JMLR: Workshop and Conference Proceedings vol 35:1–26, 2014

Resourceful Contextual Bandits*

Ashwinkumar Badanidiyuru ASHWINKUMARBV@GMAIL.COM
Department of Computer Science, Cornell University, Ithaca NY, USA.

John Langford JCL@MICROSOFT.COM
Microsoft Research, New York, NY, USA.

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

Abstract

We study contextual bandits with ancillary constraints on resources, which are common in real-world applications such as choosing ads or dynamic pricing of items. We design the first algorithm for solving these problems that improves over a trivial reduction to the non-contextual case. We consider very general settings for both contextual bandits (arbitrary policy sets, Dudik et al. (2011)) and bandits with resource constraints (bandits with knapsacks, Badanidiyuru et al. (2013a)), and prove a regret guarantee with near-optimal statistical properties.

1. Introduction

Contextual bandits is a machine learning framework in which an algorithm makes sequential decisions according to the following protocol: in each round, a context arrives, then the algorithm chooses an action from the fixed and known set of possible actions, and then the reward for this action is revealed; the reward may depend on the context, and can vary over time. Contextual bandits is one of the prominent directions in the literature on online learning with exploration-exploitation tradeoff; many problems in this space are studied under the name *multi-armed bandits*.

A canonical example of contextual bandit learning is choosing ads for a search engine. Here, the goal is to choose the most profitable ad to display to a given user based on a search query and the available information about this user, and optimize the ad selection over time based on user feedback such as clicks. This description leaves out *many* important details, one of which is that every ad is associated with a budget which constrains the maximum amount of revenue which that ad can generate. In fact, this issue is so important that in some formulations it is the primary problem (e.g., Devanur and Vazirani, 2004).

The optimal solution with budget constraints fundamentally differs from the optimal solution without constraints. As an example, suppose that one ad has a high expected revenue but a small budget such that it can only be clicked on once. Should this ad be used immediately? From all reasonable perspectives, the answer is “no”. From the user’s or advertiser’s perspective, we prefer that this ad be displayed for the user with the strongest interest rather than for a user who simply has more interest than in other options. From a platform’s viewpoint, it is better to have more ads in the system, since they effectively increases the price paid in a second price auction. And from everyone’s viewpoint, it is simply odd to burn out the budget of an ad as soon as it is available. Instead, a small budget should be parceled out over time.

* This is the full version of a paper in the *26th Conf. on Learning Theory (COLT)*, 2014. The main results have been obtained while A. Badanidiyuru was a research intern at Microsoft Research New York City. A. Badanidiyuru was also partially supported by NSF grant AF-0910940.

To address these issues, we consider a generalization of contextual bandits in which there are one or several *resources* that are consumed by the algorithm. This formulation has many natural applications. *Dynamic ad allocation* follows the ad example described above: here, resources correspond to advertisers' budgets. In *dynamic pricing*, a store with a limited supply of items to sell can make customized offers to customers. In *dynamic procurement*, a contractor with a batch of jobs and a limited budget can experiment with prices offered to the workers, e.g. workers in a crowdsourcing market. The above applications have been studied on its own, but never in models that combine contexts and limited resources.

We obtain the first known algorithm for contextual bandits with resource constraints that improves over a trivial reduction to the non-contextual version of the problem. As such, we merge two lines of work on multi-armed bandits: contextual bandits and bandits with resource constraints. While significant progress has been achieved in each of the two lines of work (in particular, optimal solutions have been worked out for very general models), the specific approaches break down when applied to our model.

Our model. We define ***resourceful contextual bandits*** (in short: RCB), a common generalization of two general models for contextual bandits and bandits with resource constraints: respectively, contextual bandits with arbitrary policy sets (e.g., Langford and Zhang, 2007; Dudik et al., 2011) and bandits with knapsacks (Badanidiyuru et al., 2013a).

There are several resources that are consumed by the algorithm, with a separate constraint on each. In each round, the algorithm receives a reward and consumes some amount of each resource, in a manner that depends on the context and the chosen action, and may be randomized. We consider a stationary environment: in each round, the context and the mapping from actions to rewards and resource consumption is sampled independently from a fixed joint distribution, called the *outcome distribution*. Rewards and consumption of various resources can be correlated in an arbitrary way. The algorithm stops as soon as any constraint is violated. Initially the algorithm is given no information about the outcome distribution (except the distribution of context arrivals). In particular, expected rewards and resource consumptions are not known.

An algorithm is given a finite set Π of *policies*: mappings from contexts to actions. We compete against algorithms that must commit to some policy in Π before each round. Our benchmark is a hypothetical algorithm that knows the outcome distribution and makes optimal decisions given this knowledge and the restriction to policies in Π. The benchmark's expected total reward is denoted $\texttt{OPT}(\Pi)$. *Regret* of an algorithm is defined as $\texttt{OPT}(\Pi)$ minus the algorithm's expected total reward.

For normalization, per-round rewards and resource consumptions lie in $[0, 1]$. We assume that the distribution of context arrivals is known to the algorithm.

Discussion of the model. Allowing stochastic resource consumptions and arbitrary correlations between per-round rewards and per-round resource consumptions is essential: this is why our model subsumes diverse applications such as the ones discussed above,[1] and many extensions thereof. Further discussion of the application domains can be found in Appendix A.

Intuitively, the policy set Π consists of all policies that can possibly be learned by a given learning method, such as linear estimation or decision trees. Restricting to Π allows meaningful performance guarantees even if competing against *all* possible policies is intractable. The latter is common in real-life applications, as the set of possible contexts can be very large.

1. For example, in dynamic pricing the algorithm receives a reward and consumes a resource only if an item is sold.

Our benchmark can change policies from one round to another without restriction. As we prove, this is essentially equivalent in power to the best fixed *distribution* over policies. However, the best fixed policy may perform substantially worse.[2]

Our stopping condition corresponds to hard constraints: an advertiser cannot exceed his budget, a store cannot sell more items than it has in stock, etc. An alternative stopping condition is to restrict the algorithm to actions that cannot possibly violate any constraint if chosen in the current round, and stop if there is no such action. This alternative is essentially equivalent to the original version.[3] Moreover, we can w.l.o.g. allow our benchmark to use this alternative.

Our contributions: main algorithm. We design an algorithm, called `Mixture_Elimination`, and prove the following guarantee on its regret.

Theorem 1 *For all* `RCB` *problems with K actions, d resources, time horizon T, and for all policy sets Π. Algorithm* `Mixture_Elimination` *achieves expected total reward*

$$\texttt{REW} \geq \texttt{OPT}(\Pi) - O\left(1 + \tfrac{1}{B}\,\texttt{OPT}(\Pi)\right)\sqrt{dKT\;\log\left(dKT\,|\Pi|\right)}, \tag{1}$$

where $B = \min_i B_i$ is the smallest of the resource constraints $B_1\,,\;\dots\;,B_d$.

This regret guarantee is optimal in several regimes. First, we achieve an optimal square-root "scaling" of regret: if all constraints are scaled by the same parameter $\alpha > 0$, then regret scales as $\sqrt{\alpha}$. Second, if $B = T$ (i.e., there are no constraints), we recover the optimal $\tilde{O}(\sqrt{KT})$ regret. Third, we achieve $\tilde{O}(\sqrt{KT})$ regret for the important regime when $\texttt{OPT}(\Pi)$ and B are at least a constant fraction of T. In fact, Badanidiyuru et al. (2013a) provide a complimentary $\Omega(\sqrt{KT})$ lower bound for this regime, which holds in a very strong sense: for any given tuple $(K, B, \texttt{OPT}(\Pi), T)$.

The $\sqrt{\log|\Pi|}$ term in Theorem 1 is unavoidable (Dudik et al., 2011). The dependence on the *minimum* of the constraints (rather than, say, the maximum or some weighted combination thereof) is also unavoidable (Badanidiyuru et al., 2013a). For strongest results, one can rescale per-round rewards and per-round consumption of each resource so that they can be as high as 1.[4]

Note that the regret bound in Theorem 1 does not depend on the number of contexts, only on the number of policies in Π. In particular, it tolerates infinitely many contexts. On the other hand, if the set X of contexts is not too large, we can also obtain a regret bound with respect to the best policy among *all* possible policies. Formally, take $\Pi = \{\text{all policies}\}$ and observe that $|\Pi| \leq K^{|X|}$.

Further, Theorem 1 extends to policy sets Π that consist of *randomized policies*: mappings from contexts to distributions over actions. This may significantly reduce $|\Pi|$, as a given randomized policy might not be representable as a distribution over a small number of deterministic policies.[5] We assume deterministic policies in the rest of the paper.

2. The expected total reward of the best fixed policy can be half as large as that of the best distribution. This holds for several different domains including dynamic pricing / procurement, even without contexts (Badanidiyuru et al., 2013a). Note that without resource constraints, the two benchmarks are equivalent.
3. Each budget constraint changes by at most one, which does not affect our regret bounds in any significant way.
4. E.g., if per-round consumption of each resource is deterministically at most $\frac{1}{10}$, then multiplying all per-round consumption by 10 would increase the B by a factor of 10, and hence result in a stronger regret bound.
5. We can reduce `RCB` with randomized policies to `RCB` with deterministic policies simply by replacing each context x with a vector $(a_{(x,\,\pi)} : \pi \in \Pi)$ such that $a_{(x,\,\pi)} = \pi(x)$, and encoding the randomization in policies through the randomization in the context arrivals. While this blows up the context space, it does not affect our regret bound.

Computational issues. This paper is focused on proving the existence of solutions to this problem, and the mathematical properties of such a solution. The algorithm is specified as a mathematically well-defined mapping from histories to actions; we do not provide a computationally efficient implementation. Such "information-theoretical" results are common for the first solutions to new, broad problem formulations (e.g. Kleinberg et al., 2008; Kleinberg and Slivkins, 2010; Dudik et al., 2011). In particular, in the prior work for RCB without resource constraints there exists an algorithm with $\tilde{O}(\sqrt{KT})$ regret (Auer et al., 2002; Dudik et al., 2011), but for all known *computationally efficient* algorithms regret scales with T as $T^{2/3}$ (Langford and Zhang, 2007).

Our contributions: partial lower bound. We derive a partial lower bound: we prove that RCB is essentially hopeless for the regime $\texttt{OPT}(\Pi) \leq B \leq \sqrt{KT}/2$. The condition $\texttt{OPT}(\Pi) \leq B$ is satisfied, for example, in dynamic pricing with limited supply.

Theorem 2 *Any algorithm for* RCB *incurs regret* $\Omega(\texttt{OPT}(\Pi))$ *in the worst case over all problem instances such that* $\texttt{OPT}(\Pi) \leq B \leq \sqrt{KT}/2$ *(using the notation from Theorem 1).*

The above lower bound is specific to the general ("contextual") case of RCB. In fact, it points to a stark difference between RCB and the non-contextual version: in the latter, $o(\texttt{OPT})$ regret is achievable as long as (for example) $B \geq \log T$ (Badanidiyuru et al., 2013a).

While Theorem 2 is concerned with the regime of small B, note that in the "opposite" regime of very large B, namely $B \gg \sqrt{KT}$, the regret achieved in Theorem 1 is quite low: it can be expressed as $\tilde{O}(\sqrt{KT} + \epsilon \cdot \texttt{OPT}(\Pi))$, where $B = \frac{1}{\epsilon}\sqrt{KT}$.

Our contributions: discretization. In some applications of RCB, such as dynamic pricing and dynamic procurement, the action space is a continuous interval of prices. Theorem 1 usefully applies whenever the policy set Π is chosen so that the number of distinct actions used by policies in Π is finite and small compared to T. (Because one can w.l.o.g. remove all other actions.) However, one also needs to handle problem instances in which the policies in Π use prohibitively large or infinite number of actions.

We consider a paradigmatic example of RCB with an infinite action space: contextual dynamic pricing with a single product and prices in the $[0, 1]$ interval. We derive a corollary of Theorem 1 that applies to an arbitrary finite policy set Π.

Theorem 3 *Consider contextual dynamic pricing with a limited supply B of a single product. Let Π be the policy set and T be the time horizon. Then algorithm* Mixture_Elimination*, parameterized by a suitably chosen policy set Π' achieves expected total reward*

$$\texttt{REW} \geq \texttt{OPT}(\Pi) - O\left((TB)^{1/3} \cdot \log\left(TB|\Pi|\right) \right). \tag{2}$$

Here Π' depends only on the original policy set Π and the parameters B, T.

The regret bound is optimal (up to log factors) for the important regime $B \geq \Omega(T)$, due to the $\Omega(B^{2/3})$ lower bound from Babaioff et al. (2012) for the non-contextual case. Note that the theorem makes no assumptions on Lipschitz-continuity w.r.t. prices.[6]

6. This is not very surprising, as an optimal algorithm for the non-contextual case from Babaioff et al. (2012) does not assume Lispshitz-continuity, either. However, Lipschitz-continuity has been essential in some of the preceding work on dynamic pricing, e.g. Besbes and Zeevi (2009).

We use *discretization*: we reduce the original problem to one in which actions (i.e., prices) are multiples of some carefully chosen $\epsilon > 0$. Our approach proceeds as follows. For each $\epsilon > 0$ we define a "discretized" policy set Π_ϵ that approximates Π and only uses prices that are multiples of ϵ. We use Theorem 1 to obtain a regret bound relative to Π_ϵ; here the ϵ controls the tradeoff between the number of actions (and therefore the regret) and the "discretization error" of Π_ϵ. The technical difficulty here is to bound the discretization error; then we optimize the choice of ϵ to obtain the regret bound relative to Π.

Extending the discretization approach beyond dynamic pricing with a single product is problematic even without contexts, see Section 6 for further discussion.

Discussion: main challenges in `RCB`. The central issue in bandit problems is the tradeoff between *exploration*: acquiring new information, and *exploitation*: making seemingly optimal decisions based on this information. In this paper, we resolve the explore-exploit tradeoff in the presence of contexts and resource constraints. Each of the three components (explore-exploit tradeoff, contexts, and resource constraints) presents its own challenges, and we need to deal with all these challenges simultaneously. Below we describe these individual challenges one by one.

A well-known naive solution for explore-exploit tradeoff, which we call *pre-determined exploration*, decides in advance to allocate some rounds to exploration, and the remaining rounds to exploitation. The decisions in the exploration rounds do not depend on the observations, whereas the observations from the exploitation rounds do not impact future decisions. While this approach is simple and broadly applicable, it is typically inferior to more advanced solutions based on *adaptive exploration* – adapting the exploration schedule to the observations, so that many or all rounds serve both exploration and exploitation.[7] Thus, the general challenge in most explore-exploit settings is to design an appropriate adaptive exploration algorithm.

Resource constraints are difficult to handle for the following three reasons. First, an algorithm's ability to exploit is constrained by resource consumption for the purpose of exploration; the latter is stochastic and therefore difficult to predict in advance. Second, the expected per-round reward is no longer the right objective to optimize, as the action with the highest expected per-round reward could consume too much resources. Instead, one needs to take into account the expected reward over the entire time horizon. Third, the best fixed policy is no longer the right benchmark; instead, the algorithm should search over *distributions* over policies, which is a much larger search space.

In contextual bandit problems, an algorithm effectively chooses a policy $\pi \in \Pi$ in each round. Naively, this can be reduced to a non-contextual bandit problem in which "actions" correspond to policies. In particular, the main results in Badanidiyuru et al. (2013a) directly apply to this reduced problem. However, the action space in the reduced problem has size $|\Pi|$; accordingly, regret scales as $\sqrt{|\Pi|}$ in the worst case. The challenge in contextual bandits is to reduce this dependence. In particular, note that we replace $\sqrt{|\Pi|}$ with $\log |\Pi|$, an exponential improvement.

Related work. Multi-armed bandits have been studied since Thompson (1933) in Operations Research, Economics, and several branches of Computer Science, see (Gittins et al., 2011; Bubeck and Cesa-Bianchi, 2012) for background. This paper unifies two active lines of work on bandits: contextual bandits and bandits with resource constraints.

Contextual Bandits (Auer, 2002; Langford and Zhang, 2007) add contextual side information which can be used in prediction. This is a necessary complexity for virtually all applications of

7. For example, the difference in regret between pre-determined and adaptive exploration is $\tilde{O}(\sqrt{KT})$ vs. $O(K \log T)$ for stochastic K-armed bandits, and $\tilde{O}(T^{3/4})$ vs. $\tilde{O}(B^{2/3})$ for dynamic pricing with limited supply.

bandits since it is far more common to have relevant contextual side information than no such information. Several versions have been studied in the literature, see (Bubeck and Cesa-Bianchi, 2012; Dudik et al., 2011; Slivkins, 2011) for a discussion. For contextual bandits with policy sets, there exist two broad families of solutions, based on multiplicative weight algorithms (Auer et al., 2002; McMahan and Streeter, 2009; Beygelzimer et al., 2011) or confidence intervals (Dudik et al., 2011; Agarwal et al., 2012). We rework the confidence interval approach, incorporating and extending the ideas from the work on resource-constrained bandits (Badanidiyuru et al., 2013a).

Prior work on resource-constrained bandits includes dynamic pricing with limited supply (Babaioff et al., 2012; Besbes and Zeevi, 2009, 2012), dynamic procurement on a budget (Badanidiyuru et al., 2012; Singla and Krause, 2013; Slivkins and Vaughan, 2013), dynamic ad allocation with advertisers' budgets (Slivkins, 2013), and bandits with a single deterministic resource (Guha and Munagala, 2007; Gupta et al., 2011; Tran-Thanh et al., 2010, 2012). Badanidiyuru et al. (2013a) define and optimally solve a common generalization of all these settings: the non-contextual version of RCB. An extensive discussion of these and other applications, including applications to repeated auctions and network routing, can be found in (Badanidiyuru et al., 2013a).

To the best of our knowledge, the only prior work that explicitly considered contextual bandits with resource constraints is György et al. (2007). This paper considers a somewhat incomparable setting with a single unconstrained resource. The regret bound is against the best fixed policy, and scales as a root of the number of policies.

Our setting can be seen as a contextual bandit version of *stochastic packing* (e.g. Devanur and Hayes, 2009; Devanur et al., 2011). The difference is in the feedback structure: in stochastic packing, full information about each round is revealed before that round.

While we approximate our benchmark $\mathtt{OPT}(\Pi)$ with a linear program optimum, our algorithm and analysis are conceptually very different from the vast literature on approximately solving linear programs, and in particular from LP-based work on bandit problems such as Guha et al. (2010).

In concurrent and independent work, Agrawal and Devanur (2014) study a model for contextual bandits with resource constraints that is incomparable with ours. The dependence of expected rewards and resource consumptions on contexts and arms is more restrictive, namely linear. Whereas the model for resource constraints is more general, e.g., allowing diminishing returns.

Organization of the paper. In the body of the paper we define the main algorithm, prove its correctness, and describe the key steps of regret analysis. The appendices contain a discussion of the application domains (Appendix A), missing proofs from the body of the paper (Appendix B-C), the proof of the lower bound (Appendix D) and a discretization result for contextual dynamic pricing (Appendix E). The state-of-art for RCB and the directions for further work are discussed in Conclusions (Sections 6) and Appendix F.

2. Problem formulation and preliminaries

We consider an online setting where in each round an algorithm observes a context x from a possibly infinite known set of possible contexts X and chooses an action a from a finite known set A. The world then specifies a reward $r \in [0, 1]$ and the resource consumption. There are d resources that can be consumed, and the resource consumption is specified by numbers $c_i \in [0, 1]$ for each resource i. Thus, the world specifies the vector $(r; c_1, \ldots, c_d)$, which we call the *outcome vector*; this vector can depend on the the chosen action a and the round. There is a known hard constraint $B_i \in \mathbb{R}_+$ on the consumption of each resource i; we call it a *budget* for resource i. The algorithm stops at the earliest time τ when any budget constraint is violated; its total reward is the sum of the rewards in all rounds strictly preceding τ. The goal of the algorithm is to maximize the expected total reward.

We are only interested in regret at a specific time T (*time horizon*) which is known to the algorithm. Formally, we model time as a specific resource with budget T and a deterministic consumption of 1 for every action. So $d \geq 2$ is the number of all resources, *including time*. W.l.o.g., $B_i \leq T$ for every resource i.

We assume that an algorithm can choose to skip a round without doing anything. Formally, we posit a *null action*: an action with 0 reward and 0 consumption of all resources except the time. This is for technical convenience, so as to enable Lemma 5.

Stochastic assumptions. We assume that there exists an unknown distribution $D(x, \boldsymbol{r}, \boldsymbol{c_i})$, called the *outcome distribution*, from which each round's observations are created independently and identically, where the vectors are indexed by individual actions. In particular, context x is drawn from the marginal distribution $\mathcal{D}_{\mathtt{X}}(\cdot)$, and the observed reward and resource consumptions for each action a are drawn from the conditional distribution $D(\boldsymbol{r}_a, \boldsymbol{c_{i}}_a|x)$. We assume that the marginal distribution over contexts $D(x)$ is known.

Policy sets and the benchmark. An algorithm is given a finite set Π of *policies* – mappings from contexts to actions. Our benchmark is a hypothetical algorithm that knows the outcome distribution D, and makes optimal decisions given this knowledge. The benchmark is restricted to policies in Π: before each round, it must commit to some policy $\pi \in \Pi$, and then choose action $\pi(x)$ upon arrival of any given context x. The expected total reward of the benchmark is denoted $\mathtt{OPT}(\Pi)$. *Regret* of an algorithm is $\mathtt{OPT}(\Pi)$ minus the algorithm's expected total reward.

Uniform budgets. We say that the budgets are *uniform* if $B_i = B$ for each resource i. Any problem instance can be reduced to one with uniform budgets by dividing all consumption values for every resource i by B_i/B, where $B = \min_i B_i$. (That is tantamount to changing the units in which we measure consumption of resource i.) We assume uniform budgets B from here on.

Notation. Let $r(\pi) = E_{(x,\boldsymbol{r})\sim D}[\boldsymbol{r}_{\pi(x)}]$ and $c_i(\pi) = E_{(x,\boldsymbol{c_i})\sim D}[\boldsymbol{c_i}_{\pi(x)}]$ be the expected per-round reward and the expected per-round consumption of resource i for policy π. Similary, define $r(P) = E_{\pi\sim P}[r(\pi)]$ and $c_i(P) = E_{\pi\sim P}[c_i(\pi)]$ as the natural extension to a distribution P over policies.

The tuple $\mu = (\,(r(\pi); c_1(\pi)\,,\,\ldots\,, c_d(\pi))\,:\,\pi \in \Pi\,)$ is called the *expected-outcomes tuple* .

For a distribution P over policies, let $P(\pi)$ is the probability that P places over policy π. By a slight abuse of notation, let $P(a|x) = \sum_{\pi(x)=a} P(\pi)$ be the probability that P places on action a given context x. Thus, each context x induces a distribution $P(\cdot|x)$ over actions.

2.1. Linear approximation and the benchmark

We set up a linear relaxation that will be crucial throughout the paper. As a by-product, we (effectively) reduce our benchmark $\mathtt{OPT}(\Pi)$ to the best fixed distribution over policies.

A given distribution P over policies defines an algorithm $\mathtt{ALG}_P$: in each round a policy π is sampled independently from P, and the action $a = \pi(x)$ is chosen. The *value* of P is the total reward of this algorithm, in expectation over the outcome distribution.

As the value of P is difficult to characterize exactly, we approximate it (generalizing the approach from (Babaioff et al., 2012; Badanidiyuru et al., 2013a) for the non-contextual version). We use a linear approximation where all rewards and consumptions are deterministic and the time is continuous. Let $r(P, \mu)$ and $c_i(P, \mu)$ be the expected per-round reward and the expected per-round consumption of resource i for policy $\pi \sim P$, given expected-outcomes tuple μ. Then the linear approximation corresponds to the solution of a simple linear program:

$$\begin{array}{llcll} \text{Maximise} & t\, r(P,\mu) & & & \text{in } t \in \mathbb{R} \\ \text{subject to} & t\, c_i(P,\mu) & \leq & B & \text{for each } i \\ & t & \geq & 0. & \end{array} \tag{3}$$

The solution to this LP, which we call the *LP-value* of P, is

$$\texttt{LP}(P,\mu) = r(P,\mu) \;\; \min_i B/c_i(P,\mu). \tag{4}$$

Denote $\texttt{OPT}_{\texttt{LP}} = \sup_P \texttt{LP}(P,\mu)$, where the supremum is over all distributions P over Π.

Lemma 4 $\texttt{OPT}_{\texttt{LP}} \geq \texttt{OPT}(\Pi)$.

Therefore, it suffices to compete against the best fixed distribution over Π, as approximated by $\texttt{OPT}_{\texttt{LP}}$, even though our benchmark $\texttt{OPT}(\Pi)$ allows unrestricted changes over time. Note that proving regret bounds relative to $\texttt{OPT}_{\texttt{LP}}$ rather than to $\texttt{OPT}(\Pi)$ only makes our results stronger.

A distribution P over Π that attains the supremum value $\texttt{OPT}_{\texttt{LP}}$ is called *LP-optimal*. Such P is called *LP-perfect* if furthermore $|\texttt{support}(P)| \leq d$ and $c_i(P,\mu) \leq B/T$ for each resource i. We find it useful to consider LP-perfect distributions throughout the paper.

Lemma 5 *An LP-perfect distribution exists for any instance of* RCB.

Lemma 4 and Lemma 5 are proved for the non-contextual version of RCB in Badanidiyuru et al. (2013a). The general case can be reduced to the non-contextual version via a standard reduction where actions in the new problem correspond to policies in Π in the original problem. For Lemma 5, Badanidiyuru et al. (2013a) obtain an LP-perfect distribution by mixing an LP-optimal distribution with the "null action"; this is why we allow the null action in the setting.

3. The algorithm: `Mixture_Elimination`

The algorithm's goal is to converge on a LP-perfect distribution over policies. The general design principle is to explore as much as possible while avoiding obviously suboptimal decisions.

Overview of the algorithm. In each round t, the following happens.

1. *Compute estimates.* We compute high-confidence estimates for the per-round reward $r(\pi)$ and per-round consumption $c_i(\pi)$, for each policy $\pi \in \Pi$ and each resource i. The collection $\mathcal{I}$ of all expected-outcomes tuple that are consistent with these high-confidence estimates is called the *confidence region*.

2. *Avoid obviously suboptimal decisions.* We prune away all distributions P over policies in Π that are not LP-perfect with high confidence. More precisely, we prune all P that are not LP-perfect for any expected-outcomes tuple in the confidence region $\mathcal{I}$; the remaining distributions are called *potentially LP-perfect*. Let $\mathcal{F}$ be the convex hull of the set of all potentially LP-perfect distributions.

3. *Explore as much as possible.* We choose a distribution $P \in \mathcal{F}$ which is *balanced*, in the sense that no action is starved; see Equation (5) for the precise definition. Note that balanced distributions are typically *not* LP-perfect.

Algorithm 1 `Mixture_Elimination`

Parameters: #actions K, time horizon T, budget B, benchmark set Π, context distribution $\mathcal{D}_{\mathtt{X}}$.
Data structure: "confidence region" $\mathcal{I} \leftarrow \{$all feasible expected-outcomes tuples$\}$.

For each round $t = 1 \ldots T$ **do**
  $\mathbf{\Delta}_t = \{$distributions P over Π: P is LP-perfect for some $\mu \in \mathcal{I}\}$.
  Let $\mathcal{F}_t$ be the convex hull of $\mathbf{\Delta}_t$.
  Let $\alpha_{\pi,t} = \max_{P \in \mathcal{F}_t} P(\pi), \quad \forall \pi \in \Pi$.
  Choose a "balanced" distribution $P_t \in \mathcal{F}_t$: any $P \in \mathcal{F}_t$ such that $\forall \pi \in \Pi$

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}_{\mathtt{X}}} \left[\frac{1}{(1-q_0)\, P(\pi(x)|x) + \frac{q_0}{K}} \right] \leq \frac{2K}{\alpha_{\pi,t}}, \text{ where } q_0 = \min\left(\tfrac{1}{2}, \sqrt{\tfrac{K}{T} \log(K\,T|\Pi|)} \right). \tag{5}$$

  Observe context x_t; **choose** action a_t to "play":
    with probability q_0, draw a_t u.a.r. in A; else, draw $\pi \sim P_t$ and let $a_t = \pi(x_t)$.
  Observe outcome vector $(r, c_1, \ldots, c_d)$.
  Halt if one of the resources is exhausted.
  Eliminate expected-outcomes tuples from $\mathcal{I}$ that violate equations (6-7)

4. *Select an action.* We choose policy $\pi \in \Pi$ independently from P. Given context x, the action a is chosen as $a = \pi(x)$. The algorithm adds some random noise: with probability q_0, the action a is instead chosen uniformly at random, for some parameter q_0.

The algorithm halts as soon as the time horizon is met, or one of the resources is exhausted.

The pseudocode can be found in Algorithm 1.

Some details. After each round t, we estimate the per-round consumption $c_i(\pi)$ and the per-round reward $r(\pi)$, for each policy $\pi \in \Pi$ and each resource i, using the following unbiased estimators:

$$\widetilde{c}_i(\pi) = \frac{c_i\ \mathbf{1}_{\{a=\pi(x)\}}}{P[a = \pi(x)\,|\,x]} \text{ and } \widetilde{r}(\pi) = \frac{r\ \mathbf{1}_{\{a=\pi(x)\}}}{P[a = \pi(x)\,|\,x]}.$$

The corresponding time-averages up to round t are denoted

$$\hat{c}_{t,i}(\pi) = \tfrac{1}{t-1} \sum_{s=1}^{t-1} \widetilde{c}_{s,i}(\pi) \text{ and } \hat{r}_t(\pi) = \tfrac{1}{t-1} \sum_{s=1}^{t-1} \widetilde{r}_s(\pi).$$

We show that with high probability these time-averages are close to their respective expectations. To express the confidence term in a more lucid way, we use the following shorthand, called *confidence radius*: $\mathtt{rad}_t(\nu) = \sqrt{C_{\mathtt{rad}}\,\nu/t}$, where $C_{\mathtt{rad}} = \Theta(\log(d\,T\,|\Pi|))$ is a parameter which we will fix later. We show that w.h.p. the following holds:

$$|r(\pi) - \hat{r}_t(\pi)| \leq \mathtt{rad}_t\left(K/\alpha_{\pi,t}\right), \tag{6}$$

$$|c_i(\pi) - \hat{c}_{t,i}(\pi)| \leq \mathtt{rad}_t\left(K/\alpha_{\pi,t}\right) \quad \text{for all } i. \tag{7}$$

(Here $\alpha_{\pi,t} = \max_{P \in \mathcal{F}_t} P(\pi)$, as in Algorithm 1.)

4. Correctness of the algorithm

We need to prove that in each round t, some $P \in \mathcal{F}_t$ satisfies (5), and Equations (6-7) hold for all policies $\pi \in \Pi$ with high probability.

Notation. Recall that P_t is the distribution over Π chosen in round t of the algorithm, and q_0 is the noise probability. The "noisy version" of P_t is defined as

$$P'_t(a|x) = (1 - q_0)\, P_t(a|x) + q_0/K \qquad (\forall x \in X, a \in A).$$

Then action a_t in round t is drawn from distribution $P'_t(\cdot|x_t)$.

Lemma 6 *In each round t, some $P \in \mathcal{F}_t$ satisfies (5).*

Proof First we prove that $\mathcal{F}_t$ is compact; here each distribution over Π is interpreted as a $|\Pi|$-dimensional vector, and compactness is w.r.t. the Borel topology on $\mathbb{R}^{|\Pi|}$. This can be proved via standard real analysis arguments; we provide a self-contained proof in Appendix B.

In what follows we extend the minimax argument from Dudik et al. (2011). Our proof works for any $q_0 \in [0, \frac{1}{2}]$ and any compact and convex set $\mathcal{F} \subset \mathcal{F}_\Pi$.

Denote $\alpha_\pi = \max_{P \in \mathcal{F}} P(\pi)$, for each $\pi \in \Pi$. Let $\mathcal{F}_\Pi$ be the set of all distributions over Π.

Equation (5) holds for a given $P \in \mathcal{F}$ if and only if for every distribution $Z \in \mathcal{F}_\Pi$ we have that

$$f(P, Z) \triangleq \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X} \mathop{\mathbb{E}}_{\pi \sim Z} \left[\frac{\alpha_\pi}{P'(\pi(x)|x)} \right] \leq 2K,$$

where P' is the noisy version of P. It suffices to show that

$$\min_{P \in \mathcal{F}} \max_{Z \in \mathcal{F}_\Pi} f(P, Z) \leq 2K. \tag{8}$$

We use a min-max argument: noting that f is a convex function of P and a concave function of Z, by the Sion's minimax theorem (Sion, 1958) we have that

$$\min_{P \in \mathcal{F}} \max_{Z \in \mathcal{F}_\Pi} f(P, Z) = \max_{Z \in \mathcal{F}_\Pi} \min_{P \in \mathcal{F}} f(P, Z). \tag{9}$$

For each policy $\pi \in \Pi$, let $\beta_\pi \in \operatorname{argmax}_{\beta \in \mathcal{F}} \beta(\pi)$ be a distribution which maximizes the probability of selecting π. Such distribution exists because $\beta \mapsto \beta(\pi)$ is a continuous function on a compact set $\mathcal{F}$. Recall that $\alpha_\pi = \beta_\pi(\pi)$.

Given any $Z \in \mathcal{F}_\Pi$, define distribution $P_Z \in \mathcal{F}_\Pi$ by $P_Z(\pi) = \sum_{\phi \in \Pi} Z(\phi)\, \beta_\phi(\pi)$. Note that P_Z is a convex combination of distributions in $\mathcal{F}$. Since $\mathcal{F}$ is convex, it follows that $P_Z \in \mathcal{F}$. Also, note that $P_Z(a|x) \geq \sum_{\pi \in \Pi:\, \pi(x)=a} Z(\pi)\, \alpha_\pi$. Letting P'_Z be the noisy version of P_Z, we have:

$$\begin{aligned}
\min_{P \in \mathcal{F}} f(P, Z) &\leq f(P_Z, Z) = \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X} \left[\sum_\pi \frac{Z(\pi)\, \alpha_\pi}{P'_Z(\pi(x)|x)} \right] \\
&= \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X} \left[\sum_{a \in A} \sum_{\pi \in \Pi:\, \pi(x)=a} \frac{Z(\pi)\, \alpha_\pi}{P'_Z(a|x)} \right] = \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X} \left[\sum_{a \in X} \frac{\sum_{\pi \in \Pi:\, \pi(x)=a} Z(\pi)\, \alpha_\pi}{(1-q_0) P_Z(a|x) + q_0/K} \right] \\
&\leq \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X} \left[\sum_{a \in X} \frac{1}{1-q_0} \right] = \frac{K}{1-q_0} \leq 2K.
\end{aligned}$$

Thus, by Equation (9) we obtain Equation (8). ∎

To analyze Equations (6-7), we will use Bernstein's inequality for martingales (Freedman, 1975), via the following formulation from Bubeck and Slivkins (2012):

Lemma 7 *Let $\mathcal{G}_0 \subseteq \mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_n$ be a filtration, and $X_1, \ldots, X_n$ be real random variables such that X_t is $\mathcal{G}_t$-measurable, $\mathbb{E}(X_t|\mathcal{G}_{t-1}) = 0$ and $|X_t| \leq b$ for some $b > 0$. Let $V_n = \sum_{t=1}^n \mathbb{E}(X_t^2|\mathcal{G}_{t-1})$. Then with probability at least $1-\delta$ it holds that*

$$\textstyle\sum_{t=1}^n X_t \leq \sqrt{4V_n \log(n\delta^{-1}) + 5b^2 \log^2(n\delta^{-1})}.$$

Lemma 8 *With probability at least $1-\frac{1}{T}$, Equations (6-7) hold for all rounds t and policies $\pi \in \Pi$.*

Proof Let us prove Equation (6). (The proof of (7) is similar.) Fix round t and policy $\pi \in \Pi$. We bound the conditional variance of the estimators $\widetilde{r}_t(\pi)$. Specifically, let $\mathcal{G}_t$ be the σ-algebra induced by all events up to (but not including) round t. Then

$$\mathbb{E}\left[\widetilde{r}_t(\pi)^2 \,|\, \mathcal{G}_t\right] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathtt{X}},\, a\sim P'_t}\left[\frac{r_t^2\, \mathbf{1}_{\{\pi(x)=a\}}}{P'_t(a|x)^2}\right] \leq \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathtt{X}}}\left[\frac{1}{P'_t(\pi(x)|x)}\right] \leq \frac{2K}{\alpha_{\pi,t}}.$$

The last inequality holds by the algorithm's choice of distribution P_t. Since the confidence region $\mathcal{I}$ in our algorithm is non-increasing over time, it follows that $\alpha_{\pi,t}$ is non-increasing in t, too. We conclude that $\mathtt{Var}\left[\widetilde{r}_s(\pi) \,|\, \mathcal{G}_s\right] \leq 2K/\alpha_{\pi,t}$ for each round $s \leq t$. Therefore, noting that $\widetilde{r}_t(\pi) \leq 1/P'(\pi(x_t)|x_t) \leq K/q_0$, we obtain Equation (6) by applying Lemma 7 with $X_t = \widetilde{r}_t(\pi) - r(\pi)$. □

5. Regret analysis: proof of Theorem 1

We provide the key steps of the proof; the details can be found in Section C.

Let $\mathcal{I}_t$ and $\mathbf{\Delta}_t$ be, resp., the confidence region $\mathcal{I}$ and the set $\mathbf{\Delta}$ of potentially LP-perfect distributions computed in round t. Let $\mathtt{Conv}(\mathbf{\Delta}_t)$ be the convex hull of $\mathbf{\Delta}_t$.

First we bound the deviations within the confidence region.

Lemma 9 *For any two expected-outcomes tuples $\mu', \mu'' \in \mathcal{I}_t$ and a distribution $P \in \mathtt{Conv}(\mathbf{\Delta}_t)$:*

$$|c_i(P,\mu') - c_i(P,\mu'')| \leq \mathtt{rad}_t\,(dK) \qquad \textit{for each resource } i \tag{10}$$

$$|r(P,\mu') - r(P,\mu'')| \leq \mathtt{rad}_t\,(dK) \tag{11}$$

Proof Let us prove Equation (11). (Equation (10) is proved similarly.) By definition of $\mathcal{I}_t$:

$$\begin{aligned}|r(P,\mu') - r(P,\mu'')| &\leq \textstyle\sum_{\pi\in\Pi} P(\pi)\; |r(\pi,\mu') - r(\pi,\mu'')| \\ &\leq \textstyle\sum_{\pi\in\Pi} P(\pi)\; \mathtt{rad}_t\,(K/\alpha_{\pi,t})\,.\end{aligned}$$

It remains to prove that the right-hand side is at most $\mathtt{rad}_t(dK)$. By linearity, it suffices to prove this for $P \in \mathbf{\Delta}_t$. So let us assume $P \in \mathbf{\Delta}_t$ from here on. Recall that $|\mathtt{support}(P)| \leq d$ since P is LP-perfect, and $P(\pi) \leq \alpha_{\pi,t}$ for any policy $\pi \in \Pi$. Therefore:

$$\begin{aligned}\textstyle\sum_{\pi\in\Pi} P(\pi)\; \mathtt{rad}_t\,(K/\alpha_{\pi,t}) &\leq \textstyle\sum_{\pi\in\Pi}\; \mathtt{rad}_t\,(K P(\pi)) \\ &\leq \mathtt{rad}_t\left(dK \textstyle\sum_{\pi\in\Pi} P(\pi)\right) = \mathtt{rad}_t\,(dK)\,.\end{aligned}$$

□

Using Lemma 9 and a long computation (fleshed out in Section C), we prove the following.

Lemma 10 *For any two expected-outcomes tuples* $\mu', \mu'' \in \mathcal{I}_t$ *and a distribution* $P \in \texttt{Conv}(\mathbf{\Delta}_t)$*:*

$$\texttt{LP}(P, \mu') - \texttt{LP}(P, \mu'') \leq (\tfrac{1}{B}\,\texttt{LP}(P, \mu') + 2) \cdot T \cdot \texttt{rad}_t(dK).$$

Let $\texttt{REW}_t$ and $C_{t,i}$ be, respectively, the (realized) total reward and average consumption of resource i up to and including round t. Recall that P'_t is the noisy version of distribution P_t chosen by the algorithm in round t. Given P_t, the expected revenue and resource-i consumption in round t is, respectively, $r(P'_t, \mu)$ and $c_i(P'_t, \mu)$. Denote $\overline{r}_t = \frac{1}{t}\sum_{i=1}^{t} r(P'_t, \mu)$ and $\overline{c}_{i,t} = \frac{1}{t}\sum_{i=1}^{t} c_i(P'_t, \mu)$.

Analysis of a clean execution. Henceforth, without further notice, we assume a ***clean execution*** where several high-probability conditions are satisfied. Formally, the algorithm's execution is *clean* if in each round t Equations (6-7) are satisfied, and moreover $\min\left(|\frac{1}{t}\,\texttt{REW}_t - \overline{r}_t|,\ |C_{t,i} - \overline{c}_{t,i}|\right) \leq \texttt{rad}_t(1)$.

In particular, the set $\mathbf{\Delta}_t$ of potentially LP-perfect distributions indeed contains a LP-perfect distribution. By Lemma 8 and Azuma-Hoeffding Inequality, clean execution happens with probability at least $1 - \frac{1}{T}$. Thus, it suffices to lower-bound the total reward $\texttt{REW}_T$ for a clean execution.

Lemma 11 *For any distribution* $P' \in \texttt{Conv}(\mathbf{\Delta}_t)$ *and any expected-outcomes tuple* $\mu \in \mathcal{I}_t$*,*

$$\min_{P \in \mathbf{\Delta}_t} \texttt{LP}(P, \mu) \leq \texttt{LP}(P', \mu) \leq \max_{P \in \mathbf{\Delta}_t} \texttt{LP}(P, \mu). \tag{12}$$

The following lemma captures a crucial argument. Denote

$$\Phi_t = \left(2 + \tfrac{1}{B}\left[\max_{P \in \mathcal{F}_\Pi,\, \mu \in \mathcal{I}_t} \texttt{LP}(P, \mu)\right]\right) \cdot T \cdot \texttt{rad}_t(dK)$$

Lemma 12 *For any expected-outcomes tuple* $\mu^*, \mu^{**} \in \mathcal{I}_t$ *and distributions* $P', P'' \in \texttt{Conv}(\mathbf{\Delta}_t)$*:*

$$|\texttt{LP}(P', \mu^*) - \texttt{LP}(P'', \mu^{**})| \leq 3\Phi_t. \tag{13}$$

Proof Assume $P', P'' \in \mathbf{\Delta}_t$. In particular, P', P'' are LP-perfect for some expected-outcomes tuples $\mu', \mu'' \in \mathcal{I}_t$, resp. Also, some distribution $P^* \in \mathbf{\Delta}_t$ is LP-perfect for μ^* (by Lemma 5). Therefore:

$$\begin{aligned}
\texttt{LP}(P', \mu^*) &\geq \texttt{LP}(P', \mu') - \Phi_t && \text{(by Lemma 10: } P = P') \\
&\geq \texttt{LP}(P^*, \mu') - \Phi_t \\
&\geq \texttt{LP}(P^*, \mu^*) - 2\Phi_t && \text{(by Lemma 10: } P = P^*) \\
&\geq \texttt{LP}(P'', \mu^*) - 2\Phi_t.
\end{aligned}$$

We proved Equation (13) for $P', P'' \in \mathbf{\Delta}_t$. Thus:

$$\max_{P \in \mathbf{\Delta}_t} \texttt{LP}(P, \mu^*) - \min_{P \in \mathbf{\Delta}_t} \texttt{LP}(P, \mu^*) \leq 2\Phi_t. \tag{14}$$

Next we generalize to $P', P'' \in \texttt{Conv}(\mathbf{\Delta}_t)$.

$$\begin{aligned}
\texttt{LP}(P', \mu^*) &\geq \min_{P \in \mathbf{\Delta}_t} \texttt{LP}(P, \mu^*) && \text{(by Lemma 11)} \\
&\geq \max_{P \in \mathbf{\Delta}_t} \texttt{LP}(P, \mu^*) - 2\Phi_t && \text{(by Equation (14))} \\
&\geq \texttt{LP}(P'', \mu^*) - 2\Phi_t && \text{(by Lemma 11).}
\end{aligned}$$

We proved Equation (13) for $\mu^* = \mu^{**}$. We obtain the general case by plugging in Lemma 10. □

In the remainder of the proof, we upper-bound Φ_t in terms of

$$\Psi_t = (2 + \tfrac{1}{B}\,\texttt{OPT}_{\texttt{LP}}) \cdot T \cdot \texttt{rad}_t(dK).$$

Corollary 13 $\Phi_t \leq 2\Psi_t$, *assuming that* $B \geq 6 \cdot T \cdot \texttt{rad}_t(dK)$.

Proof Follows from Lemma 12 via a simple computation, see Section C. □

Corollary 14 $\texttt{LP}(P_t, \mu) \geq \texttt{OPT}_{\texttt{LP}} - 12\,\Psi_t$, *where* μ *is the actual expected-outcomes tuple.*

Proof Follows from Lemma 12 and Corollary 13, observing that $P_t \in \texttt{Conv}(\mathbf{\Delta}_t)$ and $\texttt{OPT}_{\texttt{LP}} = \texttt{LP}(P^*, \mu)$ for some $P^* \in \mathbf{\Delta}_t$. □

In the remainder of the proof (which is fleshed out in Section C) we build on the above lemmas and corollaries to prove the following sequence of claims:

$$\begin{aligned} \texttt{REW}_t &\geq \frac{t}{T}\,(\texttt{OPT}_{\texttt{LP}} - O(\Psi_t)) \\ C_{t,i} &\leq B/T + O(\texttt{rad}_t(dK)) \\ \texttt{REW}_T &\geq \texttt{OPT}_{\texttt{LP}} - O(\Psi_T). \end{aligned} \tag{15}$$

To complete the proof of Theorem 1, we re-write the last equation as $\texttt{REW}_T \geq f(\texttt{OPT}_{\texttt{LP}})$ for an appropriate function $f()$, and observe that $f(\texttt{OPT}_{\texttt{LP}}) \geq f(\texttt{OPT})$ because function $f()$ is increasing.

6. Conclusions and open questions

We define a very general setting for contextual bandits with resource constraints (denoted RCB). We design an algorithm for this problem, and derive a regret bound which achieves the optimal root-T scaling in terms of the time horizon T, and the optimal $\sqrt{\log|\Pi|}$ scaling in terms of the policy set Π. Further, we consider discretization issues, and derive a specific corollary for contextual dynamic pricing with a single product; we obtain a regret bound that applies to an arbitrary policy set Π and is near-optimal in the regime $B \geq \Omega(T)$. Finally, we derive a partial lower bound which establishes a stark difference from the non-contextual version. These results set the stage for further study of RCB, as discussed below.

The main open question is to combine provable regret bounds and a computationally efficient (CE) implementation. While we focused on the statistical properties, we believe our techniques are unlikely to lead to CE implementations. Achieving near-optimal regret bounds in a CE way has been a major open question for contextual bandits with policy sets (without resource constraints). Very recently, this question has been resolved in the positive in a simultaneous and independent work (Agarwal et al., 2013). One can hope to achieve the corresponding advance on RCB, perhaps by building on the techniques from Agarwal et al. (2013). As a stepping stone, one can target CE algorithms with weaker regret bounds, such as $T^{2/3}$ rather than $\sqrt{T}$ dependence on time, perhaps building on the corresponding results for the non-constrained version (Langford and Zhang, 2007).

Computational issues aside, several open questions concern the regret bounds for RCB and extensions thereof, see Appendix F.

References

Alekh Agarwal, Miroslav Dudik, Satyen Kale, and John Langford. Contextual bandit learning under the realizability assumption. In *15th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2012.

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)*, 2013.

Shipra Agrawal and Nikhil R. Devanur. Bandits with concave rewards and convex knapsacks. In *15th*, 2014.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. of Machine Learning Research (JMLR)*, 3:397–422, 2002. Preliminary version in *41st IEEE FOCS*, 2000.

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

Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. In *13th ACM Conf. on Electronic Commerce (EC)*, 2012.

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

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

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. A technical report on `arxiv.org`., May 2013b.

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

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

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Efficient optimal leanring for contextual bandits. In *14th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011.

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.

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

Nikhil Devanur and Vijay Vazirani. The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results. In *36th ACM Symp. on Theory of Computing (STOC)*, 2004.

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

Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *12th ACM Conf. on Electronic Commerce (EC)*, pages 29–38, 2011.

Miroslav Dudik, 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.

D. A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3:100–118, 1975.

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

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

Sudipta Guha, Kamesh Munagala, and Peng Shi. Approximation algorithms for restless bandit problems., 2010. Combined final version of papers in *IEEE FOCS 2007* and *ACM-SIAM SODA 2009*.

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

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

Robert Kleinberg and Aleksandrs Slivkins. Sharp Dichotomies for Regret Minimization in Metric Spaces. In *21st ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2010.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-Armed Bandits in Metric Spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008.

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

Brendan McMahan and Matthew Streeter. Tighter bounds for multi-armed bandits with expert advice. In *22nd Conf. on Learning Theory (COLT)*, 2009.

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

Maurice Sion. On general minimax theorems. *Pac. J. Math.*, 8:171176, 1958.

Aleksandrs Slivkins. Contextual Bandits with Similarity Information. In *24th Conf. on Learning Theory (COLT)*, 2011. To appear in J. of Machine Learning Research (JMLR), 2014.

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

Aleksandrs Slivkins and Jennifer Wortman Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges. *SIGecom Exchanges*, 12(2), December 2013. Position Paper and survey.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285294, 1933.

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

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

Appendix A: RCB: applications and special cases

In this section we discuss the application domains of resource-constrained (contextual) bandits in more detail. We focus on the three main application domains: dynamic pricing, dynamic procurement, and dynamic ad allocation. A more extensive discussion of these and other application domains (in the non-contextual version) can be found in Badanidiyuru et al. (2013a,b).

Dynamic pricing with limited supply. The algorithm is a monopolistic seller with a limited inventory. In the basic version, there is a limited supply of identical items. In each round, a new customer arrives, the algorithm picks a price, and offers one item for sale at this price. The customer then either buys the item at this price, or rejects the offer and leaves. The "context" represents the available information about the current customer, such as demographics, location, etc. The probability of selling at a given price for a given context (a.k.a. the *demand distribution*) is fixed over time, but not known to the algorithm. The algorithm optimizes the revenue; it does not derive any utility from the left-over items.

We represent this problem as an instance of RCB as follows. "Actions" are the possible prices, and the "resource constraint" is the number of items. In each round, the outcome vector is a pair (reward, items sold); if the offered price is p, the outcome vector is $(p, 1)$ if there is a sale, and $(0, 0)$ otherwise.

Many generalizations of dynamic pricing have been studied in the literature. In particular, RCB subsumes a number of extensions. First, an algorithm can sell multiple items to the same customer, possibly with volume discounts or surcharges. Second, an algorithm can have multiple products for sale, with limited inventory of each. Third, it may be advantageous to offer bundles consisting of different products, possibly with non-additive pricing (mirroring the non-additive valuations of the customers).

Dynamic procurement on a budget. The algorithm is a monopolistic buyer with a limited budget. The basic version is as follows. In each round, a new customer arrives, the algorithm picks a price, and offers to buy one item at this price. Then the customer either accepts the offer and sells the item at this price, or rejects the offer and leaves. The "context" is the available information on the current customer. The probability of buying at a given price for a given context (a.k.a. the "supply distribution") is fixed over time, but not known to the algorithm. The algorithm maximizes the number of items bought; it has no utility for the left-over money.

An alternative interpretation is that the algorithm is a contractor which hires workers to perform tasks, e.g. in a crowdsourcing market. In each round, a new worker arrives, the algorithm picks a price, and offers the worker to perform one task for this price; the worker then either accepts and performs the task at this price, or rejects and leaves. The relevant "context" for a worker in a crowdsourcing market may include, for example, age, location, language, and task preferences.

Here, "actions" correspond to the possible prices, and the "resource constraint" is the buyer's budget. In each round, the outcome vector is a pair (items bought, money spent); if the offered price is p, then the outcome vector is $(1, p)$ if the offer is accepted, and $(0, 0)$ otherwise.

Dynamic procurement is a rich problem space, both for buying items and for hiring workers (see (Slivkins and Vaughan, 2013) for a discussion of the application to crowdsourcing markets). In particular, RCB subsumes a number of extensions of this basic setting. First, the algorithm may offer several tasks to the same worker, possibly at a discount. Second, there may be multiple types of tasks, each having a different value for the contractor; moreover, there may be additional budget constraints on each task type, or on various *subsets* of task types. Third, a given worker can be offered a bundle of tasks, consisting of tasks of multiple types, possibly with non-additive pricing. Fourth, there is a way to model the presence of competition (other contractors).

Dynamic ad allocation with budgets. The algorithm is an advertising platform. In the basic version, there is a fixed collection of ads to choose from. In each round, a user arrives, and the algorithm chooses one ad to display to this user. The user either clicks on this ad, or leaves without clicking. The algorithm receives a payment if and only if the ad is clicked; the payment for a given ad is fixed over time and known to the algorithm. The "context" is the available information about the user and the page on which the ad is displayed. The click probability for a given ad and a given context is constant over time, but not known to the algorithm.

Each ad belongs to some advertiser (who is the one paying the algorithm when this ad is clicked). Each advertiser may own multiple ads, and has a budget constraint: a maximal amount of money that can be spent on all his ads. Moreover, an advertiser may specify additional budget constraints on various subsets of the ads. The algorithm maximizes its revenue; it derives no utility from the left-over budgets.

Here, "actions" correspond to ads, and each budget corresponds to a separate resource. In a round when the chosen ad a is clicked, the reward is the corresponding payment v, and the resource consumption is v for each budget that involves a, and 0 for all other budgets. If the ad is not clicked, the reward and the consumption of each resource is 0.

RCB also subsumes more advanced versions in which multiple non-zero outcomes are possible in each round. For example, the ad platform may record what happens *after* the click, e.g. the time spent on the page linked from the ad and whether this interaction has resulted in a sale.

Appendix B: Compactness of $\mathcal{F}_t$

Recall that $\mathbf{\Delta}_t$ is the set of distributions over Π that is computed by our algorithm in each round t, and $\mathcal{F}_t = \texttt{Conv}(\mathbf{\Delta}_t)$ is the convex hull of $\mathbf{\Delta}_t$. In this appendix we prove that $\mathcal{F}_t$ is compact for each t. Here each distribution over Π is interpreted as a $|\Pi|$-dimensional vector, and compactness is with respect to the Borel topology on $\mathbb{R}^{|\Pi|}$.

Lemma 15 *$\mathcal{F}_t$ is compact for each t, relative to the Borel topology on $\mathbb{R}^{|\Pi|}$.*

Since a convex hull of compact set is compact, it suffices to prove that $\mathbf{\Delta}_t$ is compact, i.e. that it is closed and bounded. Each distribution is contained in a unit cube, hence bounded. Thus, it suffices to prove that $\mathbf{\Delta}_t$ is a closed subset of $\mathbb{R}^{|\Pi|}$.

We use the following general lemma, which can be proved via standard real analysis arguments.

Lemma 16 *Consider the following setup:*

- *$\mathcal{X}, \mathcal{Y}$ are compact subsets of finite-dimensional real spaces $\mathbb{R}^{d_X}$ and $\mathbb{R}^{d_Y}$, respectively.*
- *Functions $f, g_1, \ldots, g_d : \mathcal{X} \times \mathcal{Y} \to [0,1]$ are continuous w.r.t. product topology on $\mathcal{X} \times \mathcal{Y}$.*
- *$H(y) = \{x \in \mathcal{X} : f(x,y) = \sup_{x' \in \mathcal{X}} f(x',y)$ and $g(x,y) \leq 0\}$, for each $y \in \mathcal{Y}$.*

Then $H(\mathcal{Y}) = \cup_{y \in \mathcal{Y}} H(y)$ is a closed subset of $\mathbb{R}^{d_X}$.

We apply this lemma to prove that $\mathbf{\Delta}_t$ is a closed subset of $\mathbb{R}^{|\Pi|}$. Specifically, we take $\mathcal{X}$ to be the set of all distributions over policies with support at most d, and $\mathcal{Y}$ be the confidence region in round t of the algorithm. It is easy to see that both sets are closed and bounded by definition, therefore compact. Further, for each distribution $P \in \mathcal{X}$, each expected-outcomes tuple $\mu \in \mathcal{Y}$, and each resource i we define $f(P,\mu)$ to be the corresponding LP-value, and $g_i(P,\mu) = c_i(P,\mu) - B/T$. Then $P \in H(\mu)$ if and only if P is an LP-perfect distribution with respect to μ, and $H(\mathcal{Y}) = \mathbf{\Delta}_t$.

This completes the proof of Lemma 15.

B.1. Proof of Lemma 16

Suppose $x^* \in \mathbb{R}^{d_X}$ is an accumulation point of $H(\mathcal{Y})$, i.e. there is a sequence $x_1, x_2, \ldots \in H(\mathcal{Y})$ such that $x_j \to x^*$. Note that $x^* \in \mathcal{X}$ since $\mathcal{X}$ is closed. We need to prove that $x^* \in H(\mathcal{Y})$.

For each j, there is $y_j \in \mathcal{Y}$ such that $x_j \in H(y_j)$. Recall that $\mathcal{Y}$ is closed and bounded. Since $\mathcal{Y}$ is bounded, sequence $\{y_j\}_{j \in \mathbb{N}}$ contains a convergent subsequence. Since $\mathcal{Y}$ is closed, $\mathcal{Y}$ contains the limit of this subsequence. From here on, let us focus on this convergent subsequence.

Thus, we have proved that there exists a sequence of pairs $\{(x_j, y_j)\}_{j \in \mathbb{N}}$ such that

- $x_j \in H(y_j)$ and $y_j \in \mathcal{Y}$ for all $j \in \mathbb{N}$,
- $x_j \to x^* \in \mathcal{X}$ and $y_j \to y^* \in \mathcal{Y}$.

We will prove that $x^* \in H(y^*)$. For that, we need to prove two things: (i) $g_i(x^*, y^*) \leq 0$ for each i, and (ii) $f(x^*, y^*) = \sup_{x \in \mathcal{X}} f(x, y^*)$.

First, $g_i(x^*, y^*) \leq 0$ for each i because $g_i(x^*, y^*) = \lim_j g_i(x_j, y_j) \leq 0$ by continuity of g_i.

Second, we claim that $f(x^*, y^*) = \sup_{x \in \mathcal{X}} f(x, y^*)$. For the sake of contradiction, suppose $\epsilon \triangleq \sup_{x \in \mathcal{X}} f(x, y^*) - f(x^*, y^*) > 0$. By continuity of f, the following holds:

- $f(x^{**}, y^*) = \sup_{x \in \mathcal{X}} f(x, y^*)$ for some $x^{**} \in \mathcal{X}$.
- there exists an open neighborhood S of (x^*, y^*) on which $|f(x,y) - f(x^*, y^*)| < \epsilon/4$.

- there is an open neighborhood S' of (x^{**}, y^*) on which $|f(x,y) - f(x^{**}, y^*)| < \epsilon/4$.

In particular, there are open balls $B_{\mathtt{X}}, B'_{\mathtt{X}} \subset \mathcal{X}$ and $B_{\mathtt{Y}} \subset \mathcal{Y}$ such that $(x^*, y^*) \in B_{\mathtt{X}} \times B_{\mathtt{Y}} \subset S$ and $(x^{**}, y^*) \in B'_{\mathtt{X}} \times B_{\mathtt{Y}} \subset S'$. For a sufficiently large j it holds that $x_j \in B_{\mathtt{X}}$ and $y_j \in B_{\mathtt{Y}}$. It follows that $(x_j, y_j) \in S$ and $(x^{**}, y_j) \in S'$. Therefore:

$$\begin{aligned} f(x^*, y^*) &> f(x_j, y_j) - \epsilon/4 && \text{(since } (x_j, y_j) \in S) \\ &\geq f(x^{**}, y_j) - \epsilon/4 && \text{(using the optimality of } x_j) \\ &> f(x^{**}, y^*) - \epsilon/2 && \text{(since } (x^{**}, y_j) \in S'). \end{aligned}$$

We obtain a contradiction which completes the proof.

Appendix C: Regret analysis: details for the proof of Theorem 1

C.1. Proof of Lemma 10

We restate the lemma for convenience.

Lemma For any two expected-outcomes tuples $\mu', \mu'' \in \mathcal{I}_t$ and a distribution $P \in \mathtt{Conv}(\mathbf{\Delta}_t)$:

$$\mathtt{LP}(P, \mu') - \mathtt{LP}(P, \mu'') \leq (\tfrac{1}{B}\, \mathtt{LP}(P, \mu') + 2) \cdot T \cdot \mathtt{rad}_t(dK).$$

Proof For brevity, we will denote:

$$\begin{aligned} \mathtt{LP}' &= \mathtt{LP}(P, \mu') &\quad \text{and} \quad \mathtt{LP}'' &= \mathtt{LP}(P, \mu'') \\ r' &= r(P, \mu') &\quad \text{and} \quad r'' &= r(P, \mu'') \\ c'_i &= c_i(P, \mu') &\quad \text{and} \quad c''_i &= c_i(P, \mu''). \end{aligned}$$

By symmetry, it suffices to prove the upper bound for $\mathtt{LP}' - \mathtt{LP}''$. Henceforth, assume $\mathtt{LP}' > \mathtt{LP}''$.

We consider two cases, depending on whether

$$T \leq B/c''_i \quad \text{for all resources } i. \tag{16}$$

Case 1. Assume Equation (16) holds. Then $\mathtt{LP}'' = T\, r''$. Therefore by Lemma 9

$$\mathtt{LP}' - \mathtt{LP}'' \leq T\, r' - T\, r'' \leq T\, \mathtt{rad}_t(dK).$$

Case 2. Assume Equation (16) fails. Then $\mathtt{LP}'' = B\, r''/c''_i$ for some resource i. We consider two subcases, depending on whether

$$T \leq B/c'_j \quad \text{for all resources } j. \tag{17}$$

Subcase 1. Assume Equation (17) holds. Then:

$$\mathtt{LP}' = T\, r' \tag{18}$$

$$\mathtt{LP}'' \leq T \cdot \min(r',\ r'') \leq \mathtt{LP}' \tag{19}$$

Equation (19) follows from (18) and $\mathtt{LP}' > \mathtt{LP}''$.

For $\delta \in [0, c''_i)$, define

$$\begin{aligned} r(\delta) &= r'' + \delta \\ c_i(\delta) &= c''_i - \delta \\ f(\delta) &= B\, r(\delta)/c_i(\delta). \end{aligned}$$

Then $f()$ is monotonically and continuously increasing function, with $f(\delta) \to \infty$ as $\delta \to c''_i$. For convenience, define $f(c''_i) = \infty$.

Let $\delta_0 = \min(c''_i,\, \texttt{rad}_t(dK))$. By Lemma 9, we have $f(\delta_0) \geq Br'/c'_i$. Therefore:

$$f(0) = \texttt{LP}'' < \texttt{LP}' \leq Br'/c'_i \leq f(\delta_0).$$

Thus, by Equation (19), we can fix $\delta \in [0, \delta_0)$ such that $f(\delta) = T \cdot \min(r',\; r'')$.

$$\begin{aligned} \texttt{LP}'' &= B\, \frac{r''}{c''_i} = B\, \frac{r(\delta) - \delta}{c_i(\delta) + \delta} \\ &\geq B\, \frac{r(\delta) - \delta}{c_i(\delta)} \left(1 - \frac{\delta}{c_i(\delta)}\right). \\ f(\delta) - \texttt{LP}'' &\leq \frac{B}{c_i(\delta)}\delta + B\frac{r(\delta)}{c_i(\delta)^2}\delta \\ &= \left(1 + \frac{r(\delta)}{c_i(\delta)}\right) \frac{B\, \delta}{c_i(\delta)} \\ &= \left(1 + \frac{f(\delta)}{B}\right) \frac{f(\delta)\, \delta}{r(\delta)} \\ &\leq \left(1 + \frac{T\, r'}{B}\right) \frac{T\, r''\, \delta}{r(\delta)} \\ &\leq \left(1 + \frac{\texttt{LP}'}{B}\right) T\, \delta \\ &\leq \big(\texttt{LP}'/B + 1\big) \cdot T \cdot \texttt{rad}_t(dK). \\ \texttt{LP}' - f(\delta) &= T\, r' - T\, \min(r, r') \\ &\leq T \cdot \texttt{rad}_t(dK) \\ \texttt{LP}' - \texttt{LP}'' &= \big(\texttt{LP}' - f(\delta)\big) + \big(f(\delta) - \texttt{LP}''\big) \\ &\leq \big(\texttt{LP}'/B + 2\big) \cdot T \cdot \texttt{rad}_t(dK). \end{aligned}$$

Subcase 2. Assume Equation (17) fails. Then $\texttt{LP}' = B\, r'/c'_j$ for some resource j. Note that $c'_i \leq c'_j$ and $c''_j \leq c''_i$ by the choice of i and j.

From these inequalities and Lemma 9 we obtain $c''_i \leq c'_j + \mathtt{rad}_t(dK)$. Therefore,

$$\begin{aligned} B\frac{r''}{c''_i} &\geq B\,\frac{r' - \mathtt{rad}_t(dK)}{c'_j + \mathtt{rad}_t(dK)} \qquad \text{(by Lemma 9)} \\ &\geq B\,\frac{r' - \mathtt{rad}_t(dK)}{c'_j}\left(1 - \frac{\mathtt{rad}_t(dK)}{c'_j}\right). \\ \mathtt{LP}' - \mathtt{LP}'' &= B\frac{r'}{c'_j} - B\frac{r''}{c''_i} \\ &\leq \left(\frac{B}{c'_j} + B\frac{r'}{(c'_j)^2}\right)\mathtt{rad}_t(dK) \\ &\leq \left(T + \mathtt{LP}'\frac{T}{B}\right)\mathtt{rad}_t(dK) \\ &\leq \big(\mathtt{LP}'/B + 1\big)\cdot T\cdot \mathtt{rad}_t(dK). \qquad \square \end{aligned}$$

C.2. Proof of Lemma 11

The proof consists of two parts. The second inequality in Equation (12) follows easily because the distribution which maximizes $\mathtt{LP}(P, \mu)$ by definition belongs to $\mathbf{\Delta}_t$, and so

$$\mathtt{LP}(P', \mu) \leq \max_{P \in \mathtt{Conv}(\mathbf{\Delta}_t)} \mathtt{LP}(P, \mu) = \max_{P \in \mathbf{\Delta}_t} \mathtt{LP}(P, \mu).$$

To prove the first inequality in Equation (12), we first argue that $\mathtt{LP}(P, \mu)$ is a quasi-concave function of P. Denote $\eta_i(P, \mu) = B \cdot r(P, \mu)/c_i(P, \mu)$ for each resource i. Then η_i is a quasi-concave function of P since each *level set* (the set of distributions P that satisfy $\eta_i(P, \mu) \geq \alpha$ for some $\alpha \in \mathbb{R}$) is a convex set. Therefore $\mathtt{LP}(P, \mu) = \min_i \eta_i(P, \mu)$ is a quasi-concave function of P as a minimum of quasi-concave functions.

Since $P' \in \mathtt{Conv}(\mathbf{\Delta}_t)$, it is a convex combination $P' = \sum_{Q \in \mathbf{\Delta}_t} \alpha_Q\, Q$ with $\sum_{Q \in \mathbf{\Delta}_t} \alpha_Q = 1$. Therefore:

$$\begin{aligned} \mathtt{LP}(P', \mu) &= \mathtt{LP}\left(\sum_{Q \in \mathbf{\Delta}_t} \alpha_Q\, Q,\ \mu\right) \\ &\geq \min_{Q \in \mathbf{\Delta}_t, \alpha_Q > 0} \mathtt{LP}(Q, \mu) \qquad \text{By definition of quasi-concave functions} \\ &\geq \min_{Q \in \mathbf{\Delta}_t} \mathtt{LP}(Q, \mu). \qquad \square \end{aligned}$$

C.3. Remainder of the proof after Lemma 12

We start with Corollary 13, which we restate here for convenience.

Corollary $\Phi_t \leq 2\Psi_t$, assuming that $B \geq 6 \cdot T \cdot \mathtt{rad}_t(dK)$.

Proof Let $\gamma = \max_{P \in \mathcal{F}_\Pi, \mu \in \mathcal{I}_t} \mathtt{LP}(P, \mu)$. Note that $\gamma \leq T$. Then from Lemma 12 we obtain:

$$\gamma - \mathtt{OPT}_{\mathtt{LP}} \leq 3(\tfrac{\gamma}{B} + 2) \cdot T \cdot \mathtt{rad}_t(dK) \leq \tfrac{\gamma}{2} + 6 \cdot T \cdot \mathtt{rad}_t(dK). \tag{20}$$

Using (20) and Lemma 12 we get the desired bound:

$$\begin{aligned}\Phi_t &\leq (\tfrac{\gamma}{B}+2)\cdot T\cdot \mathtt{rad}_t(dK)\\ &\leq \left(\frac{2\,\mathtt{OPT_{LP}}+12\cdot T\cdot \mathtt{rad}_t(dK)}{B}+2\right)\cdot T\cdot \mathtt{rad}_t(dK)\\ &\leq \left(\frac{2\,\mathtt{OPT_{LP}}}{B}+4\right)\cdot T\cdot \mathtt{rad}_t(dK) = 2\Psi_t. \qquad \square\end{aligned}$$

In the remainder of this appendix, we prove the claims in Equation (15) one by one.

Corollary 17 $\mathtt{REW}_t \geq \frac{t}{T}\,(\mathtt{OPT_{LP}} - O(\Psi_t))$ *for each round* $t \leq \tau$.

Proof From Lemma 14 we obtain

$$\begin{aligned}T\,r(P'_t,\mu) &\geq (1-q_0)\,\mathtt{LP}(P_t,\mu)\\ &\geq (1-q_0)\,(\mathtt{OPT_{LP}}-12\Psi_t)\\ &\geq \mathtt{OPT_{LP}}-13\Psi_t.\end{aligned}$$

Summing up and taking average over rounds, we obtain:

$$T\,\overline{r}_t \geq \mathtt{OPT_{LP}} - \tfrac{13}{t}\textstyle\sum_{s=1}^t\ \Psi_s \geq \mathtt{OPT_{LP}} - O(\Psi_t).$$

By definition of clean execution, we obtain:

$$\mathtt{REW}_t \geq t(\overline{r}_t - \mathtt{rad}_t(\overline{r_t})) \geq \tfrac{t}{T}(\mathtt{OPT_{LP}} - O(\Psi_t)). \qquad \square$$

Corollary 18 $C_{t,i} \leq B/T + O(\mathtt{rad}_t(dK))$ *for each round* $t \leq \tau$.

Proof Let μ be the (actual) expected-outcomes tuple, and recall that P_t is LP-optimal for some expected-outcomes tuple $\mu' \in \mathbf{\Delta}_t$. Then, by Lemma 9, it follows that $c_i(P_t,\mu) \leq c_i(P_t,\mu') + \mathtt{rad}_t(dK)$. Furthermore since P_t is LP-optimal for μ' we have $c_i(P_t,\mu') \leq \frac{B}{T}$. Therefore:

$$\begin{aligned}c_i(P_t,\mu) &\leq \tfrac{B}{T} + \mathtt{rad}_t(dK)\\ c_i(P'_t,\mu) &\leq (1-q_0)\,c_i(P_t,\mu)+q_0\\ &\leq \tfrac{B}{T} + O(\mathtt{rad}_t(dK)).\end{aligned}$$

Now summing and taking average we obtain $\overline{c}_{t,i} \leq \frac{B}{T} + O(\mathtt{rad}_t(dK))$. Using the definition of clean execution, it follows that

$$C_{t,i} \leq \overline{c}_{t,i} + \mathtt{rad}_t(\overline{c}_{t,i}) \leq \tfrac{B}{T} + O(\mathtt{rad}_t(dK)). \qquad \square$$

Lemma 19 $\mathtt{REW}_T \geq \mathtt{OPT}_{\mathtt{LP}} - O(\Psi_T)$.

Proof Either $\tau = T$ or some resource i gets exhausted, in which case (using Corollary 18)

$$\begin{aligned} \tau = \frac{B}{C_{\tau,i}} &\geq \frac{B}{\frac{B}{T} + \mathtt{rad}_\tau(dK)} \\ \Rightarrow \tau\tfrac{B}{T} + \tau\mathtt{rad}_\tau(dK) &\geq B \\ \Rightarrow \tau\tfrac{B}{T} + T\mathtt{rad}_T(dK) &\geq B \\ \Rightarrow \tau &\geq T\left(1 - \tfrac{T}{B}\,\mathtt{rad}_T(dK)\right). \end{aligned} \tag{21}$$

Using this lower bound and Corollary 17, we obtain the desired bound on the total revenue $\mathtt{REW}_T$.

$$\begin{aligned} \mathtt{REW}_T = \mathtt{REW}_\tau &\geq \frac{\tau}{T}\,(\,\mathtt{OPT}_{\mathtt{LP}} - O(\Psi_\tau)\,) \\ &\geq \mathtt{OPT}_{\mathtt{LP}}(1 - \tfrac{T}{B}\,\mathtt{rad}_T(dK)) - \frac{O(\tau\,\Psi_\tau)}{T} \\ &\geq \mathtt{OPT}_{\mathtt{LP}} - \Psi_T - \frac{O(\tau\,\Psi_\tau)}{T}. \end{aligned}$$

In the above, the first inequality holds by Corollary 17, the second by Equation (21), and the third by definition of Ψ_T.

Finally, we note that $\tau\,\Psi_\tau$ is an increasing function of τ, and substitute $\tau\Psi_\tau \leq T\Psi_T$. □

We complete the proof of Theorem 1 as follows. Re-writing Lemma 19 as $\mathtt{REW}_T \geq f(\mathtt{OPT}_{\mathtt{LP}})$, for an appropriate function $f()$, note that $\mathtt{REW}_T \geq f(\mathtt{OPT})$ because function $f()$ is increasing.

Appendix D: Lower bound: proof of Theorem 2

In fact, we prove a stronger theorem that implies Theorem 2.

Theorem 20 *Fix any tuple (K, T, B) such that $K \in [2, T]$ and $B \leq \sqrt{KT}/2$. Any algorithm for* RCB *incurs regret $\Omega(\mathtt{OPT}(\Pi))$ in the worst case over all problem instances with K actions, time horizon T, smallest budget B, and policy sets Π such that $\mathtt{OPT}(\Pi) \leq B$.*

We will use the following lemma (which follows from simple probability arguments).

Lemma 21 *Consider two collections of n balls $\mathcal{I}_1$ and $\mathcal{I}_2$, each numbered from 1 to n. Let $\mathcal{I}_1$ consists of all red balls, while $\mathcal{I}_2$ consist of $n-1$ red balls and 1 green ball (with labels chosen uniformly at random). In this setting, let an algorithm is given access to random samples from one of $\mathcal{I}_i$ with replacement. The algorithm is allowed to first look at the ball's number and then decide whether to inspect it's color. Then any algorithm $\mathcal{A}$ which with probability at least $\frac{1}{2}$ can distinguish between $\mathcal{I}_1$ and $\mathcal{I}_2$ must inspect color of at least $n/2$ balls in expectation.*

In the remainder of this section we prove Theorem 20.

Let us define a family of problem instances as follows. Let the set of arms be $\{a_1, a_2, \ldots, a_K\}$. There are T/B different contexts labelled $\{x_1, ..., x_{T/B}\}$ and there is a uniform distribution over contexts. The policy set Π consists of $T\,(K-1)/B$ policies $\pi_{i,j}$, where $2 \leq i \leq K$ and $1 \leq j \leq T/B$. Define them as follows: $\pi_{i,j}(x_l) = a_i$ for $l = j$, and $\pi_{i,j}(x_l) = a_1$ for $l \neq j$.

There is just one resource constraint B (apart from time). Pulling arm a_1 always costs 0 and arm $a_i, i \neq 1$ always costs 1. Now consider the following problem instances:

- Let $\mathcal{F}_0$ be the instance in which every arm always gives a reward 0. Note that $\texttt{OPT}(\mathcal{F}_0) = 0$.
- Let $\mathcal{F}_{i,j}$ be the instance in which arm a_i on context x_j gives reward 1, otherwise every arm on every context gives reward 0. Note that in this case the optimal distribution over policies is just to follow $\pi_{i,j}$ and gets reward $\approx B$.

Now consider any algorithm $\mathcal{A}$ and let the expected number of times it pulls arm a_i be p_i on input $\mathcal{F}_0$. Let $i', i' \neq 1$ be the arm for which this is minimum. Then by simple linearity of expectation we get that $B \geq (K-1)p_{i'}$. It is also simple to see that for the algorithm to get a regret better than $\Omega(\texttt{OPT})$ it should be able to distinguish between $\mathcal{F}_0$ and $\mathcal{F}_{i',\cdot}$ at least with probability $\frac{1}{2}$. From lemma 21 this can be done iff $p_{i'} \geq T/(2B)$. Combining the two equations we get $B \geq (K-1)T/(2B)$. Solving for B we get $B \geq \sqrt{KT}/2$.

Appendix E: Discretization: proof of Theorem 3

We consider contextual dynamic pricing with B copies of a single product. The action space consists of all prices $p \in [0,1]$. We obtain regret bounds relative to an arbitrary policy set Π.

We will need some notation. Let $S(p|x)$ be the probability of a sale for price p and context x; note that $S(p|x)$ is non-increasing in p, for any given x. For a randomized policy π, define $S(\pi|x) = \mathbb{E}_{p\sim\pi(x)}[\,S(p|x)\,]$. Let $f_\epsilon(p)$, $p \geq 0$ be the largest price $p' \leq p$ such that $p' \in \epsilon\mathbb{N}$. For notational convenience, we use a special price $p = \infty$ which corresponds to skipping a round.

As discussed in the Introduction, for each $\epsilon > 0$ we define a ϵ-discretized policy set Π_ϵ that approximates Π and only uses multiples of ϵ as prices. The idea is that whenever a policy $\pi \in \Pi$ picks some price p, we replace it with the discretized price $p' = f_\epsilon(p)$. To account for a possible decrease in the sale probability $S(\cdot)$, we skip the corresponding round with probability $1 - S(p)/S(p')$.

Formally, for each $\pi \in \Pi$ we define a randomized policy π_ϵ as follows: for each context x,

$$\pi_\epsilon(x) = \begin{cases} f_\epsilon(\pi(x)) & \text{with probability } \frac{S(\pi|x)}{S(f_\epsilon(\pi(x))\,|x)} \\ \infty & \text{with the remaining probability.} \end{cases} \tag{22}$$

The ϵ-discretized policy set is defined as $\Pi_\epsilon = \{\pi_\epsilon : \pi \in \Pi\}$.

In addition to the discretized prices, the π_ϵ has the following useful properties:

- $\pi(x) \geq \pi_\epsilon(x) \geq \pi(x) - \epsilon$ for all contexts x,
- $S(\pi_\epsilon|x) = S(\pi|x)$ for all contexts x.

The key technical step is to bound the discretization error of Π_ϵ compared to Π, as quantified by the difference in $\texttt{OPT}_{\texttt{LP}}(\cdot)$.

Lemma 22 $\texttt{OPT}_{\texttt{LP}}(\Pi) - \texttt{OPT}_{\texttt{LP}}(\Pi_\epsilon) \leq \epsilon B$*, for each* $\epsilon > 0$.

Proof Fix $\epsilon > 0$ and a distribution P over policies in Π. Define the ϵ-discretized distribution P_ϵ over Π_ϵ in a natural way: for each policy $\pi \in \Pi$, select π_ϵ with probability $P(\pi)$. It suffices to prove that $\texttt{LP}(P_\epsilon) \geq \texttt{LP}(P) - \epsilon B$.

Let $r(P)$ and $c(P)$ denote the expected per-round reward and the expected per-round consumption for distribution P. Then

$$\begin{aligned}
c(P_\epsilon) &= \mathop{\mathbb{E}}_{x,\pi}\left[\, S(\pi_\epsilon|x)\,\right] = \mathop{\mathbb{E}}_{x,\pi}\left[\, S(\pi_\epsilon|x)\,\right] = c(P) \\
r(P_\epsilon) &= \mathop{\mathbb{E}}_{x,\pi}\left[\, f_\epsilon(\pi(x)) \cdot S(\pi_\epsilon|x)\,\right] \\
&\geq \mathop{\mathbb{E}}_{x,\pi}\left[\, (\pi(x)-\epsilon) \cdot S(\pi_\epsilon|x)\,\right] \\
&\geq \mathop{\mathbb{E}}_{x,\pi}\left[\, \pi(x) \cdot S(\pi|x)\,\right] - \epsilon \mathop{\mathbb{E}}_{x,\pi}\left[\, S(\pi_\epsilon|x)\,\right] \\
&= r(P) - \epsilon\, c(P_\epsilon). \\
\texttt{LP}(P_\epsilon) &= r(P_\epsilon)\, \frac{B}{c(P_\epsilon)} \geq r(P)\, \frac{B}{c(P)} - \epsilon B. \\
&\geq \texttt{LP}(P) - \epsilon B. \qquad \square
\end{aligned}$$

Let $\texttt{REW}(\Pi')$ be the expected total reward when algorithm `Mixture_Elimination` is run with policy set Π' which uses only K distinct actions. Recall that we actually prove a somewhat stronger version of Theorem 1: the same regret bound (1), but with respect to $\texttt{OPT}_{\texttt{LP}}(\Pi')$ rather than $\texttt{OPT}(\Pi')$. In our setting we have $d=1$ (single constraint) and $\texttt{OPT}_{\texttt{LP}}(\Pi') \leq B$. Therefore we have

$$\texttt{REW}(\Pi') \geq \texttt{OPT}_{\texttt{LP}}(\Pi') - O\left(\sqrt{KT\ \log\left(KT\,|\Pi'|\right)}\right).$$

Plugging in $\Pi' = \Pi_\epsilon$ and $K = \frac{1}{\epsilon}$, and using Lemma 22, we obtain

$$\texttt{REW}(\Pi_\epsilon) \geq \texttt{OPT}_{\texttt{LP}}(\Pi) - \epsilon B - O\left(\sqrt{\tfrac{T}{\epsilon}\ \log\left(\tfrac{T}{\epsilon}\,|\Pi_\epsilon|\right)}\right), \qquad \text{for each } \epsilon > 0. \tag{23}$$

We obtain Theorem 3 choosing $\epsilon = (\frac{T}{B^2})^{1/3}$ and noting $|\Pi_\epsilon| \leq |\Pi|$.

Appendix F: Open questions

As discussed in the Conclusions, the main open question concerns combining provable regret bounds and a computationally efficient implementation. Computational issues aside, several open questions concern our regret bounds.

First, it is desirable to achieve the same regret bounds without assuming a known time horizon T (as it is in most bandit problems in the literature). This may be difficult because time is one of the resource constraints in our problem, and our techniques rely on knowing all resource constraints in advance. More generally, one can consider a version of `RCB` in which some of the resource constraints are not fully revealed to an algorithm; instead, the algorithm receives updated estimates of these constrains over time.

Second, while our main regret bound in Theorem 1 is optimal in the important regime when $\texttt{OPT}(\Pi)$ and B are at least a constant fraction of T, it is not tight for some other regimes. For a concrete comparison, consider problem instances with a constant number of resources (d), a constant number of actions (K), and $\texttt{OPT}(\Pi) \geq \Omega(B)$. Then, ignoring logarithmic factors, we obtain regret $\texttt{OPT}(\Pi)\,\sqrt{T}/B$, whereas the lower bound in Badanidiyuru et al. (2013a) is $\texttt{OPT}(\Pi)/\sqrt{B}$. So there is a gap when $B \ll T$. Likewise, our result for contextual dynamic pricing with a single product

(Theorem 3) is tight in the regime $B \geq \Omega(T)$, but not if $B \ll T$: ignoring the logarithmic factors, our upper bound is $(TB)^{1/3}$, whereas the lower bound from Babaioff et al. (2012) is $B^{2/3}$. In both cases, both upper and lower bounds can potentially be improved.

Third, it is important to extend the discretization approach beyond contextual dynamic pricing with a single product. However, this is problematic even without contexts: discretization issues are unresolved whenever one has multiple resource constraints (see Badanidiyuru et al. (2013a)), essentially because there is no general technique to usefully upper-bound the discretization error. Further, while Badanidiyuru et al. (2013a) resolve discretization for (non-contextual) dynamic procurement with a single budget, they find that the seemingly natural mesh in which all prices are multiples of some ϵ is *not* the right mesh for this problem.

Fourth, if there are no contexts or resource constraints then one can achieve $O(\log T)$ regret with an instance dependent constant; it is not clear whether one can meaningfully extend this result to contextual bandits with resource constraints.

The model of `RCB` can be extended in several directions, two of which we outline below. The most immediate extension is to unknown distribution of context arrivals. We conjecture the present results can be extended, perhaps with a rather tedious amount of work, building on the techniques from Dudik et al. (2011). The most important extension, in our opinion, would be from a stationary environment to one controlled by an adversary (which can be restricted in some natural way). We are not aware of any prior work in this direction, even for the non-contextual version.