Journal of Machine Learning Research 25 (2024) 1-37 Submitted 8/24; Revised 11/24; Published 12/24

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

Aleksandrs Slivkins slivkins@microsoft.com
Microsoft Research NYC

Xingyu Zhou xingyu.zhou@wayne.edu
Wayne State University, Detroit

Karthik Abinav Sankararaman karthikabinavs@gmail.com
Meta

Dylan J. Foster dylanfoster@microsoft.com
Microsoft Research NYC

Editor: Kevin Jamieson

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 statistically optimal under mild assumptions. 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.

Keywords: multi-armed bandits, contextual bandits, bandits with knapsacks, regression oracles, primal-dual algorithms

1. Introduction

Our scope. 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") for some parameter $B_i \leq T$. We focus on the stochastic environment, wherein the context and the arms' outcome vectors are drawn from a fixed joint distribution, independently 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), the special case with no contexts and some additional simplifications. 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. 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.[1] Contextual bandits with knapsacks (CBwK), a common generalization of contextual bandits and BwK, has also been explored in prior work.

In contextual bandits, even without resources, one typically specifies some additional structure. This is necessary for tractability, both statistical and computational, when one has a large number of possible contexts (as is the case in many/most applications). We adopt one standard approach which assumes access to *regression oracle*, a subroutine for solving certain supervised regression problems (Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2022). A contextual bandit algorithm calls a regression oracle to approximate the observed rewards/losses as a function of the corresponding context-arm pairs; this function is used to predict rewards/losses in the future.[2] This approach is computationally efficient, allows for strong provable guarantees, and tends to be superior in experiments compared to other approaches (see Section 1.1).

Our contributions. We design the first algorithm for CBwLC with regression oracles (in fact, this constitutes the first such algorithm for CBwK). To handle contexts via the regression-oracle approach, we build on the SquareCB algorithm from Foster and Rakhlin (2020). SquareCB estimates actions' rewards using the regression function and converts them into a distribution over actions that optimally balances exploration and exploitation. To handle resource constraints, we build on the LagrangeBwK framework of Immorlica, Sankararaman, Schapire, and Slivkins (2019, 2022). LagrangeBwK solves the simpler problem of BwK by setting up a repeated zero-sum game between two bandit algorithms: the "primal" algorithm which chooses among arms and the "dual" algorithm which chooses among resources. The payoffs in this game are given by a natural Lagriangian relaxation of the original constrained problem. Note that each of the two algorithms solves a bandit problem without resource constraints (but the payoff distribution changes over time, as it is driven by the other algorithm).

We make three technical contributions. First, we develop LagrangeCBwLC, an extension of the LagrangeBwK framework from BwK with hard-stopping to CBwLC. The main challenge is to bound constraint violations without hard-stopping (which trivially prevents them). This necessitates a subtle change in the algorithm (a re-weighting of the Lagrangian payoffs) and some new tricks in the analysis. The framework does not specify a particular primal algorithm, but instead assumes that it satisfies a certain regret bound. Second, we design a suitable primal algorithm that handles contexts via a regression oracle. This algorithm

1. 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.
2. Formally, a regression oracle returns a *regression function* that maps context-arm pairs to real values and belongs to some predetermined (and typically simple) function class.

builds on the SquareCB technique and we formally interpret it as an instance of SquareCB for a suitably defined contextual bandit problem. 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 the (optimal) $\widetilde{\mathcal{O}}(\sqrt{T})$ outcome-regret bound whenever $B > \Omega(T)$ and a minor non-degeneracy assumption holds. We also attain outcome-regret $\widetilde{\mathcal{O}}(T^{3/4})$ for general CBwLC problems. We emphasize that these are the first regret bound for CBwLC with regression oracles. We also obtain $\widetilde{\mathcal{O}}(\sqrt{T})$ regret for contextual BwK with hard-stopping.

Our proof leverages the inherent *modularity* of the techniques. A key conceptual contribution here is to identify the pieces and connecting them to one another. In particular, LagrangeCBwLC permits the use of any application-specific primal bandit algorithm with a particular regret guarantee,[3] and our SquareCB-based primal algorithm satisfies this guarantee when it has access to a suitable regression oracle. We provide two "theoretical interfaces" for LagrangeCBwLC that a primal algorithm can plug into, depending on whether the non-degeneracy assumption holds. We incorporate the original analysis of SquareCB as a theorem which we invoke when analyzing our primal algorithm. This theorem requires a regression oracle with a particular guarantee on the squared regression error; prior work on regression provides such oracles under various conditions. This is how our analysis for the stochastic environment comes together. We then re-use this whole machinery for the analysis of the switching environment.

Special cases. The LagrangeCBwLC framework 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 and to convex optimization (where rewards and resource consumption are convex/concave functions of an arm). However, the basic version of BwLC (*i.e.,* stochastic environment without additional structure) was already solved in prior work (Agrawal and Devanur, 2014, 2019), achieving optimal outcome-regret.

Our result for CBwK with hard-stopping builds on LagrangeBwK (as a special case of LagrangeCBwLC). Again, our analysis is modular: we encapsulate prior work on LagrangeBwK as a theorem that our primal algorithm plugs into.

Our result for the switching environment is new even for BwK, *i.e.,* when one only has packing constraints and no contexts. 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).

3. The LagrangeBwK framework permits a similar modularity for BwK, and Immorlica et al. (2019, 2022) and Castiglioni et al. (2022) use this modularity to derive several extensions.

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. We focus on CB with regression oracles, a promising emerging paradigm (Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2022). 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). [4]

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.[5] 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 $\widetilde{\mathcal{O}}(\sqrt{KT})$ 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 UCB-BwK, implements the paradigm of *optimism in the face of uncertainty.* Most work on BwK posits hard-stopping (as defined earlier). A detailed survey of BwK and its extensions can be found in Slivkins (Ch.11, 2019).

4. A *classification oracle* solves a different problem compared to a regression oracle: it is a subroutine for computing an optimal policy (mapping from contexts to arms) within a given class of policies.
5. *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 UCB-BwK 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 UCB-BwK 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 UCB-BwK 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.

More recently, Efroni et al. (2020); Ding et al. (2021); Zhou and Ji (2022); Ghosh et al. (2022) studied various extensions of BwLC, essentially following LagrangeBwK framework. They build on the same tools from constrained convex optimization as we do (*e.g.,* Corollary 15) in order to bound the constraint violations. However, they use specific primal and dual algorithms, and their analyses are tailored to these algorithms.[6] In contrast, our meta-theorem allows for arbitrary plug-in algorithms with suitable regret guarantees. Moreover, these papers only handle the "nice" case with Slater's constraint, whereas we also handle the general case. Finally, the regret bounds in these papers are suboptimal for large d, the number of constraints, scaling as $\sqrt{d}$ rather than $\sqrt{\log d}$, even in the non-contextual case.

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

6. The primal algorithms are based on "optimistic" bonuses, and the dual algorithms are based on gradient descent; Zhou and Ji (2022) also consider a primal algorithm based on Thompson Sampling.

et al., 2022; Fikioris and Tardos, 2023).[7] 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.

Liu et al. (2022) achieves a vanishing-regret result for Adversarial `BwK` against a standard benchmark when one has bounded pathlength and total variation.[8] This result is incomparable to our results for the switching environment, which are parameterized by the (unknown) number of switches and holds against a non-standard benchmark. Moreover, our approach extends to contextual bandits with regression oracles, whereas theirs does not.

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.[9] 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.

1.2 Concurrent work

Han et al. (2023) focus on `CBwK` with hard-stopping in the stochastic environment and obtain a result similar to Theorem 17(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 `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.

1.3 Organization

Section 2 introduces the `CBwLC` problem. Sections 3 and 4 provide our Lagrangian framework (`LagrangeCBwLC`) and the associated modular guarantees for this framework. The material specific to regression oracles is encapsulated in Section 5, including the setup and the `SquareCB`-based primal algorithm. Section 6 extends our results to the switching environment, defining a novel benchmark and building on the machinery from the previous sections.

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

7. Fikioris and Tardos (2023) is concurrent and independent work with respect to ours.
8. In fact, they consider the strongest possible standard benchmark: the optimal dynamic policy. The pathlength (or an upper bound thereon) must be known to the algorithm.
9. In particular, $B > \Omega(T)$ is explicitly assumed in, *e.g.,* Besbes and Zeevi (2009); Wang et al. (2014); Balseiro and Gur (2019); Castiglioni et al. (2022); Gaitonde et al. (2023).

all arms, rounds, and resources.[10] 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* $\vec{o}_t = (r_t; c_{t,1}, \ldots, c_{t,d}) \in [0,1] \times [-1,+1]^d$. For each resource $i \in [d]$ we are required to (approximately) satisfy the constraint

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

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 $V_i(T)$. [11]

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 $\vec{o}_t = \mathbf{M}_t(a_t)$. 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 $\mathcal{D}^{\mathsf{out}}$. In Section 6 we consider a generalization in which the distribution $\mathcal{D}^{\mathsf{out}}$ 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.

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.*[12] *In particular, i.i.d. context arrivals are needed to make the linear program (2.5) well-defined.*

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,*[13] *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.*

10. Throughout, $[n]$, $n \in \mathbb{N}$ stands for the set $\{1, 2, \ldots, n\}$.
11. A similar bi-objective approach is taken in Agrawal and Devanur (2014, 2019) and Agrawal et al. (2016).
12. 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.
13. Existence of a "null arm" is equivalent to the algorithm being able to skip rounds.

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 $\mathcal{D}^{\mathtt{out}}$. Our benchmark is the best algorithm for a given problem instance:

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

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

Additional notation. For the round-t outcome matrix $\mathbf{M}_t$, the row for arm $a \in [K]$ is denoted

$$\mathbf{M}_t(a) = (r_t(a);\, c_{t,1}(a), \ldots, c_{t,d}(a)),$$

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.

Expected reward and resource-i consumption for a given context-arm pair $(x, a) \in \mathcal{X} \times [K]$ is $r(x, a) = \mathbb{E}\left[r_t(a) \mid x_t = x \right]$ and $c_i(x, a) = \mathbb{E}\left[c_{t,i}(a) \mid x_t = x \right]$, where the expectation is over the marginal distribution of $\mathbf{M}_t$ conditional on $x_t = x$.

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 following happens in each round t: the algorithm deterministically chooses some distribution D_t over policies, draws a policy $\pi_t \sim D_t$ independently at random, observes context x_t (*after* choosing D_t), and chooses an arm $a_t = \pi_t(x_t)$. The *history* of the first t rounds is $\mathcal{H}_t := \left(D_s,\, \pi_s,\, x_s,\, a_s,\, r_s \right)_{s\in[t]}$.

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

$$r_t(D) := \mathop{\mathbb{E}}_{\pi\sim D}\left[r_t(\pi(x_t)) \right] \text{ and } c_{t,i}(D) := \mathop{\mathbb{E}}_{\pi\sim D}\left[c_{t,i}(\pi(x_t)) \right] \qquad \textit{(given } (x_t, \mathbf{M}_t)\textit{)} \tag{2.3}$$

$$r(D) := \mathbb{E}\left[r_t(\pi(x_t)) \right] \text{ and } c_i(D) := \mathbb{E}\left[c_{t,i}(\pi(x_t)) \right], \tag{2.4}$$

where the expectation is over both policies $\pi \sim D$ and context-matrix pairs $(x_t, \mathbf{M}_t) \sim \mathcal{D}^{\mathtt{out}}$. Note that $\mathbb{E}\left[r_t(a_t) \mid \mathcal{H}_{t-1} \right] = r(D_t)$ and likewise $\mathbb{E}\left[c_{t,i}(a_t) \mid \mathcal{H}_{t-1} \right] = c_i(D_t)$. If distribution D is played in all rounds $t \in [T]$, the expected constraint-i violation is denoted as $V_i(D) := \sigma_i \left(T \cdot c_i(D) - B \right)$.

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 in Eqs. (2.3) and (2.4) can be overloaded naturally to input a policy, an arm, or a distribution over arms.[14]

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).

Linear relaxation. We use 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:

$$\begin{array}{ll} \text{maximize} & r(D) \\ \text{subject to} & D \in \Delta_\Pi \\ & V_i(D) := \sigma_i \left(T \cdot c_i(D) - B \right) \leq 0 \qquad \forall i \in [d]. \end{array} \tag{2.5}$$

The value of this linear program is denoted $\mathtt{Opt}_{\mathsf{LP}}$. It is easy to see that $T \cdot \mathtt{Opt}_{\mathsf{LP}} = \mathtt{Opt}$. [15]

The Lagrange function associated with the linear program (2.5) is defined as follows:

$$\mathcal{L}_{\mathsf{LP}}(D, \lambda) := r(D) \; + \; \textstyle\sum_{i \in [d]} \; \sigma_i \cdot \lambda_i \left(1 - \frac{T}{B} \, c_i(D) \right), \quad D \in \Delta_\Pi, \, \lambda \in \mathbb{R}^d_+. \tag{2.6}$$

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

$$\mathtt{Opt}_{\mathsf{LP}} = \sup_{D \in \Delta_\Pi} \inf_{\lambda \in \mathbb{R}^d_+} \mathcal{L}_{\mathsf{LP}}(D, \lambda). \tag{2.7}$$

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;[16] $\mathbf{Alg}_{\mathsf{Dual}}$ 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

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

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

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

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

14. For the non-contextual problem, for example, $r(a)$ is the expected reward of arm a, and $r(D) = \mathbb{E}_{a \in D} \left[r(a) \right]$ is the expected reward for a distribution D over arms.
15. Indeed, to see that $T \cdot \mathtt{Opt}_{\mathsf{LP}} \geq \mathtt{Opt}$, 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. For the other direction, consider an LP-optimizing distribution D and observe that using this distribution in each round constitutes a feasible algorithm for the benchmark Eq. (2.2).
16. The terms 'primal' and 'dual' here refer to the duality in linear programming. For the LP-relaxation (2.5), 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 $\vec{o}_t$ 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.

Algorithm 1: `LagrangeCBwLC` framework

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 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 5, and we rescale the time consumption in Theorem 9, 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 $\vec{o}_t$ 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 $\mathcal{L}_{\mathsf{LP}}(D, \eta \cdot \lambda)$. 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,[17] but for general instances of `BwLC` we choose $\eta > 1$ in a fashion that depends on the problem instance.

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

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:

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

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

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

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

$$T/B \cdot \eta \cdot R(T,\delta) := \overline{\mathbf{Reg}}_{\mathsf{Prim}}(T,\delta) + \overline{\mathbf{Reg}}_{\mathsf{Dual}}(T,\delta) + 2\,R_{\mathsf{conc}}(T,\delta), \tag{3.5}$$

where $R_{\mathsf{conc}}(T,\delta) = O\left(T/B \cdot \eta \cdot \sqrt{T\log(dT/\delta)} \right)$ 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 $R(T,\delta) = O(\sqrt{KT\log(dT/\delta)}\,)$.*[18] *Several other applications of* `LagrangeBwK` *framework (and, by extension, of* `LagrangeCBwLC`*) are discussed in (Immorlica et al., 2022; Castiglioni et al., 2022). In Section 5, 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 11 and its corollaries.*

Our guarantees. Our main guarantee for `LagrangeCBwLC` holds whenever some solution for the LP (2.5) is feasible *by a constant margin.* 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 $\sigma_i\left(\frac{T}{B}\,c_i(D)-1\right)\le-\zeta$, and we need some D to be ζ-feasible with $\zeta>0$. This is a very common assumption in convex analysis, known as *Slater's condition.* It holds without loss of generality when all

18. 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 $\overline{\mathbf{Reg}}_{\mathsf{Dual}}(T,\delta) = O(\,T/B\cdot\eta\cdot\sqrt{T\log d}\,)$.

constraints are packing constraints ($\sigma_i \equiv 1$) and there is a null arm (*i.e.,* it is feasible to do nothing); it is a mild "non-degeneracy" assumption in the general case of BwLC. We obtain, essentially, the best possible guarantee when Slater's condition holds and $B > \Omega(T)$. We also obtain a non-trivial (but weaker) guarantee when $\zeta = 0$, *i.e.,* we are only guaranteed *some* feasible solution. Importantly, the ζ-feasible solution is only needed for the analysis: our algorithm does not need to know it (but it does need to know ζ).

Theorem 9. *Suppose some solution for LP (2.5) is ζ-feasible, for a known margin $\zeta \geq 0$. Fix some $\delta > 0$ and consider the setup in Eqs.* (3.3) *to* (3.5) *with "combined" regret bound $R(T, \delta)$.*

(a) If $\zeta > 0$, consider LagrangeCBwLC *with parameter $\eta = 2/\zeta$. With probability at least $1 - O(\delta)$,*

$$\max_{i \in [d]} \left(\texttt{Opt} - \texttt{Rew},\ V_i(T) \right) \leq O\left(T/B \cdot 1/\zeta \cdot R(T, \delta) \right). \tag{3.6}$$

(b) Consider LagrangeCBwLC *with parameter $\eta = \frac{B}{T}\sqrt{\frac{T}{R(T,\delta)}}$. With probability at least $1 - O(\delta)$,*

$$\max_{i \in [d]} \left(\texttt{Opt} - \texttt{Rew},\ V_i(T) \right) \leq O\left(\sqrt{T \cdot R(T, \delta)} \right). \tag{3.7}$$

The guarantee in part (a) is the best possible for BwLC, in the regime when $B > \Omega(T)$ and ζ is a constant. To see this, consider non-contextual BwLC with K arms: we obtain regret rate $\widetilde{\mathcal{O}}(\sqrt{KT})$ by Remark 7. This regret rate is the best possible in the worst case, even without resource constraints, due to the lower bound in (Auer et al., 2002). However, our guarantee is suboptimal when $B = o(T)$, compared to $\widetilde{\mathcal{O}}(\sqrt{KT})$ outcome-regret achieved in Agrawal and Devanur (2014, 2019) via a different approach.

To characterize the regret rate in part (b), 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. [19] Then the right-hand side of Eq. (3.7) becomes $\widetilde{\mathcal{O}}\left(\Psi^{1/4} \cdot T^{3/4} \right)$.

Remark 10. *When $B > \Omega(T)$ and the margin $\zeta > 0$ is a constant, the guarantee in Theorem 9(a) can be improved to zero constraint violation, with the same regret rate.*[20] *The algorithm is modified slightly, by rescaling the budget parameter B and the consumption values $c_{t,i}$ passed to the algorithm, and one need to account for this rescaling in the analysis. See Appendix C for details.*

The two guarantees in Theorem 9 can be viewed as "theoretical interfaces" to LagrangeCBwLC framework. We obtain them as special cases of a more general analysis (Theorem 12), which is deferred to the next section. The main purpose of these guarantees is to enable applications to regression oracles (Section 5) and also to the switching environment (Section 6, via

19. Here and elsewhere, $\widetilde{\mathcal{O}}(\cdot)$ notation hides $\log(Kdt/\delta)$ factors when the left-hand side depends on both T and δ.

20. Here, zero constraint violation happens with high probability rather than almost surely. Some recent work on the special case of online bidding under constraints (*e.g.,* Feng et al., 2023; Lucier et al., 2024) achieves zero constraint violation almost surely even when the algorithm must continue till time T.

Theorem 12). An additional application — to bandit convex optimization, which may be of independent interest — is spelled out in Appendix A.

For the last result in this section, we restate another "theoretical interface", which concerns the simpler CBwK problem and gives an $T/B \cdot R(T,\delta)$ regret rate with parameter $\eta = 1$ whenever hard-stopping is allowed.[21] We invoke this result in Section 5 along with Theorem 9.

Theorem 11 (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. Analysis of LagrangeCBwLC

We obtain Theorem 9 in a general formulation: with an arbitrary choice for the parameter η (which we tune optimally to obtain Theorem 9) and with realized primal/dual regret (rather than upper bounds thereon). The latter is needed to handle the switching environment in Section 6.

Theorem 12. *Suppose some solution for LP (2.5) is ζ-feasible, for a known margin $\zeta \in [0,1)$. Run* LagrangeCBwLC *with some primal/dual algorithms and parameter $\eta \geq 1$; write $\eta' = \eta \cdot T/B$ as shorthand. Fix some $\delta > 0$ and denote*

$$R := \left(\mathbf{Reg}_{\mathsf{Prim}}(T) + \mathbf{Reg}_{\mathsf{Dual}}(T) + 2\, R_{\mathsf{conc}}(T,\delta) \right) / \eta'. \tag{4.1}$$

Then:

(a) For $\zeta > 0$ and any $\eta \geq \frac{2}{\zeta}$,

$$\Pr\left[\texttt{Opt} - \texttt{Rew} \leq 3\eta' R \;\; \text{and} \;\; \max_{i \in [d]} V_i(T) \leq 4R + \eta' R \right] \geq 1 - O(\delta).$$

(b) For $\zeta = 0$ and any $\eta \geq 1$,

$$\Pr\left[\texttt{Opt} - \texttt{Rew} \leq 3\eta' R \;\; \text{and} \;\; \max_{i \in [d]} V_i(T) \leq \frac{T}{\eta'} + 2R + \eta' R \right] \geq 1 - O(\delta).$$

Note that our regret bound $\texttt{Opt} - \texttt{Rew} \leq 3\eta' R$ holds for any $\zeta \geq 0$. We use $\zeta > 0$ to provide a sharper bound on constraint violations.

4.1 Tools from Optimization (for the proof of Theorem 12)

Our proof builds on some techniques from prior work on linear optimization. When put together, these techniques provide a crucial piece of the overall argument.

Specifically, we formulate two lemmas that connect approximate saddle points, Slater's condition, and the maximal constraint violation for a given distribution $D \in \Delta_\Pi$,

$$V_{\max}(D) := \max_{i \in [d]} \left[\sigma_i \left(T \cdot c_i(D) - B \right) \right]_+ . \tag{4.2}$$

21. 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.

Using the notation from Eq. (2.6), let us define

$$\mathcal{L}_{\mathsf{LP}}^{\eta}(D,\lambda) := \mathcal{L}_{\mathsf{LP}}(D,\eta\lambda). \tag{4.3}$$

The first lemma is on the properties of approximate saddle points of $\mathcal{L}_{\mathsf{LP}}^{\eta}$. A *$\nu$-approximate saddle point* of $\mathcal{L}_{\mathsf{LP}}^{\eta}$ is a pair $(D',\lambda') \in \Delta_\Pi \times \Delta_d$ such that

$$\begin{aligned} \mathcal{L}_{\mathsf{LP}}^{\eta}(D',\lambda') &\geq \mathcal{L}_{\mathsf{LP}}^{\eta}(D,\lambda') - \nu, \quad \forall D \in \Delta_\Pi \\ \mathcal{L}_{\mathsf{LP}}^{\eta}(D',\lambda') &\leq \mathcal{L}_{\mathsf{LP}}^{\eta}(D',\lambda) + \nu, \quad \forall \lambda \in \Delta_d. \end{aligned}$$

Lemma 13. *Let (D',λ') be a ν-approximate saddle point of $\mathcal{L}_{\mathsf{LP}}^{\eta}$. Then it satisfies the following properties for any feasible solution $D \in \Delta_\Pi$ of LP in (2.5):*

(a) $r(D') \geq r(D) - 2\nu$.

(b) $r(D) - r(D') + \frac{\eta}{B}\, V_{\max}(D') \leq 2\nu$.

Theorem 13 follows from Lemmas 2-3 in Agarwal et al. (2017); we provide a standalone proof in Appendix B.1 for completeness.

The second lemma is on bounding the constraint violation under Slater's condition (*i.e.*, $\zeta > 0$).

Lemma 14 (Implications of Slater's condition)**.** *Consider the linear program in (2.5) and suppose Slater's condition holds, i.e., some distribution $\hat{D} \in \Delta_\Pi$ is ζ-feasible, $\zeta > 0$. Suppose for some numbers $C \geq 2/\zeta$, $\gamma > 0$ and distribution $\widetilde{D} \in \Delta_\Pi$ the following holds:*

$$r(D^*) - r(\widetilde{D}) + \tfrac{C}{B}\, V_{\max}(\widetilde{D}) \leq \gamma,$$

where D^* is an optimal solution of (2.5). Then $\frac{C}{B}\, V_{\max}(\widetilde{D}) \leq 2\gamma$.

Similar results have appeared in (Efroni et al., 2020, Theorem 42 and Corollary 44), which are variants of results in (Beck, 2017, Theorem 3.60 and Theorem 8.42), respectively. For completeness, we provide a standalone proof in Appendix B.2.

We use Theorem 13(b) and Theorem 14 through the following corollary.

Corollary 15. *Suppose some distribution over policies is ζ-feasible, for some $\zeta \geq 0$. Let (D',λ') be a ν-approximate saddle point of $\mathcal{L}_{\mathsf{LP}}^{\eta}$. Then, even for $\zeta = 0$, we have*

$$\frac{\eta}{B}\, V_{\max}(D') \leq 2\nu + 1. \tag{4.4}$$

Moreover, if $\zeta > 0$, then a sharper bound is possible:

$$\frac{\eta}{B}\, V_{\max}(D') \leq 4\nu \text{ whenever } \eta \geq 2/\zeta. \tag{4.5}$$

Proof The first statement trivially follows from Theorem 13(b). For the $\zeta > 0$ case, we invoke Theorem 13(b) with $D = D^*$ and Theorem 14 with $\widetilde{D} = D'$, $C = \eta$ and $\gamma = 2\nu$. ■

4.2 Proof of Theorem 12

We divide the proof into three steps: convergence, regret, and constraint violation. We note that the Slater's condition is only used in the third step.

Step 1 (convergence via no-regret dynamics). Consider the average play of $\mathbf{Alg}_{\mathsf{Prim}}$ and $\mathbf{Alg}_{\mathsf{Dual}}$: respectively, $\overline{D}_T = \frac{1}{T}\sum_{t\in[T]} D_t$ and $\overline{\lambda}_T = \frac{1}{T}\sum_{t\in[T]} \lambda_t$. We show that with probability at least $1-O(\delta)$,

$$(\overline{D}_T, \overline{\lambda}_T) \text{ is a } \nu\text{-approximate saddle point of the expected Lagrangian } \mathcal{L}^{\eta}_{\mathsf{LP}}, \tag{4.6}$$

with

$$\nu = \frac{1}{T}\cdot\left(\mathbf{Reg}_{\mathsf{Prim}}(T) + \mathbf{Reg}_{\mathsf{Dual}}(T) + 2R_{\mathsf{conc}}(T,\delta)\right). \tag{4.7}$$

This step is standard as in Freund and Schapire (1996) for a deterministic payoff matrix and in Immorlica et al. (2022) for a random payoff matrix. We provide a proof in Appendix B.3. The rest of the analysis conditions on the high-probability event (4.6).

Step 2 (regret analysis). Since $(\overline{D}_T, \overline{\lambda}_T)$ is a ν-approximate saddle point, Lemma 13(a) implies

$$r(\overline{D}_T) \geq r(D^*) - 2\nu = \mathtt{Opt}_{\mathsf{LP}} - 2\nu, \tag{4.8}$$

where D^* is an optimal solution of (2.5). With this, we obtain the regret bound as follows.

$$\begin{aligned}
\mathtt{Opt} - \mathtt{Rew} &\leq T\cdot \mathtt{Opt}_{\mathsf{LP}} - \textstyle\sum_{t\in[T]} r_t(a_t) \\
&\leq T\cdot \mathtt{Opt}_{\mathsf{LP}} - T\cdot r(\overline{D}_T) + R_{\mathsf{conc}}(T,\delta) \\
&\overset{(i)}{\leq} 2T\cdot\nu + R_{\mathsf{conc}}(T,\delta) \\
&\overset{(ii)}{\leq} 3\eta' R
\end{aligned}$$

where (i) holds by Eq. (4.8); (ii) holds by definition of ν in Eq. (4.7) and $R = R(T,\delta)$ in Eq. (3.5).

Step 3 (constraint violations). We first note that

$$\begin{aligned}
V_i(T) = \sigma_i\left(\textstyle\sum_{t\in[T]} c_{t,i} - B\right) &\leq T\sigma_i\left(c_i(\overline{D}_T) - \tfrac{B}{T}\right) + R_{\mathsf{conc}}(T,\delta) \\
&\leq V_{\max}(\overline{D}_T) + R_{\mathsf{conc}}(T,\delta).
\end{aligned} \tag{4.9}$$

Thus, it remains to bound $V_{\max}(\overline{D}_T)$. To this end, recall that we condition on (4.6), we can now invoke Theorem 15, for $\zeta = 0$ and $\zeta > 0$, respectively.

Case 1: $\zeta = 0$. By Eq. (4.4) in Theorem 15, recalling that $\eta' = \eta\cdot T/B$, we have

$$V_{\max}(\overline{D}_T) \leq B\cdot\frac{1+2\nu}{\eta} = \frac{(1+2\nu)T}{\eta'},$$

Hence, by Eq. (4.9) and the definitions of ν in Eq. (4.7) and $R = R(T,\delta)$ in Eq. (3.5), we have

$$V_i(T) \leq \frac{(1+2\nu)T}{\eta'} + R_{\mathsf{conc}}(T,\delta) \leq \frac{T}{\eta'} + 2R + \eta' R.$$

Case 2: $\zeta > 0$. By Eq. (4.5) in Theorem 15, we have

$$V_{\max}(\overline{D}_T) \leq B \cdot \frac{4\nu}{\eta} = \frac{4\nu T}{\eta'}.$$

Hence, using Equations (3.5), (4.7) and (4.9) as in Case 1, we have $V_i(T) \leq 4R + \eta' R$.

5. 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 below.

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 regression-context $z_t \in \mathcal{Z}$, arm $a_t \in [K]$, score $y_t \in [a,b]$, and auxiliary data aux_t (if any).
3. the algorithm receives the new datapoint $(z_t, a_t, y_t, \mathsf{aux}_t)$.

(We call z_t a *regression-context* to distinguish it from contexts in contextual bandits.)

We assume access to an algorithm for online regression with context space $\mathcal{Z} = \mathcal{X}$, scores y_t equal to rewards (resp., consumption of a given resource i), and no auxiliary data aux_t. It can be an arbitrary algorithm for this problem, subject to a performance guarantee stated below in Eq. (5.5) which asserts that the algorithm can approximate 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 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 5.2 and 5.3.

5.1 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. (5.1)). These estimates are then converted into a distribution over arms in Eq. (5.2); 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 $\mathbb{E}\,[\,\mathcal{L}_t(a,\lambda) \mid x_t\,]$ with

$$\widehat{\mathcal{L}}_t(a) := \widehat{f}_{t,1}(x_t,a) + \eta \cdot \textstyle\sum_{i\in[d]} \sigma_i \cdot \lambda_{t,i} \left(1 - \frac{T}{B} \quad \cdot \widehat{f}_{t,i+1}(x_t,a)\right). \tag{5.1}$$

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

$$p_t(a) = 1/(\; c_t^{\mathrm{norm}} + \gamma \cdot \max_{a'\in[K]} \widehat{\mathcal{L}}_t(a') - \widehat{\mathcal{L}}_t(a)\;). \tag{5.2}$$

// 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 $\vec{o}_t = (r_t; c_{t,1}\,,\;\ldots\;, c_{t,d}) \in [0,1]^{d+1}$.
For each oracle $\mathcal{O}_i$, $i \in [d+1]$: pass a new datapoint $(\,x_t, a_t, (\vec{o}_t)_i\,)$.

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. (5.1). For the probabilities in Eq. (5.2), 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 $(z_t, a_t, y_t, \mathsf{aux}_t)$ for each round t, where the regression-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 $\mathsf{aux}_t = \vec{o}_t$ is the outcome vector. For the purpose of this problem definition, regression-contexts z_t are adversarially chosen, possibly depending on the history (because the dual vector λ_t is generated by the dual algorithm). The *Lagrange oracle* $\mathcal{O}_{\mathsf{Lag}}$ 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, $\mathbf{Alg}_{\mathsf{Prim}}$ is an instantiation of `SquareCB` algorithm equipped, with an oracle $\mathcal{O}_{\mathsf{Lag}}$ for solving the Lagrange regression problem defined above.

5.2 Provable guarantees for Algorithm 2

Formulating our guarantees for Algorithm 2 requires some care, as they relies on performance of the regression oracles $\mathcal{O}_i$, $i \in [d+1]$. (The notions of Lagrange regression/oracle are not needed to state these guarantees; we only invoke them in the analysis in Section 5.4).

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 regression-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 $y_t = (\vec{o}_t)_i$, 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

$$f_i^*(x,a) = \mathbb{E}\left[\, (\vec{o}_t)_i \mid x_t = x,\ a_t = a \,\right] \quad \forall x \in \mathcal{X},\ a \in [K]. \tag{5.3}$$

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

$$\mathbf{Err}_i(\mathcal{O}_i) := \textstyle\sum_{t\in[T]} \left(\widehat{f}_{t,i}(x_t,a_t) - f_i^*(x_t,a_t) \right)^2, \quad \forall i \in [d+1]. \tag{5.4}$$

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

$$\forall \delta \in (0,1) \quad \exists U_\delta > 0 \quad \forall i \in [d+1] \quad \Pr\left[\, \mathbf{Err}_i(\mathcal{O}_i) \le U_\delta \,\right] \ge 1-\delta. \tag{5.5}$$

Now we are ready to spell out our primal/dual guarantee:

Theorem 16. *Suppose $\mathbf{Alg}_{\mathsf{Prim}}$ is given by Algorithm 2, invoked with a regression oracle $\mathbf{Alg}_{\mathsf{Est}}$ that satisfies Eq. (5.5). 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 $\mathbf{Alg}_{\mathsf{Dual}}$ 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` (Theorem 9(ab) and Theorem 11), highlighting the modularity of our approach. In particular, we obtain optimal $\sqrt{T}$ scaling of regret under Slater's condition (and $B \ge \Omega(T)$) and for contextual `BwK`, via Theorem 11. Let us spell out these corollaries for the sake of completeness.

Corollary 17. *Consider* `LagrangeCBwLC` *with primal and dual algorithms as in Theorem 16, and write $\Phi = dU\log(dT/\delta)$. Let $\mathsf{reg}_{\mathsf{out}} := \max_{i\in[d]}\left(\mathsf{Opt} - \mathsf{Rew}, V_i(T)\right)$ denote the outcome-regret.*

(a) Suppose the LP (2.5) has a ζ-feasible solution, $\zeta \in (0,1)$. Set the parameter to $\eta = 2/\zeta$. Then $\mathsf{reg}_{\mathsf{out}} \le O\left({}^{T}\!/_{B} \cdot {}^{1}\!/_{\zeta} \cdot \sqrt{\Phi T}\right)$ with probability at least $1 - O(\delta)$.

(b) Suppose the LP (2.5) has a feasible solution. Set the algorithm's parameter as $\eta = \frac{B}{T}\sqrt{\frac{T}{R(T,\delta)}}$. Then $\mathsf{reg}_{\mathsf{out}} \le O\left(\Phi^{1/4} \cdot T^{3/4}\right)$ with probability at least $1 - O(\delta)$.

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

5.3 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. (5.5) 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. (5.5) 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 $U_0 \leq \mathcal{O}(b \log(T/b))$. 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. (5.5) 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. (5.2). "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 regression-context.

3. Instead of computing distribution p_t via Eq. (5.2) and binary search for c^{norm}, one can do the following (cf. Foster and Rakhlin (2020)): Let $b_t = \text{argmax}_{a \in [K]} \widehat{\mathcal{L}}_t(a)$. 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 18.

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.

5.4 Proof of Theorem 16

We incorporate the existing analysis of SquareCB from Foster and Rakhlin (2020) by applying it to the Lagrange oracle $\mathcal{O}_{\mathsf{Lag}}$, and restating it in our notation as Theorem 18. Define the squared regression error for $\mathcal{O}_{\mathsf{Lag}}$ as

$$\mathbf{Err}(\mathcal{O}_{\mathsf{Lag}}) = \textstyle\sum_{t\in[T]}\left(\widehat{\mathcal{L}}_t(a_t) - \mathbb{E}\left[\mathcal{L}_t(a_t,\lambda_t)\right] \right)^2. \tag{5.6}$$

The main guarantee for SquareCB posits a known high-probability upper-bound on this quantity:

$$\forall \delta\in(0,1) \quad \exists U_\delta^{\mathsf{Lag}} > 0 \quad \Pr\left[\mathbf{Err}(\mathcal{O}_{\mathsf{Lag}}) \leq U_\delta^{\mathsf{Lag}}\right] \geq 1-\delta. \tag{5.7}$$

Theorem 18 (Implied by Foster and Rakhlin (2020))**.** *Consider Algorithm 2 with Lagrange oracle that satisfies Eq. (5.7). Fix $\delta\in(0,1)$, let $U = U_\delta^{\mathsf{Lag}}$ be the upper bound from Eq. (5.7). Set the parameter $\gamma = \sqrt{AT/U}$. Then with probability at least $1-O(\delta T)$ we have*

$$\forall \tau\in[T] \quad \mathbf{Reg}_{\mathsf{Prim}}(\tau) \leq O\left(\sqrt{T(U+1)\log(dT/\delta)}\right). \tag{5.8}$$

Remark 19. *The original guarantee stated in Foster and Rakhlin (2020) is for $\tau = T$ in Theorem 18. 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.*

Remark 20. *Recall that regression-contexts $z_t = (x_t,\lambda_t)$ in Lagrange regression are treated as adversarially chosen, because the dual vector λ_t is generated by the dual algorithm. In particular, one cannot immediately analyze $\mathbf{Alg}_{\mathsf{Prim}}$ via the technique of Simchi-Levi and Xu (2022), which assumes* stochastic *regression-context arrivals. The analysis from Foster and Rakhlin (2020) that we invoke handles adversarial regression-context arrivals.*

To complete the proof, it remains to derive Eq. (5.7) from Eq. (5.5), *i.e.,* upper-bound $\mathbf{Err}(\mathcal{O}_{\mathsf{Lag}})$ using respective upper bounds for the individual oracles $\mathcal{O}_i$. Represent $\mathbf{Err}(\mathcal{O}_{\mathsf{Lag}})$ as

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

where $\Phi_t = \widehat{f}_{t,1}(x_t,a_t) - r(x_t,a_t)$ and $\Psi_{t,i} = c_i(x_t,a_t) - \widehat{f}_{t,i+1}(x_t,a_t)$. 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,

$$\mathbf{Err}(\mathcal{O}_{\mathsf{Lag}}) \leq 2(\eta\cdot T/B)^2 \textstyle\sum_{i\in[d+1]} \mathbf{Err}_i(\mathcal{O}_i). \tag{5.9}$$

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, (5.7) holds with $U_\delta^{\mathsf{Lag}} = (d+1)(\eta\cdot T/B)^2\, U_{\delta/(d+1)}$. Finally, we plug this U_δ^{Lag} into (5.8), and then normalize $\mathbf{Reg}_{\mathsf{Prim}}$ according to (3.5) to obtain $R(T,\delta)$.

6. Non-stationary environments

In this section, we generalize the preceding results by allowing the outcome distribution $\mathcal{D}^{\mathsf{out}}$ to change over time. In each round $t \in [T]$, the pair $(x_t, \mathbf{M}_t)$ is drawn independently from some outcome distribution $\mathcal{D}^{\mathsf{out}}_t$. The sequence of distributions $(\mathcal{D}^{\mathsf{out}}_1, \ldots, \mathcal{D}^{\mathsf{out}}_T)$ 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 $\mathcal{D}^{\mathsf{out}}_t \neq \mathcal{D}^{\mathsf{out}}_{t-1}$; we refer to these as *environment-switches*. The algorithm does not know when the environment-switches occur. We refer to such problem instances as the *switching environment*.

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 $\mathcal{D}^{\mathsf{out}}_t$ defines a version of the linear program (2.5); call it LP_t. Let $D^*_t \in \Delta_\Pi$ be an optimal solution to LP_t, and $\mathtt{Opt}_{\mathsf{LP},\,t}$ be its value. Our benchmark is $\mathtt{Opt}_{\mathsf{pace}} := \sum_{t\in[T]} \mathtt{Opt}_{\mathsf{LP},\,t}$. The intuition is that the benchmark would like to pace the resource consumption uniformly over time. We term $\mathtt{Opt}_{\mathsf{pace}}$ the *pacing benchmark*. Accordingly, we are interested in the *pacing regret*,

$$\mathsf{reg}_{\mathsf{pace}} := \max_{i\in[d]} \left(\mathtt{Opt}_{\mathsf{pace}} - \mathsf{Rew}(\mathsf{ALG}),\ V_i(T) \right). \tag{6.1}$$

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).[22]

To derive bounds on the pacing regret, we take advantage of the modularity of `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). In the "back-end" of the analysis we invoke Theorem 12.

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

$$\mathbf{Reg}_{\mathsf{Prim}}(\mathcal{T}) := \left[\max_{\pi\in\Pi} \textstyle\sum_{t\in\mathcal{T}} \mathcal{L}_t(\pi(x_t), \lambda_t) \right] - \textstyle\sum_{t\in\mathcal{T}} \mathcal{L}_t(a_t, \lambda_t). \tag{6.2}$$

Next, an *S-switch sequence* is an increasing sequence of rounds $\vec{\tau} = (\, \tau_j \in [2, T] :\ j \in [S]\,)$, with a convention that $\tau_0 = 1$ and $\tau_{S+1} = T + 1$. The primal regret for $\vec{\tau}$ is defined as the sum over the intervals between these rounds:

$$\mathbf{Reg}_{\mathsf{Prim}}(\vec{\tau}) := \textstyle\sum_{j\in[S+1]} \mathbf{Reg}_{\mathsf{Prim}}\left([\tau_{j-1},\ \tau_j - 1] \right). \tag{6.3}$$

If $\vec{\tau}$ is the sequence of all environment-switches, then these are *stationarity intervals*, in the sense that the environment is stochastic throughout each interval. For the dual regret,

22. 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).

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

$$\Pr\left[\,\mathbf{Reg}_{\mathsf{Prim}}(\vec{\tau}) \leq {}^{T}\!/_{B} \cdot \eta \cdot R_1^S(T,\delta) \quad \text{and} \quad \mathbf{Reg}_{\mathsf{Dual}}(\vec{\tau}) \leq {}^{T}\!/_{B} \cdot \eta \cdot R_2^S(T,\delta)\,\right] \geq 1-\delta, \tag{6.4}$$

for known functions $R_1^S(T,\delta)$ and $R_2^S(T,\delta)$ and failure probability $\delta \in (0,1)$. Similar to Eq. (3.5), we define "combined" regret bound

$$R^S(T,\delta) := R_1^S(T,\delta) + R_2^S(T,\delta) + \sqrt{ST\log(KdT/\delta)}, \tag{6.5}$$

where the last term accounts for concentration.

Remark 21. *We do not explicitly assume that S is known. Instead, we note that achieving a particular regret bound (6.4) may require the algorithm to know S or an upper bound thereon. For the non-contextual setting, implementing $\mathbf{Alg}_{\mathsf{Prim}}$ as algorithm EXP3.S (Auer et al. (2002)) achieves regret bound $R_1^S(T,\delta) = \widetilde{\mathcal{O}}(\sqrt{KST})$ if S is known, and $R_1^S(T,\delta) = \widetilde{\mathcal{O}}(S\cdot\sqrt{KT})$ against an unknown S. Below, we also obtain $R_1^S(T,\delta) \sim \sqrt{ST}$ scaling for* `CBwLC` *via a variant of Algorithm 2 (with known S). For the dual player, the Fixed-Share algorithm from Herbster and Warmuth (1998) achieves $R_2^S(T,\delta) = O\left(\sqrt{ST\log d}\right)$ when S is known in advance, and $R_2^S(T,\delta) = O\left(S\sqrt{T\log d}\right)$, when S is not known.*

Theorem 22. *Consider* `CBwLC` *with S environment-switches. Suppose each linear program LP_t, $t \in [T]$ has a ζ-feasible solution, for some known margin $\zeta \in [0,1)$. Fix $\delta > 0$. Consider* `LagrangeCBwLC` *with primal/dual algorithms which satisfy regret bound (6.4). Use the notation in (6.5).*

(a) If $\zeta > 0$, use `LagrangeCBwLC` *with parameter $\eta = 2/\zeta$. Then with probability at least $1-O(S\delta)$,*

$$\mathsf{reg}_{\mathsf{pace}} \leq O\left({}^{T}\!/_{B} \cdot {}^{1}\!/_{\zeta} \cdot R^S(T,\delta)\right).$$

(b) Use `LagrangeCBwLC` *with parameter $\eta = \frac{B}{T}\sqrt{\frac{T}{R^S(T,\delta)}}$. Then with probability at least $1-O(S\delta)$,*

$$\mathsf{reg}_{\mathsf{pace}} \leq O\left(\sqrt{T\cdot R^S(T,\delta)}\right).$$

Proof Let $\vec{\tau}$ be the S-switch sequence that comprises all environment-switches. Since the problem is stochastic when restricted to each time interval $\mathcal{I}_j := [\,t_{j-1},\, t_j - 1\,]$, $j \in [S+1]$, we can invoke Theorem 12 specialized to this interval, with "effective" time horizon of $|\mathcal{I}_j|$ (see Theorem 24). Then, we sum up the resulting regret bound over all the intervals. Note that the concentration terms from Theorem 12 sum up as $\Lambda := \sum_{j\in[S+1]} R_{\mathsf{conc}}(|\mathcal{I}_j|,\delta) \leq \sqrt{ST\log(KdT/\delta)}$.

Thus, for part (a) we invoke Theorem 12(a) for each interval $\mathcal{I}_j$, $j \in [S]$. Summing up over the intervals, we obtain, with probability at least $1-O(S\delta)$, that

$$\begin{aligned}\mathsf{reg}_{\mathsf{pace}} &\leq O\left(\mathbf{Reg}_{\mathsf{Prim}}(\vec{\tau}) + \mathbf{Reg}_{\mathsf{Dual}}(\vec{\tau}) + \Lambda\right)\\ &\leq O\left({}^{T}\!/_{B} \cdot {}^{1}\!/_{\zeta} \cdot R^S(T,\delta)\right),\end{aligned}$$

where for the second inequality we invoke the regret bound in (6.4) for sequence $\vec{\tau}$.

For part (b), we likewise invoke Theorem 12(b) for each interval $\mathcal{I}_j$, $j \in [S]$. Summing up over the intervals, we obtain, with probability at least $1 - O(S\delta)$, that

$$\begin{aligned}
\forall i \in [d] \quad V_i(T) &\leq O\left(\mathbf{Reg}_{\mathsf{Prim}}(\vec{\tau}) + \mathbf{Reg}_{\mathsf{Dual}}(\vec{\tau}) + \Lambda + B/\eta\right) \\
&\leq O\left({}^{T}\!/_{B} \cdot \eta \cdot R^S(T,\delta) + B/\eta\right), \\
\mathtt{Opt}_{\mathsf{pace}} - \mathsf{Rew}(\mathsf{ALG}) &\leq O\left(\mathbf{Reg}_{\mathsf{Prim}}(\vec{\tau}) + \mathbf{Reg}_{\mathsf{Dual}}(\vec{\tau}) + \Lambda\right) \\
&\leq O\left({}^{T}\!/_{B} \cdot \eta \cdot R^S(T,\delta)\right). \\
\text{(Consequently)} \quad \mathtt{reg}_{\mathsf{pace}} &\leq O\left({}^{T}\!/_{B} \cdot \eta \cdot R^S(T,\delta) + B/\eta\right),
\end{aligned}$$

which is at most $O\left(\sqrt{T \cdot R^S(T,\delta)}\right)$ when η is as specified. ■

Remark 23. *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 regret bounds in Theorem 22 become $\mathtt{reg}_{\mathsf{pace}} \leq \widetilde{\mathcal{O}}\left({}^{T}\!/_{B} \cdot {}^{1}\!/_{\zeta} \cdot \sqrt{\Psi \cdot ST}\right)$ for part (a), and $\mathtt{reg}_{\mathsf{pace}} \leq \widetilde{\mathcal{O}}\left((S\Psi)^{1/4} \cdot T^{3/4}\right)$ for part (b).*

Remark 24. *To apply Theorem 12 to every given stationarity interval, the following two features are essential. First, Theorem 12 carries over* as if *the algorithm is run on this interval rather than the full time horizon. This is due to a non-trivial property of* `LagrangeCBwLC`*: it has no memory (and no knowledge of T) outside of its primal/dual algorithms. Second, Theorem 12 invokes* realized *primal (resp., dual) regret, rather than an upper bound thereon like in Theorem 9. This allows us to leverage the "aggregate" bound on primal (resp., dual) regret over the entire switching environment, as per Eq. (6.3). Using Theorem 9 directly would require similar upper bounds for every stationarity interval, which we do not immediately have.*[23]

Remark 25. *Even in the special case of packing constraints and hard-stopping (*i.e., *skipping the remaining rounds once some resource is exceeded), it is essential for Theorem 22 that* `LagrangeCBwLC` *continues until the time horizon. Our analysis would not work (even for this special case) if* `LagrangeCBwLC` *is replaced with some algorithm whose guarantees for the stationary environment assume hard-stopping. This is because such guarantees would not bound constraint violations within a given stationarity interval in the switching environment.*

Remark 26. *As an optimization, we may reduce the dependence on S in Theorem 22 by ignoring shorter environment-switches. Let the sequence $\vec{\tau}$ and the stationarity intervals $\mathcal{I}_j$ be defined as in the proof. The time intervals that last $\leq L$ rounds collectively take up $\Phi(L) = \sum_{j \in [S]} |\mathcal{I}_j| \cdot \mathbf{1}_{\{|\mathcal{I}_j| \leq L\}}$ rounds. We focus on environment-switches t_j such that $\Phi(|\mathcal{I}_j|) > R$, for some parameter R; we call them R-*significant*. Theorem 22 can be restated so that S is replaced with the number of R-significant environment-switches, for some R that does not exceed the stated regret bound.*

23. Regret on a given stationarity interval cannot immediately be upper-bounded by the aggregate regret in (6.3). This is because per-interval regret can in principle be negative for some (other) stationarity intervals. Besides, this approach would be inefficient even if it does work, resulting in an extra factor of S in the final regret bound.

Primal and dual algorithms. We now turn to the task of developing primal algorithms that can be applied within LagrangeCBwLC in the non-stationary *contextual* setting. To generalize the regression-based machinery from Section 5, define the correct regression function $f^*_{t,i}$ according to the right-hand side of Eq. (5.3) for each round $t \in [T]$ The estimation error $\mathbf{Err}_i(\mathcal{O}_i)$ is like in Eq. (5.4), but replacing f^*_i with $f^*_{t,i}$ for each round t. In formulae, for each $i \in [d+1]$ we have

$$f^*_{t,i}(x,a) = \mathbb{E}\left[\, (\vec{o}_t)_i \mid x_t = x,\, a_t = a \,\right] \quad \forall x \in \mathcal{X},\, a \in [K].$$
$$\mathbf{Err}_i(\mathcal{O}_i) := \textstyle\sum_{t\in[T]} \left(\widehat{f}_{t,i}(x_t,a_t) - f^*_{t,i}(x_t,a_t) \right)^2 .$$

We posit the high-probability error bound (5.5), as in Theorem 16. (A particular error bound of this form may depend on S, the number of environment-switches; achieving it may require the algorithm to know S or an upper bound thereon.)

Theorem 27. *Consider* CBwLC *with S environment-switches such that each linear program* LP_t, $t \in [T]$ *has a ζ-feasible solution, for some known $\zeta \in [0,1)$. Consider* $\mathbf{Alg}_{\mathsf{Prim}}$ *as in as in Theorem 16, with the high-probability error bound $U = U_{\delta/(d+1)}$ defined, for this S, via Eq. (5.5). Let* $\mathbf{Alg}_{\mathsf{Dual}}$ *be the Fixed-Share algorithm, as per Theorem 21. Then* LagrangeCBwLC *with these primal and dual algorithms satisfies the guarantees in Theorem 22(ab) with* $R^S(T,\delta) = O\left(\sqrt{dTU \log(dT/\delta)} \right)$.

Proof The full power of SquareCB analysis from Foster and Rakhlin (2020) implies Theorem 18 even with environment-switches, with Eq. (5.8) replaced by

$$\mathbf{Reg}_{\mathsf{Prim}}(\vec{\tau}) \leq O\left(\sqrt{T(U+1)\log(dT/\delta)} \right), \tag{6.6}$$

for any S-switch sequence $\vec{\tau}$ and any S.

To bound $U^{\mathsf{Lag}}_{\delta}$ in Eq. (5.7) (which is assumed by Theorem 18), we observe that Eq. (5.9) holds (and its proof carries over word-by-word from Section 5.4). Plugging this back into Eq. (6.6) and normalizing accordingly, we see that Eqs. (6.4) and (6.5) hold with $R^S(T,\delta) = O\left(\sqrt{dTU \log(dT/\delta)} \right)$. ■

Remark 28. *To obtain Eq. (5.5), 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 the class of all sequences of functions $(f_1\,,\;\ldots\;, f_T) \in \mathcal{F}^T$ with at most S switches. This achieves Eq. (5.5) with $U_\delta = O(S\cdot\log|\mathcal{F}|) + \log(2/\delta)$. Plugging this in, we obtain $R^S(T,\delta) = O\left(\sqrt{d \log|\mathcal{F}| \cdot \log(dT/\delta)} \right)$.*

7. 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 and (essentially) attains optimal regret bounds.

While our main results (Theorem 9(a) and corollaries) assume a known margin ζ in the Slater condition, it is desirable to recover similar results without knowing ζ in advance. Several follow-up papers achieve this (Guo and Liu, 2024; Castiglioni et al., 2024; Bernasconi et al., 2024), albeit with a worse dependence on the margin.[24] Aggarwal et al. (2024) achieves the same for the special case of auto-bidding, with the same dependence on ζ as ours.

Given the results in Section 6, more advanced guarantees for a non-stationary environment may be within reach. First, one would like to improve dependence on the number of switches, particularly when the changes are of small magnitude. Second, one would like to replace an assumption on the environment (at most S environment-switches) with assumptions on the benchmark. 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.

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.

Gagan Aggarwal, Giannis Fikioris, and Mingfei Zhao. No-regret algorithms in non-truthful auctions with budget and roi constraints. *arXiv preprint arXiv:2404.09832*, 2024.

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.

24. These papers are follow-up relative to the conference version of our paper, and concurrent work relative to the present version. Outcome-regret scales as $1/\zeta^2$ in Guo and Liu (2024); Castiglioni et al. (2024) and as $1/\zeta^3$ in Bernasconi et al. (2024), as compared with $1/\zeta$ dependence in Theorem 9(a) for a known ζ.

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*.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Martino Bernasconi, Matteo Castiglioni, and Andrea Celli. No-regret is not enough! bandits with general constraints through adaptive regret minimization. *arXiv preprint arXiv:2405.06575*, 2024.

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.

Matteo Castiglioni, Andrea Celli, and Christian Kroer. Online learning under budget and roi constraints via weak adaptivity. In *41st Intl. Conf. on Machine Learning (ICML)*, 2024.

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.

Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. Provably efficient safe exploration via primal-dual policy optimization. In *International conference on artificial intelligence and statistics*, pages 3304–3312. PMLR, 2021.

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.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

Zhe Feng, Swati Padmanabhan, and Di Wang. Online bidding algorithms for return-on-spend constrained advertisers. In *32nd The Web Conference (formerly known as* WWW*)*, pages 3550–3560, 2023.

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.

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.

Arnob Ghosh, Xingyu Zhou, and Ness Shroff. Provably efficient model-free constrained rl with linear function approximation. *Advances in Neural Information Processing Systems*, 35:13303–13315, 2022.

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.

Hengquan Guo and Xin Liu. Stochastic constrained contextual bandits via lyapunov optimization based estimation to decision framework. In *37th Conf. on Learning Theory (COLT)*, pages 2204–2231. PMLR, 2024.

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.

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.

Brendan Lucier, Sarath Pattathil, Aleksandrs Slivkins, and Mengxiao Zhang. Autobidders with budget and ROI constraints: Efficiency, regret, and pacing dynamics. In *37th Conf. on Learning Theory (COLT)*, 2024.

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. *Mathematics of Operation Research*, 47(3):1904–1931, 2022.

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`.

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.

Xingyu Zhou and Bo Ji. On kernelized multi-armed bandits with constraints. *Advances in neural information processing systems*, 35:14–26, 2022.

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. 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 $f_t : \mathcal{A} \to [0,1]$ and functions $g_{t,i} : \mathcal{A} \to [-1,1]$, 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

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

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 (2.5) has a* ζ*-feasible solution for some known* $\zeta \in [0,1)$*. 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 (A.1). The guarantees in Theorem 9(ab) apply with this* $R(T,\delta)$*.*

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.*

Appendix B. Details for the Proof of Theorem 12

B.1 Proof of Theorem 13

This lemma follows from Lemmas 2-3 in Agarwal et al. (2017). We prove it here for completeness.

Lemma (Theorem 13, restated)**.** *Let (D', λ') be a ν-approximate saddle point of $\mathcal{L}^{\eta}_{\mathsf{LP}}$. Then it satisfies the following properties for any feasible solution $D \in \Delta_\Pi$ of LP in (2.5):*

(a) $r(D') \geq r(D) - 2\nu$.

(b) $r(D) - r(D') + \frac{\eta}{B} V_{\max}(D') \leq 2\nu$.

We show the following claim, which will imply Theorem 13.

Claim 31. *For any feasible D, we have*

$$r(D') - \frac{\eta}{B} V_{\max}(D') + \nu \geq \mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda') \geq r(D) - \nu. \tag{B.1}$$

Proof We first show the following upper bound on $\mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda')$. In particular, for any $\lambda \in \Lambda$

$$\begin{aligned}
r(D') + \sum_{i \in [d]} \eta \lambda'_i \cdot \sigma_i \left(1 - \frac{T}{B} c_i(D')\right) &= \mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda') \\
&\leq \mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda) + \nu \\
&= r(D') + \sum_{i \in [d]} \eta \cdot \lambda_i \cdot \sigma_i \left(1 - \frac{T}{B} c_i(D')\right) + \nu \\
&\overset{(i)}{\leq} r(D') - \eta \max_{i \in [d]} \left[\sigma_i \left(\frac{T}{B} c_i(D') - 1\right)\right]_+ + \nu,
\end{aligned}$$

where (i) holds by choosing a specific $\lambda \in \Delta_d$ as follows

$$\lambda = \begin{cases} 0 & \text{if } \forall i \in [d], \sigma_i \left(1 - \frac{T}{B} c_i(D')\right) \geq 0 \\ e_{i^*} & \text{otherwise, where } i^* = \operatorname{argmin}_{i \in [d]} \sigma_i \left(1 - \frac{T}{B} c_i(D')\right), \end{cases}$$

where e_i denotes the unit vector for the i-th dimension, and $[x]_+ := \max\{x, 0\}$.

We also establish a lower bound for $\mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda')$. Note that for any feasible D, since $\lambda' \geq 0$, we have

$$\mathcal{L}^{\eta}_{\mathsf{LP}}(D, \lambda') = r(D) + \sum_{i \in [d]} \eta \lambda'_i \cdot \sigma_i \left(1 - \frac{T}{B} c_i(D)\right) \geq r(D).$$

Moreover, by the approximate saddle point of (D', λ'),

$$\mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda') \geq \mathcal{L}^{\eta}_{\mathsf{LP}}(D, \lambda') - \nu.$$

Putting them together yields that $\mathcal{L}^{\eta}_{\mathsf{LP}}(D', \lambda') \geq r(D) - \nu$. ■

Now let us use the claim to prove Theorem 13.

Part (a) follows since the LHS of (B.1) can be further upper bounded by $r(D') + \nu$.

Part (b) follows by the rearrangement of (B.1).

B.2 Proof of Theorem 14

Similar results have appeared in (Efroni et al., 2020, Theorem 42 and Corollary 44), which are variants of results in (Beck, 2017, Theorem 3.60 and Theorem 8.42), respectively. We provide a proof for completeness. We prove the following, which implies Theorem 14.

Lemma 32. *Consider the linear program in (2.5) and suppose Slater's condition holds, i.e., some distribution $\hat{D} \in \Delta_\Pi$ is ζ-feasible, $\zeta > 0$. Then:*

(a) Let λ^* be any optimal dual solution of the dual problem of (2.5), then $\|\lambda^*\|_1 \leq \frac{r(D^*)-r(\hat{D})}{\zeta} \leq \frac{1}{\zeta}$.

(b) Further, suppose the following holds for some $C \geq 2\|\lambda^*\|_1$, $\widetilde{D} \in \Delta_\Pi$ and $\gamma > 0$:

$$r(D^*) - r(\widetilde{D}) + \tfrac{C}{B} V_{\max}(\widetilde{D}) \leq \gamma$$

where D^* is an optimal solution of (2.5). Then $\frac{C}{B} V_{\max}(\widetilde{D}) \leq 2\gamma$.

Proof Let $q(\lambda) := \max_{D\in\Delta_\Pi} \mathcal{L}_{\mathsf{LP}}(D,\lambda)$ be the dual function. Consider an optimal dual solution $\lambda^* \in \operatorname{argmin}_{\lambda\in R^d_+} q(\lambda)$. By strong duality from Slater's condition, we have $q(\lambda^*) = r(D^*) < \infty$.

To prove part (a), we note that for any optimal dual solution λ^*, we have

$$r(D^*) = q(\lambda^*) \geq r(\hat{D}) + \sum_{i\in[d]} \lambda_i^* \cdot \sigma_i \left(1 - \frac{T}{B} c_i(\hat{D})\right) \geq r(\hat{D}) + \sum_i \lambda_i^* \zeta = r(\hat{D}) + \|\lambda^*\|_1 \zeta,$$

which gives our first result.

We turn to prove part (b). For $\tau \in \mathbb{R}^d$, define

$$u(\tau) := \max_{D\in\Delta_\Pi} \left\{ r(D) \mid V'(D) \leq -\tau \right\},$$

where $V'(D) = [V_1'(D), \ldots, V_d'(D)]^\top$. Note that for any $D \in \Delta_\Pi$

$$u(0) = r(D^*) = q(\lambda^*) \geq \mathcal{L}_{\mathsf{LP}}(D, \lambda^*)$$

Hence, we have for any D such that $V'(D) \leq -\tau$

$$\begin{aligned} u(0) - \tau^\top \lambda^* &\geq \mathcal{L}_{\mathsf{LP}}(D, \lambda^*) - \tau^\top \lambda^* \\ &= r(D) - \sum_i \lambda_i^* V_i'(D) - \tau^\top \lambda^* \\ &\geq r(D). \end{aligned}$$

Maximizing the RHS over all D such that $V'(D) \leq -\tau$, gives

$$u(0) - \tau^\top \lambda^* \geq u(\tau).$$

Set $\tau = \widetilde{\tau} := -\left[\left[\sigma_1\left(\frac{T}{B}c_1(\widetilde{D})-1\right)\right]_+,\ldots,\left[\sigma_d\left(\frac{T}{B}c_d(\widetilde{D})-1\right)\right]_+\right]^\top$ in the above inequality, we have

$$u(0) - \widetilde{\tau}^\top \lambda^* \geq u(\widetilde{\tau}) \geq u(0) = r(D^*) \geq r(\widetilde{D}),$$

which gives

$$r(D^*) - r(\widetilde{D}) \geq \widetilde{\tau}^\top \lambda^* \geq -\left\|\lambda^*\right\|_1 \left\|\widetilde{\tau}\right\|_\infty,$$

where the last step follows from Hölder's inequality.

From this result, we have

$$\begin{aligned}(C - \left\|\lambda^*\right\|_1)\left\|\widetilde{\tau}\right\|_\infty &= -\left\|\lambda^*\right\|_1\left\|\widetilde{\tau}\right\|_\infty + C\left\|\widetilde{\tau}\right\|_\infty \\ &\leq r(D^*) - r(\widetilde{D}) + C\left\|\widetilde{\tau}\right\|_\infty \leq \gamma,\end{aligned}$$

where the last step follows the assumption in Lemma 14. Hence, we finally obtain

$$\max_{i\in[d]}\left[\sigma_i\left(\frac{T}{B}c_i(\widetilde{D})-1\right)\right]_+ = \left\|\widetilde{\tau}\right\|_\infty \leq \frac{\gamma}{C - \left\|\lambda^*\right\|_1} \leq \frac{2\gamma}{C},$$

which follows from $C \geq 2\left\|\lambda^*\right\|_1$, hence finishing the proof. ■

B.3 Convergence to an approximate saddle point

We need to prove that (4.6) holds with probability at least $1 - O(\delta)$. That is:

$$\mathcal{L}^\eta_{\mathsf{LP}}(\overline{D}_T, \overline{\lambda}_T) \geq \mathcal{L}^\eta_{\mathsf{LP}}(D, \overline{\lambda}_T) - \nu, \quad \forall D \in \Delta_\Pi \tag{B.2}$$

$$\mathcal{L}^\eta_{\mathsf{LP}}(\overline{D}_T, \overline{\lambda}_T) \leq \mathcal{L}^\eta_{\mathsf{LP}}(\overline{D}_T, \lambda) + \nu, \quad \forall \lambda \in \Delta_d. \tag{B.3}$$

To establish (B.2), we note that for any $D \in \Delta_\Pi$, with probability $1 - O(\delta)$, we have

$$\begin{aligned}\mathcal{L}^\eta_{\mathsf{LP}}(D, \overline{\lambda}_T) &= \frac{1}{T}\sum_t \mathcal{L}^\eta_{\mathsf{LP}}(D, \lambda_t) \\ &\overset{(i)}{\leq} \frac{1}{T}\sum_t \mathcal{L}_t(D, \lambda_t) + \frac{1}{T}\cdot R_{\mathsf{conc}}(T,\delta) \\ &\overset{(ii)}{\leq} \frac{1}{T}\sum_t \mathcal{L}_t(a_t, \lambda_t) + \frac{1}{T}\cdot\left(\mathbf{Reg}_{\mathsf{Prim}}(T) + R_{\mathsf{conc}}(T,\delta)\right) \\ &\overset{(iii)}{\leq} \frac{1}{T}\sum_t \mathcal{L}_t(a_t, \overline{\lambda}_T) + \frac{1}{T}\cdot\left(\mathbf{Reg}_{\mathsf{Dual}}(T) + \mathbf{Reg}_{\mathsf{Prim}}(T) + R_{\mathsf{conc}}(T,\delta)\right) \\ &\overset{(iv)}{\leq} \mathcal{L}^\eta_{\mathsf{LP}}(\overline{D}_T, \overline{\lambda}_T) + \frac{1}{T}\cdot\left(\mathbf{Reg}_{\mathsf{Dual}}(T) + \mathbf{Reg}_{\mathsf{Prim}}(T) + 2R_{\mathsf{conc}}(T,\delta)\right) \\ &\overset{(v)}{=} \mathcal{L}^\eta_{\mathsf{LP}}(\overline{D}_T, \overline{\lambda}_T) + \nu,\end{aligned}$$

where (i) and (iv) follows from the concentration of Azuma-Hoeffding inequality; (ii) and (iii) hold by the definitions of primal and dual regrets in Eq. (3.3); (v) holds by definition of ν in (4.7).

We establish (B.3) using a similar analysis. In particular, for any $\lambda \in \Delta_d$, we have with probability $1 - O(\delta)$

$$\begin{aligned}
\mathcal{L}^{\eta}_{\mathsf{LP}}(\overline{D}_T, \lambda) &\overset{(i)}{\geq} \frac{1}{T}\sum_t \mathcal{L}_t(a_t, \lambda) - \frac{1}{T}\cdot R_{\mathsf{conc}}(T,\delta) \\
&\overset{(ii)}{\geq} \frac{1}{T}\sum_t \mathcal{L}_t(a_t, \lambda_t) - \frac{1}{T}\cdot (\mathbf{Reg}_{\mathsf{Dual}}(T) + R_{\mathsf{conc}}(T,\delta)) \\
&\overset{(iii)}{\geq} \frac{1}{T}\sum_t \mathcal{L}_t(\overline{D}_T, \lambda_t) - \frac{1}{T}\cdot (\mathbf{Reg}_{\mathsf{Prim}}(T) + \mathbf{Reg}_{\mathsf{Dual}}(T) + R_{\mathsf{conc}}(T,\delta)) \\
&\overset{(iv)}{\geq} \frac{1}{T}\sum_t \mathcal{L}^{\eta}_{\mathsf{LP}}(\overline{D}_T, \lambda_t) - \frac{1}{T}\cdot (\mathbf{Reg}_{\mathsf{Prim}}(T) + \mathbf{Reg}_{\mathsf{Dual}}(T) + 2R_{\mathsf{conc}}(T,\delta)) \\
&= \mathcal{L}^{\eta}_{\mathsf{LP}}(\overline{D}_T, \bar{\lambda}_T) - \frac{1}{T}\cdot (\mathbf{Reg}_{\mathsf{Prim}}(T) + \mathbf{Reg}_{\mathsf{Dual}}(T) + 2R_{\mathsf{conc}}(T,\delta)) \\
&\overset{(v)}{=} \mathcal{L}^{\eta}_{\mathsf{LP}}(\overline{D}_T, \bar{\lambda}_T) - \nu,
\end{aligned}$$

where (i) and (iv) follows from the concentration of Azuma-Hoeffding inequality; (ii) and (iii) hold by the definitions of primal and dual regrets in Eq. (3.3); (v) holds by definition of ν in (4.7).

Appendix C. Zero Constraint Violation

Let us provide a variant of Theorem 9(a) with zero constraint violation, fleshing out Theorem 10.

We use `LagrangeCBwLC` algorithm with two modifications. First, the budget parameter is scaled down as $B' = B(1-\epsilon)$, for some $\epsilon \in (0, 1/2]$. Second, for each *covering* resource $i \in [d]$ and each round t, the consumption reported back to `LagrangeCBwLC` is scaled down as $c'_{t,i} = c_{t,i} - 2\epsilon B/T$. [25] For packing resources i, reported consumption stays the same: $c'_{t,i} = c_{t,i}$. The modified algorithm, called `LagrangeCBwLC.rescaled`, has two parameters: $\eta \geq 1$ as before and $\epsilon \in (0, 1/2]$ for rescaling. Effectively, the modifications have made all resources slightly more constrained.

Theorem 33. *Suppose some solution for LP (2.5) is ζ-feasible, for a known margin $\zeta > 0$. Fix some $\delta > 0$ and consider the setup in Eqs. (3.3) to (3.5) with "combined" regret bound $R(T,\delta)$. Consider algorithm* `LagrangeCBwLC.rescaled` *with parameters $\eta = 4/\zeta$ and $\epsilon = 16 \cdot \frac{TR(T,\delta)}{\zeta B^2}$. Assume the budget is large enough so that $\epsilon \leq \zeta/2$. Then with probability at least $1 - O(\delta)$ we have $\min_{i\in[d]} V_i(T) \leq 0$ and*

$$\mathtt{Opt} - \mathtt{Rew} \leq O\left(\left({}^{T}\!/_{B} \cdot {}^{1}\!/_{\zeta} \right)^2 \cdot R(T,\delta) \right). \tag{C.1}$$

25. Note that $c'_{t,i} \in [-2, 1]$, whereas the original model has $c_{t,i} \in [-1, 1]$. However, the analysis leading to Theorem 9(a) carries over without modifications, and it is only the constants in $O(\cdot)$ that change slightly.

Remark 34. *The regret bound is the same as in Theorem 9(a), up to the factor of* $T/B \cdot 1/\zeta$ *(and the paradigmatic case is that this factor is an absolute constant).*

In the rest of this appendix, we prove Theorem 33. An execution of `LagrangeCBwLC.rescaled` on the original problem instance $\mathcal{I}$ can be interpreted as an execution of `LagrangeCBwLC` on a modified problem instance $\mathcal{I}'$ which has budget B' and realized consumptions $c'_{t,i}$ as defined above, and the same realized rewards as in $\mathcal{I}$. Note that $\mathcal{I}'$ is slightly more constrained in each resource compared to $\mathcal{I}$, *i.e.*, a slightly "harder" instance.

Let us write down a version of the LP (2.5) for the modified instance $\mathcal{I}'$:

$$\begin{array}{ll} \text{maximize} & r(D) \\ \text{subject to} & D \in \Delta_\Pi \\ & V'_i(D) := \sigma_i\left(\, T \cdot c'_i(D) - B' \,\right) \leq 0 \qquad \forall i \in [d], \end{array} \tag{C.2}$$

where $c'_i(D) := \mathbb{E}\left[\, c'_{t,i}(\pi(x_t)) \,\right]$ and the expectation is over $\pi \sim D$ and $(x_t, \mathbf{M}_t) \sim \mathcal{D}^{\mathsf{out}}$. Let $\mathtt{Opt}'_{\mathtt{LP}}$ be the value of this LP, and recall that $\mathtt{Opt}' = T \cdot \mathtt{Opt}'_{\mathtt{LP}}$. Note that for each resource $i \in [d]$,

$$V'_i(D) = \sigma_i\left(\, T \cdot c_i(D) - B \,\right) + \epsilon B = V_i(D) + \epsilon B \qquad \forall D \in \Delta_\Pi. \tag{C.3}$$

(Indeed, this holds for both packing and covering resources i.)

We claim that $\mathcal{I}'$ satisfies Slater condition with margin $\zeta' = \zeta/2$. Take $\hat{D}$ be the ζ-feasible solution to $\mathcal{I}$ guaranteed by the theorem statement. Then $V_i(\hat{D}) \leq -\zeta B$ for each resource $i \in [d]$. So by Eq. (C.3) we have $V'_i(D) \leq \epsilon B - \zeta B \leq \zeta B/2$ because $\epsilon \leq \zeta/2$ by assumption and $B' = B(1-\epsilon)$. Claim proved.

Thus, we can now invoke Theorem 9(a) for `LagrangeCBwLC` and the modified problem instance $\mathcal{I}'$. So, with probability at least $1 - O(\delta)$, we have

$$\max_{i \in [d]} \left(\, \mathtt{Opt}' - \mathtt{Rew},\; V'_i(T) \,\right) \leq O\left(\, T/B' \cdot 1/\zeta' \cdot R(T,\delta) \,\right) \leq O\left(\, T/B \cdot 1/\zeta \cdot R(T,\delta) \,\right), \tag{C.4}$$

where $\mathtt{Opt}'$ and $V'_i(T)$ are, resp., the benchmark (2.2) and the constraint violation (2.1) for the modified problem instance. It remains to "massage" this guarantee to obtain regret bound (C.1) and no constraint violations for the original problem instance. In what follows, let us condition on the event in (C.4).

To analyze regret, we bound the difference $\mathtt{Opt}_{\mathtt{LP}} - \mathtt{Opt}'_{\mathtt{LP}}$. We construct a feasible solution to instance $\mathcal{I}'$ via a mixture of D^*, the optimal solution for the original LP (2.5), and the ζ-feasible solution $\hat{D}$ to $\mathcal{I}$. Specifically, consider $D' := \left(\, 1 - \epsilon/\zeta \,\right) D^* + \epsilon/\zeta\, \hat{D}$. Hence, we have $V'_i(D') \leq 0$ for all $i \in [d]$, i.e., it is a feasible solution of the new LP (C.2). Consequently,

$$\mathtt{Opt}'_{\mathtt{LP}} \geq r(D') \geq \left(\, 1 - \epsilon/\zeta \,\right) \mathtt{Opt}_{\mathtt{LP}},$$

so $\Delta := \mathtt{Opt}_{\mathtt{LP}} - \mathtt{Opt}'_{\mathtt{LP}} \leq \epsilon/\zeta$. Finally,

$$\begin{aligned} \mathtt{Opt} - \mathtt{Rew} &\leq \mathtt{Opt}' - \mathtt{Rew} + T\Delta \\ &= O\left(\, T/B \cdot 1/\zeta \cdot R(T,\delta) + T^2/B^2 \cdot 1/\zeta^2 \cdot R(T,\delta) \,\right), \end{aligned}$$

where we plugged in (C.4) and the definition of ϵ.

To analyze constraint violation, use (C.4) in a more explicit version from Theorem 12, we have

$$\begin{aligned} V_i'(T) &\leq 4 \cdot {}^{T}\!/\!{}_{B'} \cdot {}^{1}\!/\!{}_{\zeta'} \cdot R(T, \delta) \\ &\leq 16 \cdot {}^{T}\!/\!{}_{B} \cdot {}^{1}\!/\!{}_{\zeta} \cdot R(T, \delta) = \epsilon B. \\ V_i(T) &= \sigma_i \left(\textstyle\sum_{t \in [T]} c_{t,i} - B \right) \\ &= \sigma_i \left(\textstyle\sum_{t \in [T]} c'_{t,i} - B' \right) - B\epsilon \leq 0. \end{aligned}$$