Contextual Bandits with Packing and Covering Constraints: A Modular Lagrangian Approach via Regression

Aleksandrs Slivkins SLIVKINS@MICROSOFT.COM
Microsoft Research NYC, New York, NY, USA.

Karthik Abinav Sankararaman KARTHIKABINAVS@GMAIL.COM
Meta

Dylan J. Foster DYLANFOSTER@MICROSOFT.COM
Microsoft Research NYC, New York, NY, USA.

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

We consider *contextual bandits with linear constraints* (CBwLC), a variant of contextual bandits in which the algorithm consumes multiple resources subject to linear constraints on total consumption. This problem generalizes *contextual bandits with knapsacks* (CBwK), allowing for packing and covering constraints, as well as positive and negative resource consumption. We provide the first algorithm for CBwLC (or CBwK) that is based on regression oracles. The algorithm is simple, computationally efficient, and admits vanishing regret. It is statistically optimal for the variant of CBwK in which the algorithm must stop once some constraint is violated. Further, we provide the first vanishing-regret guarantees for CBwLC (or CBwK) that extend beyond the stochastic environment. We side-step strong impossibility results from prior work by identifying a weaker (and, arguably, fairer) benchmark to compare against. Our algorithm builds on LagrangeBwK (Immorlica et al., 2019, 2022), a Lagrangian-based technique for CBwK, and SquareCB (Foster and Rakhlin, 2020), a regression-based technique for contextual bandits. Our analysis leverages the inherent modularity of both techniques.

1. Introduction

We consider a problem called *contextual bandits with linear constraints* (CBwLC). In this problem, an algorithm chooses from a fixed set of K arms and consumes $d \geq 1$ constrained *resources*. In each round t, the algorithm observes a context x_t, chooses an arm a_t, receives a reward $r_t \in [0, 1]$, and also consumes some bounded amount of each resource. (So, the outcome of choosing an arm is a $(d + 1)$-dimensional vector.) The consumption of a given resource could also be negative, corresponding to replenishment thereof. The algorithm proceeds for T rounds, and faces a constraint on the total consumption of each resource i: either a *packing constraint* ("at most B_i") or a *covering constraint* ("at least B_i"). We focus on the stochastic environment, wherein the context and the arms' outcome vectors are drawn i.i.d. in each round. On a high level, the challenge is to simultaneously handle bandits with contexts and resource constraints.

CBwLC subsumes two well-studied bandit problems: *contextual bandits*, the special case with no resources, and *bandits with knapsacks* (BwK), a simpler special case with no contexts. Specifically, BwK is a special case with no contexts, only packing constraints, non-negative resource consumption, and a *null arm* that allows one to skip a round.[1] While *contextual* bandits with knapsacks (CBwK) have been explored in prior work, our results for this special case are new.

1. Most prior work on BwK assumes *hard-stopping*: the algorithm must stop (or, alternatively, permanently switch to the null arm) as soon as one of the constraints is violated. Note that hard-stopping may not be feasible without the null arm, and is not meaningful when one has covering constraints (since they are usually not satisfied initially). Further background and references on BwK can be found in Section 1.1.

We adopt an approach to contextual bandits which assumes access to a subroutine for solving certain supervised regression problems, also known as a *regression oracle* (Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2020). This approach is computationally efficient, allows for strong provable guarantees, and tends to be superior in experiments compared to other approaches such as ones based on classification oracles (*e.g.,* Langford and Zhang, 2007; Agarwal et al., 2014).[2]

We design the first algorithm for `CBwLC` with regression oracles (in fact, this constitutes the first such algorithm for `CBwK`). To handle contexts, we build on the `SquareCB` algorithm from Foster and Rakhlin (2020): a randomization technique that converts actions' estimated rewards into a distribution over actions that optimally balances exploration and exploitation. To handle resource constraints, we build on the `LagrangeBwK` algorithm of Immorlica, Sankararaman, Schapire, and Slivkins (2019, 2022). This algorithm solves the simpler problem of `BwK` via a repeated zero-sum game between the "primal" algorithm which chooses arms and the "dual" algorithm which chooses resources, with game payoffs given by a natural Lagriangian relaxation.

We make three technical contributions. First, we develop `LagrangeCBwLC`, an extension of the `LagrangeBwK` framework from `BwK` with hard-stopping to `CBwLC`. Bounding constraint violations without stopping the algorithm necessitates a subtle change in the algorithm (a re-weighting of the Lagrangian payoffs) and some new tricks in the analysis. Second, we design a new "primal" algorithm for this framework, which uses the `SquareCB` method to explore in an oracle-efficient fashion. Third, we extend our guarantees beyond stochastic environments, allowing for a bounded number of "switches" from one stochastic environment to another (henceforth, the *switching environment*).

We measure performance in terms of 1) regret relative to the best algorithm, and 2) maximum violation of each constraint at time T. We bound the maximum of these quantities, henceforth called *outcome-regret*. Our main result attains outcome-regret $\widetilde{\mathcal{O}}(T^{3/4})$ for general `CBwLC` problems. We emphasize that it is the first regret bound for `CBwLC` with regression oracles; whether this regret bound can be improved is an open question. Moreover, we obtain the (optimal) $\widetilde{\mathcal{O}}(\sqrt{T})$ regret bound whenever there exists an optimal distribution that satisfies all constraints by a known constant margin. We also obtain $\widetilde{\mathcal{O}}(\sqrt{T})$ regret for contextual `BwK`.

Our presentation leverages the inherent *modularity* of the techniques. In particular, the `LagrangeBwK` reduction permits the use of any application-specific primal algorithm with a particular regret guarantee,[3] and the `SquareCB` method gives such a primal algorithm whenever one has access to a regression oracle with a particular guarantee on the squared regression error. We incorporate their respective analyses as specific theorems, and re-use much of the technical setup. We provide two new "theoretical interfaces" for `LagrangeCBwLC` that our `SquareCB`-based primal algorithm can plug into, as well as any primal algorithm from prior work on `LagrangeBwK`. We re-use this machinery for the analysis of the switching environment. A key conceptual contribution here is to identify the pieces and how they connect to one another.

The `LagrangeCBwLC` reduction is of independent interest for even for the simpler problem of `CBwLC` without contexts (henceforth, `BwLC`). This is due to two extensions which appear new even without contexts: to the switching environment (defined above), and to convex optimization (where rewards and resource

2. Regression oracles return *regression function* that map context-arm pairs to real values, and aim to approximate the reward/loss of a given context-action pair. A *classification oracle* solves a different problem: it is a subroutine for computing an optimal policy (mapping from contexts to arms) within a given class of policies. More background on contextual bandits with either oracle can be found in Section 1.1.

3. Immorlica et al. (2019, 2022) and Castiglioni et al. (2022) use this modularity to derive extensions to `BwK` with *e.g.,* full feedback, combinatorial semi-bandits, bandit convex optimization, repeated Stackelberg games, or budget-constrained bidding in first-price auctions.

consumption are convex/concave functions of an arm). However, the $T^{3/4}$ scaling of outcome-regret is suboptimal for `BwLC` in the stochastic environment with K arms (see Section 1.1).

Our result for the switching environment is new even for `BwK`, *i.e.,* when one only has packing constraints. This result builds on our analysis for `LagrangeCBwLC`: crucially, the algorithm continues till round T. Prior analyses of `LagrangeBwK` with hard-stopping do not appear to suffice. We obtain regret bounds relative to a non-standard, yet well-motivated benchmark, bypassing strong impossibility results from prior work on Adversarial `BwK` (see Section 1.1 for background).

1.1. Additional background and related work

Contextual bandits and `BwK` generalize (stochastic) multi-armed bandits, *i.e.,* the special case without contexts or resource constraints. Further background on bandit algorithms can be found in books (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2020; Slivkins, 2019).

Contextual bandits (`CB`). While various versions of the contextual bandit problem have been studied over the past three decades, most relevant are the approaches based on computational oracles (see Footnote 2). We focus on `CB` with regression oracles, a promising emerging paradigm (Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2020). `CB` with classification oracles is an earlier approach, studied in Langford and Zhang (2007) and follow-up work, *e.g.,* Dudík et al. (2011); Agarwal et al. (2014).

Contextual bandits with regression oracles are practical to implement, and can leverage the fact that regression algorithms are common in practice. In addition, `CB` with regression oracles tend to have superior statistical performance compared to `CB` with classification oracles, as reported in extensive real-data experiments (Foster et al., 2018, 2021b; Bietti et al., 2021).

`CB` with regression oracles are desirable from a theoretical perspective, as they admit *unconditionally* efficient algorithms for various standard function classes under realizability.[4] In contrast, statistically optimal guarantees for `CB` with classification oracles are only computationally efficient conditionally. Specifically, one needs to assume that the oracle is an exact optimizer for all possible datasets, even though this is typically an NP-hard problem. This assumption is needed even if the CB algorithm is run on an instance that satisfies realizability.

Linear `CB` (Li et al., 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011), a well-studied special case of the regression-based approach to `CB`, posits realizability for linear regression functions. Analyses tend to focus on the high-confidence region around regression-based estimates. This variant is less relevant to our paper.

Bandits with Knapsacks (`BwK`) are more challenging compared to stochastic bandits for two reasons. First, instead of *per-round* expected reward one needs to think about the *total* expected reward over the entire time horizon, taking into account the resource consumption. Moreover, instead of the best arm one is interested in the best fixed *distribution* over arms, which can perform much better. Both challenges arise in the "basic" special case when one has only two arms and only one resource other than the time itself.

The `BwK` problem was introduced and optimally solved in Badanidiyuru et al. (2013, 2018), achieving $\widetilde{O}(\sqrt{KT})$ regret for K arms when budgets are $B_i = \Omega(T)$. Agrawal and Devanur (2014, 2019) and Immorlica et al. (2019, 2022) provide alternative regret-optimal algorithms. In particular, the algorithm in Agrawal and Devanur (2014, 2019), which we refer to as `UcbBwK`, implements the paradigm of *optimism in the face of uncertainty*. Most work on `BwK` posits hard-stopping (see Footnote 1). A detailed survey of `BwK` and its extensions can be found in Slivkins (Ch.11, 2019).

4. *I.e.,* assuming that a given class of regression functions contains one that correctly describes the problem instance.

The contextual version of BwK (CBwK) was first studied in Badanidiyuru et al. (2014). They consider CBwK with classification oracles, and obtain an algorithm that is regret-optimal but not computationally efficient. Agrawal et al. (2016) provide a regret-optimal and oracle-efficient algorithm for the same problem, which combines UcbBwK and with the oracle-efficient contextual bandit method Agarwal et al. (2014). Agrawal and Devanur (2016) provide a regression-based approach for the special case of linear CBwK, combining UcbBwK and the optimistic approach for linear contextual bandits (Li et al., 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011). Other regression-based methods for contextual BwK have not been studied.

Many special cases of CBwK have been studied for their own sake, most notably dynamic pricing (*e.g.,* Besbes and Zeevi, 2009; Babaioff et al., 2015; Wang et al., 2014) and online bidding under budget (*e.g.,* Balseiro and Gur, 2019; Balseiro et al., 2022; Gaitonde et al., 2023). For the latter, Gaitonde et al. (2023) achieve vanishing regret against a benchmark similar to ours.

CBwLC beyond (contextual) BwK. Agrawal and Devanur (2014, 2019) solve CBwLC without contexts (BwLC), building on UcbBwK and achieving regret $\widetilde{\mathcal{O}}(\sqrt{KT})$. In fact, their result extends to arbitrary convex constraints and (in Agrawal et al., 2016) to CBwK with classification oracles. However, their technique does not appear to connect well with regression oracles.

A notable special case involving covering constraints is online bidding under return-on-investment constraint (*e.g.,* Balseiro et al., 2022; Golrezaei et al., 2021b,a).

The version of BwK that allows negative resource consumption has not been widely studied. A very recent algorithm in Kumar and Kleinberg (2022) admits a regret bound that depends on several instance-dependent parameters, but no worst-case regret bound is provided.

Adversarial BwK. The adversarial version of BwK, introduced in Immorlica et al. (2019, 2022), is even more challenging compared to the stochastic version due to the *spend-or-save dilemma*: essentially, the algorithm does not know whether to spend its budget now or to save it for the future. The algorithms are doomed to approximation ratios against standard benchmarks, as opposed to vanishing regret, even for a switching environment with just a single switch (Immorlica et al., 2022). The approximation-ratio version is by now well-understood (Immorlica et al., 2019, 2022; Kesselheim and Singla, 2020; Castiglioni et al., 2022; Fikioris and Tardos, 2023).[5] Interestingly, all algorithms in these papers build on versions of LagrangeBwK. On the other hand, obtaining vanishing regret against some reasonable-but-weaker benchmark (such as ours) is articulated as a major open question (Immorlica et al., 2022). We are not aware of any such results in prior work.

Large vs. small budgets. Our guarantees are most meaningful in the regime of "large budgets", where $B := \min_{i\in[d]} B_i > \Omega(T)$. This is the main regime of interest in all prior work on BwK and its special cases.[6] That said, our guarantees are non-trivial even if $B = o(T)$.

The small-budget regime, $B = o(T)$, has been studied since Babaioff et al. (2015). In particular, (Badanidiyuru et al., 2013, 2018) derive optimal upper/lower regret bounds in this regime for BwK with hard-stopping. The respective *lower* bounds are specific to hard-stopping, and do not directly apply when a BwK algorithm can continue till round T.

Concurrent work. Han et al. (2023) focus on CBwK with hard-stopping in the stochastic environment and obtain a result similar to Theorem 15(c), also using an algorithm based on LagrangeBwK and SquareCB. The main technical difference is that they do not explicitly express their algorithm as an instantiation of

5. Fikioris and Tardos (2023) is concurrent and independent work with respect to ours.
6. In particular, $B > \Omega(T)$ is explicitly assumed in, *e.g.,* Besbes and Zeevi (2009); Wang et al. (2014); Balseiro and Gur (2019); Immorlica et al. (2022); Castiglioni et al. (2022); Gaitonde et al. (2023).

LagrangeBwK, and accordingly do not take advantage of its modularity. Their treatment does not extend to the full generality of CBwLC, and does not address the switching environment. We emphasize that our results are simultaneous and independent with respect to theirs.

Concurrent work of Liu et al. (2022) achieves a vanishing-regret result for Adversarial BwK. This result is incomparable to our results for switching environments. Their regret bound is parameterized by (known) pathlength and (unknown) total variation, and holds against the standard benchmark (the optimal dynamic policy), whereas ours is parameterized by the (unknown) number of switches, and holds against a non-standard benchmark. Their algorithm is a version of UcbBwK with sliding-window estimators.

1.2. Organization

Section 2 introduces the CBwLC problem. Section 3 provide our Lagrangian framework (LagrangeCBwLC) and the associated modular guarantees. The material specific to regression oracles is encapsulated in Section 4, including the setup and the SquareCB-based primal algorithm. Due to the page limit, some material is moved to the appendices. Particularly, Appendix A extends our results to the switching environment, defining a novel benchmark and building on the machinery from the previous sections. Appendix B contains the analysis for Section 3. Appendix C applies our machinery to bandit convex optimization.

2. Model and preliminaries

Contextual Bandits with Linear Constraints (CBwLC). There are $K \geq 2$ arms, $T \geq 2$ rounds, and $d \geq 1$ *resources*. We use $[K]$, $[T]$, and $[d]$ to denote, respectively, the sets of all arms, rounds, and resources.[7] In each round $t \in [T]$, an algorithm observes a context $x_t \in \mathcal{X}$ from a set $\mathcal{X}$ of possible contexts, chooses an arm $a_t \in [K]$, receives a reward $r_t \in [0,1]$, and consumes some amount $c_{t,i} \in [-1,+1]$ of each resource i. Consumptions are observed by the algorithm, so that the outcome of choosing an arm is the *outcome vector* $\boldsymbol{o}_t = (r_t; c_{t,1}\,, \dots, c_{t,d}) \in [0,1] \times [-1,+1]^d$. For each resource $i \in [d]$ we are required to (approximately) satisfy the constraint

$$V_i(T) := \sigma_i \left(\textstyle\sum_{t\in[T]} c_{t,i} - B_i \right) \leq 0, \tag{2.1}$$

where $B_i \in [0,T]$ is the budget and $\sigma_i \in \{-1,+1\}$ is the *constraint sign*. Here $\sigma_i = 1$ (resp., $\sigma_i = -1$) corresponds to a packing (resp., covering) constraint, which requires that the total consumption never exceeds (resp., never falls below) B_i. Informally, the goal is to minimize both regret (on the total reward) and the constraint violations $V_i(T)$. [8]

We define counterfactual outcomes as follows. The *outcome matrix* $\mathbf{M}_t \in [-1,1]^{K\times(d+1)}$ is chosen in each round $t \in [T]$, so that its rows $\mathbf{M}_t(a)$ correspond to arms $a \in [K]$ and the outcome vector is defined as $\boldsymbol{o}_t = \mathbf{M}_t(a_t)$. Thus, the row $\mathbf{M}_t(a)$ represents the outcome the algorithm *would have* observed in round t if it has chosen arm a.

We focus on *Stochastic* CBwLC throughout the paper unless stated otherwise: in each round t, the pair $(x_t, \mathbf{M}_t)$ is drawn independently from some fixed distribution $\mathcal{D}^{\text{out}}$. In Appendix A we consider a generalization in which the distribution $\mathcal{D}^{\text{out}}$ can change over time.

The special case of CBwLC without contexts (equivalently, with only one possible context, $|\mathcal{X}| = 1$) is called *bandits with linear constraints* (BwLC). We also refer to it as the non-contextual problem.

7. Throughout, $[n]$, $n \in \mathbb{N}$ stands for the set $\{1, 2\,, \dots, n\}$.
8. A similar bi-objective approach is taken in Agrawal and Devanur (2014, 2019) and Agrawal et al. (2016).

Remark 1 *Rewards and resource consumptions can be mutually correlated. This is essential in most motivating examples of* `BwK`*,* e.g., *Badanidiyuru et al. (2018) and (Slivkins, 2019, Ch. 10).*

Remark 2 *We assume i.i.d. context arrivals. While many analyses in contextual bandits seamlessly carry over to adversarial chosen context arrivals, this is not the case for our problem.*[9]

Remark 3 `BwLC` *differs from Bandits with Knapsacks (*`BwK`*) in several ways. First,* `BwK` *only allows packing constraints ($\sigma_i \equiv 1$), whereas* `BwLC` *also allows covering constraints ($\sigma_i = -1$). Second, we allow resource consumption to be both positive and negative, whereas on* `BwK` *it must be non-negative. Third,* `BwK` *assumes that some arm in $[K]$ is a "null arm": an arm with zero reward and consumption of each resource,*[10] *whereas* `BwLC` *does not. Moreover, most prior work on* `BwK` *posits* hard-stopping*: the algorithm must stop — in our terms, permanently switch to the null arm — as soon as one of the constraints is violated.*

Let $B = \min_{i \in [d]} B_i$ be the smallest budget. Without loss of generality, we rescale the problem so that all budgets are B: we divide the per-round consumption of each resource i by B_i/B.

Without loss of generality, we assume that one of the resources is the *time resource*: it is deterministically consumed by each action at the rate of B/T, with a packing constraint ($\sigma_i = 1$).

Formally, an instance of `CBwLC` is specified by parameters T, B, K, d, constraint signs $\sigma_1, \ldots, \sigma_d$, and outcome distribution $\mathcal{D}^{\texttt{out}}$. Our benchmark is the best algorithm for a given problem instance:

$$\texttt{Opt} := \sup_{\text{algorithms } \texttt{ALG} \text{ with } \mathbb{E}[V_i(T)] \leq 0 \text{ for all resources } i} \mathbb{E}[\texttt{Rew}(\texttt{ALG})], \tag{2.2}$$

where $\texttt{Rew}(\texttt{ALG}) = \sum_{t \in [T]} r_t$ is the algorithm's total reward (we write `Rew` when the algorithm is clear from the context). The goal is to minimize *regret*, defined as $\texttt{Opt} - \texttt{Rew}(\texttt{ALG})$, as well as constraint violations $V_i(T)$. For most lucid results we upper-bound the maximum of these quantities, $\max_{i \in [d]} (\, \texttt{Opt} - \texttt{Rew}(\texttt{ALG}),\; V_i(T)\,)$, called the *outcome-regret*.

Notation. A *policy* is a deterministic mapping from contexts to arms. The set of all policies is denoted Π. Without loss of generality, we assume that the algorithm chooses a_t according some policy $\pi_t \in \Pi$ in each round t.

Consider a distribution D over policies. Suppose this distribution is "played" in some round t, *i.e.,* a policy π_t is drawn independently from D, and then an arm is chosen as $a_t = \pi_t(x_t)$. The expected reward and resource-i consumption for D are denoted $r(D) := \mathbb{E}_{\pi \sim D}\left[\, r_t(\pi(x_t))\,\right]$ and $c_i(D) := \mathbb{E}_{\pi \sim D}\left[\, c_{t,i}(\pi(x_t))\,\right]$. If D is played in all rounds $t \in [T]$, the expected constraint violation is denoted as $V_i(D) := \sigma_i\,(\, T \cdot c_i(D) - B\,)$.

Note that a policy can be interpreted as a singleton distribution that chooses this policy almost surely, and an arm can be interpreted as a policy that always chooses this arm. So, the notation $r(\cdot)$ and $c_i(\cdot)$ can be overloaded in the natural way for policies $\pi \in \Pi$, arms $a \in [K]$, and distributions over arms D. In particular, for the non-contextual problem we have $r(D) = \mathbb{E}[r(a)]$ and $c_i(D) = \mathbb{E}[c_i(a)]$, where D is a distribution over arms and the expectations are over arms a drawn from D (and the random outcomes).

Let Δ_S denote the set of all distributions over set S. We write $\Delta_n = \Delta_{[n]}$ for $n \in \mathbb{N}$ as a shorthand. We identify Δ_K (resp., Δ_d) with the set of all distributions over arms (resp., resources).

9. Indeed, with adversarial context arrivals algorithms cannot achieve sublinear regret, and instead are doomed to a constant approximation ratio. To see this, focus on `CBwK` and consider a version of the "spend or save" dilemma from Section 1.1. There are three types of contexts which always yield, resp., high, low, and medium rewards. The contexts are "medium" in the first $T/2$ rounds, and either all "high" or all "low" afterwards. The algorithm would not know whether to spend all its budget in the first half, or save it for the second half.
10. Existence of a "null arm" is equivalent to the algorithm being able to skip rounds.

For the round-t outcome matrix $\mathbf{M}_t$, the row for arm $a \in [K]$ is denoted $\mathbf{M}_t(a) = (r_t(a);\, c_{t,1}(a)\,,\, \dots\,,\, c_{t,d}(a))$, so that $r_t(a)$ is the reward and $c_{t,i}(a)$ is the consumption of each resource i if arm a is chosen.

Linear relaxation. We make use of a standard linear relaxation of `CBwLC`, which optimizes over distributions over policies, $D \in \Delta_\Pi$, maximizing the expected reward $r(D)$ subject to the constraints:

$$\begin{aligned} &\text{maximize} && r(D) \\ &\text{subject to} && D \in \Delta_\Pi \\ &&& V_i(D) := \sigma_i\,(\,T \cdot c_i(D) - B\,) \leq 0 \qquad \forall i \in [d]. \end{aligned} \tag{2.3}$$

The value of this linear program is denoted $\mathtt{Opt}_{\mathtt{LP}}$. It is easy to see that $T \cdot \mathtt{Opt}_{\mathtt{LP}} \geq \mathtt{Opt}$. [11] We make use of the value $\mathtt{Opt}_{\mathtt{LP}}$ throughout our analysis.

The Lagrange function associated with the linear program (2.3) is defined as follows:

$$\mathcal{L}_{\mathtt{LP}}(D,\lambda) := r(D) \;+\; \textstyle\sum_{i\in[d]}\; \sigma_i \cdot \lambda_i \left(1 - \tfrac{T}{B}\, c_i(D)\,\right), \quad D \in \Delta_\Pi,\ \lambda \in \mathbb{R}^d_+. \tag{2.4}$$

A standard result concerning *Lagrange duality* states that the maximin value of $\mathcal{L}_{\mathtt{LP}}$ coincides with $\mathtt{Opt}_{\mathtt{LP}}$. For this result, D ranges over all distributions over policies and λ ranges over all of $\mathbb{R}^d_+$:

$$\mathtt{Opt}_{\mathtt{LP}} = \sup_{D\in\Delta_\Pi}\, \inf_{\lambda\in\mathbb{R}^d_+} \mathcal{L}_{\mathtt{LP}}(D,\lambda). \tag{2.5}$$

3. Lagrangian framework for `CBwLC`

We provide a new algorithm design framework, `LagrangeCBwLC`, which generalizes the `LagrangeBwK` framework from Immorlica et al. (2019, 2022). We consider a repeated zero-sum game between two algorithms: a *primal algorithm* $\mathbf{Alg}_{\mathsf{Prim}}$ that chooses arms $a \in [K]$, and a *dual algorithm* $\mathbf{Alg}_{\mathsf{Dual}}$ that chooses distributions $\lambda \in \Delta_d$ over resources;[12] $\mathbf{Alg}_{\mathsf{Dual}}$ goes first, and $\mathbf{Alg}_{\mathsf{Prim}}$ can react to the chosen λ. The round-t payoff (*reward* for $\mathbf{Alg}_{\mathsf{Prim}}$, and *cost* for $\mathbf{Alg}_{\mathsf{Dual}}$) is defined as

$$\mathcal{L}_t(a,\lambda) = r_t(a) \;+\; \eta \cdot \textstyle\sum_{i\in[d]} \sigma_i \cdot \lambda_i \left(1 - \tfrac{T}{B}\, c_{t,i}(a)\,\right). \tag{3.1}$$

Here, $\eta \geq 1$ is a parameter specified later. For a distribution over policies, $D \in \Delta_\Pi$, denote $\mathcal{L}_t(D,\lambda) = \mathbb{E}_{\pi\sim D}\left[\,\mathcal{L}_t(\pi(x_t),\,\lambda)\,\right]$. The purpose of the definition Eq. (3.1) is to ensure that

$$\mathbb{E}\left[\,\mathcal{L}_t(D,\lambda)\,\right] = \mathcal{L}_{\mathtt{LP}}(D, \eta\cdot\lambda), \tag{3.2}$$

where the expectation is over the context x_t and the outcome $\boldsymbol{o}_t$. The repeated game is summarized in Algorithm 1.

Remark 4 *Beyond incorporating contexts, the main change compared to* `LagrangeBwK` *(Immorlica et al., 2019, 2022) is that we scale the constraint terms in the Lagrangian by the parameter $\eta \geq 1$. This parameter is the "lever" that allows us as to extend the algorithm from* `BwK` *to* `BwLC`*, accommodating general constraints. This modification effectively rescale the dual vectors from distributions $\lambda \in \Delta_d$ to vectors $\eta \cdot \lambda \in \mathbb{R}^d_+$. An equivalent reformulation of the algorithm could instead rescale all* rewards *to lie in the interval $[0, 1/\eta]$. This reformulation is instructive because the scale of rewards can be arbitrary as far as the original problem is concerned, but it leads to some notational difficulties in the analysis, which is why we*

11. Indeed, consider any algorithm in the supremum in Eq. (2.2). Let D_π be the expected fraction of rounds in which a given policy $\pi \in \Pi$ is chosen. Then distribution $D \in \Delta_\Pi$ satisfies the constraints in the LP.
12. The terms 'primal' and 'dual' here refer to the duality in linear programming. For the LP-relaxation (2.3), primal variables correspond to arms, and dual variables (*i.e.,* variables in the dual LP) correspond to resources.

Given : K arms, d resources, and ratio T/B, as per the problem definition;
parameter $\eta \geq 1$; algorithms $\mathbf{Alg}_{\mathsf{Prim}}$, $\mathbf{Alg}_{\mathsf{Dual}}$.
for *rounds* $t \in [T]$ **do**
Dual algorithm $\mathbf{Alg}_{\mathsf{Dual}}$ outputs a distribution $\lambda_t \in \Delta_d$ over resources.
Primal algorithm $\mathbf{Alg}_{\mathsf{Prim}}$ receives (x_t, λ_t) and outputs an arm $a_t \in [K]$.
Arm a_t is played and outcome vector $\boldsymbol{o}_t$ is observed (and passed to both algorithms).
Lagrange payoff $\mathcal{L}_t(a_t, \lambda_t)$ is computed as per Eq. (3.1),
and reported to $\mathbf{Alg}_{\mathsf{Prim}}$ as reward and $\mathbf{Alg}_{\mathsf{Dual}}$ as cost.
end

Algorithm 1: `LagrangeCBwLC` framework

did not choose it for presentation. Interestingly, setting $\eta = 1$, like in (Immorlica et al., 2019, 2022), does not appear to suffice even for `BwK` *if hard-stopping is not allowed (i.e., Algorithm 1 must continue as defined till round T).*

Lastly, we mention two further changes compared to `LagrangeBwK`*: we allow $\mathbf{Alg}_{\mathsf{Prim}}$ to respond to the chosen λ_t, which is crucial to handle contexts in Section 4, and we rescale the time consumption in Theorem 12, which allows for improved regret bounds.*

Remark 5 *A version of* `LagrangeBwK` *with parameter $\eta = {}^{T}/_{B}$ was recently used in Castiglioni et al. (2022). Their analysis is specialized to* `BwK` *and targets (improved) approximation ratios for the adversarial version. An important technical difference is that their algorithm does not make use of the* time resource*, a dedicated resource that track the time consumption.*

Remark 6 *In* `LagrangeCBwLC`*, the dual algorithm $\mathbf{Alg}_{\mathsf{Dual}}$ receives* full feedback *on its Lagrange costs: indeed, the outcome vector $\mathbf{o}_t$ allows Algorithm 1 to reconstruct $\mathcal{L}_t(a_t, i)$ for each resource $i \in [d]$. $\mathbf{Alg}_{\mathsf{Dual}}$ could also receive the context x_t, but our analysis does not make use of this.*

The intuition behind `LagrangeCBwLC` is as follows. If $\mathbf{Alg}_{\mathsf{Prim}}$ and $\mathbf{Alg}_{\mathsf{Dual}}$ satisfy certain regret-minimizing properties, the repeated game converges to a Nash equilibrium for the rescaled Lagrangian $\mathcal{L}_{\text{LP}}(D, \eta \cdot \lambda)$. The specific definition (3.1), for an appropriate choice of η, ensures that the strategy of $\mathbf{Alg}_{\mathsf{Prim}}$ in the Nash equilibrium is (near-)optimal for the problem instance by a suitable version of Lagrange duality. For `BwK` with hard-stopping problem, $\eta = 1$ suffices,[13] but for general instances of `BwLC` we choose $\eta > 1$ in a fashion that depends on the problem instance.

Primal/dual regret. We provide general guarantees for `LagrangeCBwLC` when invoked with arbitrary primal and dual algorithms $\mathbf{Alg}_{\mathsf{Prim}}$ and $\mathbf{Alg}_{\mathsf{Dual}}$ satisfying suitable regret bounds. We define the *primal problem* (resp., *dual problem*) as the online learning problem faced by $\mathbf{Alg}_{\mathsf{Prim}}$ (resp., $\mathbf{Alg}_{\mathsf{Dual}}$) from the perspective of the repeated game in `LagrangeCBwLC`. The primal problem is a bandit problem where algorithm's action set is the set of all arms, and the Lagrange payoffs are rewards. The dual problem is a full-feedback online learning problem where algorithm's "actions" are the resources in `CBwLC`, with Lagrange payoffs are costs. The *primal regret* (resp., *dual regret*) is the regret relative to the best-in-hindsight action in the respective problem. Formally, these quantities are as follows:

$$\begin{aligned}
\mathbf{Reg}_{\mathsf{Prim}}(T) &:= \left[\max_{\pi\in\Pi} \textstyle\sum_{t\in[T]} \mathcal{L}_t(\pi(x_t), \lambda_t)\right] - \textstyle\sum_{t\in[T]} \mathcal{L}_t(a_t, \lambda_t). \\
\mathbf{Reg}_{\mathsf{Dual}}(T) &:= \textstyle\sum_{t\in[T]} \mathcal{L}_t(a_t, \lambda_t) - \left[\max_{i\in[d]} \textstyle\sum_{t\in[T]} \mathcal{L}_t(a_t, i)\right].
\end{aligned} \tag{3.3}$$

13. Because $\texttt{Opt}_{\text{LP}} = \sup_{D\in\Delta_K} \inf_{\lambda\in\Delta_d} \mathcal{L}_{\text{LP}}(D, \lambda)$ when $\sigma_i \equiv 1$ and there is a null arm (Immorlica et al., 2019, 2022).

We assume that the algorithms under consideration provide high-probability upper bounds on the primal and dual regret:

$$\Pr\left[\forall\tau\in[T]\quad \mathbf{Reg}_{\mathsf{Prim}}(\tau)\leq\overline{\mathbf{Reg}}_{\mathsf{Prim}}(T,\delta)[\tau]\quad\text{and}\quad \mathbf{Reg}_{\mathsf{Dual}}(\tau)\leq\overline{\mathbf{Reg}}_{\mathsf{Dual}}(T,\delta)[\tau]\,\right]\geq 1-\delta, \tag{3.4}$$

where $\overline{\mathbf{Reg}}_{\mathsf{Prim}}(T,\delta)$ and $\overline{\mathbf{Reg}}_{\mathsf{Dual}}(T,\delta)$ are known upper bounds, non-decreasing in T, and $\delta\in(0,1)$ is the failure probability. Our theorems use "combined regret" $R(T,\delta)$ defined by

$$T/B\cdot\eta\cdot R(T,\delta):=\overline{\mathbf{Reg}}_{\mathsf{Prim}}(T,\delta)+\overline{\mathbf{Reg}}_{\mathsf{Dual}}(T,\delta)+\sqrt{T\log(dT/\delta)}, \tag{3.5}$$

where the third summand accounts for concentration.

Remark 7 *The range of Lagrange payoffs is proportional to $T/B\cdot\eta$, which is why we separate out this factor on the left-hand side of Eq. (3.5). For the non-contextual version with K arms, standard results yield $R(T,\delta)=O(\sqrt{KT\log(dT/\delta)}\,)$.*[14] *Several other applications of* `LagrangeBwK` *framework (and, by extension, of* `LagrangeCBwLC`*) are discussed in (Immorlica et al., 2022; Castiglioni et al., 2022). In Section 4, we provide a new primal algorithm for* `CBwLC` *with regression oracles. Most applications, including ours, feature $\widetilde{\mathcal{O}}(\sqrt{T})$ scaling for $R(T,\delta)$.*

Remark 8 *For our results, (3.5) with $\tau=T$ suffices. We only use the full power of Eq. (3.5) to incorporate the prior-work results on* `CBwK` *with hard-stopping,* i.e., *Theorem 13 and its corollaries.*

Our guarantees. Our first guarantee for `LagrangeCBwLC` is the most general, and places no assumption on the `CBwLC` instance beyond existence of a feasible solution in Eq. (2.3). This assumption holds without loss of generality when all constraints are packing constraints ($\sigma_i\equiv 1$) and there is a null arm (*i.e.,* it is feasible to do nothing), but is substantially more general, and accommodates covering constraints.

Theorem 9 *Suppose the LP (2.3) has a feasible solution. Fix some $\delta>0$ and use the setup in Eqs.* (3.3) *to* (3.5)*. Consider algorithm* `LagrangeCBwLC` *with parameter $\eta=\sqrt{B/R(T,\delta)}$. With probability at least $1-O(\delta)$, it satisfies*

$$\max\left(\,\mathtt{Opt}-\mathtt{Rew},\; V_i(T)\,\right)\leq O\left(\sqrt{T\cdot R(T,\delta)}\right)\quad\text{for each resource } i\in[d]. \tag{3.6}$$

Remark 10 *Consider the paradigmatic regime when $R(T,\delta)=\widetilde{\mathcal{O}}(\sqrt{\Psi\cdot T})$ for some parameter Ψ that does not depend on T. In this case, the right-hand side of Eq. (3.6) becomes $\widetilde{\mathcal{O}}\left(\Psi^{1/4}\cdot T^{3/4}\right)$.*

Remark 11 *The main purpose of Theorem 9 is to enable applications to regression oracles and to the switching environment (see, resp., Section 4 and Appendix A). An additional application to bandit convex optimization, which may be of independent interest, is given in Appendix C for details. However, we note that the $T^{3/4}$ scaling of outcome-regret is suboptimal for* `BwLC` *in the stochastic environment with K arms, since $\widetilde{\mathcal{O}}(\sqrt{KT})$ outcome-regret is achieved in Agrawal and Devanur (2014, 2019) with an algorithm based on optimism under uncertainty.*

Next, we obtain an improved $T/B\cdot R(T,\delta)$ regret rate under the more restrictive assumption that some *optimal* solution for the LP (2.3) is feasible *by a constant margin*. This is a reasonable assumption when the algorithm's goal is just to satisfy the constraints (so, formally, all actions get the same reward). More generally, this may be a reasonable assumption when the optimal solution is a distribution over policies

14. Using algorithms EXP3.P (Auer et al., 2002) for $\mathbf{Alg}_{\mathsf{Prim}}$ and Hedge (Freund and Schapire, 1997) for $\mathbf{Alg}_{\mathsf{Dual}}$, we obtain Eq. (3.4) with $\overline{\mathbf{Reg}}_{\mathsf{Prim}}(T,\delta)=O(\,T/B\cdot\eta\cdot\sqrt{KT\log(K/\delta)}\,)$ and $\overline{\mathbf{Reg}}_{\mathsf{Dual}}(T,\delta)=O(\,T/B\cdot\eta\cdot\sqrt{T\log(d/\delta)}\,)$.

with largest expected reward. Formally, a distribution $D \in \Delta_\Pi$ is called *ζ-feasible*, $\zeta \in [0,1)$ if for each non-time resource $i \in [d]$ it satisfies $\sigma_i\left(\frac{T}{B}\, c_i(D) - 1\right) \leq -\zeta$.

For this result, we modify the algorithm slightly. Namely, we redefine the time resource to consume $\frac{B}{T}\cdot(1-\zeta)$ per round for any arm, rather than $\frac{B}{T}$.

Theorem 12 *Suppose some optimal solution for LP (2.3) is ζ-feasible, for a known margin $\zeta > 0$. Fix some $\delta > 0$ and consider the setup in Eqs.* (3.3) *to* (3.5)*. Consider algorithm* `LagrangeCBwLC` *with parameter $\eta = 2/\zeta$, where the time resource is redefined to consume $\frac{B}{T}\cdot(1-\zeta)$ per round. With probability at least $1 - O(\delta)$, this algorithm satisfies*

$$\max\left(\texttt{Opt} - \texttt{Rew},\ V_i(T)\right) \leq \widetilde{\mathcal{O}}\left({}^{T}\!/_{B} \cdot {}^{1}\!/_{\zeta} \cdot R(T,\delta)\right) \quad \textit{for each resource } i \in [d]. \tag{3.7}$$

As an example, by Remark 7, we obtain $\widetilde{\mathcal{O}}(\sqrt{KT})$ regret rate for non-contextual `BwLC` when $B > \Omega(T)$ and ζ is a constant. Such regret rate is the best possible, in the worst case, even without resource constraints (Auer et al., 2002).

These two theorems can be viewed as "theoretical interfaces" to `LagrangeCBwLC` framework. We obtain them as corollaries of a more general analysis, which is deferred to Appendix B. For the last result, we restate another interface, which concerns the simpler `CBwK` problem and gives an ${}^{T}\!/_{B} \cdot R(T,\delta)$ regret rate whenever hard-stopping is allowed.[15] We invoke this result in Section 4.

Theorem 13 (Immorlica et al. (2019, 2022)) *Consider* `CBwK` *with hard-stopping. Fix some $\delta > 0$ and consider the setup in Eqs.* (3.3) *to* (3.5)*. Consider algorithm* `LagrangeCBwLC` *with parameter $\eta = 1$. With probability at least $1 - O(\delta)$, we have* $\texttt{Opt} - \texttt{Rew} \leq {}^{T}\!/_{B} \cdot O(R(T,\delta))$.

4. Contextual `BwLC` via regression oracles

In this section, we instantiate the `LagrangeCBwLC` framework with $\mathbf{Alg}_{\mathsf{Prim}}$ as `SquareCB`, a regression-based technique for contextual bandits from Foster and Rakhlin (2020). In particular, we assume access to a subroutine ("oracle") for solving the online regression problem, defined per the following protocol.

Problem protocol: Online regression

Parameters: K arms, T rounds, context space $\mathcal{Z}$, range $[a,b] \subset \mathbb{R}$.
In each round $t \in [T]$:

1. the algorithm outputs a regression function $f_t:\ \mathcal{Z} \times [K] \to [a,b]$.
   `// Informally,` $f_t(x_t,a_t)$ `must approximate the expected score` $\mathbb{E}[y_t \mid x_t, a_t]$.
2. adversary chooses context $z_t \in \mathcal{Z}$, arm $a_t \in [K]$, score $y_t \in [a,b]$,
   and auxiliary data $\texttt{aux}_t$ (if any).
3. the algorithm receives the new datapoint $(z_t, a_t, y_t, \texttt{aux}_t)$.

We assume access to an online regression algorithm with context space $\mathcal{Z} = \mathcal{X}$, scores y_t equal to rewards (resp., consumption of a given resource i), and no auxiliary data $\texttt{aux}_t$. It can be an arbitrary algorithm for this problem, subject to a performance guarantee stated below in Eq. (4.4) which asserts that the algorithm is able the scores y_t well. We refer to this algorithm, which we denote by $\mathbf{Alg}_{\mathsf{Est}}$, as the *online regression oracle*, and invoke it as a subroutine. Our algorithm for the `CBwLC` framework will be efficient whenever

15. Recall that under hard-stopping the algorithm effectively stops as soon as some constraint is violated, and therefore all constraint violations are bounded by 1.

the per-round update for the oracle is computationally efficient, *e.g.,* the update time does not depend on the time horizon T. For simplicity, we use the same oracle for rewards and for each resource $\in [d]$. However, our algorithm and analysis can easily accommodate a different oracle for each component of the outcome vector.

The quality of the oracle is typically measured in terms of squared regression error, which in turn can be upper-bounded whenever the conditional mean scores are well modeled by a given class $\mathcal{F}$ of regression functions; this is detailed in Sections 4.2 and 4.3.

4.1. Regression-based primal algorithm

Our primal algorithm, given in Algorithm 2, is parameterized by an online regression oracle $\mathbf{Alg}_{\mathsf{Est}}$. We create $d+1$ instances of this oracle, denoted $\mathcal{O}_i$, for $i \in [d+1]$, which we apply separately to rewards and to each resource; we use range $[0,1]$ for rewards and $[-1,+1]$ for resources. At each step t, given the regression functions $\widehat{f}_{t,i}$ produced by these oracle instances, Algorithm 2 first estimates the expected Lagrange payoffs in a plug-in fashion (Eq. (4.1)). These estimates are then converted into a distribution over arms in Eq. (4.2); this technique, known as *inverse gap weighting* optimally balances exploration and exploitation, as parameterized by a scalar $\gamma > 0$.

Given : T/B ratio, K arms, d resources as per the problem definition;
parameter $\eta \geq 1$ from `LagrangeCBwLC`;
online regression oracle $\mathbf{Alg}_{\mathsf{Est}}$; parameter $\gamma > 0$.
Init : Instance $\mathcal{O}_i$ of regression oracle $\mathbf{Alg}_{\mathsf{Est}}$ for each $i \in [d+1]$.
// $\widehat{f}_1(x,a)$ and $\widehat{f}_{i+1}(x,a)$ estimate, resp., $r(x,a)$ and $c_i(x,a)$, $i \in [d]$.
for *round* $t = 1, 2, \ldots$ *(until stopping)* **do**
For each oracle $\mathcal{O}_i$, $i \in [d+1]$: update regression function $\widehat{f}_t = \widehat{f}_{t,i}$.
Input context $x_t \in \mathcal{X}$ and dual distribution $\lambda_t = (\,\lambda_{t,i} \in [d]\,) \in \Delta_d$.
For each arm a, estimate $\mathbb{E}\left[\,\mathcal{L}_t(a,\lambda) \mid x_t\,\right]$ with

$$\widehat{\mathcal{L}}_t(a) := \widehat{f}_{t,1}(x_t,a) + \eta \cdot \textstyle\sum_{i\in[d]} \sigma_i \cdot \lambda_{t,i} \left(1 - \frac{T}{B} \cdot \widehat{f}_{t,i+1}(x_t,a)\right). \tag{4.1}$$

Compute distribution over the arms, $p_t \in \Delta_K$, as

$$p_t(a) = 1/(\,c_t^{\text{norm}} + \gamma \cdot \max_{a' \in [K]} \widehat{\mathcal{L}}_t(a') - \widehat{\mathcal{L}}_t(a)\,). \tag{4.2}$$

// c_t^{norm} is chosen so that $\sum_a p_t(a) = 1$, via binary search.
Draw arm a_t independently from p_t.
Output arm a_t, input outcome vector $\boldsymbol{o}_t = (r_t; c_{t,1}\,,\ \ldots\ , c_{t,d}) \in [0,1]^{d+1}$.
For each oracle $\mathcal{O}_i$, $i \in [d+1]$: pass a new datapoint $(\,x_t, a_t,\ (\boldsymbol{o}_t)_i\,)$.
end

Algorithm 2: Regression-based implementation of $\mathbf{Alg}_{\mathsf{Prim}}$

The per-round running time of $\mathbf{Alg}_{\mathsf{Prim}}$ is dominated by $d+1$ oracle calls and $K(d+1)$ evaluations of the regression functions $\widehat{f}_i$ in Eq. (4.1). For the probabilities in Eq. (4.2), it takes $O(K)$ time to compute the $\max$ expressions, and then $O(K \log \frac{1}{\epsilon})$ time to binary-search for c^{norm} up to a given accuracy ϵ.

It is instructive (and essential for the analysis) to formally realize $\mathbf{Alg}_{\mathsf{Prim}}$ as an instantiation of `SquareCB` (Foster and Rakhlin, 2020), a contextual bandit algorithm with makes use of a regression oracle following the protocol described in the prequel. Define the *Lagrange regression* as an online regression problem with data points of the form $(z_t, a_t, y_t, \mathtt{aux}_t)$ for each round t, where the context $z_t = (x_t, \lambda_t)$ consists of both the CBwK context x_t and the dual vector λ_t, the score $y_t = \mathcal{L}_t(a_t, \lambda_t)$ is the Lagrangian payoff as defined by Eq. (3.1), and the auxiliary data $\texttt{aux}_t = \boldsymbol{o}_t$ is the outcome vector. The *Lagrange oracle* $\mathcal{O}_{\texttt{Lag}}$ is an algorithm for this problem (*i.e.,* an online regression oracle) which, for each round t, uses the estimated Lagrangian payoff Eq. (3.1) as a regression function. Thus, $\textbf{Alg}_{\textsf{Prim}}$ is an instantiation of SquareCB algorithm equipped, with an oracle $\mathcal{O}_{\texttt{Lag}}$ for solving the Lagrange regression problem defined above.

4.2. Provable guarantees

Let us formalize the online regression problem faced by a given oracle $\mathcal{O}_i$, $i \in [d+1]$. In each round t of this problem, the context x_t is drawn independently from some fixed distribution, and the arm a_t is chosen arbitrarily, possibly depending on the history. The score is $y_t = (\boldsymbol{o}_t)_i$, the i-th component of the realized outcome vector for the (x_t, a_t) pair. Let f_i^* be the "correct" regression function, given by

$$f_i^*(x, a) = \mathbb{E}\left[\,(\boldsymbol{o}_t)_i \mid x_t = x,\; a_t = a\,\right] \quad \forall x \in \mathcal{X},\; a \in [K]. \tag{4.3}$$

Following the literature on online regression, we evaluate the performance of $\mathcal{O}_i$ in terms of *squared regression error*:

$$\textbf{Est}_i(\mathcal{O}_i) := \textstyle\sum_{t\in[T]} \left(\, \widehat{f}_{t,i}(x_t, a_t) - f_i^*(x_t, a_t) \,\right)^2, \quad \forall i \in [d+1]. \tag{4.4}$$

We rely on a known uniform high-probability upper-bound on these errors:

$$\forall \delta \in (0,1) \quad \exists U_\delta > 0 \quad \forall i \in [d+1] \quad \Pr\left[\,\textbf{Est}_i(\mathcal{O}_i) \le U_\delta\,\right] \ge 1 - \delta. \tag{4.5}$$

Theorem 14 *Suppose* $\textbf{Alg}_{\textsf{Prim}}$ *is given by Algorithm 2, invoked with a regression oracle* $\textbf{Alg}_{\textsf{Est}}$ *that satisfies Eq. (4.5). Fix an arbitrary failure probability* $\delta \in (0,1)$*, let* $U = U_{\delta/(d+1)}$*, and set the parameter* $\gamma = \frac{B}{T}\sqrt{\frac{KT}{d+1}/U}$*. Let* $\textbf{Alg}_{\textsf{Dual}}$ *be the exponential weights algorithm ("Hedge") (Freund and Schapire, 1997). Then Eqs. (3.4) and (3.5) are satisfied with* $R(T,\delta) = O\left(\sqrt{dTU\log(dT/\delta)}\right)$.

This guarantee directly plugs into each of the three "theoretical interfaces" of LagrangeCBwLC (Theorems 9, 12 and 13), highlighting the modularity of our approach. In particular, we obtain optimal $\sqrt{T}$ scaling of regret for contextual BwK, via Theorem 13. Let us spell out these corollaries for the sake of completeness.

Corollary 15 *Consider* LagrangeCBwLC *with primal and dual algorithms as in Theorem 14, and write* $\Phi = dU\log(dT/\delta)$*. Let* $\texttt{reg}_{\texttt{out}} := \max_{i\in[d]}\left(\,\texttt{Opt} - \texttt{Rew}, V_i(T)\,\right)$ *denote the outcome-regret.*

(a) Suppose the LP (2.3) has a feasible solution. Set the algorithm's parameter as $\eta = \sqrt{B/R(T,\delta)}$*. Then* $\texttt{reg}_{\texttt{out}} \le O\left(\,\Phi^{1/4}\cdot T^{3/4}\,\right)$.

(b) Suppose some optimal solution for LP (2.3) is ζ*-feasible,* $\zeta > 0$*. Set* $\eta = 2/\zeta$ *and redefine the time resource to consume* $\frac{B}{T}\cdot(1-\zeta)$ *per round. Then* $\texttt{reg}_{\texttt{out}} \le O\left(\,{}^{T}\!/_{B}\cdot{}^{1}\!/_{\zeta}\cdot\sqrt{\Phi T}\,\right)$

(c) Consider CBwK *with hard-stopping and set* $\eta = 1$*. Then* $\texttt{Opt} - \texttt{Rew} \le O\left(\,{}^{T}\!/_{B}\cdot\sqrt{\Phi T}\,\right)$*, and (by definition of hard-stopping) the constraint violations are bounded as* $V_i(T) \le 1$.

4.3. Discussion

Generality. Online regression algorithms typically restrict themselves to a particular class of regression functions, $\mathcal{F} \subset \{\,\mathcal{X} \times [K] \to \mathbb{R}\,\}$, so that $f_t \in \mathcal{F}$ for all rounds $t \in [T]$. Typically, such algorithms ensure that Eq. (4.5) holds for a given index $i \in [d+1]$ whenever a condition known as *realizability* is satisfied: $f_i^* \in \mathcal{F}$. Under this condition, standard algorithms obtain Eq. (4.5) with $U_\delta = U_0 + \log(2/\delta)$, where $U_0 < \infty$ reflects the intrinsic statistical capacity of class $\mathcal{F}$ (Vovk, 1998a; Azoury and Warmuth, 2001; Vovk, 2006; Gerchinovitz, 2013; Rakhlin and Sridharan, 2014). Standard examples include:

- Finite classes, for which Vovk (1998a) achieves $U_0 = \mathcal{O}(\log|\mathcal{F}|)$.
- Linear classes, where for a known feature map $\phi(x,a) \in \mathbb{R}^b$ with $\|\phi(x,a)\|_2 \leq 1$, regression functions are of the form $f(x,a) = \theta \cdot \phi(x,a)$, for some $\theta \in \mathbb{R}^b$ with $\|\theta\|_2 \leq 1$. Here, the Vovk-Azoury-Warmuth algorithm (Vovk, 1998b; Azoury and Warmuth, 2001) achieves $U_0 \leq \mathcal{O}(d \log(T/d))$. If d is very large, one could use Online Gradient Descent (e.g., Hazan (2016)) and achieve $U_0 \leq \mathcal{O}(\sqrt{T})$.

We emphasize that Eq. (4.5) can also be ensured via *approximate* versions of realizability, with the upper bound U_δ depending on the approximation quality. The literature on online regression features various such guarantees, which seamlessly plug into our theorem. See Foster and Rakhlin (2020) for further background.

`SquareCB` allows for various extensions to large, structured action sets. Any such extensions carry over to `LagrangeCBwLC`. Essentially, one needs to efficiently implement computation and sampling of an appropriate exploration distribution that generalizes Eq. (4.2). "Practical" extensions are known for action sets with linear structure (Foster et al., 2020; Zhu et al., 2021), and those with Lipschitz-continuity (via uniform discretization) (Foster et al., 2021a). More extensions to general action spaces, RL, and beyond are in (Foster et al., 2021a).

Implementation details. Several remarks are in order regarding the implementation.

1. While our theorem sets the parameter γ according to the known upper bound U_δ, in practice it may be advantageous to treat γ as a hyperparameter and tune it experimentally.

2. In practice, one could potentially implement the Lagrange oracle by applying $\mathbf{Alg}_{\mathsf{Est}}$ to the entire Lagrange payoffs $\mathcal{L}_t(a_t, \lambda_t)$ directly, with (x_t, λ_t) as a context.

3. Instead of computing distribution p_t via Eq. (4.2) and binary search for c^{norm}, one can do the following (cf. Foster and Rakhlin (2020)): Let $b_t = \mathrm{argmax}_{a\in[K]} \widehat{\mathcal{L}}_t(a)$. Set $p_t(a) = 1/\left(K + \gamma \cdot (\widehat{\mathcal{L}}_t(b_t) - \widehat{\mathcal{L}}_t(a))\right)$, for all $a \neq b_t$, and set $p_t(b_t) = 1 - \sum_{a\neq b_t} p_t(a)$. This attains the same regret bound (up to absolute constants) as in Theorem 16.

4. In some applications, the outcome vector is determined by an observable "fundamental outcome" of lower dimension. For example, in dynamic pricing an algorithm offers an item for sale at a given price p, and the "fundamental outcome" is whether there is a sale. The corresponding outcome vector is $(p, 1) \cdot \mathbf{1}_{\text{sale}}$, *i.e.,* a sale brings reward p and consumes 1 unit of resource. In such applications, it may be advantageous to apply regression directly to the fundamental outcomes.

4.4. Analysis: Proof of Theorem 14

We incorporate the existing analysis of `SquareCB` (Foster and Rakhlin, 2020) by applying it to the Lagrange oracle $\mathcal{O}_{\mathtt{Lag}}$, and restating it in our notation as Theorem 16. Define the squared regression error for $\mathcal{O}_{\mathtt{Lag}}$ as

$$\mathbf{Est}(\mathcal{O}_{\mathtt{Lag}}) = \textstyle\sum_{t\in[T]} \left(\widehat{\mathcal{L}}_t(a_t) - \mathcal{L}(a_t, \lambda_t) \right)^2. \tag{4.6}$$

The main guarantee for `SquareCB` posits a known high-probability upper-bound in the regression error:

$$\forall \delta \in (0,1) \quad \exists U_\delta^{\mathtt{Lag}} > 0 \quad \Pr\left[\mathbf{Est}(\mathcal{O}_{\mathtt{Lag}}) \leq U_\delta^{\mathtt{Lag}} \right] \geq 1 - \delta. \tag{4.7}$$

Theorem 16 (Foster and Rakhlin (2020)) *Consider Algorithm 2 with Lagrange oracle that satisfies Eq. (4.7). Fix $\delta \in (0,1)$, let $U = U_\delta^{Lag}$ be the upper bound from Eq. (4.7). Set the parameter $\gamma = \sqrt{AT/U}$. Then with probability at least $1 - O(\delta T)$ we have*

$$\forall \tau \in [T] \quad \mathbf{Reg}_{\mathsf{Prim}}(\tau) \leq O\left(\sqrt{TU \log(dT/\delta)} \right). \tag{4.8}$$

Remark 17 *The original guarantee stated in Foster and Rakhlin (2020) is for $\tau = T$ in Theorem 16. To obtain the guarantee for all τ, as stated, it suffices to replace Freedman inequality in the analysis in Foster and Rakhlin (2020) with its anytime version.*

To complete the proof, it remains to derive Eq. (4.7) from Eq. (4.5), *i.e.,* upper-bound the error on $\mathcal{O}_{\texttt{Lag}}$ using respective upper bounds that $\mathbf{Alg}_{\mathsf{Est}}$ for the individual outcomes $i \in [d+1]$. Represent $\mathbf{Est}(\mathcal{O}_{\texttt{Lag}})$ as

$$\mathbf{Est}(\mathcal{O}_{\texttt{Lag}}) = \textstyle\sum_{t \in [T]} \left(\Phi_t + \eta \cdot \frac{T}{B} \sum_{i \in [d]} \lambda_{t,i} \Psi_{t,i} \right)^2,$$

where $\Phi_t = f_{t,1}(x_t, a_t) - r(x_t, a_t)$ and $\Psi_{t,i} = c_i(x_t, a_t) - f_{t,i+1}(x_t, a_t)$. For each round t, we have

$$\begin{aligned} \left(\Phi_t + \eta \cdot \tfrac{T}{B} \textstyle\sum_{i \in [d]} \lambda_{t,i} \Psi_{t,i} \right)^2 &\leq 2\, \Phi_t^2 + 2\, (\eta \cdot T/B)^2 \left(\textstyle\sum_{i \in [d]} \lambda_{t,i} \Psi_{t,i} \right)^2 \\ &\leq 2\, \Phi_t^2 + 2\, (\eta \cdot T/B)^2 \textstyle\sum_{i \in [d]} \lambda_{t,i} \Psi_{t,i}^2, \end{aligned}$$

where the latter inequality follows from Jensen's inequality. Summing this up over all rounds t gives

$$\mathbf{Est}(\mathcal{O}_{\texttt{Lag}}) \leq 2(\eta \cdot T/B)^2 \textstyle\sum_{i \in [d+1]} \mathbf{Est}_i(\mathcal{O}_i).$$

The $(\eta \cdot T/B)^2$ scaling is due to the fact that consumption is scaled by $\eta \cdot T/B$ in the Lagrangian, and the error is quadratic. Consequently, (4.7) holds with $U_\delta^{\texttt{Lag}} = (d+1)(\eta \cdot T/B)^2\, U_{\delta/(d+1)}$.

5. Conclusions and open questions

We solve CBwLC via a Lagrangian approach to handle resource constraints, and a regression-based approach to handle contexts. Our solution emphasizes modularity of both approaches.

While we obtain optimal $\sqrt{T}$ scaling in regret for the special case of contextual BwK, the main open question concerns obtaining a similar guarantee to the full generality of CBwLC. Our current result achieves $\sqrt{T}$ under a strong feasibility assumption, and $T^{3/4}$ in general.

Given the results in Appendix A, more advanced guarantees relative to the pacing benchmark may be within reach. Most immediately, one would like to improve dependence on the number of switches, particularly when the changes are of small magnitude. Moreover, one would like to make assumptions on the benchmark rather than the environment. Similar extensions are known for adversarial bandits (*i.e.,* without resources).

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *25th Advances in Neural Information Processing Systems (NIPS)*, pages 2312–2320, 2011.

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.

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

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

Shipra Agrawal and Nikhil R. Devanur. Bandits with global convex constraints and objective. *Operations Research*, 67(5):1486–1502, 2019. Preliminary version in *ACM EC 2014*.

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.

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.

Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, June 2001.

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.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *54th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2013.

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

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *J. of the ACM*, 65(3):13:1–13:55, 2018. Preliminary version in *FOCS 2013*.

Santiago R. Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Manag. Sci.*, 65(9):3952–3968, 2019. Preliminary version in *ACM EC 2017*.

Santiago R. Balseiro, Haihao Lu, and Vahab S. Mirrokni. The best of many worlds: Dual mirror descent for online allocation problems. *Operations Research*, 2022. Forthcoming. Preliminary version in *ICML 2020*.

Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009.

Alberto Bietti, Alekh Agarwal, and John Langford. A contextual bandit bake-off. *J. of Machine Learning Research (JMLR)*, 22:133:1–133:49, 2021.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. Published with *Now Publishers* (Boston, MA, USA). Also available at `https://arxiv.org/abs/1204.5721`.

Sébastien Bubeck, Ofer Dekel, Tomer Koren, and Yuval Peres. Bandit convex optimization: \(\sqrt{T}\) regret in one dimension. In *28th Conf. on Learning Theory (COLT)*, pages 266–278, 2015.

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.

Matteo Castiglioni, Andrea Celli, and Christian Kroer. Online learning with knapsacks: the best of both worlds. In *39th Intl. Conf. on Machine Learning (ICML)*, 2022.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual Bandits with Linear Payoff Functions. In *14th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011.

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.

Giannis Fikioris and Éva Tardos. Approximately stationary bandits with knapsacks. In *36th Conf. on Learning Theory (COLT)*, 2023.

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.

Dylan J. Foster and Alexander Rakhlin. Beyond UCB: optimal and efficient contextual bandits with regression oracles. In *37th Intl. Conf. on Machine Learning (ICML)*, 2020.

Dylan J. Foster, Alekh Agarwal, Miroslav Dudík, Haipeng Luo, and Robert E. Schapire. Practical contextual bandits with regression oracles. In *35th Intl. Conf. on Machine Learning (ICML)*, pages 1534–1543, 2018.

Dylan J Foster, Claudio Gentile, Mehryar Mohri, and Julian Zimmert. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021a.

Dylan J. Foster, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. In *34th Conf. on Learning Theory (COLT)*, 2021b. Extended Abstract. The full paper appears at `https://arxiv.org/abs/2010.03104`.

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

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.

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

Jason Gaitonde, Yingkai Li, Bar Light, Brendan Lucier, and Aleksandrs Slivkins. Budget pacing in repeated auctions: Regret and efficiency without convergence. In *14th Innovations in Theoretical Computer Science Conf. (ITCS)*, 2023.

S. Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. *Journal of Machine Learning Research*, 14:729–769, 2013.

Negin Golrezaei, Patrick Jaillet, Jason Cheuk Nam Liang, and Vahab Mirrokni. Bidding and pricing in budget and roi constrained markets. *arXiv preprint arXiv:2107.07725*, 2021a.

Negin Golrezaei, Ilan Lobel, and Renato Paes Leme. Auction design for roi-constrained buyers. In *30th The Web Conference (formerly known as* WWW*)*, pages 3941–3952, 2021b.

Yuxuan Han, Jialin Zeng, Yang Wang, Yang Xiang, and Jiheng Zhang. Optimal contextual bandits with knapsacks under realizibility via regression oracles. In *26th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2023. Available at `arxiv.org/abs/2210.11834` since October 2022.

Elad Hazan. *Introduction to Online Convex Optimization*. Foundations and Trends in Optimization, 2016.

Elad Hazan and Kfir Y. Levy. Bandit convex optimization: Towards tight bounds. In *27th Advances in Neural Information Processing Systems (NIPS)*, pages 784–792, 2014.

Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

Nicole Immorlica, Karthik Abinav Sankararaman, Robert Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. In *60th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2019.

Nicole Immorlica, Karthik Abinav Sankararaman, Robert Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. *J. of the ACM*, August 2022. Preliminary version in *60th IEEE FOCS*, 2019.

Thomas Kesselheim and Sahil Singla. Online learning with vector costs and bandits with knapsacks. In *33rd Conf. on Learning Theory (COLT)*, pages 2286–2305, 2020.

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

Raunak Kumar and Robert Kleinberg. Non-monotonic resource utilization in the bandits with knapsacks problem. In *35th Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, Cambridge, UK, 2020. Versions available at `https://banditalgs.com/` since 2018.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *19th Intl. World Wide Web Conf. (WWW)*, 2010.

Shang Liu, Jiashuo Jiang, and Xiaocheng Li. Non-stationary bandits with knapsacks. In *35th Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Alexander Rakhlin and Karthik Sridharan. Online nonparametric regression. In *Conference on Learning Theory*, 2014.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability, 2020. Working paper, available at `https://arxiv.org/abs/2003.12699`.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, November 2019. Published with *Now Publishers* (Boston, MA, USA). Also available at `https://arxiv.org/abs/1904.07272`. Latest online revision: Jan 2022.

V. Vovk. A game of prediction with expert advice. *J. Computer and System Sciences*, 56(2):153–173, 1998a.

Vladimir Vovk. Competitive on-line linear regression. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 364–370, Cambridge, MA, USA, 1998b. MIT Press.

Vladimir Vovk. Metric entropy in competitive on-line prediction. *CoRR*, abs/cs/0609045, 2006.

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.

Yinglun Zhu, Dylan J. Foster, Paul Mineiro, and John Langford. Contextual bandits in large action spaces: Made practical. In *39th Intl. Conf. on Machine Learning (ICML)*, 2021.

Appendix A. Non-stationary environments

In this section, we generalize the preceding results by allowing the outcome distribution $\mathcal{D}^{\mathtt{out}}$ to change over time. In each round $t \in [T]$, the pair $(x_t, \mathbf{M}_t)$ is drawn independently from some outcome distribution $\mathcal{D}^{\mathtt{out}}_t$. The sequence of distributions $(\mathcal{D}^{\mathtt{out}}_1, \ldots, \mathcal{D}^{\mathtt{out}}_T)$ is chosen in advance by an adversary (and not revealed to the algorithm). We parameterize our results in terms of the number of switches: rounds $t \geq 2$ such that $\mathcal{D}^{\mathtt{out}}_t \neq \mathcal{D}^{\mathtt{out}}_{t-1}$; we refer to these as *environment-switches*. The algorithm does not know when the environment switches occur.

We measure regret against a benchmark that chooses the best distribution over policies for each round t separately. In detail, note that each outcome distribution $\mathcal{D}^{\mathtt{out}}_t$ defines a version of the linear program (2.3); call it $\mathtt{LP}_t$. Let $D^*_t \in \Delta_\Pi$ be an optimal solution to $\mathtt{LP}_t$, and $\mathtt{Opt}_{\mathtt{LP},t}$ be its value. Our benchmark is $\mathtt{Opt}_{\mathtt{pace}} := \sum_{t\in[T]} \mathtt{Opt}_{\mathtt{LP},t}$. The intuition is that the benchmark would like to pace the resource consumption uniformly over time. We term $\mathtt{Opt}_{\mathtt{pace}}$ the *pacing benchmark*.

We view the pacing benchmark as a reasonable target for an algorithm that wishes to keep up with a changing environment. However, this benchmark gives up on "strategizing for the future", such as underspending now for the sake of overspending later. On the other hand, this property is what allows us to obtain vanishing regret bounds w.r.t. this benchmark. In contrast, the standard benchmarks require moving from regret to approximation ratios once one considers non-stationary environments (Immorlica et al., 2019, 2022).[16]

To derive bounds on the pacing regret, we take advantage of the modularity of $\mathtt{LagrangeCBwLC}$ framework and availability of "advanced" bandit algorithms that can be "plugged in" as $\mathbf{Alg}_{\mathsf{Prim}}$ and $\mathbf{Alg}_{\mathsf{Dual}}$. We use algorithms for adversarial bandits that do not make assumptions on the adversary, and yet compete with a benchmark that allows a bounded number of switches (and the same for the full-feedback problem).

To proceed, we must redefine primal and dual regret to accommodate for switches. First, we redefine the primal regret in Eq. (3.3) for an arbitrary subset of rounds $\mathcal{T} \subset [T]$:

$$\mathbf{Reg}_{\mathsf{Prim}}(\mathcal{T}) := \left[\, \max_{\pi\in\Pi} \textstyle\sum_{t\in\mathcal{T}} \mathcal{L}_t(\pi(x_t), \lambda_t) \,\right] - \textstyle\sum_{t\in\mathcal{T}} \mathcal{L}_t(a_t, \lambda_t). \tag{A.1}$$

Next, an S*-switch sequence* is an increasing sequence of rounds $\boldsymbol{\tau} = (\,\tau_j \in [T] :\ 0 \leq j \leq S+1\,)$, where $\tau_0 = 1$ and $\tau_{S+1} = T$. The primal regret for $\boldsymbol{\tau}$ is defined as the sum over the intervals,

$$\mathbf{Reg}_{\mathsf{Prim}}(\boldsymbol{\tau}) := \textstyle\sum_{j\in[S]} \mathbf{Reg}_{\mathsf{Prim}}\left(\,[\tau_{j-1},\, \tau_j - 1]\,\right) + \mathbf{Reg}_{\mathsf{Prim}}\left(\,[\tau_S,\, \tau_{S+1}]\,\right). \tag{A.2}$$

For the dual regret, $\mathbf{Reg}_{\mathsf{Dual}}(\mathcal{T})$ and $\mathbf{Reg}_{\mathsf{Dual}}(\boldsymbol{\tau})$ are defined similarly. We assume a suitable generalization of Eq. (3.5). For every $S \in [T-1]$ and every S-switching sequence $\boldsymbol{\tau}$, we assume

$$\Pr\left[\,\mathbf{Reg}_{\mathsf{Prim}}(\boldsymbol{\tau}) \leq R^S_1(T,\delta) \quad \text{and} \quad \mathbf{Reg}_{\mathsf{Dual}}(\boldsymbol{\tau}) \leq R^S_2(T,\delta)\,\right] \geq 1-\delta, \tag{A.3}$$

for known functions $R^S_1(T,\delta)$ and $R^S_2(T,\delta)$ and failure probability $\delta \in (0,1)$. Similar to Eq. (3.5), we define "combined regret" $R^S(T,\delta)$ by

$$T/B \cdot \eta \cdot R^S(T,\delta) := R^S_1(T,\delta) + R^S_2(T,\delta) + 2T/B \cdot \sqrt{ST\log(KdT/\delta)}. \tag{A.4}$$

Remark 18 *For the non-contextual setting, implementing $\mathbf{Alg}_{\mathsf{Prim}}$ as algorithm EXP3.S (Auer et al. (2002)) achieves regret bound $R^S_1(T) = \tilde{O}(\sqrt{KST})$ if S is known, and $R^S_1(T) = \tilde{O}(S \cdot \sqrt{KT})$ against an unknown S. Below, we also obtain $R^S_1(T) \sim \sqrt{ST}$ scaling for $\mathtt{CBwLC}$ via a variant of Algorithm 2 (with known S). For the dual player, the Fixed-Share algorithm from from Herbster and Warmuth (1998) achieves $R^S_2(T) = O(\sqrt{ST\log d})$.*

16. This holds even for the special case of only packing constraints and a null arm, and even against the best fixed policy (let alone the best fixed distribution over policies, a more appropriate benchmark for a constrained problem).

Theorem 19 *Fix $\delta > 0$, assume (A.3), and use the notation in (A.4). Suppose each linear program $\texttt{LP}_t$, $t \in [T]$ has a feasible solution. Let S be the number of environment-switches (which need not be known to the algorithm). Set parameter $\eta = \sqrt{B/R^S(T,\delta)}$. With probability at least $1 - O(\delta T)$ we obtain*

$$\max\left(\texttt{Opt}_{\texttt{pace}} - \texttt{Rew},\ {}^{T}\!/\!_{B} \cdot V_i(T)\right) \le \widetilde{\mathcal{O}}\left(S \cdot \sqrt{T \cdot R^S(T,\delta)}\right) \quad \text{for each } i \in [d]. \tag{A.5}$$

Remark 20 *Consider the paradigmatic regime when $R^S(T,\delta) = \widetilde{\mathcal{O}}(\sqrt{\Psi \cdot ST})$ for some Ψ that does not depend on T. Then the right-hand side of Eq. (A.5) becomes $\widetilde{\mathcal{O}}\left(S \cdot (S\Psi)^{1/4} \cdot T^{3/4}\right)$.*

Proof Sketch. Let $\boldsymbol{\tau}$ be an S-switch sequence such that $\tau_1, \ldots, \tau_S$ are the environment-switches. We apply the respective part of Theorem 9 separately to each time interval $[t_i - t_{i-1} - 1]$, using the primal/dual regret bounds w.r.t. sequence $\boldsymbol{\tau}$. When restricted to any such interval, the problem is stochastic, and the primal/dual regret in Eq. (A.3) specializes to what is required in Eq. (3.4).

Remark 21 *When specialized to packing constraints, it is essential for Theorem 19 that* `LagrangeCBwLC` *continues until the time horizon (as opposed to stopping and skipping the remaining rounds once some resource is exceeded, as in Immorlica et al. (2019, 2022)). This allows us to apply Theorem 9 to any given time interval for which the problem is stochastic.*

Remark 22 *As an optimization, we may reduce the dependence on S in Theorem 19 by ignoring some of the shorter environment-switches. Let the sequence $\boldsymbol{\tau}$ be defined as in the proof sketch. Let $\ell_i = t_i - t_{i-1}$ be the length of respective stationarity intervals. Those of length at most L rounds collectively take up $\Phi(L) = \sum_{i \in [S+1]} \ell_i \cdot \mathbf{1}_{\{\ell_i \le L\}}$ rounds. We focus on environment-switches $t = t_i$ such that $\Phi(\ell_i) > R$, for some parameter T_0; we call them R-*significant*. Theorem 19 can be restated so that S is the number of R-significant environment-switches, for some R that does not exceed the regret bound.*[17]

Primal and dual algorithms. We now term to the task of developing primal algorithms that can be applied within `LagrangeCBwLC` in the non-stationary setting. To generalize the regression-based machinery from Section 4, define the correct regression function $f^*_{t,i}$ according to the right-hand side of Eq. (4.3), for each round $t \in [T]$ and each component i of the outcome vector. The estimation error $\mathbf{Est}_i(\mathcal{O}_i)$ is redefined in a natural way: as in Eq. (4.4), but f^*_i is replaced with $f^*_{t,i}$ for each round t. We are interested in the high-probability error bound (4.5), as in Theorem 14.

Theorem 23 *Let S be a known upper bound on the number of environment-switches. Consider $\mathbf{Alg}_{\mathsf{Prim}}$ as in as in Theorem 14, with the high-probability error bound $U = U_{\delta/(d+1)}$ defined, for this S, via Eq. (4.5). Let $\mathbf{Alg}_{\mathsf{Dual}}$ be Fixed-Share (Herbster and Warmuth, 1998). Then the algorithm satisfies the guarantee in Theorem 19* with $R^S(T,\delta) = O\left({}^{T}\!/\!_{B} \cdot \sqrt{dTU \log(dT/\delta)}\right)$.

Remark 24 *To obtain Eq. (4.5), we assume that each $f^*_{t,i}$ belongs to some known class $\mathcal{F}$ of regression functions. In particular, if $\mathcal{F}$ is finite, the regression oracle can be implemented via Vovk's algorithm (Vovk, 1998a), applied to an "extended" version of the function class $\mathcal{F}$ that accommodates for the S switches. This achieves Eq. (4.5) with $U_\delta = O(S \cdot \log|\mathcal{F}|) + \log(2/\delta)$. Plugging this in, we obtain $R^S(T,\delta) = O\left({}^{T}\!/\!_{B} \cdot \sqrt{ST \cdot d\log|\mathcal{F}| \cdot \log(dT/\delta)}\right)$.*

17. *E.g.*, we could take $R \sim T^{3/4}$ for part (a), and $R \sim T^{2/3}$ for part (b).

Appendix B. Analysis of `LagrangeCBwLC`

We obtain Theorems 9 and 12 as corollaries of a more general result, Theorem 25, which we now state and prove. This result posits that some optimal LP-solution is ζ-feasible, where ζ may be 0, and is stated in terms of an arbitrary choice for the parameter η.

Theorem 25 *Fix some $\delta > 0$ and consider the setup in Eqs. (3.3) to (3.5). Abbreviate $R = R(T, \delta)$. Suppose some optimal solution for LP (2.3) is ζ-feasible, for some known, constant $\zeta \geq 0$. Redefine the time resource to always consume $\frac{B}{T} \cdot (1 - \zeta)$ per round. Consider `LagrangeCBwLC` with an arbitrary parameter $\eta \in \left(0, \frac{B}{2R}\right)$. Then with probability at least $1 - O(\delta)$, the total reward `Rew` and constraint violations $V_i(T)$ for each non-time resource $i \in [d]$ satisfy*

$$\texttt{Opt} - \texttt{Rew} \leq 2\eta \cdot {}^{T}\!/_{B} \cdot R, \quad \textit{and} \quad V_i(T) \leq 2B/\eta - B\zeta + \eta \cdot {}^{T}\!/_{B} \cdot R. \tag{B.1}$$

Theorems 9 and 12 follow from the special cases in which (respectively) $\zeta = 0$ and $\zeta > 0$ is an absolute constant. More generally, taking $\eta = {}^{B}\!/_{T} \cdot \frac{-\zeta B + \sqrt{B^2 \cdot \zeta^2 + 4TR}}{R}$ in Eq. (B.1) yields

$$\max_{i \in [d]} \left(\texttt{Opt} - \texttt{Rew}, V_i(T) \right) \leq \mathcal{O}\left(\min\left({}^{T}\!/_{B} \cdot {}^{R}\!/_{\zeta}, \;\; \sqrt{TR} \right) \right). \tag{B.2}$$

B.1. Proof of Theorem 25

The remainder of this section is dedicated to proving Theorem 25. Fix $\eta \geq 1$. As a shorthand, let $R_{\mathsf{conc}}(T, \delta) := c \cdot \sqrt{T \log(dT/\delta)}$, for a suitably large absolute constant $c > 0$, be the term from Azuma-Hoeffding inequality. Let $\varepsilon_T := \frac{1}{B} \cdot R(T, \delta)$. For $D \in \Delta_\Pi$, $\lambda \in \Delta_d$, let $\mathcal{L}_{\texttt{ALG}}(D, \lambda) := \mathcal{L}_{\texttt{LP}}(D, \eta \cdot \lambda)$ denote the expected Lagrangian payoff, as per Eq. (3.2). Its maximin value is

$$\texttt{Opt}_{\texttt{Lag}}^{\eta} := \sup_{D \in \Delta_\Pi} \inf_{\lambda \in \Delta_d} \mathcal{L}_{\texttt{ALG}}(D, \lambda). \tag{B.3}$$

First, we show that $\texttt{Opt}_{\texttt{Lag}}^{\eta}$ is close to $\texttt{Opt}_{\texttt{LP}}$. This is the only place where ζ-feasibility is used directly.

Lemma 26 *For any given $\eta \geq 0$, $\zeta \in \mathbb{R}_+$, we have $\texttt{Opt}_{\texttt{Lag}}^{\eta} \geq \texttt{Opt}_{\texttt{LP}} + \eta\zeta$.*

Proof Let $D^* \in \Delta_\Pi$ be a ζ-feasible solution for LP (2.3). Then

$$\begin{aligned}
\texttt{Opt}_{\texttt{Lag}}^{\eta} - \eta\zeta &= \sup_{D \in \Delta_\Pi} \inf_{\lambda \in \Delta_d} \mathcal{L}_{\texttt{LP}}(D, \eta \cdot \lambda) - \eta\zeta \\
&\geq \inf_{\lambda \in \Delta_d} \mathcal{L}_{\texttt{LP}}(D^*, \eta \cdot \lambda) - \eta\zeta \\
&= \inf_{\lambda \in \Delta_d} \left\{ r(D^*) + \eta \left[\textstyle\sum_{i \in [d]} \lambda_i (\sigma_i (1 - {}^{T}\!/_{B} \cdot c_i(D^*)) - \zeta)) \right] \right\} \\
&\geq r(D^*) \qquad \textit{(by ζ-feasibility and redefined time resource)} \\
&\geq \texttt{Opt}_{\texttt{LP}}.
\end{aligned}$$

■

Next, we invoke the standard analysis of regret minimization in repeated zero-sum games (Freund and Schapire, 1996, 1999).[18]

18. Immorlica et al. (2022) spells out the variant when the game matrix is drawn i.i.d. in each round.

Lemma 27 *Define the primal (resp., dual) average play as $\overline{D}_T = \frac{1}{T}\sum_{t=1}^{T} D_t$ and $\bar{\lambda}_T = \frac{1}{T}\sum_{t=1}^{T}\lambda_t$. Then with probability at least $1-\delta$, we have $\inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(\overline{D}_T,\lambda) \geq \texttt{Opt}_{\texttt{Lag}} - \eta\varepsilon_T$.*

Proof From Eq. (3.4), denoting $R = R(T,\delta)$ we have,

$$\textstyle\sum_{t\in[T]}\mathcal{L}_{\texttt{ALG}}(D_t,\lambda_t) + \eta\cdot{}^{T}\!/_{B}\cdot R \geq \sup_{D\in\Delta_\Pi}\sum_{t\in[T]}\mathcal{L}_{\texttt{ALG}}(D,\lambda_t) = T\cdot\sup_{D\in\Delta_\Pi}\mathcal{L}_{\texttt{ALG}}(D,\bar{\lambda}_T)$$
$$\textstyle\sum_{t\in[T]}\mathcal{L}_{\texttt{ALG}}(D_t,\lambda_t) - \eta\cdot{}^{T}\!/_{B}\cdot R \leq \inf_{\lambda\in\Delta_d}\sum_{t\in[T]}\mathcal{L}_{\texttt{ALG}}(D_t,\lambda) = T\cdot\inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(\overline{D}_T,\lambda).$$

Rearranging gives

$$\inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(\overline{D}_T,\lambda) + \eta\varepsilon_T \geq \sup_{D\in\Delta_\Pi}\mathcal{L}_{\texttt{ALG}}(D,\bar{\lambda}_T) \geq \sup_{D\in\Delta_\Pi}\inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(D,\lambda).$$

■

Next, we prove the following property on the LP-feasibility of primal distributions.

Lemma 28 *Fix $\eta>0$ and $\zeta\in\mathbb{R}_+$, and any $\varepsilon\leq\eta^{-1}$. Consider distribution $D\in\Delta_\Pi$ satisfying*

$$\inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(D,\lambda) \geq \texttt{Opt}_{\texttt{Lag}} - \eta\varepsilon. \tag{B.4}$$

Then $V_i(D) \leq B\left(2\eta^{-1}-\zeta\right)$ for every resource $i\in[d]$.

Proof Suppose $V_i(D)\geq\alpha B$ for some $\alpha\geq 0$ and resource $i\in[d]$.We will prove that $\alpha\leq 2\eta^{-1}-\zeta$. Indeed, by choosing $\lambda=e_i$, the unit vector in the i-th dimension, we have

$$\begin{aligned}\inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(D,\lambda) &\leq r(D) + \eta\cdot\sigma_i(1 - {}^{T}\!/_{B}\cdot c_i(D))\\ &\leq r(D) - \eta\alpha \leq 1-\eta\alpha.\end{aligned}$$

Using Eq. (B.4), we get $\texttt{Opt}_{\texttt{Lag}} - \eta\varepsilon \leq 1-\eta\alpha$. Rearranging, gives

$$\eta\alpha \leq 1 - \texttt{Opt}_{\texttt{Lag}} + \eta\varepsilon \leq 1 - \texttt{Opt}_{\texttt{LP}} - \eta\zeta + \eta\varepsilon.$$

The last inequality uses Lemma 26. Since $\texttt{Opt}_{\texttt{LP}}\geq 0$, we have,

$$1 - \texttt{Opt}_{\texttt{LP}} - \eta\zeta + \eta\varepsilon \leq 1 - \eta\zeta + \eta\varepsilon.$$

It follows that $\alpha\leq\eta^{-1}+\epsilon-\zeta\leq 2\eta^{-1}-\zeta$, where the last inequality holds because $\varepsilon\leq\eta^{-1}$. ■

Lower-bounding rewards. Let us prove that $\texttt{Opt} - \texttt{Rew}$ is at most the right-hand side of Eq. (B.1). We assume that $\eta\in[0,1/\varepsilon_T]$. With probability at least $1-\delta$ we have,

$$\begin{aligned}\texttt{Rew} := \textstyle\sum_{t\in[T]} r_t(a_t) &\geq \textstyle\sum_{t\in[T]} r(a_t) - R_{\mathsf{conc}}(T,\delta) &&\text{(by Azuma-Hoeffding inequality)}\\ &\geq T\cdot r(\overline{D}_T) - \eta\cdot{}^{T}\!/_{B}\cdot R(T,\delta).\end{aligned}$$

Without loss of generality, let the time resource be $i=1$. Then $\frac{T}{B}c_1(\overline{D}_T) = 1-\zeta$ and $\sigma_1 = 1$. From the definition of $\mathcal{L}_{\texttt{ALG}}$ in Eq. (3.2) we have $r(\overline{D}_T) = \mathcal{L}_{\texttt{ALG}}(\overline{D}_T,e_1) - \eta\zeta$. Therefore,

$$\begin{aligned}\mathcal{L}_{\texttt{ALG}}(\overline{D}_T,e_1) &\geq \inf_{\lambda\in\Delta_d}\mathcal{L}_{\texttt{ALG}}(\overline{D}_T,\lambda)\\ &\geq \texttt{Opt}_{\texttt{Lag}} - \eta\varepsilon_T &&\text{(by Lemma 27)}\\ &\geq \texttt{Opt}_{\texttt{LP}} + \eta\zeta - \eta\varepsilon_T &&\text{(by Lemma 26).}\end{aligned}$$

In addition, we have

$$r(\overline{D}_T) \geq \texttt{Opt}_{\texttt{LP}} - \eta\varepsilon_T,$$

and

$$\begin{aligned}\texttt{Rew} &\geq T \cdot \texttt{Opt}_{\texttt{LP}} - \eta \cdot T \cdot \varepsilon_T - \eta \cdot {}^{T}\!/\!_{B} \cdot R(T,\delta) \\ &\geq T \cdot \texttt{Opt}_{\texttt{LP}} - 2\eta \cdot {}^{T}\!/\!_{B} \cdot R(T,\delta).\end{aligned}$$

Upper-bounding the constraint violations. Fix resource $i \in [d]$. We now upper-bound $V_i(T)$ as per Eq. (B.1). Again, we assume $\eta \in [0, 1/\varepsilon_T]$. By Lemma 27, the primal average play $\overline{D}_T$ satisfies the premise of Lemma 28 with $\epsilon = \epsilon_T$. It follows that $V_i(D) \leq B\left(2\eta^{-1} - \zeta\right)$.

From Azuma-Hoeffding inequality, we have with probability at least $1-\delta$,

$$V_i(T) \leq \sigma_i\Big(\textstyle\sum_{t\in[T]} c_i(\overline{D}_T) - B\Big) + R_{\mathsf{conc}}(T,\delta) \leq \sigma_i\big(T \cdot c_i(\overline{D}_T) - B\big) + \eta \cdot {}^{T}\!/\!_{B} \cdot R(T,\delta).$$

Finally, $V_i(T) \leq 2B\eta^{-1} - B\zeta + \eta \cdot {}^{T}\!/\!_{B} \cdot R(T,\delta)$.

Appendix C. Bandit Convex Optimization with Linear Constraints

In this section, we spell out an additional application of `LagrangeCBwLC` to bandit convex optimization (`BCO`) with linear constraints. We consider `CBwLC` with concave rewards, convex consumption of packing resources, and concave consumption of covering resources. Essentially, we follow an application of `LagrangeBwK` from Immorlica et al. (2022, Section 7.4), which applies to `BwK` with concave rewards and convex resource consumption; we spell out the details for the sake of completeness. Without resource constraints, `BCO` has been studied in a long line of work starting from Kleinberg (2004); Flaxman et al. (2005) and culminating in Bubeck et al. (2015); Hazan and Levy (2014); Bubeck et al. (2017).

Formally, we consider *Bandit Convex Optimization with Linear Constraints* (`BCOwLC`), a common generalization of `BwLC` and `BCO`. We define `BCOwLC` as a version of BwK, where the set of arms $\mathcal{A}$ is a convex subset of $\mathbb{R}^b$. For each round t, there is a concave function $f_t : \mathcal{X} \to [0,1]$ and functions $g_{t,i} : \mathcal{X} \to [-1,1]$, for each resource i, so that the reward for choosing action $a \in \mathcal{A}$ in this round is $f_t(a)$ and consumption of each resource i is $g_{t,i}(a)$. Each function $g_{t,i}$ is convex (resp., concave) if resource i is a packing (resp., covering) resource. In the stochastic environment, the tuple of functions $(f_t; g_{t,1}, \ldots, g_{t,d})$ is sampled independently in each round t from some fixed distribution (which is not known to the algorithm). In the switching environment, there are at most S rounds when that distribution changes.

The primal algorithm $\mathbf{Alg}_{\mathsf{Prim}}$ in `LagrangeCBwLC` faces an instance of BCO with an adaptive adversary, by definition of Lagrange payoffs (3.1). We use a BCO algorithm from Bubeck et al. (2017), which satisfies the high-probability regret bound against an adaptive adversary. It particular, it obtains Eq. (3.4) with

$$\overline{\mathbf{Reg}}_{\mathsf{Prim}}(T,\delta) = O\left({}^{T}\!/\!_{B} \cdot \eta \cdot \sqrt{\Phi T}\right), \quad \text{where} \quad \Phi = b^{19}\log^{14}(T)\log(1/\delta). \tag{C.1}$$

We apply Theorem 9 with this primal algorithm, and with Hedge for the dual algorithm.

Corollary 29 *Consider* `BCOwLC` *with a convex set of arms $\mathcal{A} \subset \mathbb{R}^b$ such that LP (2.3) has a feasible solution. Suppose the primal algorithm is from Bubeck et al. (2017) and the dual algorithm is the exponential weights algorithm ("Hedge") (Freund and Schapire, 1997). Then Eqs.* (3.4) *and* (3.5) *are satisfied with $R(T,\delta) = O\left(\sqrt{\Phi T}\right)$, where Φ is as in (C.1). Further,* `LagrangeCBwLC` *with parameter*

$\eta = \sqrt{B/R(T,\delta)}$ *satisfies*

$$\max_{i\in[d]} \left(\texttt{Opt} - \texttt{Rew}, V_i(T) \right) \leq O\left(\Phi^{1/4} \cdot T^{3/4} \right).$$

Remark 30 *This application of the* `LagrangeCBwLC` *framework is admissible because the analysis does not make use of the fact that the action space is finite. In particular, we never take union bounds over actions, and we can replace* max *and sums over actions with* sup *and integrals.*