial bandits [12], as well as for Stochastic BwK if all budgets are $\Omega(T)$ [18]. One can achieve sublinear regret for Stochastic BwK if all budgets are $\Omega(T^\alpha)$, $\alpha \in (0, 1)$ [18].

Extending this idea, we prove an even stronger lower bound on the competitive ratio:

$$\texttt{OPT}_{\texttt{FD}} / \mathbb{E}[\texttt{REW}] \geq \Omega(\log T). \tag{I.1}$$

Like the simple example above, the lower-bounding construction involves only two arms and only one resource, and forces the algorithm to make a huge commitment without knowing the future.

Algorithmic contributions. Our main result is an algorithm which nearly matches (I.1), achieving

$$\mathbb{E}[\texttt{REW}] \geq \frac{1}{O(\log T)} \left(\texttt{OPT}_{\texttt{FD}} - o(\texttt{OPT}_{\texttt{FD}}) \right). \tag{I.2}$$

We put forward a new algorithm for BwK, called `LagrangeBwK`, that unifies the stochastic and adversarial versions. It has a natural game-theoretic interpretation for Stochastic BwK, and admits a simpler analysis compared to the prior work. For Adversarial BwK, we use `LagrangeBwK` as a subroutine, though with a different parameter and a different analysis, to derive two algorithms: a simple one that achieves (I.2), and a more involved one that achieves the same competitive ratio with high probability. Absent resource consumption, we recover the optimal $\tilde{O}(\sqrt{KT})$ regret for adversarial bandits.

`LagrangeBwK` is based on a new perspective on Stochastic BwK. We reframe a standard linear relaxation for Stochastic BwK in a way that gives rise to a repeated zero-sum game, where the two players choose among arms and resources, respectively, and the payoffs are given by the Lagrange function of the linear relaxation. Our algorithm consists of two online learning algorithms playing this repeated game. We analyze `LagrangeBwK` for Stochastic BwK, building on the tools from regret minimization in stochastic games, and achieve a near-optimal regret bound when the optimal value and the budgets are $\Omega(T)$.[3]

Extensions. We obtain several extensions, where we derive improved performance guarantees for some scenarios. These extensions showcase the *modularity* of `LagrangeBwK`, in the sense that the two players can be implemented as arbitrary algorithms for adversarial online learning that admit a given regret bound. Each extension follows from the main results, with a different choice of the players' algorithms.

We tackle four well-known scenarios: *full feedback* [59, 44, 9], where the algorithm observes the outcomes of all possible actions after each round; *combinatorial semi-bandits* [50, 55, 11], where actions are feasible subsets of "atoms" whose individual outcomes are observed and add up to the action's total outcome; *contextual bandits* [58, 39, 3], where a *context* is observed before each round, and the algorithm competes against the best policy in a given policy class; *bandit convex optimization* [57, 42, 26], where the rewards are convex functions from arms to reals.

[3]This regime is of primary importance in prior work, *e.g.,* [23, 86].

Discussion. LagrangeBwK has numerous favorable properties. As just discussed, it is simple, unifying, modular, and yields strong performance guarantees in multiple settings. It is the first "black-box reduction" from bandits to BwK: we take a bandit algorithm and use it as a subroutine for BwK. This is a very natural algorithm for the stochastic version once the single-shot game is set up; indeed, it is immediate from prior work that the repeated game converges to the optimal distribution over arms. Its regret analysis for Stochastic BwK is extremely clean. Compared to prior work [18, 4], we side-step the intricate analysis of sensitivity of the linear program to non-uniform stochastic deviations that arise from adaptive exploration.

LagrangeBwK has a primal-dual interpretation, as arms and resources correspond respectively to primal and dual variables in the linear relaxation. Two players in the repeated game can be seen as the respective *primal algorithm* and *dual algorithm*. Compared to the rich literature on *primal-dual algorithms* [88, 28, 64] (including the more recent literature on stochastic online packing problems [36, 7, 37, 40, 66]) LagrangeBwK has a very specific and modular structure dictated by the repeated game.

Logarithmic competitive ratios are common and well-accepted in the area of approximation algorithms, and particularly in online algorithms (see Related Work for citations).

Benchmarks. We argue that the best fixed distribution over arms is an appropriate benchmark for Adversarial BwK. First, consider the total expected reward of the *best dynamic policy*, denote it $\texttt{OPT}_{\texttt{DP}}$. (The best dynamic policy is the best algorithm, in hindsight, that is allowed to switch arms arbitrarily across time-steps.) This is the strongest possible benchmark, but it is *too* strong for Adversarial BwK. Indeed, we show a simple example with just one resource (with budget B), where competitive ratio against this benchmark is at least $\frac{T}{B^2}$ for any algorithm. Second, consider the total expected reward of the *best fixed arm*, denote it $\texttt{OPT}_{\texttt{FA}}$. It is a traditional benchmark in multi-armed bandits, but is uninteresting for Adversarial BwK. We show that the competitive ratio is at least $\Omega(K)$ in the worst case, and this is matched, in expectation, by a trivial algorithm that samples one arm at random and sticks with it forever.

For Stochastic BwK, these three benchmarks are related as follows. The best fixed distribution is still the main object of interest, as far as the design and analysis of algorithms is concerned. However, all results – both ours and prior work – are almost automatically extended to compete against the best dynamic policy. The best fixed arm is a much weaker benchmark than the best fixed distribution: there are simple examples when their expected reward differs by a factor of two, in multiple special cases of interest [18].

Map of the paper. After "related work" and "preliminaries", we present our results in the following order. We develop algorithm LagrangeBwK and analyze it for Stochastic BwK in Section IV. We analyze this algorithm for the adversarial setting in Section V, and derive a simple algorithm that achieves (I.2). We develop the high-probability algorithm in Section VI. Lower bounds are presented in Section VII. Open questions are presented in Section VIII.

The detailed analysis of the high-probability algorithm, the proofs for the lower bounds, and the discussion of the extensions can be found in the full version [53].

II. Related work

The literature on regret-minimizing online learning algorithms is vast; see [30, 25, 51] for background. Most relevant are two algorithms for adversarial rewards/costs: Hedge for full feedback [45], and EXP3 for bandit feedback [12]; both are based on the weighted majority algorithm from [59].

Stochastic BwK was introduced and optimally solved in [18]. Subsequent work extended these results to soft supply/budget constraints [4], a more general notion of rewards[4] [4], combinatorial semi-bandits [75], and contextual bandits [19, 6, 5]. Several special cases with budget/supply constraints were studied previously: dynamic pricing [23, 16, 24, 86], dynamic procurement [17, 79] (a version of dynamic pricing where the algorithm is a buyer rather than a seller), dynamic ad allocation [80, 35], and a version with a single resource and unlimited time [49, 82, 83, 38]. While all this work is on regret minimization, [47, 48] studied closely related Bayesian formulations.

Stochastic BwK was optimally solved using three different algorithms [18, 4], with extremely technical and delicate analyses. All three algorithms involve inherently 'stochastic' techniques such as "successive elimination" and "optimism under uncertainty", and do not appear to extend to the adversarial version. One of them, PrimalDualBwK from [18], is a primal-dual algorithm superficially similar to ours. Indeed, it decouples into two online learning algorithms: a "primal" algorithm which chooses among arms, and a "dual" algorithm similar to ours, which chooses among resources. However, the two algorithms are not playing a repeated game in any meaningful sense, let alone a zero-sum game. The primal algorithm operates under a much richer input: it takes the entire outcome vector for the chosen arm, as well as the "dual distribution" – the distribution over resources chosen by the dual algorithm. Further, the primal algorithm is very problem-specific: it interprets the dual distribution as a vector of costs over resources, and chooses arms with largest reward-to-cost ratios, estimated using "optimism under uncertainty".

Our approach to using regret minimization in games can be traced to [43, 45] (see Ch. 6 in [76]), who showed how a repeated zero-sum game played by two agents yields an approximate Nash equilibrium. This approach has been used

[4]The total reward is determined by the time-averaged outcome vector, but can be an arbitrary Lischitz-concave function thereof.

as a unifying algorithmic framework for several problems: boosting [43], linear programs [9], maximum flow [34], and convex optimization [1, 85]. While we use a result with the $1/\sqrt{t}$ convergence rate for the equilibrium property, recent literature obtains faster convergence for cumulative payoffs (but not for the equilibrium property) under various assumptions (e.g., [69, 81, 87]).

Repeated Lagrangian games, in conjunction with regret minimization in games, have been used in a series of recent papers [72, 52, 74, 56, 2, 73], as an algorithmic tool to solve convex optimization problems; application domains range from differential privacy to algorithmic fairness to learning from revealed preferences. All these papers deal with deterministic games (*i.e.,* same game matrix in all rounds). Reframing the problem in terms of repeated Lagrangian games is a key technical insight in this work. Most related to our paper are [74, 73], where a repeated Lagrangian game is used as a subroutine (the "inner loop") in an online algorithm; the other papers solve an offline problem. We depart from this prior work in several respects: we use a stochastic game, we deal with some subtleties specific to Stochastic BwK, and we provide a very different analysis for our main results on Adversarial BwK, where we cannot rely on the standard machinery.

Online packing problems (*e.g.,* [29, 37], see [28] for a survey) can be seen as a special case of Adversarial BwK with a much more permissive feedback model: the algorithm observes full feedback (the outcomes for all actions) before choosing an action. Online packing subsumes various *online matching* problems, including the *AdWords problem* [65] motivated by ad allocation (see [64] for a survey). While we derive $O(\log T)$ competitive ratio against $\texttt{OPT}_{\texttt{FD}}$, online packing admits a similar result against $\texttt{OPT}_{\texttt{DP}}$.

Another related line of work concerns online convex optimization with constraints [62, 63, 33, 68, 32]. Their setting differs from ours in several important respects. First, the action set is a convex subset of $\mathbb{R}^K$ (and the algorithms rely on the power to choose arbitrary actions in this set). In particular, there is no immediate way to handle discrete action sets.[5] Second, convexity/concavity is assumed on the rewards and resource consumption. Third, in addition to bandit feedback, full feedback is observed for the resource consumption, and (in all papers except [32]) one also observes either full feedback on rewards or the rewards gradient around the chosen action. Fourth, their algorithm only needs to satisfy the budget constraints at the time horizon (whereas in BwK the budget constraints hold for all rounds). Fifth, their fixed-distribution benchmark is weaker than ours: essentially, its time-averaged consumption must be small enough at each round t. Due to these differences, their setting admits sublinear regret in the adversarial setting.

[5]Unless there is full feedback, in which case one can use a standard reduction whereby actions in online convex optimization correspond to distributions over actions in a K-armed bandit problem.

Logarithmic competitive ratios are quite common in prior work on approximation algorithms and online algorithms, *e.g.,* in the context of the set cover problem [60, 54], buy-at-bulk network design [14], sparsest cut [10], and dial-a-ride problem [31], the online k-server problem [20], online packing/covering problems [15], online set cover [8], online network design [84], and online paging [41].

Simultaneous work. Two very recent papers came to our attention after the initial publication of this paper on `arxiv.org`. Rivera et al. [71] consider online convex optimization with knapsacks (essentially, the full-feedback version of our extension to bandit convex optimization). Focusing on the stochastic version, they design an algorithm similar to `LagrangeBwK`, with a similar regret bound and analysis. They also claim an extension to bandit feedback, without providing any details (such as a precise statement of Lemma III.1 in terms of the regret property (III.2)).

Rangi et al. [70] consider Adversarial BwK in the special case when there is only one constrained resource, including time. They attain sublinear regret, *i.e.,* a regret bound that is sublinear in T. They also assume a known lower bound $c_{\min} > 0$ on realized per-round consumption of each resource, and their regret bound scales as $1/c_{\min}$. They also achieve polylog(T) instance-dependent regret for the stochastic version using the same algorithm (matching results from prior work on the stochastic version). BwK with only one constrained resource (including time) is a much easier problem, compared to the general case with multiple resources studied in this paper, in the following sense. First, the single-resource version admits much stronger performance guarantees (polylog(T) vs. $\sqrt{T}$ regret bounds for Stochastic BwK, and sublinear regret vs. approximation ratio for Adversarial BwK). Second, the optimal all-knowing time-invariant policy is the best fixed arm, rather than the best fixed distribution over arms.

III. Preliminaries

We use bold fonts to represent vectors and matrices. We use standard notation whereby, for a positive integer K, $[K]$ stands for $\{1, 2, \ldots, K\}$, and Δ_K denotes the set of all probability distributions on $[K]$. Some of the notation introduced further is summarized in Appendix D.

Bandits with Knapsacks (BwK). There are T rounds, K possible actions and d resources, indexed as $[T], [K], [d]$, respectively. In each round $t \in [T]$, the algorithm chooses an action $a_t \in [K]$ and receives an *outcome vector* $\boldsymbol{o}_t = (r_t; c_{t,1}, \ldots, c_{t,d}) \in [0,1]^{d+1}$, where r_t is a reward and $c_{t,i}$ is consumption of each resource $i \in [d]$. Each resource i is endowed with budget $B_i \leq T$. The game stops early, at some round $\tau_{\text{alg}} < T$, when/if the total consumption of any resource exceeds its budget. The algorithm's objective is to maximize its total reward. Without loss of generality

all budgets are the same: $B_1 = B_2 = \ldots = B_d = B$.[6]

The outcome vectors are chosen as follows. In each round t, the adversary chooses the *outcome matrix* $\boldsymbol{M}_t \in [0,1]^{K\times(d+1)}$, where rows correspond to actions. The outcome vector $\boldsymbol{o}_t$ is defined as the a_t-th row of this matrix, denoted $\boldsymbol{M}_t(a_t)$. Only this row is revealed to the algorithm. The adversary is deterministic and *oblivious*, meaning that the entire sequence $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_T$ is chosen before round 1. A problem instance of BwK consists of (known) parameters (d, K, T, B), and the (unknown) sequence $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_T$.

In the stochastic version of BwK, henceforth termed *Stochastic BwK*, each outcome matrix $\boldsymbol{M}_t$ is chosen from some fixed but unknown distribution $\mathcal{D}_{\texttt{BwK}}$ over the outcome matrices. A problem instance consists of (known) parameters (d, K, T, B), and the (unknown) distribution $\mathcal{D}_{\texttt{BwK}}$.

Following prior work [18, 4], we assume, w.l.o.g., that one of the resources is a *dummy resource* similar to time; formally, each action consumes B/T units of this resource per round (we only need this for Stochastic BwK). Further, we posit that one of the actions is a *null action*, which lets the algorithm skips a round: it has 0 reward and consumes 0 amount of each resource other than the dummy resource.

Benchmarks. Let $\texttt{REW(ALG)} = \sum_{t\in[\tau_{\text{alg}}]} r_t$ be the total reward of algorithm ALG in the BwK problem. Our benchmark is the *best fixed distribution*, a distribution over actions which maximizes $\mathbb{E}[\texttt{REW}(\cdot)]$ for a particular problem instance. The expected total reward of this distribution is denoted $\texttt{OPT}_{\texttt{FD}}$.

For Stochastic BwK, one can compete with the *best dynamic policy*: an algorithm that maximizes $\mathbb{E}[\texttt{REW}(\cdot)]$ for a particular problem instance. Essentially, this algorithm knows the latent distribution $\mathcal{D}_{\texttt{BwK}}$ over outcome matrices. Its expected total reward is denoted $\texttt{OPT}_{\texttt{DP}}$.

Adversarial online learning. To state the framework of "regret minimization in games" below, we need to introduce the protocol of *adversarial online learning*, see Figure 1.

In this protocol, the adversary can use previously chosen arms to choose the payoff vector $\boldsymbol{f}_t$, but not the algorithm's random seed. The distribution $\boldsymbol{f}_t$ is chosen as a deterministic function of history. (The history at round t consists, for each round $s < t$, of the chosen action a_s and the observed feedback in this round.) We focus on two feedback models: *bandit feedback* (no auxiliary feedback) and *full feedback* (the entire payoff vector $\boldsymbol{f}_t$). The version for costs can be defined similarly, by setting the payoffs to be the negative of costs.

We are interested in adversarial online learning algorithms

[6]To see that this is indeed w.l.o.g., for each resource i, divide all per-round consumptions $c_{t,i}$ by B_i/B, where $B := \min_{i\in[d]} B_i$ is the smallest budget. In the modified problem instance, all consumptions still lie in $[0,1]$, and all the budgets are equal to B.

Given: action set A, payoff range $[b_{\min}, b_{\max}]$.
In each round $t \in [T]$,

1. the adversary chooses a payoff vector $\boldsymbol{f}_t \in [b_{\min}, b_{\max}]^K$;
2. the algorithm chooses a distribution $\boldsymbol{p}_t$ over A, without observing $\boldsymbol{f}_t$,
3. algorithm's chosen action $a_t \in A$ is drawn independently from $\boldsymbol{p}_t$;
4. payoff $f_t(a_t)$ is received by the algorithm.

Figure 1: Adversarial online learning

with known upper bounds on *regret*,

$$R_{\texttt{AOL}}(T) := \left[\max_{a\in A} \sum_{t\in[T]} f_t(a)\right] - \left[\sum_{t\in[T]} f_t(a_t)\right]. \quad \text{(III.1)}$$

The benchmark here is the total payoff of the best arm, according to the payoff vectors actually chosen by the adversary. More precisely, we assume high-probability regret bounds of the following form:

$$\forall \delta > 0$$
$$\Pr\left[\, R_{\texttt{AOL}}(T) \le (b_{\max} - b_{\min})\, R_\delta(T) \,\right] \ge 1-\delta, \quad \text{(III.2)}$$

for some function $R_\delta(\cdot)$. We will actually use a stronger version implied by (III.2),[7]

$$\Pr\left[\, \forall \tau \in [T] \quad R_{\texttt{AOL}}(\tau) \le (b_{\max} - b_{\min})\, R_{\delta/T}(T) \,\right] \ge 1-\delta \qquad \forall \delta > 0. \quad \text{(III.3)}$$

Algorithms EXP3.P [12] for bandit feedback, and Hedge [44] for full feedback, satisfy (III.2) with, resp.,

$$R_\delta(T) = O\left(\sqrt{|A|\, T \log(T/\delta)}\right) \quad \text{and}$$
$$R_\delta(T) = O\left(\sqrt{T \log(|A|/\delta)}\right). \quad \text{(III.4)}$$

Regret minimization in games. We build on the framework of *regret minimization in games*. A *zero-sum game* $(A_1, A_2, \boldsymbol{G})$ is a game between two players $i \in \{1,2\}$ with action sets A_1 and A_2 and payoff matrix $\boldsymbol{G} \in \mathbb{R}^{A_1\times A_2}$. If each player i chooses an action $a_i \in A_i$, the outcome is a number $G(a_1, a_2)$. Player 1 receives this number as *reward*, and player 2 receives it as *cost*. A *repeated zero-sum game* $\mathcal{G}$ with action sets A_1 and A_2, time horizon T and game matrices $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_T \in \mathbb{R}^{A_1\times A_2}$ is a game between two algorithms, $\texttt{ALG}_1$ and $\texttt{ALG}_2$, which proceeds over T rounds such that each round t is a zero-sum game $(A_1, A_2, \boldsymbol{G}_t)$.

[7]Regret bound (III.3) follows from (III.2) using a simple "zeroing-out" trick: for a given round $\tau \in [T]$, the adversary can set all future payoffs to some fixed value $x \in [b_{\min}, b_{\max}]$, in which case $R_{\texttt{AOL}}(\tau) = R_{\texttt{AOL}}(T)$.

The goal of $\texttt{ALG}_1$ is to maximize the total reward, and the goal of $\texttt{ALG}_2$ is to minimize the total cost.

The game $\mathcal{G}$ is called *stochastic* if the game matrix $\boldsymbol{G}_t$ in each round t is drawn independently from some fixed distribution. For such games, we are interested in the *expected game*, defined by the expected game matrix $\boldsymbol{G} = \mathbb{E}[\boldsymbol{G}_t]$. We can relate the algorithms' performance to the minimax value of $\boldsymbol{G}$.

Lemma III.1. *Consider a stochastic repeated zero-sum game between algorithms $\texttt{ALG}_1$ and $\texttt{ALG}_2$, with payoff range $[b_{\min}, b_{\max}]$. Assume that each $\texttt{ALG}_j$, $j \in \{1,2\}$ is an algorithm for adversarial online learning, as per Figure 1, which satisfies regret bound (III.2) with $R_\delta(T) = R_{j,\delta}(T)$.*

Let τ be some fixed round in the repeated game. For each algorithm $\texttt{ALG}_j$, $j \in \{1,2\}$, let A_j be its action set, let $p_{t,j} \in \Delta_{A_j}$ be the distribution chosen in each round t, and let $\bar{\boldsymbol{p}}_j = \frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,j}$ be the average play distribution at round τ. Let v^* be the minimax value for the expected game $\boldsymbol{G} = \mathbb{E}[\boldsymbol{G}_t]$.

Then for each $\delta > 0$, with probability at least $1 - 2\delta$,

$$\forall \boldsymbol{p}_2 \in \Delta_{A_2} \quad \bar{\boldsymbol{p}}_1^{\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_2 \geq v^* - \tfrac{1}{\tau}(b_{\max} - b_{\min}) \cdot \left(R_{1,\,\delta/T}(T) + R_{2,\,\delta/T}(T) + 4\sqrt{2T\log(T/\delta)} \right). \tag{III.5}$$

Eq. (III.5) states that the average play of player 1 is approximately optimal against any distribution chosen by player 2.[8] This lemma is well-known for the deterministic case (*i.e.,* when $\boldsymbol{G}_t = \boldsymbol{G}$ for each round t), and folklore for the stochastic case. We provide a proof in Appendix D for the sake of completeness.

IV. A New Algorithm for Stochastic BwK

We present a new algorithm for Stochastic BwK, based on the framework of regret minimization in games. This is a very natural algorithm once the single-shot game is set up, and it allows for a very clean regret analysis. We will also use this algorithm as a subroutine for the adversarial version.

On a high level, we define a stochastic zero-sum game for which a mixed Nash equilibrium corresponds to an optimal solution for a linear relaxation of the original problem. Our algorithm consists of two regret-minimizing algorithms playing this game. The framework of regret minimization in games guarantees that the average primal and dual play distributions ($\bar{\boldsymbol{p}}_1$ and $\bar{\boldsymbol{p}}_2$ in Lemma III.1) approximate the mixed Nash equilibrium in the expected game, which correspondingly approximates the optimal solution.

A. Linear relaxation and Lagrange functions

We start with a linear relaxation of the problem that all prior work relies on. This relaxation is stated in terms of expected rewards/consumptions, *i.e.,* implicitly, in terms of the expected outcome matrix $\boldsymbol{M} = \mathbb{E}[\boldsymbol{M}_t]$. We explicitly formulate the relaxation in terms of $\boldsymbol{M}$, and this is essential for the subsequent developments. For ease of notation, we write the a-th row of $\boldsymbol{M}$, for each action $a \in [K]$, as

$$\boldsymbol{M}(a) = (r^{\boldsymbol{M}}(a);\, c_1^{\boldsymbol{M}}(a),\, \ldots,\, c_d^{\boldsymbol{M}}(a)),$$

so that $r^{\boldsymbol{M}}(a)$ is the expected reward and $c_i^{\boldsymbol{M}}(a)$ is the expected consumption of each resource i.

Essentially, the relaxation assumes that each instantaneous outcome matrix $\boldsymbol{M}_t$ is equal to the expected outcome matrix $\boldsymbol{M} = \mathbb{E}[\boldsymbol{M}_t]$. The relaxation seeks the best distribution over actions, focusing on a single round with budgets rescaled as B/T. This leads to the following linear program (LP):

$$\begin{array}{lll} \text{maximize} & \sum_{a\in[K]} X(a)\, r^{\boldsymbol{M}}(a) & \\ \text{such that} & & \\ & \sum_{a\in[K]} X(a) = 1 & \\ \forall i \in [d] & \sum_{a\in[K]} X(a)\, c_i^{\boldsymbol{M}}(a) \leq B/T & \text{(IV.1)} \\ \forall a \in [K] & 0 \leq X(a) \leq 1. & \end{array}$$

We denote this LP by $\texttt{LP}_{\boldsymbol{M},B,T}$. The solution $\boldsymbol{X}$ is the best fixed distribution over actions, according to the relaxation. The value of this LP, denoted $\texttt{OPT}_{\texttt{LP}}(\boldsymbol{M}, B, T)$, is the expected per-round reward of this distribution. It is also the total reward of $\boldsymbol{X}$ in the relaxation, divided by T. We know from [18] that

$$T \cdot \texttt{OPT}_{\texttt{LP}}(\boldsymbol{M}, B, T) \geq \texttt{OPT}_{\texttt{DP}} \geq \texttt{OPT}_{\texttt{FD}}, \tag{IV.2}$$

where $\texttt{OPT}_{\texttt{DP}}$ and $\texttt{OPT}_{\texttt{FD}}$ are the total expected rewards of, respectively, the best dynamic policy and the best fixed distribution. In words, $\texttt{OPT}_{\texttt{DP}}$ is sandwiched between the total expected reward of the best fixed distribution and that of its linear relaxation.

Associated with the linear program $\texttt{LP}_{\boldsymbol{M},B,T}$ is the *Lagrange function* $\mathcal{L} = \mathcal{L}_{\boldsymbol{M},B,T}$. It is a function $\mathcal{L} : \Delta_K \times \mathbb{R}^d_{\geq 0} \to \mathbb{R}$ defined as

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\lambda}) := \sum_{a\in[K]} X(a)\, r^{\boldsymbol{M}}(a) + \sum_{i\in[d]} \lambda_i \left[1 - \frac{T}{B} \sum_{a\in[K]} X(a)\, c_i^{\boldsymbol{M}}(a) \right]. \tag{IV.3}$$

The values $\lambda_1, \ldots, \lambda_d$ in Eq. (IV.3) are called the *dual variables*, as they correspond to the variables in the dual LP. Lagrange functions are meaningful due to their max-min property (*e.g.,* Theorem D.2.2 in [21]):

$$\min_{\boldsymbol{\lambda}\geq 0} \max_{\boldsymbol{X}\in\Delta_K} \mathcal{L}(\boldsymbol{X}, \boldsymbol{\lambda}) = \max_{\boldsymbol{X}\in\Delta_K} \min_{\boldsymbol{\lambda}\geq 0} \mathcal{L}(\boldsymbol{X}, \boldsymbol{\lambda}) = \texttt{OPT}_{\texttt{LP}}(\boldsymbol{M}, B, T). \tag{IV.4}$$

This property holds for our setting because $\texttt{LP}_{\boldsymbol{M},B,T}$ has at least one feasible solution (namely, one that puts probability

[8] If each player j chooses distribution $p_j \in \Delta_{A_j}$, and the game matrix is $\boldsymbol{G}$, then expected reward/cost is $\boldsymbol{p}_1^{\mathrm{T}} \boldsymbol{G} \boldsymbol{p}_2$.

one on the null action), and the optimal value of the LP is bounded.

Remark IV.1. *We use the linear program $\text{LP}_{\boldsymbol{M},B,T}$ and the associated Lagrange function $\mathcal{L}_{\boldsymbol{M},B,T}$ throughout the paper. Both are parameterized by an outcome matrix $\boldsymbol{M}$, budget B and time horizon T. In particular, we can plug in an arbitrary $\boldsymbol{M}$, and we heavily use this ability throughout. For the adversarial version, it is essential to plug in parameter $T_0 \leq T$ instead of the time horizon T. For the analysis of the high-probability result in Adversarial BwK, we use a rescaled budget $B_0 \leq B$ instead of budget B.*

B. Our algorithm: repeated Lagrangian game

The Lagrange function $\mathcal{L} = \mathcal{L}_{\boldsymbol{M},B,T}$ from (IV.3) defines the following zero-sum game: the *primal player* chooses an arm a, the *dual player* chooses a resource i, and the payoff is a number

$$\mathcal{L}(a,i) = r^{\boldsymbol{M}}(a) + 1 - \tfrac{T}{B}\, c_i^{\boldsymbol{M}}(a). \tag{IV.5}$$

The primal player receives this number as a reward, and the dual player receives it as cost. This game is termed the *Lagrangian game* induced by $\mathcal{L}_{\boldsymbol{M},B,T}$. This game will be crucial throughout the paper.

The Lagrangian game is related to the original linear program as follows:

Lemma IV.2. *Assume one resource is the dummy resource. Consider the linear program $\text{LP}_{\boldsymbol{M},B,T}$, for some outcome matrix $\boldsymbol{M}$. Then the value of this LP equals the minimax value v^* of the Lagrangian game induced by $\mathcal{L}_{\boldsymbol{M},B,T}$. Further, if $(\boldsymbol{X},\boldsymbol{\lambda})$ is a mixed Nash equilibrium in the Lagrangian game, then $\boldsymbol{X}$ is an optimal solution to the LP.*

The proof can be found in Appendix B. The idea is that because of the special structure of the LP, the second equality in (IV.4) also holds when the dual vector $\boldsymbol{\lambda}$ is restricted to distributions.

Consider a repeated version of the Lagrangian game. Formally, the *repeated Lagrangian game* with parameters $B_0 \leq B$ and $T_0 \leq T$ is a repeated zero-sum game between the *primal algorithm* that chooses among arms and the *dual algorithm* that chooses among resources. Each round t of this game is the Lagrangian game induced by the Lagrange function $\mathcal{L}_t := \mathcal{L}_{\boldsymbol{M}_t,B_0,T_0}$, where $\boldsymbol{M}_t$ is the round-t outcome matrix. Note that we use parameters B_0, T_0 instead of budget B and time horizon T.[9]

Remark IV.3. *Consider repeated Lagrangian game for Stochastic BwK (with $B_0 = B$ and $T_0 = T$). The payoffs in the expected game are defined by the expected Lagrange function $\mathcal{L} := \mathbb{E}[\mathcal{L}_t]$. By linearity, $\mathcal{L}$ is the Lagrange function for the expected outcome matrix $\boldsymbol{M} = \mathbb{E}[\boldsymbol{M}_t]$:*

$$\mathcal{L} := \mathbb{E}[\mathcal{L}_t] = \mathcal{L}_{\boldsymbol{M},B,T}. \tag{IV.6}$$

Our algorithm, called LagrangeBwK, is very simple: it is a repeated Lagrangian game in which the primal algorithm receives bandit feedback, and the dual algorithm receives full feedback.

To set up the notation, let a_t and i_t be, respectively, the chosen arm and resource in round t. The payoff is therefore $\mathcal{L}_t(a_t, i_t)$. It can be rewritten in terms of the observed outcome vector $\boldsymbol{o}_t = (r_t; c_{t,1}, \ldots, c_{t,d})$ (which corresponds to the a_t-th row of the instantaneous outcome matrix $\boldsymbol{M}_t$):

$$\mathcal{L}_t(a_t, i_t) = r_t + 1 - \tfrac{T_0}{B_0}\, c_{t,i_t} \in [-\tfrac{T_0}{B_0} + 1, 2]. \tag{IV.7}$$

Note that the payoff range is $[b_{\min}, b_{\max}] = [-\frac{T_0}{B_0} + 1]$.

With this notation, the pseudocode for LagrangeBwK is summarized in Algorithm 1. The pseudocode is simple and self-contained, without referring to the formalism of repeated games and Lagrangian functions. Note that the algorithm is implementable, in the sense that the outcome vector $\boldsymbol{o}_t$ revealed in each round t of the BwK problem suffices to generate full feedback for the dual algorithm.

Algorithm 1: Algorithm LagrangeBwK.

input: parameters B_0, T_0, primal algorithm ALG_1, dual algorithm ALG_2.

```
// ALG_1, ALG_2 are adversarial
//    online learning algorithms
//    with bandit feedback
//    and full feedback, resp.
```

for *round* $t = 1, 2, 3, \ldots$ **do**

1) ALG_1 returns arm $a_t \in [K]$, algorithm ALG_2 returns resource $i_t \in [d]$.
2) arm a_t is chosen, outcome vector $\boldsymbol{o}_t = (r_t(a_t); c_{t,1}(a_t), \ldots, c_{t,d}(a_t)) \in [0,1]^{d+1}$ is observed.
3) The payoff $\mathcal{L}_t(a_t, i_t)$ from (IV.7) is reported to ALG_1 as reward, and to ALG_2 as cost.
4) The payoff $\mathcal{L}_t(a_t, i)$ is reported to ALG_2 for each resource $i \in [d]$.

C. Performance guarantees

We consider algorithm LagrangeBwK with parameter $T_0 = T$. We assume the existence of the dummy resource; this is to ensure that the crucial step, Eq. (IV.13), works out even if the algorithm stops at time T, without exhausting any actual resources. We obtain a regret bound that is non-trivial whenever $B > \Omega(\sqrt{T})$, and is optimal, up to log factors, in the regime when $\min(\text{OPT}_{\text{DP}}, B) > \Omega(T)$.

[9] These parameters are needed only for the adversarial version. For Stochastic BwK we use $B_0 = B$ and $T_0 = T$.

Theorem IV.4. *Consider Stochastic BwK with K arms, d resources, time horizon T, and budget B. Assume that one resource is a dummy resource (with deterministic consumption $\frac{B}{T}$ for each arm). Fix the failure probability parameter $\delta \in (0,1)$. Consider algorithm* `LagrangeBwK` *with parameters $B_0 = B$ and $T_0 = T$.*

If EXP3.P and Hedge are used as the primal and the dual algorithms, respectively, then the algorithm achieves the following regret bound, with probability at least $1-\delta$:

$$\texttt{OPT}_{\texttt{DP}} - \texttt{REW}(\texttt{LagrangeBwK}) \leq O\left(\frac{T}{B}\sqrt{TK\log(dT/\delta)}\right). \quad \text{(IV.8)}$$

In general, suppose each algorithm $\texttt{ALG}_j$ satisfies a regret bound (III.2) with $R_\delta(T) = R_{j,\delta}(T)$ and payoff range $[b_{\min}, b_{\max}] = [-\frac{T}{B}+1, 2]$. Then with probability at least $1 - O(\delta T)$ it holds that

$$\texttt{OPT}_{\texttt{DP}} - \texttt{REW}(\texttt{LagrangeBwK}) \leq O\left(\tfrac{T}{B}\right)\left(R_{1,\frac{\delta}{T}}(T) + R_{2,\frac{\delta}{T}}(T) + \sqrt{T\log\tfrac{dT}{\delta}}\right). \quad \text{(IV.9)}$$

Remark IV.5. *To obtain (IV.8) from the "black-box" result (IV.9), we use regret bounds in Eq. (III.4).*

Remark IV.6. *From [18], the optimal regret bound for Stochastic BwK is*

$$\texttt{OPT}_{\texttt{DP}} - \mathbb{E}[\texttt{REW}] \leq \tilde{O}\left(\sqrt{K\,\texttt{OPT}_{\texttt{DP}}}\,(1+\sqrt{\texttt{OPT}_{\texttt{DP}}/B})\right).$$

Thus, the regret bound (IV.8) is near-optimal if $\min(\texttt{OPT}_{\texttt{DP}}, B) > \Omega(T)$, and non-trivial if $B > \Omega(\sqrt{T})$.

We next prove the "black-box" regret bound (IV.9). For the sake of analysis, consider a version of the repeated Lagrangian game that continues up to the time horizon T. In what follows, we separate the "easy steps" from what we believe is the crux of the proof.

Notation. Let $\boldsymbol{X}_t$ be the distribution chosen in round t by the primal algorithm $\texttt{ALG}_1$. Let $\overline{\boldsymbol{X}}_\tau := \frac{1}{\tau}\sum_{t\in[\tau]}\boldsymbol{X}_t$ be the distribution of average play up to round τ. Let $\boldsymbol{M} = \mathbb{E}[\boldsymbol{M}_t]$ be the expected outcome matrix. Let $\boldsymbol{r} = (r^{\boldsymbol{M}}(a) : a \in [K])$ be the vector of expected rewards over the actions. Likewise, $\boldsymbol{c}_i = (c_i^{\boldsymbol{M}}(a) : a \in [K])$ be the vector of expected consumption of each resource $i \in [d]$.

Using Azuma-Hoeffding inequality. Consider the first τ rounds, for some $\tau \in [T]$. The average reward and resource-i consumption over these rounds are close to $\overline{\boldsymbol{X}}_\tau \cdot \boldsymbol{r}$ and $\overline{\boldsymbol{X}}_\tau \cdot \boldsymbol{c}_i$, respectively, with high probability. Specifically, a simple usage of Azuma-Hoeffding inequality (Lemma A.1) implies that

$$\tfrac{1}{\tau}\textstyle\sum_{t\in[\tau]} r_t \geq \overline{\boldsymbol{X}}_\tau \cdot \boldsymbol{r} - R_0(\tau)/\tau, \quad \text{(IV.10)}$$

$$\tfrac{1}{\tau}\textstyle\sum_{t\in[\tau]} c_{i,t} \leq \overline{\boldsymbol{X}}_\tau \cdot \boldsymbol{c}_i + R_0(\tau)/\tau, \quad \forall i \in [d], \quad \text{(IV.11)}$$

hold with probability at least $1-\delta$, where $R_0(\tau) = O(\sqrt{\tau\log(d/\delta)})$.

Regret minimization in games. Let us apply the machinery from regret minimization in games to the repeated Lagrangian game. Consider the game matrix $\boldsymbol{G}$ of the expected game. Using Eq. (IV.6) and Lemma IV.2, we conclude that the minimax value of $\boldsymbol{G}$ is $v^* = \texttt{OPT}_{\texttt{LP}}(\boldsymbol{M}, B, T)$.

We apply Lemma III.1, with a fixed stopping time $\tau \in [T]$. Recall that the payoff range is $b_{\max} - b_{\min} = \frac{T}{B} + 1$. Thus, with probability at least $1-2\delta$ it holds that

$$\boldsymbol{\lambda} \in \Delta_d : \quad \overline{\boldsymbol{X}}_\tau^{\mathrm{T}}\,\boldsymbol{G}\,\boldsymbol{\lambda} \geq v^* - \tfrac{1}{\tau}(\tfrac{T}{B}+1)\cdot\texttt{reg}(T), \quad \text{(IV.12)}$$

where the regret term is $\texttt{reg}(T) := R_{1,\delta/T}(T) + R_{2,\delta/T}(T) + 4\sqrt{2T\log(T/\delta)}$.

Crux of the proof. Let us condition on the event that (IV.10), (IV.11), and (IV.12) hold for each $\tau \in [T]$. By the union bound, this event holds with probability at least $1-3\delta T$.

Let τ denote the *stopping time* of the algorithm, the first round when the total consumption of some resource exceeds its budget. Let i be the resource for which this happens; hence,

$$\textstyle\sum_{t\in[\tau]} c_{i,t} > B. \quad \text{(IV.13)}$$

Let us use Eq. (IV.12) with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(i)}$, the point distribution for this resource. Then by Eq. (IV.6) we have,

$$\overline{\boldsymbol{X}}_\tau^{\mathrm{T}}\,\boldsymbol{G}\,\boldsymbol{\lambda}^{(i)} = \mathcal{L}_{\boldsymbol{M},B,T}(\overline{\boldsymbol{X}}_\tau, \boldsymbol{\lambda}^{(i)})$$

By definition of Lagrange function this equals,

$$= \overline{\boldsymbol{X}}_\tau \cdot \boldsymbol{r} + 1 - \tfrac{T}{B}\,\overline{\boldsymbol{X}}_\tau \cdot \boldsymbol{c}_i$$

Plugging in (IV.10) and (IV.11), this is upper-bounded by

$$\leq \tfrac{1}{\tau}\left(\left(\textstyle\sum_{t\in[\tau]} r_t\right) - \left(\tfrac{T}{B}\textstyle\sum_{t\in[\tau]} c_{i,t}\right) + \tau - (1+\tfrac{T}{B})\,R_0(\tau)\right)$$

Plugging in Eq. (IV.13), it can further be upper-bounded by,

$$\leq \tfrac{1}{\tau}\left(\left(\textstyle\sum_{t\in[\tau]} r_t\right) + \tau - T - (1+\tfrac{T}{B})\,R_0(\tau)\right).$$

Plugging this into Eq. (IV.12) and rearranging, we obtain

$$\textstyle\sum_{t\in[\tau]} r_t \geq \tau\, v^* + T - \tau - (1+\frac{T}{B})\cdot\texttt{reg}(T).$$

Since $v^* \leq 1$ (because $v^* = \texttt{OPT}_{\texttt{LP}}$, as we've proved),

$$\texttt{REW}(\texttt{LagrangeBwK}) = \sum_{t\in[\tau]} r_t \geq T\,v^* - (1+\tfrac{T}{B})\cdot\texttt{reg}(T).$$

The claimed regret bound (IV.9) follows by Eq. (IV.2), completing the proof of Theorem IV.4.

Algorithm 2: simple algorithm for Adversarial BwK.

input: scale parameter $\kappa > 0$,
guess range $[g_{\min}, g_{\max}]$,
algorithms $\texttt{ALG}_1$, $\texttt{ALG}_2$ as in Algorithm 1
Choose u uniformly at random from $\{0, 1, \ldots, u^{\max}\}$, where $u^{\max} = \left\lceil \log_\kappa \frac{g_{\max}}{g_{\min}} \right\rceil$.
Guess the value of $\texttt{OPT}_{\texttt{FD}}$ as $\widehat{g} = g_{\min} \cdot \kappa^u$.
Run $\texttt{LagrangeBwK}$ with algorithms $\texttt{ALG}_1$, $\texttt{ALG}_2$ and parameters $B_0 = B$ and $T_0 = \widehat{g}/\kappa$.

V. A simple algorithm for Adversarial BwK

We present and analyze an algorithm for Adversarial BwK which achieves $O(d^2 \log T)$ competitive ratio, in expectation, up to a low-order additive term. Our algorithm is very simple: we randomly guess the value of $\texttt{OPT}_{\texttt{FD}}$ and run $\texttt{LagrangeBwK}$ with parameter T_0 driven by this guess. The analysis is very different, however, since we cannot rely on the machinery from regret minimization in stochastic games. The crux of the analysis (Lemma V.5) is re-used to analyze the high-probability algorithm in the next section.

The intuition for our algorithm can be explained as follows. $\texttt{LagrangeBwK}$ builds on adversarial online learning algorithms $\texttt{ALG}_j$, and appears plausibly applicable to Adversarial BwK. We analyze it for Adversarial BwK, with an arbitrary parameter T_0 (see Lemma V.5, the crux of our analysis), and find that it performs best when T_0 is tailored to $\texttt{OPT}_{\texttt{FD}}$ up to a constant multiplicative factor. This is precisely what our algorithm achieves via the random guess.

Our algorithm is presented as Algorithm 2. We randomly guess the value of $\texttt{OPT}_{\texttt{FD}}$ from within a specified range $[g_{\min}, g_{\max}]$, up to the specified multiplicative factor of $\kappa > 0$. We consider multiplicative scales $[\kappa^u, \kappa^{u+1}]$, $u \in \mathbb{N}$, and we guess uniformly at random among all possible u. Our analysis works as long as $\texttt{OPT}_{\texttt{FD}} \in [g_{\min}, g_{\max}]$ and $\kappa \geq d + 1$; then we obtain competitive ratio $\kappa^2 \left\lceil \log \frac{g_{\max}}{g_{\min}} \right\rceil$ up to a low-order additive term. As a corollary, we obtain competitive ratio $\kappa^2 \lceil \log T \rceil$ with no assumptions.

Theorem V.1. *Consider Adversarial BwK with K arms, d resources, time horizon T, and budget B. Assume that one of the arms is a* null arm *that has zero reward and zero resource consumption. Consider Algorithm 2 with scale parameter $\kappa \geq d+1$. Suppose algorithms $\texttt{ALG}_j$ that satisfy the regret bound (III.2) with $\delta = T^{-2}$ and regret term $R_\delta(T) = R_{j,\delta}(T)$, for any known payoff range $[b_{\min}, b_{\max}]$.*

If $\texttt{OPT}_{\texttt{FD}} \in [g_{\min}, g_{\max}]$ then the expected reward of Algorithm 2 satisfies

$$\mathbb{E}[\texttt{REW}] \geq (\texttt{OPT}_{\texttt{FD}} - \texttt{reg}) / \left(\kappa^2 \left\lceil \log_\kappa \tfrac{g_{\max}}{g_{\min}} \right\rceil \right), \qquad \text{(V.1)}$$

where $\texttt{reg} = (1 + \frac{\texttt{OPT}_{\texttt{FD}}}{\kappa B}) \left(R_{1,\delta/T}(T) + R_{2,\delta/T}(T) \right)$.

Taking $[g_{\min}, g_{\max}] = [1, T]$, we obtain

$$\mathbb{E}[\texttt{REW}] \geq (\texttt{OPT}_{\texttt{FD}} - \texttt{reg}) / \left(\kappa^2 \lceil \log_\kappa T \rceil \right). \qquad \text{(V.2)}$$

Remark V.2. *One can use algorithms EXP3.P for $\texttt{ALG}_1$ and Hedge for $\texttt{ALG}_2$, with regret bounds given by (III.4), and achieve the regret term $\texttt{reg} = O\left(1 + \frac{\texttt{OPT}_{\texttt{FD}}}{\kappa B}\right) \sqrt{TK \log(Td/\delta)}$. We obtain a meaningful performance guarantee as long as, say, $\texttt{reg} < \texttt{OPT}_{\texttt{FD}}/2$; this requires $\texttt{OPT}_{\texttt{FD}}$ and B to be at least $\widetilde{\Omega}(\sqrt{TK})$.*

Remark V.3. *We define the outcome matrices slightly differently compared to Section IV in that we do not posit a dummy resource. Formally, we assume that the null arm has zero consumption in every resource. This is essential for case 1 (*i.e., *when $\tau_{alg} \leq \sigma$) in the analysis of Lemma V.5.*

If a problem instance of Adversarial BwK is actually an instance of adversarial bandits, then we recover the optimal $\tilde{O}(\sqrt{KT})$ regret. (This easily follows by examining the proof of Lemma V.5.)

Lemma V.4. *Consider $\texttt{LagrangeBwK}$, with algorithms EXP3.P for $\texttt{ALG}_1$ and Hedge for $\texttt{ALG}_2$, for an instance of Adversarial BwK with zero resource consumption. This algorithm obtains $\tilde{O}(\sqrt{KT})$ regret, for any parameters $B_0, T_0 > 0$. Accordingly, so does Algorithm 2 with any scale parameter $\kappa > 0$.*

A. Analysis: proof of Theorem V.1 and Lemma V.4

Stopped linear program. Let us set up a linear relaxation that is suitable to the adversarial setting. The expected outcome matrix is no longer available. Instead, we use *average* outcome matrices:

$$\overline{\boldsymbol{M}}_\tau = \tfrac{1}{\tau} \textstyle\sum_{t \in [\tau]} \boldsymbol{M}_t, \qquad \text{(V.3)}$$

the average up to a given intermediate round $\tau \in [T]$. Similar to the stochastic case, the relaxation assumes that each instantaneous outcome matrix $\boldsymbol{M}_t$ is equal to the average outcome matrix $\overline{\boldsymbol{M}}_\tau$. What is different now is that the relaxation depends on τ: using $\overline{\boldsymbol{M}}_\tau$ is tantamount to stopping precisely at this round.

With this intuition in mind, for a particular end-time τ we consider the linear program (IV.1), parameterized by the time horizon τ and the average outcome matrix $\overline{\boldsymbol{M}}_\tau$. Its value, $\texttt{OPT}_{\texttt{LP}}(\overline{\boldsymbol{M}}_\tau, B, \tau)$, represents the per-round expected reward, so it needs to be scaled by the factor of τ to obtain the total expected reward. Finally, we maximize over τ. Thus, our linear relaxation for Adversarial BwK is defined as follows:

$$\texttt{OPT}_{\texttt{LP}}^{[T]} := \max_{\tau \in [T]} \tau \cdot \texttt{OPT}_{\texttt{LP}}(\overline{\boldsymbol{M}}_\tau, B, \tau) \geq \texttt{OPT}_{\texttt{FD}}. \qquad \text{(V.4)}$$

The inequality in (V.4) is proved in the appendix (Section C).

Regret bounds for $\texttt{ALG}_j$. Since each algorithm $\texttt{ALG}_j$, $j \in \{1, 2\}$ satisfies regret bound (III.2) with $\delta = T^{-2}$ and

$R_\delta(T) = R_{j,\delta}(T)$, it also satisfies a stronger version (III.3) with the same parameters. Recall from (IV.7) that the payoff range is $[b_{\min}, b_{\max}] = [-\frac{T_0}{B} + 1, 2]$. For succinctness, let $U_j(T|T_0) = (1+\frac{T_0}{B})\, R_{j,\,\delta/T}(T)$ denote the respective regret term in (III.3).

Let us apply these regret bounds to our setting. Let $a_t \in [K]$ and $i_t \in [d]$ be, resp., the chosen arm and resource in round t. We represent the outcomes as vectors over arms: $\boldsymbol{r}_t, \boldsymbol{c}_{t,i} \in [0,1]^K$ denote, resp., reward vector and resource-i consumption vector for a given round t. Recall that the round-t payoffs in LagrangeBwK are given by the Lagrange function $\mathcal{L}_t := \mathcal{L}_{\boldsymbol{M}_t,B,T_0}$ such that

$$\mathcal{L}_t(a,i) = r_t(a) + 1 - \tfrac{T_0}{B}\, c_{t,i}(a) \tag{V.5}$$

for each arm a and resource i. Consider the total Lagrangian payoff at a given round $\tau \in [T]$:

$$\textstyle\sum_{t\in[\tau]} \mathcal{L}_t(a_t, i_t) = \texttt{REW}_\tau + \tau - W_\tau, \tag{V.6}$$

where $\texttt{REW}_\tau = \sum_{t\in[\tau]} r_t(a_t)$ is the total reward up to round τ, and $W_\tau = \frac{T_0}{B} \sum_{t\in[\tau]} c_{t,i_t}(a_t)$ is the *consumption term*. The regret bounds sandwich (V.6) from above and below:

$$\begin{aligned}
&\left(\max_{a\in[K]} \sum_{t\in[\tau]} \mathcal{L}_t(a,i_t)\right) - U_1(T|T_0)\\
&\quad\le \texttt{REW}_\tau + \tau - W_\tau\\
&\quad\le \left(\min_{i\in[d]} \sum_{t\in[\tau]} \mathcal{L}_t(a_t,i)\right) + U_2(T|T_0).
\end{aligned} \tag{V.7}$$

This holds for all $\tau \in [T]$, with probability at least $1 - 2\delta$. The first inequality in (V.7) is due to the primal algorithm, and the second is due to the dual algorithm. Call them *primal* and *dual* inequality, respectively.

Crux of the proof. We condition on the event that (V.7) holds for all $\tau \in [T]$, which we call the *clean event*. The crux of the analysis is encapsulated in the following lemma, which analyzes an execution of LagrangeBwK with an arbitrary parameter T_0 under the clean event.

Lemma V.5. *Consider an execution of* LagrangeBwK *with $B_0 = B$ and an arbitrary parameter T_0 such that the clean event holds. Fix an arbitrary round $\sigma \in [T]$, and consider the LP value relative to this round:*

$$f(\sigma) := \texttt{OPT}_{\texttt{LP}}(\overline{\boldsymbol{M}}_\sigma, B, \sigma). \tag{V.8}$$

The algorithm's reward up to round σ satisfies

$$\texttt{REW}_\sigma \ge \min(T_0,\, \sigma\cdot f(\sigma) - dT_0) - \big(\, U_1(T|T_0) + U_2(T|T_0)\,\big). \tag{V.9}$$

Taking σ to be the maximizer in (V.4), algorithm's reward satisfies

$$\texttt{REW} \ge \min(T_0,\, \texttt{OPT}_{\texttt{FD}} - dT_0) - \big(\, U_1(T|T_0) + U_2(T|T_0)\,\big). \tag{V.10}$$

Proof: Let τ_{alg} be the stopping time of the algorithm. We consider two cases, depending on whether some resource is exhausted at time σ. In both cases, we focus on the round $\min(\tau_{\text{alg}}, \sigma)$.

Case 1: $\tau_{\text{alg}} \le \sigma$ and some resource is exhausted. Let us focus on round $\tau = \tau_{\text{alg}}$. If i is the exhausted resource, then $\sum_{t\in[\tau]} c_{t,i}(a_t) > B$. Let us apply the dual inequality in (V.7) for this resource:

$$\begin{aligned}
&\texttt{REW}_\tau + \tau - W_\tau - U_2(T|T_0)\\
&\le \textstyle\sum_{t\in[\tau]} \mathcal{L}_t(a_t,i)\\
&= \texttt{REW}_\tau + \tau - \tfrac{T_0}{B} \textstyle\sum_{t\in[\tau]} c_{t,i}(a_t)\\
&\le \texttt{REW}_\tau + \tau - T_0.
\end{aligned}$$

It follows that $W_\tau \ge T_0 - U_2(T|T_0)$.

Now, let us apply the primal inequality in (V.7) for the null arm. Recall that the reward and consumption for this arm is 0, so $\mathcal{L}_t(\texttt{null}, i_t) = 1$ for each round t. Therefore,

$$\texttt{REW}_\tau + \tau - W_\tau + U_1(T|T_0) \ge \textstyle\sum_{t\in[\tau]} \mathcal{L}_t(\texttt{null}, i_t) = \tau.$$

We conclude that $\texttt{REW}_\tau \ge W_\tau - U_1(T|T_0) \ge T_0 - U_1(T|T_0) - U_2(T|T_0)$.

Case 2: $\tau_{\text{alg}} \ge \sigma$. Let us focus on round σ. Consider the linear program $\texttt{LP}_{\overline{\boldsymbol{M}}_\sigma,B,\sigma}$, and let $\boldsymbol{X}^* \in \Delta_K$ be an optimal solution to this LP. The primal inequality in (V.7) implies

$$\begin{aligned}
&\texttt{REW}_\sigma + \sigma - W_\sigma + U_1(\sigma)\\
&\quad\ge \max_{a\in[K]} \sum_{t\in[\sigma]} \mathcal{L}_t(a,i_t)\\
&\quad\ge \textstyle\sum_{t\in[\sigma]} \sum_{a\in[K]} X^*(a)\, \mathcal{L}_t(a,i_t)\\
&\quad= \sigma + \sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{r}_t - \tfrac{T_0}{B} \sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{c}_{t,i_t}
\end{aligned}$$

Rearranging and using the fact that $\sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{r_t} = \sigma\cdot f(\sigma)$ (by optimality of $\boldsymbol{X}^*$) we get,

$$\texttt{REW}_\sigma \ge \sigma\cdot f(\sigma) - \frac{T_0}{B} \sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{c}_{t,i_t} - U_1(T|T_0). \tag{V.11}$$

$\sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{c}_{t,i} \le B$ for each resource i, since $\boldsymbol{X}^*$ is a feasible solution for $\texttt{OPT}_{\texttt{LP}}(\overline{\boldsymbol{M}}_\sigma, B, \sigma)$. Then,

$$\sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{c}_{t,i_t} \le \sum_{i\in[d]} \sum_{t\in[\sigma]} \boldsymbol{X}^*\cdot\boldsymbol{c}_{t,i} \le dB. \tag{V.12}$$

Plugging (V.12) into (V.11), we conclude that $\texttt{REW}_\sigma \ge \sigma\cdot f(\sigma) - dT_0 - U_1(T|T_0)$.

Conclusions from the two cases imply (V.10). ∎

Wrapping up. If $\texttt{OPT}_{\texttt{FD}}$ lies in the guess range $[g_{\min}, g_{\max}]$, then some guess $\widehat{g}$ is approximately correct:

$$\texttt{OPT}_{\texttt{FD}}/\kappa \leq \widehat{g} \leq \texttt{OPT}_{\texttt{FD}}.$$

With such a guess $\widehat{g}$, and provided that $\kappa \geq d+1$, we have $T_0 = \widehat{g}/\kappa \geq \texttt{OPT}_{\texttt{FD}}/\kappa^2$, and

$$\texttt{OPT}_{\texttt{FD}} - dT_0 \geq \texttt{OPT}_{\texttt{FD}}(1 - \tfrac{d}{\kappa}) \geq \texttt{OPT}_{\texttt{FD}}/\kappa.$$

So, by Lemma V.5, the algorithm's execution with this guess, assuming the clean event, satisfies (V.10) with $\min(T_0, \texttt{OPT}_{\texttt{FD}} - dT_0) \geq \texttt{OPT}_{\texttt{FD}}/\kappa^2$ and $T_0 \leq \texttt{OPT}_{\texttt{FD}}/\kappa$. The regret term for this guess is

$$U_1(T|T_0) + U_2(T|T_0) \leq \\ (1 + \tfrac{\texttt{OPT}_{\texttt{FD}}}{\kappa B}) \left(R_{1,\,\delta/T}(T) + R_{2,\,\delta/T}(T)\right).$$

To complete the proof of Theorem V.1, we obtain a suitable guess $\widehat{g}$ with probability $1/\left\lceil \log_\kappa \frac{g_{\max}}{g_{\min}} \right\rceil$.

Proof Sketch of Lemma V.4. Recall that in the adversarial bandit setting we have $\boldsymbol{c}_{i,t} = 0$ for every $i \in [d]$ and every $t \in [T]$. We re-analyze Lemma V.5 with $\sigma = T$. Notice that case 1 never occurs. Thus we obtain obtain Eq. (V.11) in case 2. Note that $\frac{T_0}{B} \sum_{t\in[\sigma]} \boldsymbol{X^*} \cdot \boldsymbol{c}_{t,i_t} = 0$ since $\boldsymbol{c}_{i,t} = 0$. Therefore, we obtain

$$\texttt{REW}_T \geq T \cdot f(T) - U_1(T|T_0).$$

We now argue that $T \cdot f(T) = \max_{a\in[K]} \sum_{t\in[T]} r_t(a)$. Let $\boldsymbol{X}^*$ be the optimal distribution over the arms. Thus $\sum_{t\in[T]} \boldsymbol{X^*} \cdot \boldsymbol{r_t} = T \cdot f(T)$. Note that since $\boldsymbol{c}_{i,t} = 0$ the only constraint on $\boldsymbol{X}^*$ is that it lies in Δ_K. Therefore the maximizer is a point distribution on $\max_{a\in[K]} \sum_{t\in[T]} r_t(a)$. This proof does not rely on any specific value for B_0, T_0. The payoff range is $[b_{\max}, b_{\min}] = [1, 2]$, which implies that $U_1(T|T_0) = \tilde{O}\left(\sqrt{KT}\right)$.

VI. High-probability algorithm for Adversarial BwK

We recover the $O(\log T)$ approximation ratio for Adversarial BwK, but with high probability rather than merely in expectation. Our algorithm uses `LagrangeBwK` as a subroutine, and re-uses the adversarial analysis thereof (Lemma V.5). We do not attempt to optimize the regret term.

The algorithm is considerably more complicated compared to Algorithm 2. Instead of making one random guess $\widehat{g}$ for the value of $\texttt{OPT}_{\texttt{LP}}^{[T]}$, we iteratively refine this guess over time. The algorithm proceeds in phases. In the beginning of each phase, we start a fresh instance of `LagrangeBwK` with parameter T_0 defined by the current value of $\widehat{g}$.[10] We update the guess $\widehat{g}$ in each round (in a way specified later), and stop the phase once $\widehat{g}$ becomes too large compared to its initial value in this phase. We invoke `LagrangeBwK` with a rescaled budget $B_0 = B/\Theta(\log T)$. Within each phase, we simulate the BwK problem with budget B_0: we stop `LagrangeBwK` once the consumption of some resource in this phase exceeds B_0. For the remainder of the phase, we play the null arm with probability $1 - \gamma_0$ and do uniform exploration with the remaining probability, for some parameter $\gamma_0 \in (0, 1)$ (here and elsewhere, *uniform exploration* refers to choosing each action with equal probability). The pseudocode is summarized in Algorithm 3.

Algorithm 3: High-probability algorithm for Adversarial BwK.

input: scale parameter κ, exploration parameter γ_0, algorithms $\texttt{ALG}_1$, $\texttt{ALG}_2$ as in Algorithm 1
Initialize $\widehat{g} = 1$.
for *each phase* **do**
  Start a fresh instance ALG of `LagrangeBwK` with parameters $B_0 = B/2\lceil \log_\kappa T \rceil$ and $T_0 = \widehat{g}/(\lceil \log_\kappa T \rceil \kappa^2)$.
  for *each round in this phase* **do**
    Recompute the global estimate $\widehat{g}$
    if $\widehat{g} > T_0/\kappa$ **then** start a new phase
    if *consumption of all resources in this phase does not exceed* B_0 **then**
      Play the action chosen by ALG, observe the outcome and report it back to ALG.
    else
      Choose the null arm with probability $1 - \gamma_0$, do uniform exploration otherwise

To complete algorithm's specification, let us define how to update the guess $\widehat{g}$ in each round t. The guess, denoted $\widehat{g}_t$, is an estimate for $\texttt{OPT}_{\texttt{LP}}^{[t]}$, as defined in (V.4). We form this estimate using a standard *inverse propensity scoring* (*IPS*) technique. Let $\boldsymbol{p}_t$ and a_t be, resp., the distribution and the arm chosen by the primal algorithm in round t. The instantaneous outcome matrix $\boldsymbol{M}_t$ is estimated by matrix $\boldsymbol{M}_t^{\texttt{ips}} \in [0, \infty)^{K\times d}$ such that each row $\boldsymbol{M}_t^{\texttt{ips}}(a)$ is defined as follows:

$$\boldsymbol{M}_t^{\texttt{ips}}(a) := \mathbf{1}_{\{a_t = a\}} \, \tfrac{1}{f_t(a_t)} \, \boldsymbol{M}_t(a).$$

For a given end-time τ, the average outcome matrix $\overline{\boldsymbol{M}}_\tau$ from (V.3) is estimated as

$$\overline{\boldsymbol{M}}_\tau^{\texttt{ips}} := \tfrac{1}{\tau} \sum_{t\in[\tau]} \boldsymbol{M}_t^{\texttt{ips}}.$$

Finally, we plug this estimate into (V.3) and define

$$\widehat{g}_t := \max_{\tau\in[T]} \; \tau \cdot \texttt{OPT}_{\texttt{LP}}(\overline{\boldsymbol{M}}_\tau^{\texttt{ips}}, B, \tau). \qquad \text{(VI.1)}$$

For the analysis, we will assume that the primal algorithm

[10] The idea of restarting the algorithm in each phase is similar to the standard "doubling trick" in the online machine learning literature, but much more delicate in our setting.

does some uniform exploration:

$$p_t(a) \geq \gamma > 0$$
for each arm $a \in [K]$ and each round $t \in [T]$. (VI.2)

Theorem VI.1. *Consider Adversarial BwK with K arms, d resources, time horizon T, and budget B; assume $B > 4\,T^{3/4}$. Suppose that one of the arms is a* null arm *that has zero reward and zero resource consumption. Let $\delta > 0$ be the failure probability parameter.*

Consider Algorithm 3 with parameters $\kappa \geq d+1$ and $\gamma_0 = T^{-1/4}$. Assume that each algorithm $\texttt{ALG}_j$, $j \in \{1,2\}$, satisfies the regret bound (III.2) with payoff range $[b_{\min}, b_{\max}] = [-\frac{T}{B}+1, 2]$ and regret term $R_\delta(T) = R_{j,\delta}(T)$. Assume that the primal algorithm $\texttt{ALG}_1$ satisfies (VI.2) with parameter $\gamma \geq T^{-1/4}$.

Then the total reward $\texttt{REW}$ collected by Algorithm 3 satisfies

$$\Pr\left[\, \texttt{REW} \geq \frac{\texttt{OPT}_{\texttt{FD}} - \texttt{reg}}{2\kappa^4 \lceil \log_\kappa T \rceil} \,\right] \;\geq\; 1 - O(\delta T), \quad \text{(VI.3)}$$

where the regret term is

$$\texttt{reg} = \tfrac{T}{B}\left(K\, T^{3/4}\, \log^{1/2}(\tfrac{1}{\delta}) + R_{1,\,\delta/T}(T) + R_{2,\,\delta/T}(T) \right).$$

Remark VI.2. *Using algorithms EXP3.P for $\texttt{ALG}_1$ and Hedge for $\texttt{ALG}_2$, we can achieve (VI.3) with*

$$\texttt{reg} = O\left(\tfrac{TK}{B}\right)\, T^{3/4}\, \sqrt{\log(T/\delta)}.$$

This is because EXP3.P, with appropriately modified uniform exploration term $\gamma = T^{-1/4}$, satisfies the regret bound (III.2) with $R_\delta(\tau) = O(T^{3/4})\sqrt{K \log \frac{T}{\delta}}$, and for Hedge we can (still) use Eq. (III.4). The theorem is meaningful whenever, say, $\texttt{reg} < \texttt{OPT}_{\texttt{FD}}/2$. The latter requires $\texttt{OPT}_{\texttt{FD}} \cdot \frac{B}{K} > \widetilde{\Omega}(T^{7/4})$.

Remark VI.3. *Like in Theorem V.1, we posit that the null arm does not consume any resources.*

We provide a proof sketch below. The detailed proof, quite lengthy and technical, can be found in the full version [53].

Proof Sketch:

The proof consists of several steps. First, we argue that the guess $\widehat{g}_t$ is close to $\texttt{OPT}_{\texttt{LP}}^{[t]}$ with high probability. This argument only relies on the uniform exploration property (VI.1) and the definition of IPS estimators, not on any properties of the algorithm. We immediately obtain concentration for the average outcome matrices; a somewhat subtle point is to derive concentration on the respective LP-values.

Next, we focus on a particular phase in the execution of the algorithm. We say that a phase is *full* if the stopping condition $\widehat{g}_t > T_0/\kappa$ has fired. We focus on the last full phase. We prove there is enough reward to be collected in this phase. Essentially, letting τ_1, τ_2 be, resp., the start and end time of this phase, we consider the BwK problem restricted to time interval $[\tau_1, \tau_2]$, and lower-bound the LP-value of this problem in terms of the LP-value of the original problem. Finally, we use the adversarial analysis of $\texttt{LagrangeBwK}$ (Lemma V.5) to guarantee that our algorithm actually collects that value.

Because of the stopping condition $\widehat{g}_t > T_0/\kappa$, there can be at most $\lceil \log_\kappa T \rceil$ phases. Therefore, rescaling the budget to $B_0/2\lceil \log_\kappa T \rceil$ guarantees that the algorithm consumes at most $B/2$ of the budget. We then argue that, with high-probability, the additional uniform exploration in each phase, consumes a budget of at most $B/2$ with high-probability. Thus, the algorithm never runs out of budget. ■

VII. Lower bounds

We provide the theorem statements and the constructions for the lower bounds on the competitive ratio that we have claimed in Section I: the $\Omega(\log T)$ lower bound w.r.t. the best fixed distribution benchmark ($\texttt{OPT}_{\texttt{FD}}$), the $\Omega(T)$ lower bound w.r.t. the best dynamic policy benchmark ($\texttt{OPT}_{\texttt{DP}}$), and the $\Omega(K)$ lower bound w.r.t. the best fixed arm benchmark ($\texttt{OPT}_{\texttt{FA}}$). All lower-bounds are for a randomized algorithm against an oblivious adversary. The proofs are deferred to the full version [53].

Theorem VII.1. *Consider Adversarial BwK with a single resource ($d = 1$) and K arms. Consider any randomized algorithm for this problem, and let $\texttt{REW}$ denote its reward. Then:*

(a) $\texttt{OPT}_{\texttt{FD}}/\,\mathbb{E}[\texttt{REW}] \geq \frac{5}{4} - o(1)$ *for some problem instance (warmup: the example in the Introduction).*

(b) $\texttt{OPT}_{\texttt{FD}}/\,\mathbb{E}[\texttt{REW}] \geq \Omega(\log T)$ *for some problem instance.*

(c) $\texttt{OPT}_{\texttt{DP}}/\,\mathbb{E}[\texttt{REW}] \geq T/B^2$ *for some problem instance, for any given budget B.*

(d) $\texttt{OPT}_{\texttt{FA}}/\,\mathbb{E}[\texttt{REW}] \geq \Omega(K)$ *for some problem instance.*

Remark VII.2. *The lower bounds for parts (a,b,c) hold (even) for problem instances with $K = 2$ arms. The lower bounds in parts (a,b) hold even for a much more permissive feedback model from the online packing literature, namely, when the algorithm observes the outcome vector for all actions in a given round, and moreover does it* before *it chooses an arm in this round.*

The constructions (for the respective parts of the theorem) are as follows:

(a) There are two arms and one resource with budget $B = \frac{T}{2}$. Arm 1 has zero rewards and zero consumption. Arm 2 has consumption 1 in each round, and offers reward $\frac{1}{2}$ in each round of the first half-time ($\frac{T}{2}$ rounds). In the second half-time, arm 1 offers either reward 1 in all rounds, or reward 0 in all rounds. More formally, there are two problem instances, call them $\mathcal{I}_1$ and $\mathcal{I}_2$, that coincide for the first half-time and differ in the second half-time.

The intuition is that given a random instance as input the algorithm needs to choose how much budget to invest in the first half-time, without knowing what comes in the second, and any choice (in expectation) leads to the claimed competitive ratio.

(b) There is one resource with budget B, and two arms, denoted A_0, A_1. Arm A_0 is the "null arm" that has zero reward and zero consumption. The consumption of arm A_1 is 1 in all rounds. The rewards of A_1 are defined as follows. We partition the time into $\frac{T}{B}$ phases of duration B each (for simplicity, assume that B divides T). We consider $\frac{T}{B}$ problem instances; for each instance $\mathcal{I}_\tau$, $\tau \in \left[\frac{T}{B}\right]$ arm A_1 has positive rewards up to and including phase τ; after that all rewards are 0. In each phase $\sigma \in [\tau]$, arm A_1 has reward σ/T in each round. The lower bound holds for any B in the interval $[\Omega(\log^3 T), O(T^{1-\alpha})]$, for some constant $\alpha \in (0,1)$.

(c) There is one resource with budget B, and two arms, denoted A_0, A_1. Arm A_0 is the 'null arm' that has zero reward and zero consumption. The consumption of arm A_1 is 1 in all rounds. The rewards of A_1 are defined as follows. We partition the time into $\frac{T}{B}$ phases of duration B each (for simplicity, assume that B divides T). We consider $\frac{T}{B}$ problem instances; for each instance $\mathcal{I}_\tau$, $\tau \in [T/B]$ arm A_1 has 0 reward in all phases except phase τ; in phase τ it has a reward of 1 in each round.

(d) There is one resource with budget B, and K arms denoted by $A_1, A_2, \ldots, A_K$. Arm A_K is the 'null arm' that has zero reward and zero consumption. There are K instances in the family. In instance $\mathcal{I}_j$, all arms $A_{j'}$ where $j' > j$ have 0 reward and 0 consumption in all time-steps. Consider an instance $\mathcal{I}_j$ for some $j \in [K-1]$ and an arm $j' \leq j$. Arm $A_{j'}$ has a reward of $\frac{1}{K^{K-j'}}$ and consumption of 1 in all time-steps in phase j' and has a reward of 0 and consumption of 0 in every other time-step. Thus the rewards and consumption are bounded in the interval $[0,1]$ for every arm and every time-step in all instances in this family.

VIII. Open Questions

We use essentially the same algorithm, `LagrangeBwK`, to solve both stochastic and adversarial version of bandits with knapsacks. Yet, we use it with different parameter T_0 and a slightly different definition of the outcome matrices. Indeed, recall that in the stochastic setting there a 'dummy resource' with strictly positive consumption for all arms, whereas in the adversarial version the null arm must have zero consumption for all resources. Can we solve both versions with *exactly* the same algorithm? One concrete goal would be to achieve $O(\log T)$ competitive ratio in the adversarial version, and $o(T)$ regret for the stochastic version in the regime $\min(B, \texttt{OPT}_{\texttt{FD}}) \geq \Omega(T)$. A similar "best of both worlds" result has been obtained for bandits without budget/supply constraints: one algorithm that achieves optimal regret rates for both adversarial bandits and stochastic bandits, without knowing which environment it is in [27, 78, 13]. Further developments focused on mostly stochastic environments with a small amount of adversarial behavior [78, 77, 61, 87]; similar questions are relevant to BwK as well, once the basic "best-of-both-worlds" question is resolved.

Acknowledgements

The authors are grateful to Robert Kleinberg, Akshay Krishnamurthy, Steven Wu, and Chicheng Zhang for many insightful conversations on online machine learning and related subjects.

References

[1] Jacob D Abernethy and Jun-Kun Wang. On frank-wolfe and equilibrium computation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6584–6593, 2017.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *Fairness, Accountability, and Transparency in Machine Learning (FATML)*, 2017.

[3] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *31st Intl. Conf. on Machine Learning (ICML)*, 2014.

[4] Shipra Agrawal and Nikhil R. Devanur. Bandits with concave rewards and convex knapsacks. In *15th ACM Conf. on Economics and Computation (ACM EC)*, 2014.

[5] Shipra Agrawal and Nikhil R. Devanur. Linear contextual bandits with knapsacks. In *29th Advances in Neural Information Processing Systems (NIPS)*, 2016.

[6] Shipra Agrawal, Nikhil R. Devanur, and Lihong Li. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. In *29th Conf. on Learning Theory (COLT)*, 2016.

[7] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

[8] Noga Alon, Baruch Awerbuch, and Yossi Azar. The online set cover problem. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 100–105. ACM, 2003.

[9] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[10] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

[11] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Minimax policies for combinatorial prediction games. In *24th Conf. on Learning Theory (COLT)*, pages 107–132, 2011.

[12] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002. Preliminary version in *36th IEEE FOCS*, 1995.

[13] Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *29th Conf. on Learning Theory (COLT)*, 2016.

[14] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 542–547. IEEE, 1997.

[15] Yossi Azar, Niv Buchbinder, TH Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, and Joseph Naor. Online algorithms for covering and packing problems with convex objectives. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 148–157. IEEE, 2016.

[16] Moshe Babaioff, Shaddin Dughmi, Robert D. Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. *ACM Trans. on Economics and Computation*, 3(1):4, 2015. Special issue for *13th ACM EC*, 2012.

[17] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In *13th ACM Conf. on Electronic Commerce (EC)*, pages 128–145, 2012.

[18] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *J. of the ACM*, 65(3), 2018. Preliminary version in *FOCS 2013*.

[19] Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. Resourceful contextual bandits. In *27th Conf. on Learning Theory (COLT)*, 2014.

[20] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. A polylogarithmic-competitive algorithm for the k-server problem. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 267–276. IEEE, 2011.

[21] Ahron Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. Siam, 2001.

[22] Dirk Bergemann and Juuso Välimäki. Bandit Problems. In Steven Durlauf and Larry Blume, editors, *The New Palgrave Dictionary of Economics, 2nd ed.* Macmillan Press, 2006.

[23] Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57:1407–1420, 2009.

[24] Omar Besbes and Assaf J. Zeevi. Blind network revenue management. *Operations Research*, 60(6):1537–1550, 2012.

[25] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1), 2012.

[26] Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *49th ACM Symp. on Theory of Computing (STOC)*, pages 72–85. ACM, 2017.

[27] Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: stochastic and adversarial bandits. In *25th Conf. on Learning Theory (COLT)*, 2012.

[28] Niv Buchbinder and Joseph Seffi Naor. The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science*, 3(2–3):93–263, 2009.

[29] Niv Buchbinder and Joseph (Seffi) Naor. Online primal-dual algorithms for covering and packing. *Math. Oper. Res.*, 34(2):270–286, May 2009.

[30] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge Univ. Press, 2006.

[31] Moses Charikar and Balaji Raghavachari. The finite capacity dial-a-ride problem. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*, pages 458–467. IEEE, 1998.

[32] Tianyi Chen and Georgios B Giannakis. Bandit convex optimization for scalable and dynamic iot management. *IEEE Internet of Things Journal*, 2018.

[33] Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.

[34] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *43rd ACM Symp. on Theory of Computing (STOC)*, pages 273–282. ACM, 2011.

[35] Richard Combes, Chong Jiang, and Rayadurgam Srikant. Bandits with budgets: Regret lower bounds and optimal algorithms. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):245–257, 2015.

[36] Nikhil R. Devanur and Thomas P. Hayes. The AdWords problem: Online keyword matching with budgeted bidders under random permutations. In *10th ACM Conf. on Electronic Commerce (EC)*, pages 71–78, 2009.

[37] Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *12th ACM Conf.*

on Electronic Commerce (EC), pages 29–38, 2011.

[38] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. Multi-armed bandit with budget constraint and variable costs. In *27th AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[39] Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal leanring for contextual bandits. In *27th Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2011.

[40] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Clifford Stein. Online stochastic packing applied to display ad allocation. In *18th Annual European Symp. on Algorithms (ESA)*, pages 182–194, 2010.

[41] Amos Fiat, Richard M Karp, Michael Luby, Lyle A McGeoch, Daniel D Sleator, and Neal E Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, 1991.

[42] Abraham Flaxman, Adam Kalai, and H. Brendan McMahan. Online Convex Optimization in the Bandit Setting: Gradient Descent without a Gradient. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 385–394, 2005.

[43] Yoav Freund and Robert E Schapire. Game theory, online prediction and boosting. In *9th Conf. on Learning Theory (COLT)*, pages 325–332, 1996.

[44] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

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

[46] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, 2011.

[47] Sudipta Guha and Kamesh Munagala. Multi-armed Bandits with Metric Switching Costs. In *36th Intl. Colloquium on Automata, Languages and Programming (ICALP)*, pages 496–507, 2007.

[48] Anupam Gupta, Ravishankar Krishnaswamy, Marco Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits. In *52nd IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 827–836, 2011.

[49] András György, Levente Kocsis, Ivett Szabó, and Csaba Szepesvári. Continuous time associative bandit problems. In *20th Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 830–835, 2007.

[50] András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. The on-line shortest path problem under partial monitoring. *J. of Machine Learning Research (JMLR)*, 8:2369–2403, 2007.

[51] Elad Hazan. Introduction to Online Convex Optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2015.

[52] Justin Hsu, Zhiyi Huang, Aaron Roth, and Zhiwei Steven Wu. Jointly private convex programming. In *27th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 580–599, 2016.

[53] Nicole Immorlica, Karthik Abinav Sankararaman, Robert Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks, 2018. Working paper. Available at `https://arxiv.org/abs/1811.11881`.

[54] David S Johnson. Approximation algorithms for combinatorial problems. *Journal of computer and system sciences*, 9(3):256–278, 1974.

[55] Satyen Kale, Lev Reyzin, and Robert E. Schapire. Non-stochastic bandit slate problems. In *24th Advances in Neural Information Processing Systems (NIPS)*, pages 1054–1062, 2010.

[56] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *35th Intl. Conf. on Machine Learning (ICML)*, pages 2564–2572, 2018.

[57] Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004.

[58] John Langford and Tong Zhang. The Epoch-Greedy Algorithm for Contextual Multi-armed Bandits. In *21st Advances in Neural Information Processing Systems (NIPS)*, 2007.

[59] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–260, 1994.

[60] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete mathematics*, 13(4):383–390, 1975.

[61] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes-Leme. Stochastic bandits robust to adversarial corruptions. In *50th ACM Symp. on Theory of Computing (STOC)*, 2018.

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

[63] Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. Stochastic convex optimization with multiple objectives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1115–1123, 2013.

[64] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8 (4):265–368, 2013.

[65] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online match-

ing. *J. ACM*, 54(5):22, 2007.

[66] Marco Molinaro and R. Ravi. Geometry of online packing linear programs. In *39th Intl. Colloquium on Automata, Languages and Programming (ICALP)*, pages 701–713, 2012.

[67] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, Cambridge, 1995.

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

[69] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *26th Conf. on Learning Theory (COLT)*, pages 993–1019, 2013.

[70] Anshuka Rangi, Massimo Franceschetti, and Long Tran-Thanh. Unifying the stochastic and the adversarial bandits with knapsack. In *28th Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 3311–3317, 2019.

[71] Adrian Rivera, He Wang, and Huan Xu. Online saddle point problem with applications to constrained online convex optimization. *arXiv preprint arXiv:1806.08301*, 2018.

[72] Ryan Rogers, Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Inducing approximately optimal flow using truthful mediators. In *16th ACM Conf. on Electronic Commerce (EC)*, pages 471–488, 2015.

[73] Aaron Roth, Aleksandrs Slivkins, Jonathan Ullman, and Zhiwei Steven Wu. Multidimensional dynamic pricing for welfare maximization. In *18th ACM Conf. on Electronic Commerce (EC)*, pages 519–536, 2017.

[74] Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Watch and learn: Optimizing from revealed preferences feedback. In *48th ACM Symp. on Theory of Computing (STOC)*, pages 949–962, 2016.

[75] Karthik Abinav Sankararaman and Aleksandrs Slivkins. Combinatorial semi-bandits with knapsacks. In *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, pages 1760–1770, 2018.

[76] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[77] Yevgeny Seldin and Gábor Lugosi. An improved parametrization and analysis of the EXP3++ algorithm for stochastic and adversarial bandits. In *30th Conf. on Learning Theory (COLT)*, 2017.

[78] Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *31th Intl. Conf. on Machine Learning (ICML)*, 2014.

[79] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *22nd Intl. World Wide Web Conf. (WWW)*, pages 1167–1178, 2013.

[80] Aleksandrs Slivkins. Dynamic ad allocation: Bandits with budgets. A technical report on `arxiv.org/abs/1306.0155`, June 2013.

[81] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E. Schapire. Fast convergence of regularized learning in games. In *28th Advances in Neural Information Processing Systems (NIPS)*, pages 2989–2997, 2015.

[82] Long Tran-Thanh, Archie Chapman, Enrique Munoz de Cote, Alex Rogers, and Nicholas R. Jennings. ϵ-first policies for budget-limited multi-armed bandits. In *24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1211–1216, 2010.

[83] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1134–1140, 2012.

[84] Seeun Umboh. Online network design algorithms via hierarchical decompositions. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1373–1387. Society for Industrial and Applied Mathematics, 2015.

[85] Jun-Kun Wang and Jacob D. Abernethy. Acceleration through optimistic no-regret dynamics. In *31st Advances in Neural Information Processing Systems (NIPS)*, pages 3828–3838, 2018.

[86] Zizhuo Wang, Shiming Deng, and Yinyu Ye. Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331, 2014.

[87] Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. In *31st Conf. on Learning Theory (COLT)*, 2018.

[88] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

APPENDIX

Our exposition in the body of the paper relies on some tools that are either known or can easily be derived using standard techniques. We state (and sometimes derive) these tools in this appendix.

A. Concentration Inequalities

Lemma A.1 (Azuma-Hoeffding inequality [67]). *Let $Y_1, Y_2, \ldots, Y_T$ be a martingale difference sequence (*i.e., *$\mathbb{E}[Y_t \mid Y_1, Y_2, \ldots, Y_{t-1}] = 0$). Suppose $|Y_t| \leq c$ for all $t \in \{1, 2, \ldots, T\}$. Let $R_{0,\delta}(T) := \sqrt{2Tc^2 \ln(1/\delta)}$. Then for every $\delta > 0$,*

$$\Pr\left[\sum_{t\in[T]} Y_t > R_{0,\delta}(T)\right] \leq \delta.$$

Lemma A.2 (Chernoff-Hoeffding bounds [67]). *Let $X_1, X_2, \ldots, X_T$ be a sequence of independent random variables such that $|X_t| \leq c$ for all $t \in \{1, 2, \ldots, T\}$. Let $Z_t := \mathbb{E}[X_t]$. Then for every $\delta > 0$,*

$$\Pr\left[\left|\sum_{t\in[T]} X_t - Z_t\right| > 3\sqrt{\left(\sum_{t\in[T]} Z_t\right) c^2 \ln(1/\delta)}\right] \leq \delta.$$

B. Lagrangians: proof of Lemma IV.2

Assume one of the resources is the dummy resource, and one of the arms is the null arm. Consider the linear program $\texttt{LP}_{\boldsymbol{M},B,T}$, for some outcome matrix $\boldsymbol{M}$. Let $\mathcal{L} = \mathcal{L}_{\boldsymbol{M},B,T}$ denote the Lagrange function.

Lemma A.3 (Lemma IV.2, restated). *Suppose $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$ is a mixed Nash equilibrium for the Lagrangian game. Then $\boldsymbol{X}^*$ is an optimal solution for the linear program (IV.1). Moreover, the minimax value of the Lagrangian game equals the LP value: $\mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*) = \texttt{OPT}_{\texttt{LP}}$.*

In what follows we prove Lemma A.3. Writing out the definition of the mixed Nash equilibrium,

$$\mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}) \geq \mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*) \geq \mathcal{L}(\boldsymbol{X}, \boldsymbol{\lambda}^*) \qquad \forall \boldsymbol{X} \in \Delta_K, \boldsymbol{\lambda} \in \Delta_d. \tag{A.1}$$

For brevity, denote $r(\boldsymbol{X}^*) = \sum_{a\in[K]} \boldsymbol{X}^*(a)\ r(a)$ and $c_i(\boldsymbol{X}^*) = \sum_{a\in[K]} \boldsymbol{X}^*(a)\ c_i(a)$.

We first state and prove the complementary slackness condition for the Nash equilibrium.

Claim A.4. *For every resource $i \in [d]$ we have,*

(a) $1 - \frac{T}{B}\, c_i(\boldsymbol{X}^*) \geq 0$,

(b) $\lambda_i^* > 0 \implies 1 - \frac{T}{B}\, c_i(\boldsymbol{X}^*) = 0$.

Proof: **Part (a).** For contradiction, consider resource i that minimizes the left-hand side in (a), and assume that the said left-hand side is strictly negative. We have two cases: either $\lambda_i^* < 1$ or $\lambda_i^* = 1$. When $\lambda_i^* < 1$, consider another distribution $\tilde{\boldsymbol{\lambda}} \in \Delta_d$ such that $\tilde{\lambda}_i = 1$ and $\tilde{\lambda}_{i'} = 0$ for every $i' \neq i$. Note that we have, $\mathcal{L}(\boldsymbol{X}^*, \tilde{\boldsymbol{\lambda}}) < \mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$. This contradicts the first inequality in (A.1).

Consider the second case, when $\lambda_i^* = 1$. Then $\mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*) = r(\boldsymbol{X}^*) + 1 - \frac{T}{B} c_i(\boldsymbol{X}^*)$. Consider any arm $a \in [K]$ such that $X^*(a) \neq 0$. Let $\tilde{\boldsymbol{X}} \in \Delta_K$ be another distribution such that $\tilde{X}(a) := 0$ and $\tilde{X}(\text{null}) := X^*(\text{null}) + X^*(a)$ and $\tilde{X}(a') = X^*(a')$ for every $a' \notin \{a, \text{null}\}$. Note that $\tilde{X}(\text{null}) \leq 1$. Since $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$ is a Nash equilibrium, we have that $\mathcal{L}(\tilde{\boldsymbol{X}}, \boldsymbol{\lambda}^*) \leq \mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$. This implies that $-X^*(a)r(a) + X^*(a)\frac{T}{B} c_i(a) \leq 0$. Re-arranging we obtain, $\frac{T}{B} c_i(a) \leq r(a) \leq 1$. Thus, we have $1 - \frac{T}{B} c_i(a) \geq 0$.

Since this holds for every $a \in [K]$ with $X^*(a) \neq 0$, we obtain a contradiction:

$$1 - \tfrac{T}{B}\, c_i(\boldsymbol{X}^*) = \textstyle\sum_{a\in[K]} X^*(a) \left(1 - \frac{T}{B} c_i(a)\right) \geq 0.$$

Part (b). For contradiction, assume the statement is false for some resource i. Then, by part (a), $\lambda_i^* > 0$ and $1 - \frac{T}{B} c_i(\boldsymbol{X}^*) > 0$, and consequently $\mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*) > r(\boldsymbol{X}^*)$. Now, consider distribution $\tilde{\boldsymbol{\lambda}}$ which puts probability 1 on the dummy resource. We then have $\mathcal{L}(\boldsymbol{X}^*, \tilde{\boldsymbol{\lambda}}) = r(\boldsymbol{X}^*) < \mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$, contradicting the first inequality in Eq. (A.1). ∎

Let $\tilde{\boldsymbol{X}}$ be some feasible solution for the linear program (IV.1). Plugging the feasibility constraints into the definition of the Lagrangian function, $\mathcal{L}(\tilde{\boldsymbol{X}}, \boldsymbol{\lambda}^*) \geq r(\tilde{\boldsymbol{X}})$. Claim A.4(a) implies that $\boldsymbol{X}^*$ is a feasible solution to the linear program (IV.1). Claim A.4(b) implies that $\mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*) = r(\boldsymbol{X}^*)$. Thus,

$$r(\boldsymbol{X}^*) = \mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*) \geq \mathcal{L}(\tilde{\boldsymbol{X}}, \boldsymbol{\lambda}^*) \geq r(\tilde{\boldsymbol{X}}).$$

So, $\boldsymbol{X}^*$ is an optimal solution to the LP. In particular, $\texttt{OPT}_{\texttt{LP}} = r(\boldsymbol{X}^*) = \mathcal{L}(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$.

C. The stopped LP for Adversarial BwK: proof of Eq. (V.4)

The proof is similar to prior work [18, 37]. Denote $\mathcal{D}_\tau$ to be the set of all distributions over the arms such that for every $\boldsymbol{p} \in \mathcal{D}_\tau$ we have the following: for every $i \in [d]$ we have $\sum_{t\in[\tau]} \boldsymbol{p} \cdot \boldsymbol{c}_{t,i} \leq B$. In other words, $\mathcal{D}_\tau$ denotes the set of distributions whose expected stopping time is at least τ. Thus it immediately follows that $\texttt{OPT}_{\texttt{LP}}(\tau) \geq \max_{\boldsymbol{p}\in\mathcal{D}_\tau} \sum_{t\in[\tau]} \boldsymbol{p} \cdot \boldsymbol{r}_t$ since for any given $\boldsymbol{p} \in \mathcal{D}_\tau$ it is feasible to $\texttt{LP}(\tau)$. Thus $\texttt{OPT}_{\texttt{LP}}(\tau)$ is at least the value of any feasible solution $\boldsymbol{p} \in \mathcal{D}_\tau$. Note that for every fixed distribution $\boldsymbol{p} \in \Delta_K$, there exists a τ such that either $\boldsymbol{p} \in \mathcal{D}_\tau$ and $\boldsymbol{p} \notin \mathcal{D}_{\tau+1}$ or $\boldsymbol{p} \in \mathcal{D}_T$. Moreover the total expected reward we can obtain using $\boldsymbol{p}$ is $\sum_{t\in[\tau]} \boldsymbol{p} \cdot \boldsymbol{r}_t$. Thus $\max_{1\leq\tau\leq T} \texttt{OPT}_{\texttt{LP}}(\tau) \geq \texttt{OPT}_{\texttt{FD}}$.

D. Regret minimization in games: proof of Lemma III.1

Let us revisit adversarial online learning, as per Figure 1. Denote the benchmark in Eq. (III.2) as

$$\texttt{OPT}_{\texttt{AOL}}(T) := \max_{a\in A} \textstyle\sum_{t\in[T]} f_t(a).$$

Recall that $[b_{\min}, b_{\max}]$ is the payoff range, and denote $\sigma = b_{\max} - b_{\min}$.

Lemma A.5. *Suppose an algorithm for adversarial online learning satisfies (III.2) for some $\delta > 0$. Then*

$$\Pr\Big[\ \forall \tau \in [T]\ \mathit{OPT}_{\mathit{AOL}}(\tau)\ -\ \textstyle\sum_{t\in[\tau]} \boldsymbol{f}_t \cdot \boldsymbol{p}_t \\ \leq \sigma \cdot \Big(R_{\delta/T}(T) + \sqrt{2T\log(T/\delta)}\Big)\ \Big] \geq 1 - 2\delta. \quad \text{(A.2)}$$

Proof: Let us use the stronger regret bound (III.3) implied by (III.2). Note that

$$\mathbb{E}[f_t(a_t) \mid a_1, a_2\,, \ \ldots\ , a_{t-1}] = \boldsymbol{f}_t \cdot \boldsymbol{p}_t.$$

Applying the Azuma-Hoeffding inequality for each $\tau \in [T]$, and taking a union bound, we have

$$\Pr\Big[\ \forall \tau \in [T] \quad \textstyle\sum_{t\in[\tau]} f_t(a_t) - \sum_{t\in[\tau]} \boldsymbol{f}_t \cdot \boldsymbol{p}_t \\ \leq \sigma \cdot \sqrt{2T\log(T/\delta)}\ \Big] \geq 1 - \delta. \quad \text{(A.3)}$$

Taking a union bound over Eq. (A.3) and Eq. (III.3) and adding the equations we get Eq. (A.2). ∎

Remark A.6. *For Hedge algorithm, regret bound Eq. (A.2) is already proved in [44].*

Let $W = \sqrt{2T\log(T/\delta)}$ denote the term from Lemma A.5 in what follows.

We now prove Lemma III.1, similar to the proof in [43] for the deterministic game. Recall that we take averages up to some fixed round $\tau \in [T]$. We prove that the following two inequalities hold, each with probability at least $1 - \delta$.

$$\frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,1}^{\mathrm{T}}\, \boldsymbol{G}_t\, \boldsymbol{p}_{t,2} \geq v^* - \sigma \cdot \frac{R_{1,\,\delta/T}(T) + 2W}{\tau}. \quad \text{(A.4)}$$

$$\frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,1}^{\mathrm{T}}\, \boldsymbol{G}_t\, \boldsymbol{p}_{t,2} \leq \overline{\boldsymbol{p}}_1^{\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_2 \\ + \sigma \cdot \frac{R_{2,\,\delta/T}(T) + 2W}{\tau} \qquad \forall \boldsymbol{p}_2 \in \Delta_{A_2}. \quad \text{(A.5)}$$

Eq. (III.5) in Lemma III.1 follows by adding Eq. (A.4) and Eq. (A.5).

First we prove Eq. (A.4). Following the set of inequalities in Section 2.5 of [43] we have the following. From Lemma A.5 we have,

$$\frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,1}^{\mathrm{T}} \boldsymbol{G}_t \boldsymbol{p}_{t,2} \\ \geq_{whp} \frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_1^{*\mathrm{T}}\, \boldsymbol{G}_t\, \boldsymbol{p}_{t,2} - \sigma \cdot \frac{R_{1,\,\delta/T}(T) + W}{\tau}$$

From Lemma A.1 this can be lower-bounded by,

$$\geq_{whp} \frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_1^{*\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_{t,2} - \sigma \cdot \frac{R_{1,\,\delta/T}(T) + 2W}{\tau}$$

From Definition of $\boldsymbol{p}_1^*$, this equals,

$$= \max_{\boldsymbol{p}_1\in\Delta_{A_1}} \frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_1{}^{\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_{t,2} - \sigma \cdot \frac{R_{1,\,\delta/T}(T) + 2W}{\tau}$$

From Definition of $\overline{\boldsymbol{p}}_2$, this equals,

$$= \max_{\boldsymbol{p}_1\in\Delta_{A_1}} \boldsymbol{p}_1{}^{\mathrm{T}}\, \boldsymbol{G}\, \overline{\boldsymbol{p}}_2 - \sigma \cdot \frac{R_{1,\,\delta/T}(T) + 2W}{\tau}$$

$$\geq \min_{\boldsymbol{p}_2\in\Delta_{A_2}} \max_{\boldsymbol{p}_1\in\Delta_{A_1}} \boldsymbol{p}_1^{\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_2 - \sigma \cdot \frac{R_{1,\,\delta/T}(T) + 2W}{\tau}$$

Here $\leq_{whp}$ denotes statements that hold with probability at least $1 - \delta$.

Now let us prove (A.5). Fix distribution $\boldsymbol{p}_2 \in \Delta_{A_2}$. Then from Lemma A.5 we have,

$$\frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,1}^{\mathrm{T}}\, \boldsymbol{G}_t\, \boldsymbol{p}_{t,2} \\ \leq_{whp} \frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,1}{}^{\mathrm{T}}\, \boldsymbol{G}_t\, \boldsymbol{p}_2 + \sigma \cdot \frac{R_{2,\,\delta/T}(T) + W}{\tau}$$

From Lemma A.1 this is upper-bounded by

$$\leq_{whp} \frac{1}{\tau}\sum_{t\in[\tau]} \boldsymbol{p}_{t,1}{}^{\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_2 + \sigma \cdot \frac{R_{2,\,\delta/T}(T) + 2W}{\tau}$$

From Definition of $\overline{\boldsymbol{p}_1}$, this equals

$$= \overline{\boldsymbol{p}_1}^{\mathrm{T}}\, \boldsymbol{G}\, \boldsymbol{p}_2 + \sigma \cdot \frac{R_{2,\,\delta/T}(T) + 2W}{\tau}$$

Taking a union bound over all the four high-probability inequalities, we get the lemma.