Combinatorial Semi-Bandits with Knapsacks

Karthik Abinav Sankararaman
University of Maryland, College Park
kabinav@cs.umd.edu

Aleksandrs Slivkins
Microsoft Research NYC
slivkins@microsoft.com

Abstract

We unify two prominent lines of work on multi-armed bandits: *bandits with knapsacks* and *combinatorial semi-bandits*. The former concerns limited "resources" consumed by the algorithm, *e.g.,* limited supply in dynamic pricing. The latter allows a huge number of actions but assumes combinatorial structure and additional feedback to make the problem tractable. We define a common generalization, support it with several motivating examples, and design an algorithm for it. Our regret bounds are comparable with those for BwK and combinatorial semi-bandits.

1 Introduction

Multi-armed bandits (*MAB*) is an elegant model for studying the tradeoff between acquisition and usage of information, a.k.a. *explore-exploit tradeoff* [Robbins, 1952, Thompson, 1933]. In each round an algorithm sequentially chooses from a fixed set of alternatives (sometimes known as *actions* or *arms*), and receives reward for the chosen action. Crucially, the algorithm does not have enough information to answer all "counterfactual" questions about what would have happened if a different action was chosen in this round. MAB problems have been studied steadily since 1930-ies, with a huge surge of interest in the last decade.

This paper combines two lines of work related to bandits: on *bandits with knapsacks* (BwK) [Badanidiyuru et al., 2013a] and on *combinatorial semi-bandits* [György et al., 2007]. BwK concern scenarios with limited "resources" consumed by the algorithm, *e.g.,* limited inventory in a dynamic pricing problem. In combinatorial semi-bandits, actions correspond to subsets of some "ground set", rewards are additive across the elements of this ground set (*atoms*), and the reward for each chosen atom is revealed (*semi-bandit feedback*). A paradigmatic example is an online routing problem, where atoms are edges in a graph, and actions are paths. Both lines of work have received much recent attention and are supported by numerous examples.

Our contributions. We define a common generalization of combinatorial semi-bandits and BwK, termed *Combinatorial Semi-Bandits with Knapsacks* (SemiBwK). Following all prior work on BwK, we focus on an i.i.d. environment: in each round, the "outcome" is drawn independently from a fixed distribution over the possible outcomes. Here the "outcome" of a round is the matrix of reward and resource consumption for all atoms.[1] We design an algorithm for SemiBwK, achieving regret rates that are comparable with those for BwK and combinatorial semi-bandits.

Specifics are as follows. As usual, we assume "bounded outcomes": for each atom and each round, rewards and consumption of each resource is non-negative and at most 1. Regret is relative to the expected total reward of the best all-knowing policy, denoted OPT. For BwK problems, this is known to be a much stronger benchmark than the traditional best-fixed-arm benchmark. We upper-bound the regret in terms of the relevant parameters: time horizon T, (smallest) budget B, number of atoms n, and OPT itself (which may be as large as nT). We obtain

$$\texttt{Regret} \leq \tilde{O}(\sqrt{n})(\mathrm{OPT}/\sqrt{B} + \sqrt{T + \mathrm{OPT}}). \quad (1.1)$$

The "shape" of the regret bound is consistent with prior work: the $\mathrm{OPT}/\sqrt{B}$ additive term appears in the optimal regret bound for BwK, and the $\sqrt{T}$ and $\sqrt{\mathrm{OPT}}$ additive terms are very common in regret bounds for MAB. The per-round running time is polynomial in n, and near-linear in n for some important special cases.

KAS's research supported in part by NSF Awards CNS 1010789 and CCF 1422569.

For the full version of this paper, please see Sankararaman and Slivkins [2017]

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

[1] Our model allows arbitrary correlations within a given round, both across rewards and consumption for the same atom and across multiple atoms. Such correlations are essential in applications such as dynamic pricing and dynamic assortment. *E.g.,* customers' valuations can be correlated across products, and algorithm earns only if it sells; see Section 4 for details.

Our regret bound is optimal up to polylog factors for paradigmatic special cases. BwK is a special case when actions are atoms. For $\mathrm{OPT} > \Omega(T)$, the regret bound becomes $\tilde{O}(T\sqrt{n/B} + \sqrt{nT})$, where n is the number of actions, which coincides with the lower bound from [Badanidiyuru et al., 2013a]. Combinatorial semi-bandits is a special case with $B = nT$. If all feasible subsets contain at most k atoms, we have $\mathrm{OPT} \leq kT$, and the regret bound becomes $\tilde{O}(\sqrt{knT})$. This coincides with the $\Omega(\sqrt{knT})$ lower bound from [Kveton et al., 2014].

Our main result assumes that the action set, *i.e.,* the family of feasible subsets of atoms, is described by a *matroid constraint*.[2] This is a rather general scenario which includes many paradigmatic special cases of combinatorial semi-bandits such as cardinality constraints, partition matroid constraints, and spanning tree constraints. We also assume that $B > \tilde{\Omega}(n + \sqrt{nT})$.

Our model captures several application scenarios, incl. dynamic pricing, dynamic assortment, repeated auctions, and repeated bidding. We work out these applications, and explain how our regret bounds improve over prior work.

Challenges and techniques. BwK problems are challenging compared to traditional MAB problems with i.i.d. rewards because it no longer suffices to look for the best action and/or optimize expected per-round rewards; instead, one essentially needs to look for a *distribution* over actions with optimal expected *total* reward across all rounds. Generic challenges in combinatorial semi-bandits concern handling exponentially many actions (both in terms of regret and in terms of the running time), and taking advantage of the additional feedback. And in SemiBwK, one needs to deal with distributions over subsets of atoms, rather than "just" with distributions over actions.

Our algorithm connects a technique from BwK and a randomized rounding technique from combinatorial optimization. (With *five* existing BwK algorithms and a wealth of approaches for combinatorial optimization, choosing the techniques is a part of the challenge.)

We build on a BwK algorithm from Agrawal and Devanur [2014a], which combines linear relaxations and a well-known "optimism-under-uncertainty" paradigm. A generalization of this algorithm to SemiBwK results in a fractional solution $\boldsymbol{x}$, a vector over atoms. Randomized rounding converts $\boldsymbol{x}$ into a distribution over feasible subsets of atoms that equals $\boldsymbol{x}$ in expectation. It is crucial (and challenging) to ensure that this distribution contains enough randomness so as to admit concentration bounds not only across rounds, but also across atoms. Our analysis "opens up" a fairly technical proof from prior work and intertwines it with a new argument based on negative correlation.

[2]Matroid is a standard notion in combinatorial optimization which abstracts and generalizes linear independence.

We present our algorithm and analysis so as to "plug in" any suitable randomized rounding technique. This makes our presentation more lucid, and also leads to faster running times for some important special cases.

Solving SemiBwK using prior work. Solving SemiBwK using an algorithm for BwK would result in a regret bound like (1.1) with n replaced with the number of actions. The latter could be on the order of n^k if each action can consist of at most k atoms, or perhaps even exponential in n.

SemiBwK can be solved as a special case of a much more general *linear-contextual* extension of BwK from Agrawal and Devanur [2014a, 2016]. In their model, an algorithm takes advantage of the combinatorial structure of actions, yet it ignores the additional feedback from the atoms. Their regret bounds have a worse dependence on the parameters, and apply for a much more limited range of parameters. Further, their per-round running time is linear in the number of actions, which is often prohibitively large.

To compare the regret bounds, let us focus on instances of SemiBwK in which at most one unit of each resource is consumed in each round. (This is the case in all our motivating applications, except repeated bidding.) Then Agrawal and Devanur [2014a, 2016] assume $B > \sqrt{n}\,T^{3/4}$, and achieve regret $\tilde{O}(n\sqrt{T}\frac{\mathrm{OPT}}{B} + n^2\sqrt{T})$. [3] It is easy to see that we improve upon the range and upon both summands. In particular, we improve both summands by the factor of $n\sqrt{n}$ in a lucid special case when $B > \Omega(T)$ and $\mathrm{OPT} < O(T)$.[4]

We run simulations to compare our algorithm against prior work on BwK and combinatorial semi-bandits.

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. Among broad directions in MAB, most relevant is MAB with i.i.d. rewards, starting from [Lai and Robbins, 1985, Auer et al., 2002].

Bandits with Knapsacks (BwK) were first introduced by Badanidiyuru et al. [2013a] as a common generalization of several models from prior work and many other motivating examples. Subsequent papers extended BwK to "smoother" resource constraints and introduced several new algorithms [Agrawal and Devanur, 2014a], and generalized BwK to

[3]Agrawal and Devanur [2014a, 2016] state regret bound with term $+n\sqrt{T}$ rather than $+n^2\sqrt{T}$, but they assume that per-round rewards lie in $[0, 1]$. Since per-round rewards can be as large as n in our setting, we need to scale down all rewards by a factor of n, apply their regret bound, and then scale back, which results in the regret bound with $+n^2\sqrt{T}$. When per-round consumption can be as large as n, regret bound from Agrawal and Devanur [2014a, 2016] becomes $\tilde{O}(n^2\,\mathrm{OPT}\,\sqrt{T}/B + n^2\sqrt{T})$ due to rescaling.

[4]In prior work on combinatorial bandits (without constraints), semi-bandit feedback improves regret bound by a factor of $\sqrt{n}$, see the discussion in Kveton et al. [2015b].

contextual bandits [Badanidiyuru et al., 2014, Agrawal et al., 2016, Agrawal and Devanur, 2016]. All prior work on BwK and special cases thereof assumed i.i.d. outcomes.

Special cases of BwK include dynamic pricing with limited supply [Babaioff et al., 2015, Besbes and Zeevi, 2009, 2012, Wang et al., 2014], dynamic procurement on a budget [Badanidiyuru et al., 2012, Singla and Krause, 2013, Slivkins and Vaughan, 2013], dynamic ad allocation with advertiser budgets [Slivkins, 2013], and bandits with a single deterministic resource [Guha and Munagala, 2007, Gupta et al., 2011, Tran-Thanh et al., 2010, 2012]. Some special cases admit instance-dependent logarithmic regret bounds [Xia et al., 2016b,a, Combes et al., 2015a, Slivkins, 2013] when there is only one bounded resource and unbounded time, or when resource constraints do not bind across arms.

Combinatorial semi-bandits were studied by György et al. [2007], in the adversarial setting. In the i.i.d. setting, in a series of works by [Anantharam et al., 1987, Gai et al., 2010, 2012, Chen et al., 2013, Kveton et al., 2015b, Combes et al., 2015b], an optimal algorithm was achieved. This result was then extended to atoms with linear rewards by Wen et al. [2015]. Kveton et al. [2014] obtained improved results for the special case when action set is described by a matroid. Some other works studied a closely related "cascade model", where the ordering of atoms matters [Kveton et al., 2015a, Katariya et al., 2016, Zong et al., 2016]. Contextual semi-bandits have been studied in [Wen et al., 2015, Krishnamurthy et al., 2016].

Randomized rounding schemes (RRS) come from the literature on approximation algorithms in combinatorial optimization (see Williamson and Shmoys [2011], Papadimitriou and Steiglitz [1982] for background). RRS were introduced in Raghavan and Tompson [1987]. Subsequent work [Gandhi et al., 2006, Asadpour et al., 2010, Chekuri et al., 2010, 2011] developed RRS which correlate the rounded random variables so as to guarantee sharp concentration bounds.

Discussion. The basic model of multi-armed bandits can be extended in many distinct directions: what auxiliary information, if any, is revealed to the algorithm before it needs to make a decision, which feedback is revealed afterwards, which "process" are the rewards coming from, do they have some known structure that can be leveraged, are there global constraints on the algorithm, etc. While many real-life scenarios combine several directions, most existing work proceeds along only one or two. We believe it is important (and often quite challenging) to unify these lines of work. For example, an important recent result of Syrgkanis et al. [2016], Rakhlin and Sridharan [2016] combined "contextual" and "adversarial" bandits.

Organization of the paper. We formally define the model, describe the algorithm and the regret bounds, overview the analysis, discuss applications and examples, and overview the simulations. Due to the page limit, many details are deferred to the full version.

2 Our model and preliminaries

Our model, called *Semi-Bandits with Knapsacks* (SemiBwK) is a generalization of multi-armed bandits (henceforth, *MAB*) with i.i.d. rewards. As such, in each round $t = 1, \ldots, T$, an algorithm chooses an action S_t from a fixed set of actions $\mathcal{F}$, and receives a reward $\mu_t(S_t)$ for this action which is drawn independently from a fixed distribution that depends only on the chosen action. The number of rounds T, a.k.a. the *time horizon*, is known.

There are d resources being consumed by the algorithm. The algorithm starts out with budget $B_j \geq 0$ of each resource j. All budgets are known to the algorithm. If in round t action $S \in \mathcal{F}$ is chosen, the outcome of this round is not only the reward $\mu_t(S)$ but the consumption $C_t(S,j)$ of each resource $j \in [d]$. We refer to $\boldsymbol{C}_t(S) = (C_t(S,j) : j \in [d])$ as the *consumption vector*.[5] Following prior work on BwK, we assume that all budgets are the same: $B_j = B$ for all resources j.[6] Algorithm stops as soon as any one of the resources goes strictly below 0. The round in which this happens is called the stopping time and denoted $\tau_{\texttt{stop}}$. The reward collected in this last round does not count; so the total reward of the algorithm is $\texttt{rew} = \sum_{t<\tau_{\texttt{stop}}} \mu_t(S_t)$.

Actions correspond to subsets of a finite ground set $\mathcal{A}$, with $n = |\mathcal{A}|$; we refer to elements of $\mathcal{A}$ as *atoms*. Thus, the set $\mathcal{F}$ of actions corresponds to the family of "feasible subsets" of $\mathcal{A}$. The rewards and resource consumption is additive over the atoms: for each round t and each atom a there is a reward $\mu_t(a) \in [0,1]$ and consumption vector $\boldsymbol{C}_t(a) \in [0,1]^d$ such that for each action $S \subset \mathcal{F}$ it holds that $\mu_t(S) = \sum_{a\in S} \mu_t(a)$ and $\boldsymbol{C}_t(S) = \sum_{a\in S} \boldsymbol{C}_t(a)$.

We assume the i.i.d. property across rounds, but allow arbitrary correlations within the same round. For a given round t we consider the $n \times (d+1)$ "outcome matrix" $(\mu_t(a), \boldsymbol{C}_t(a) : a \in \mathcal{A})$, which specifies rewards and resource consumption for all resources and all atoms. We assume that the outcome matrix is chosen independently from a fixed distribution $\mathcal{D}_{\texttt{M}}$ over such matrices, which is not revealed to the algorithm. The mean rewards and mean consumption is denoted $\mu(a) := \mathbb{E}[\mu_t(a)]$ and $\boldsymbol{C}(a) := \mathbb{E}[\boldsymbol{C}_t(a)]$. We extend the notation to actions, *i.e.,* to subsets of atoms: $\mu(S) := \sum_{a\in S} \mu(a)$ and $\boldsymbol{C}(S) := \sum_{a\in S} \boldsymbol{C}(a)$.

An instance of SemiBwK consists of the action set $\mathcal{F} \subset 2^{[n]}$, the budgets $\boldsymbol{B} = (B_j : j \in [d])$, and the distribution $\mathcal{D}_{\texttt{M}}$.

[5] We use bold font to indicate vectors and matrices.

[6] This is w.l.o.g. because we can divide all consumption of each resource j by $B_j / \min_{j' \in [d]} B_{j'}$. Effectively, B is the smallest budget in the original problem instance.

The $\mathcal{F}$ and $\boldsymbol{B}$ are known to the algorithm, and $\mathcal{D}_{\mathtt{M}}$ is not. As explained in the introduction, SemiBwK subsumes *Bandits with Knapsacks* (BwK) and semi-bandits. BwK is the special case when $\mathcal{F}$ consists of singletons, and semi-bandits is the special case when all budgets are equal to $B_j = nT$ (so that the resource consumption is irrelevant).

Following the prior work on BwK, we compete against the "optimal all-knowing algorithm": an algorithm that optimizes the expected total reward for a given problem instance; its expected total reward is denoted by OPT. As observed in Badanidiyuru et al. [2013a], OPT can be much larger (*e.g.,* factor of 2 larger) than the expected cumulative reward of the best action, for a variety of important special cases of BwK. Our goal is to minimize *regret*, defined as OPT minus algorithm's total reward.

Combinatorial constraints. Action set $\mathcal{F}$ is given by a *combinatorial constraint*, *i.e.,* a family of subsets. Treating subsets of atoms as n-dimensional binary vectors, $\mathcal{F}$ corresponds to a finite set of points in $\mathbb{R}^n$. We assume that the convex hull of $\mathcal{F}$ forms a polytope in $\mathbb{R}^n$. In other words, there exists a set of linear constraints over $\mathbb{R}^n$ whose set of feasible *integral* solutions is $\mathcal{F}$. We call such $\mathcal{F}$ *linearizable*; the convex hull is called the polytope *induced* by $\mathcal{F}$.

Our main result is for *matroid constraints*, a family of linearizable combinatorial constraints which subsumes several important special cases such as cardinality constraints, partition matroid constraints, spanning tree constraints and transversal constraints. Formally, $\mathcal{F}$ is a matroid if it contains the empty set, and satisfies two properties: (i) if $\mathcal{F}$ contains a subset S, then it also contains every subset of S, and (ii) for any two subsets $S, S' \in \mathcal{F}$ with $|S| > |S'|$ it holds that $S' \cup \{a\} \in \mathcal{F}$ for each atom $a \in S \setminus S'$. See Appendix B for more background and examples.

We incorporate prior work on randomized rounding for linear programs. Consider a linearizable action set $\mathcal{F}$ with induced polytope $P \subset [0,1]^n$. The *randomized rounding scheme* (henceforth, RRS) for $\mathcal{F}$ is an algorithm which inputs a feasible fractional solution $\boldsymbol{x} \in P$ and the linear equations describing P, and produces a random vector $\boldsymbol{Y}$ over $\mathcal{F}$. We consider RRS's such that $\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{x}$ and $\boldsymbol{Y}$ is negatively correlated (see below for definition); we call such RRS's *negatively correlated*. Several such RRS are known: *e.g.,* for cardinality constraints and bipartite matching [Gandhi et al., 2006], for spanning trees [Asadpour et al., 2010], and for matroids [Chekuri et al., 2010].

Negative correlation. Let $\mathcal{X} = (X_1, X_2, \ldots, X_m)$ denote a family of random variables which take values in $[0,1]$. Let $X := \frac{1}{m}\sum_{i=1}^m X_i$ be the average, and $\mu := \mathbb{E}[X]$.

Family $\mathcal{X}$ is called *negatively correlated* if

$$\mathbb{E}\left[\prod_{i\in S} X_i\right] \leq \prod_{i\in S} \mathbb{E}[X_i] \quad \forall S \subseteq [m] \quad (2.1)$$

$$\mathbb{E}\left[\prod_{i\in S} (1-X_i)\right] \leq \prod_{i\in S} \mathbb{E}[1-X_i] \; \forall S \subseteq [m] \quad (2.2)$$

Independent random variables satisfy both properties with equality. For intuition: if X_1, X_2 are Bernoulli and (2.1) is strict, then X_1 is more likely to be 0 if $X_2 = 1$.

Negative correlation is a generalization of independence that allows for similar *concentration bounds*, *i.e.,* high-probability upper bounds on $|X-\mu|$. However, our analysis does not invoke them directly. Instead, we use a concentration bound given a closely related property:

$$\mathbb{E}\left[\prod_{i\in S} X_i\right] \leq (\tfrac{1}{2})^{|S|} \quad \forall S \subseteq [m]. \quad (2.3)$$

Theorem 2.1. *If* (2.3)*, then for some absolute constant* c*,*

$$\Pr[X \geq \tfrac{1}{2} + \eta] \leq c \cdot e^{-2m\eta^2} \qquad (\forall \eta > 0) \quad (2.4)$$

This theorem easily follows from [Impagliazzo and Kabanets, 2010], see Appendix A in the full version.

Confidence radius. We bound deviations $|X-\mu|$ in a way that gets sharper when μ is small, without knowing μ in advance. (We use the notation $\mathcal{X}, X, \mu$ as above.) To this end, we use the notion of *confidence radius* from [Kleinberg et al., 2015, Babaioff et al., 2015, Badanidiyuru et al., 2013a, Agrawal and Devanur, 2014b][7]:

$$\mathrm{Rad}_\alpha(x,m) = \sqrt{\alpha x/m} + \alpha/m. \quad (2.5)$$

If random variables $\mathcal{X}$ are independent, then event

$$|X-\mu| < \mathrm{Rad}_\alpha(X,m) < 3\,\mathrm{Rad}_\alpha(\mu,m) \quad (2.6)$$

happens with probability at least $1 - O(e^{-\Omega(\alpha)})$, for any given $\alpha > 0$. We use this notion to define upper/lower confidence bounds on the mean rewards and mean resource consumption. Fix round t, atom a, and resource j. Let $\hat{\mu}_t(a)$ and $\hat{C}_t(a,j)$ denote the empirical average of the rewards and resource-j consumption, resp., between rounds 1 and $t-1$. Let $N_t(a)$ be the number of times atom a has been chosen in these rounds (*i.e.,* included in the chosen actions). The confidence bounds are defined as

$$\begin{aligned} C_t^{\pm}(a,j) &= \mathtt{proj}(\, \hat{C}(a,j) \pm \mathrm{Rad}_\alpha(\hat{C}(a,j), N_t(a))\,) \\ \mu_t^{\pm}(a) &= \mathtt{proj}\,(\, \hat{\mu}(a) \pm \mathrm{Rad}_\alpha(\hat{\mu}(a), N_t(a))\,) \end{aligned} \quad (2.7)$$

where $\mathtt{proj}(x) := \mathrm{argmin}_{y\in[0,1]} |y-x|$ denotes the projection into $[0,1]$. We always use the same parameter $\alpha = c_{\mathtt{conf}}\,\log(ndT)$, for an appropriately chosen absolute constant $c_{\mathtt{conf}}$. We suppress α and $c_{\mathtt{conf}}$ from the notation. We use a vector notation $\boldsymbol{\mu}_t^{\pm}$ and $\boldsymbol{C}_t^{\pm}(j)$ to denote the corresponding n-dimensional vectors over all atoms a.

By (2.6), with probability $1 - O(e^{-\Omega(\alpha)})$ we have:

$$\begin{aligned} \mu(a) &\in [\mu_t^-(a),\ \mu_t^+(a)] \\ C(a,j) &\in [C^-(a,j),\ C(a,j)^+] \end{aligned}$$

[7] For instance, Theorem 2.1 in [Badanidiyuru et al., 2013b]

3 Main algorithm

Let us define our main algorithm, called SemiBwK-RRS. The algorithm builds on an arbitrary RRS for the action set $\mathcal{F}$. It is parameterized by this RRS, the polytope $\mathcal{P}$ induced by $\mathcal{F}$ (represented as a collection of linear constraints), and a number $\epsilon > 0$. In each round t, it recomputes the upper/lower confidence bounds, as defined in (2.7), and solves the following linear program:

$$\begin{array}{ll} \text{maximize} & \boldsymbol{\mu}_t^+ \cdot \boldsymbol{x} \\ \text{subject to} & \boldsymbol{C}_t^-(j) \cdot \boldsymbol{x} \leq \frac{B(1-\epsilon)}{T}, \quad j \in [d] \\ & \boldsymbol{x} \in \mathcal{P} \end{array} \qquad (\mathrm{LP}_{\mathrm{ALG}})$$

This linear program defines a linear relaxation of the original problem which is "optimistic" in the sense that it uses upper confidence bounds for rewards and lower confidence bounds for consumption. The linear relaxation is also "conservative" in the sense that it rescales the budget by $1-\epsilon$. Essentially, this is to ensure that the algorithm does not run out of budget with high probability. Parameter ϵ will be fixed throughout. For ease of notation, we will denote $B_\epsilon := (1-\epsilon)B$ henceforth. The LP solution $\boldsymbol{x}$ can be seen as a probability vector over the atoms. Finally, the algorithm uses the RRS to convert the LP solution into a feasible action. The pseudocode is given as Algorithm 1.

Algorithm 1: SemiBwK-RRS

input: an RRS for action set $\mathcal{F}$, induced polytope $\mathcal{P}$ (as a set of linear constraints), $\epsilon > 0$.

for $t = 1, 2, \ldots, T$ **do**

1. **Recompute Confidence Bounds** as in (2.7)
2. **Obtain fractional solution** $\boldsymbol{x}_t \in [0,1]^n$ by solving the linear program $\mathrm{LP}_{\mathrm{ALG}}$.
3. **Obtain a feasible action** $S_t \in \mathcal{F}$ by invoking the RRS on vector $\boldsymbol{x}_t$.
4. **Semi-bandit Feedback**: observe the rewards/consumption for all atoms $a \in S_t$.

If action set $\mathcal{F}$ is described by a matroid constraint, we can use the negatively correlated RRS from Chekuri et al. [2010]. In particular, we obtain a complete algorithm for several combinatorial constraints commonly used in the literature on semi-bandits, such as partition matroid constraints, spanning trees. More background on matroid constraints can be found in the full version (see Appendix B).

Theorem 3.1. *Consider the* SemiBwK *problem with a linearizable action set $\mathcal{F}$ that admits a negatively correlated* RRS. *Then algorithm* SemiBwK-RRS *with this* RRS *achieves expected regret bound at most the following.*

$$O(\log(ndT))\ \sqrt{n}\left(\mathrm{OPT}/\sqrt{B} + \sqrt{T + \mathrm{OPT}}\right) \qquad (3.1)$$

Here T is the time horizon, n is the number of atoms, and B is the budget. We require $B > 3(\alpha n + \sqrt{\alpha n T})$, where $\alpha = \Theta(\log(ndT))$ is the parameter in confidence radius. Parameter ϵ in the algorithm is set to $\sqrt{\frac{\alpha n}{B}} + \frac{\alpha n}{B} + \frac{\sqrt{\alpha n T}}{B}$.

Corollary 3.2. *Consider the setting in Theorem 3.1 and assume that the action set $\mathcal{F}$ is defined by a matroid on the set of atoms. Then, using the negatively correlated* RRS *from [Chekuri et al., 2010], we obtain regret bound* (3.1)*.*

The proof of the theorem is very technical. We provide an overview below, and defer the full proof to the full version. We actually prove a slightly stronger statement involving high-probability regret rather than expected regret.

3.1 Proof overview of Main Result

First, we argue that $\mathrm{LP}_{\mathrm{ALG}}$ provides a good benchmark that we can use instead of OPT. Specifically, at any given round, the optimal value for $\mathrm{LP}_{\mathrm{ALG}}$ in each round is at least $\frac{1}{T}(1-\epsilon)\,\mathrm{OPT}$ with high probability. We prove this by constructing a series of LPs, starting with a generic linear relaxation for BwK and ending with $\mathrm{LP}_{\mathrm{ALG}}$, and showing that the optimal value does not decrease along the series.

Next we define an event that occur with high probability, henceforth called *clean event*. This event concerns total rewards, and compares our algorithm against $\mathrm{LP}_{\mathrm{ALG}}$:

$$\begin{aligned} &\left|\textstyle\sum_{t\in[T]} r_t - \sum_{t\in[T]} \boldsymbol{\mu}_t^+ \cdot \boldsymbol{x}_t\right| \\ &\qquad \leq O\left(\sqrt{\alpha n \textstyle\sum_{t\in[T]} r_t} + \sqrt{\alpha n T} + \alpha n\right). \end{aligned} \qquad (3.2)$$

We prove that it is indeed a high-probability event in three steps. First, we relate the algorithm's reward $\sum_t r_t$ to its expected reward $\sum_t \boldsymbol{\mu} \cdot S_t$, where we interpret the chosen action S_t, a subset of atoms, as a binary vector over the atoms. Then we relate $\sum_t \boldsymbol{\mu} \cdot S_t$ to $\sum_t \boldsymbol{\mu}_t^+ \cdot S_t$, replacing expected rewards with the upper confidence bounds. Finally, we relate $\sum_t \boldsymbol{\mu}_t^+ \cdot S_t$ to $\sum_t \boldsymbol{\mu}_t^+ \cdot \boldsymbol{x}_t$, replacing the output of the RRS with the corresponding expectations. Putting it together, we relate algorithm's reward to $\sum_t \boldsymbol{\mu}_t^+ \cdot \boldsymbol{x}_t$, as needed. It is essential to bound the deviations in the sharpest way possible; in particular, the naive $\tilde{O}(\sqrt{T})$ bounds are not good enough. To this end, we use several tools: the confidence radius from (2.5), the negative correlation property of the RRS, and another concentration bound from prior work.

A similar "clean event" (with a similar proof) concerns the total resource consumption of the algorithm. We condition on both clean events, and perform the rest of the analysis via a "deterministic" argument not involving probabilities. In particular, we use the second "clean event" to guarantee that the algorithm never runs out of resources.

We use negative correlation via a rather delicate argument. We extend the concentration bound in Theorem 2.1 to a

random process that evolves over time, and only assumes that property (2.3) holds within each round conditional on the history. For a given round, we start with a negative correlation property of S_t and construct another family of random variables that conditionally satisfies (2.3). The extended concentration bound is then applied to this family. The net result is a concentration bound for $\sum_t \boldsymbol{\mu}_t^+ \cdot S_t$ *as if* we had $n \times T$ independent random variables there.

3.2 Running time of the algorithm

The algorithm does two computationally intensive steps in each round: solves the linear program ($\mathrm{LP}_{\mathrm{ALG}}$) and runs the RRS. For matroid constraints, the RRS from Chekuri et al. [2010] has $O(n^2)$ running time. Hence, in the general case the computational bottleneck is solving the LP, which has n variables and $O(2^n)$ constraints. Matroids are known to admit a polynomial-time seperation oracle [*e.g.,* see Schrijver, 2002]. It follows that the entire set of constraints in $\mathrm{LP}_{\mathrm{ALG}}$ admits a polynomial-time separation oracle, and therefore we can use the Ellipsoid algorithm to solve $\mathrm{LP}_{\mathrm{ALG}}$ in polynomial time. For some classes of matroid constraints the LP is much smaller: *e.g.,* for cardinality constraints (just $d+1$ constraints) and for traversal matroids on bipartite graphs (just $2n+d$ constraints). Then near-linear-time algorithms can be used.

Our algorithm works under any negatively correlated RRS. We can use this flexibility to improve the per-round running time for some special cases. (Making decisions extremely fast is often critical in practical applications of bandits [*e.g.,* see Agarwal et al., 2016].) We obtain near-linear per-round running times for cardinality constraints and partition matroid constraints. Indeed, $\mathrm{LP}_{\mathrm{ALG}}$ can be solved in near-linear time, as mentioned above, and we can use a negatively correlated RRS from [Gandhi et al., 2006] which runs in linear time. These classes of matroid constraints are important in our applications (see Section 4).

4 Applications and special cases

Let us discuss some notable examples of SemiBwK (which generalize some of the numerous applications listed in Badanidiyuru et al. [2013a]). Our results for these examples improve exponentially over a naive application of the BwK framework. Compared to what can be derived from [Agrawal and Devanur, 2014a, 2016], our results feature a substantially better dependence on parameters, a much better per-round running time, and apply to a wider range of parameters. However, we leave open the possibility that the regret bounds can be improved for some special cases.

Dynamic pricing. The dynamic pricing application is as follows. The algorithm has d products on sale with limited supply: for simplicity, B units of each. Following Besbes and Zeevi [2012], we allow supply constraints *across* products, *e.g.,* a "gadget" that goes into multiple products. In each round t, an agent arrives (who can buy any subset of the products), the algorithm chooses a vector of prices $p_t \in [0,1]^d$ to offer the agent, and the agent chooses what to buy at these prices. For simplicity, the agent is interested in buying (or is only allowed to buy) at most one item of each product. The agent has a valuation vector over products, so that the agent buys a given product if and only if her valuation for this product is at least as high as the offered price. The entire valuation vector is drawn as an independent sample from a fixed and unknown distribution (but valuations may be correlated across products). The algorithm maximizes the total revenue from sales.

To side-step discretization issues, we assume that prices are restricted to a known finite subset $S \subset [0,1]$. Achieving general regret bounds without such restriction appears beyond reach of the current techniques for BwK.[8]

To model it as a SemiBwK problem, the set of atoms is all price-product pairs. The combinatorial constraint is that at most one price is chosen for each product. (If an action does not specify a price for some product, the default price is used.) This is a "partition matroid" constraint, see Appendix B. Rewards correspond to revenue from sales, and resources correspond to inventory constraints.

We obtain regret $\tilde{O}(d\sqrt{dB|S|} + \sqrt{T|S|})$ using Corollary 3.2, whenever $B > \tilde{\Omega}(n + \sqrt{nT})$. This is because $\mathrm{OPT} \leq dB$, since that is the maximum number of products available, and the number of atoms is $n = d|S|$.

For comparison, results of [Agrawal and Devanur, 2014a, 2016] apply only when $B > \sqrt{n}\,T^{3/4}$, and yield regret bound of $\tilde{O}(d^3|S|^2\sqrt{T})$.[9] Thus, our regret bounds feature a better dependence on the number of allowed prices $|S|$ (which can be very large) and the number of products d. Further, our regret bounds hold in a meaningful way for the much larger range of values for budget B.

For a naive application of the BwK framework, arms correspond to every possible realization of prices for the d products. Thus, we have $|S|^d$ arms, with a corresponding exponential blow-up in regret.

Dynamic assortment. The dynamic assortment problem is similar to dynamic pricing in that the algorithm is selling d products to an agent, with a limited inventory B of each product, and is interested in maximizing the total revenue from sales. As before, agents can have arbitrary val-

[8] Prior work on dynamic pricing with limited supply [*e.g.,* Besbes and Zeevi, 2009, Babaioff et al., 2015, Badanidiyuru et al., 2013a] achieves regret bounds without restricting itself to a particular finite set of prices, but only for a simple special case of (essentially) a single product.

[9] We obtain this by plugging in $\mathrm{OPT} \leq dB$ and $n = d|S|$ into their regret bound. For dynamic pricing the total per-resource consumption is bounded by 1, so we can apply their results without rescaling the consumption.

uation vectors, drawn from a fixed but unknown distribution. However, the algorithm chooses which products to offer, whereas all prices are fixed externally. There is a large number of products to choose from, and any subset of $k \ll d$ of them can be offered in any given round.

To model this as SemiBwK, atoms correspond to products, and actions correspond to subsets of at most k atoms. The combinatorial constraint forms a matroid (see Appendix B). Rewards correspond to sales, and resources correspond to products, as in dynamic pricing. Since $\mathrm{OPT} \leq \min(dB, kT)$, Corollary 3.2 yields regret $\tilde{O}(k\sqrt{dT})$ when $B > \Omega(T)$, and regret $\tilde{O}(d\sqrt{dB} + \sqrt{dT})$ in general.

In a naive application of BwK, arms are subsets of k products. Hence, we have $O(d^k)$ arms. The other parameters of the problem remain the same. This leads to regret bound $\tilde{O}(d\sqrt{Bd^k})$, with an exponential dependence on k.

Repeated auctions. Consider a repeated auction with adjustable parameters, *e.g.,* repeated second-price auction with reserve price that can be adjusted from one round to another. While prior work [Cesa-Bianchi et al., 2013, Badanidiyuru et al., 2013a] concerned running one repeated auction, we generalize this scenario to multiple repeated auctions with shared inventory (*e.g.,* the same inventory may be sold via multiple channels to different audiences).

More formally, the auctioneer is running r simultaneous repeated auctions to sell a shared inventory of d products, with limited supply B of each product (*e.g.,* different auctions can cater to different audiences). Each auction has a parameter which the algorithm can adjust over time. We assume that this parameter comes from a finite domain $S \subset [0, 1]$. For simplicity, assume the auctions are synchronized with one another. As in prior work, we assume that in every round of each auction a fresh set of participants arrives, sampled independently from a fixed joint distribution, and only a minimal feedback is observed: the products sold and the combined revenue.

Following prior work [Cesa-Bianchi et al., 2013, Badanidiyuru et al., 2013a], we only assume minimal feedback: for each auction, what were the products sold and what was the combined revenue from this auction. In particular, we do not assume that the algorithm has access to participants' bids. Not using participants' bids is desirable for privacy considerations, and in order to reduce the participants' incentives to game the learning algorithm.

To model this problem as SemiBwK, atoms are all auction-parameter pairs. The combinatorial constraint is that an action must specify at most one parameter value for each auction. This corresponds to partition matroid constraints, see Appendix B. There is a "default parameter" for each auction, in case an action does not specify the parameter. We have a resource for each product being auctioned. For simplicity, each product has supply of B. Note that $\mathrm{OPT} \leq dB$ and number of atoms is $n = r|S|$. Hence, our main result yields regret $\tilde{O}(d\sqrt{r|S|B} + \sqrt{r|S|T})$.

A naive application of the BwK framework would have arms that correspond to all possible combinations of parameters, for the total of $O(|S|^r)$ arms. Again, we have an exponential blow-up in regret. Alternatively, one may try running r seperate instances of BwK, one for each auction, but that may result result in budgets being violated since the items are shared across the auctions and it is unclear a priori how much of each item will be sold in each auction.

One can also consider a "flipped" version of the previous example, where the algorithm is a bidder rather than the auction maker. The bidder participates in r repeated auctions, *e.g.,* ad auctions for different keywords. We assume a stationary environment: bidder's utility from a given bid in a given round of a given auction is an independent sample from a fixed but unknown distribution. The only limited resource here is the bidder's budget B. Bids are constrained to lie in a finite subset S.

To model this as SemiBwK, atoms correspond to the auction-bid pairs. The combinatorial constraint is that each action must specify at most one bid for each auction. (There is a "default bid" for each auction in case an action does not specify the bid for this auction.) There is exactly one resource, which is money and the total budget is B. Note that the number of atoms is $n = r|S|$. Hence, our main result yields regret $\tilde{O}(\mathrm{OPT}\sqrt{r|S|/B} + \sqrt{r|S|T})$.

A naive application of BwK would have arms that correspond to all possible combinations of bids, for the total of $O(|S|^r)$ arms; so we have an exponential blow-up in regret.

5 Numerical Simulations

We ran some experiments on simulated datasets in order to compare our algorithm, SemiBwK-RRS, with some prior work that can be used to solve SemiBwK:

- the primal-dual algorithm for BwK from Badanidiyuru et al. [2013a], denoted pdBwK.
- an algorithm for combinatorial semi-bandits with a matroid constraint: "Optimistic Matroid Maximization" from Kveton et al. [2014], denoted OMM.
- the linear-contextual BwK algorithm from Agrawal and Devanur [2016], discussed in the Introduction, denoted linCBwK.

To speed up the computation in linCBwK, we used a heuristic modification suggested by the authors in a private communication. This modification did not substantially affect average rewards in our preliminary experiments. We also made a heuristic improvement to our algorithm, setting $\epsilon = 0$ and $\alpha = 5$. We use the same value of α for the pdBwK algorithm as well.

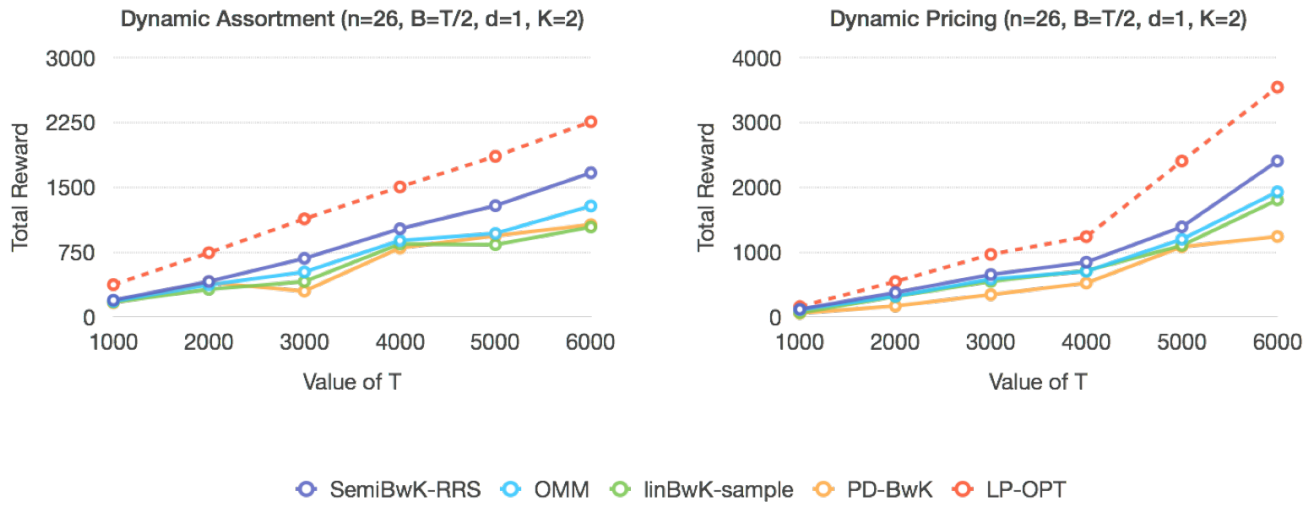

Figure 1: Experimental Results for Dynamic Assortment (left) and Dynamic Pricing (right) problems for $n = 26$.

Problem instances. We did not attempt to comprehensively cover the huge variety of problem instances in `SemiBwK`. Instead, we focus on several representative special cases. Below we describe experiments with two applications from Section 4. We also experimented with some other special cases, with qualitatively similar results; more details can be found in the full version.

The first experiment is on dynamic assortment. We have n products, and for each product i there is an atom i and a resource i. The (fixed) price for each product is generated as an independent sample from $U_{[0,1]}$, a uniform distribution on $[0, 1]$. At each round, we sample the buyers's valuation from $U_{[0,1]}$, independently for each product. If the valuation for a given product is greater than the price, one item of this product is sold (and then the reward for atom i is the price, and consumption of resource i is 1). Else, we set reward for atom i and consumption for resource i to be 0.

The second experiment is on dynamic pricing with two products. We have $n/2$ allowed prices, uniformly spaced in the $[0, 1]$ interval. Recall that atoms correspond to price-product pairs, for the total of n atoms. In each round t, the valuation $v_{t,i}$ for each product i is chosen independently from a normal distribution $\mathcal{N}(v_i^0, 1)$ truncated on $[0, 1]$. The mean valuation v_i^0 is drawn (once for all rounds) from $U_{[0,1]}$. If $v_{t,i}$ is greater than the offered price p, one item of this product is sold. Then reward for the corresponding atom (p, i) is the price p, and consumption of product i is 1. If there is no sale for this product, the reward and consumption for each atom (p, i) is set to 0.

Experimental setup and results. We choose various values of n, B and T and run our algorithms on the above two datasets assuming both a uniform matroid constraint and a partition matroid constraint. We choose $n \in \{6, 26\}$, $T \in \{1000, 2000, 3000, 4000, 5000, 6000\}$ and $B = T/2$. The maximum number of atoms in any action is set to $K = 2$. For a given algorithm, dataset and configuration of n and T, we simulate each algorithm for 20 independent runs and take the average. We calculate the total reward obtained by the algorithm at the end of T time-steps.

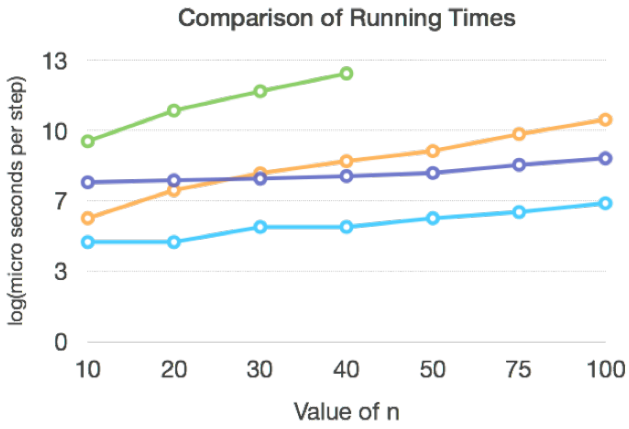

Figure 2: Variation of per-step running times as n increases for the various algorithms.

Figure 3 shows results for $n = 26$. Our algorithm achieves the best total reward among the competitors. As a benchmark, we included the performance of the fractional allocation in $\mathrm{LP}_{\mathrm{OPT}}$.

Additional experiment. `linCBwK` and `pdBwK` have running times proportional to the number of actions. We ran an additional experiment which compared per-step running times. We first calculate the average running time for every 10 steps and take the median of 50 such runs. For both Uniform matroid and Partition matroid, we run the faster RRS due to Gandhi et al. [2006]. See Figure 2 for results.

Acknowledgements. Karthik would like to thank Aravind Srinivasan for some useful discussions.

References

A. Agarwal, S. Bird, M. Cozowicz, M. Dudik, J. Langford, L. Li, L. Hoang, D. Melamed, S. Sen, R. Schapire, and A. Slivkins. Multiworld testing: A system for experimentation, learning, and decision-making, 2016. A white paper, available at `https://github.com/Microsoft/mwt-ds/raw/master/images/MWT-WhitePaper.pdf`.

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

S. Agrawal and N. R. Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006. ACM, 2014b.

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

S. Agrawal, N. R. Devanur, and L. Li. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. In *29th Conf. on Learning Theory (COLT)*, 2016.

V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

A. Asadpour, M. X. Goemans, A. Madry, S. O. Gharan, and A. Saberi. An o (log n/log log n)-approximation algorithm for the asymmetric traveling salesman problem. In *SODA*, volume 10, pages 379–389. SIAM, 2010.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

M. Babaioff, S. Dughmi, R. D. Kleinberg, and A. Slivkins. Dynamic pricing with limited supply. *ACM Trans. on Economics and Computation*, 3(1):4, 2015. Special issue for *13th ACM EC*, 2012.

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

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

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

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

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

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

S. Bubeck and N. Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5 (1), 2012.

N. Cesa-Bianchi, C. Gentile, and Y. Mansour. Regret minimization for reserve prices in second-price auctions. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2013.

C. Chekuri, J. Vondrak, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 575–584. IEEE, 2010.

C. Chekuri, J. Vondrák, and R. Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1080–1097. SIAM, 2011.

W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 151–159. JMLR Workshop and Conference Proceedings, 2013.

R. Combes, C. Jiang, and R. Srikant. Bandits with budgets: Regret lower bounds and optimal algorithms. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):245–257, 2015a.

R. Combes, M. S. T. M. Shahi, A. Proutiere, et al. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015b.

Y. Gai, B. Krishnamachari, and R. Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–9. IEEE, 2010.

Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations, Oct. 2012.

R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3):324–360, 2006.

J. Gittins, K. Glazebrook, and R. Weber. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, 2011.

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

A. Gupta, R. Krishnaswamy, M. 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.

A. György, T. Linder, G. Lugosi, and G. Ottucsák. The on-line shortest path problem under partial monitoring. *J. of Machine Learning Research (JMLR)*, 8:2369–2403, 2007.

R. Impagliazzo and V. Kabanets. Constructive proofs of concentration bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 617–631. Springer, 2010.

S. Katariya, B. Kveton, C. Szepesvári, and Z. Wen. DCM bandits: Learning to rank with multiple clicks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1215–1224, 2016.

R. Kleinberg, A. Slivkins, and E. Upfal. Bandits and experts in metric spaces. Working paper, published at `http://arxiv.org/abs/1312.1277`, 2015. Merged and revised version of conference papers in *ACM STOC 2008* and *ACM-SIAM SODA 2010*.

A. Krishnamurthy, A. Agarwal, and M. Dudík. Contextual semibandits via supervised learning oracles. In *29th Advances in Neural Information Processing Systems (NIPS)*, 2016.

B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In N. L. Zhang and J. Tian, editors, *UAI*, pages 420–429. AUAI Press, 2014.

B. Kveton, C. Szepesvari, Z. Wen, and A. Ashkan. Cascading bandits: Learning to rank in the cascade model. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 767–776. JMLR Workshop and Conference Proceedings, 2015a.

B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvri. Tight regret bounds for stochastic combinatorial semi-bandits. In G. Lebanon and S. V. N. Vishwanathan, editors, *AISTATS*, JMLR Workshop and Conference Proceedings. JMLR.org, 2015b.

T. L. Lai and H. Robbins. Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.

P. Raghavan and C. D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

A. Rakhlin and K. Sridharan. BISTRO: an efficient relaxation-based method for contextual bandits. In *33nd Intl. Conf. on Machine Learning (ICML)*, 2016.

H. Robbins. Some Aspects of the Sequential Design of Experiments. *Bull. Amer. Math. Soc.*, 58:527–535, 1952.

K. A. Sankararaman and A. Slivkins. Semi-bandits with knapsacks. *CoRR*, abs/1705.08110, 2017. URL `http://arxiv.org/abs/1705.08110`.

A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2002.

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

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

A. Slivkins and J. W. Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges. *SIGecom Exchanges*, 12(2), December 2013.

V. Syrgkanis, A. Krishnamurthy, and R. E. Schapire. Efficient algorithms for adversarial contextual learning. In *33nd Intl. Conf. on Machine Learning (ICML)*, 2016.

W. 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.

L. Tran-Thanh, A. Chapman, E. M. de Cote, A. Rogers, and N. R. Jennings. ϵ-first policies for budget-limited multi-armed bandits. In *24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1211–1216, 2010.

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

Z. Wang, S. Deng, and Y. Ye. Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331, 2014.

Z. Wen, B. Kveton, and A. Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In F. R. Bach and D. M. Blei, editors, *ICML*, JMLR Workshop and Conference Proceedings, pages 1113–1122. JMLR.org, 2015.

D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

Y. Xia, W. Ding, X.-D. Zhang, N. Yu, and T. Qin. Budgeted bandit problems with continuous random costs. In *Asian Conference on Machine Learning*, pages 317–332, 2016a.

Y. Xia, T. Qin, W. Ma, N. Yu, and T.-Y. Liu. Budgeted multi-armed bandits with multiple plays. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2210–2216. AAAI Press, 2016b. ISBN 978-1-57735-770-4. URL http://dl.acm.org/citation.cfm?id=3060832.3060930.

S. Zong, H. Ni, K. Sung, N. R. Ke, Z. Wen, and B. Kveton. Cascading bandits for large-scale recommendation problems. 2016.