A Lyapunov-Based Methodology for Constrained Optimization with Bandit Feedback

Semih Cayci,[1*] **Yilin Zheng,**[2*] **Atilla Erilmaz**[2]

[1]Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801.
[2] Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210
scayci@illinois.edu, zheng.1443@osu.edu, eryilmaz.2@osu.edu

Abstract

In a wide variety of applications including online advertising, contractual hiring, and wireless scheduling, the controller is constrained by a stringent budget constraint on the available resources, which are consumed in a random amount by each action, and a stochastic feasibility constraint that may impose important operational limitations on decision-making. In this work, we consider a general model to address such problems, where each action returns a random reward, cost, and penalty from an unknown joint distribution, and the decision-maker aims to maximize the total reward under a budget constraint B on the total cost and a stochastic constraint on the time-average penalty. We propose a novel low-complexity algorithm based on Lyapunov optimization methodology, named `LyOn`, and prove that for K arms it achieves $O(\sqrt{KB\log B})$ regret and zero constraint-violation when B is sufficiently large. The low computational cost and sharp performance bounds of `LyOn` suggest that Lyapunov-based algorithm design methodology can be effective in solving constrained bandit optimization problems.

1 Introduction

Multi-armed bandits (MAB) have been predominantly used to model exploration-and-exploitation problems since its inception (Robbins 1952; Lai and Robbins 1985; Berry and Fristedt 1985). As a consequence of the universality of the dilemma, bandit algorithms have found a broad range of applications from medical trials and adaptive routing to server allocation (Bubeck, Cesa-Bianchi et al. 2012). In many applications of interest, the controller is required to satisfy multiple constraints while achieving the optimal expected total reward under a given finite budget.

To take one example, in fair resource allocation problems, such as task scheduling or contractual hiring, each arm (e.g., a user or a social group) must receive at least a given fraction of the total budget (e.g., total time) while maximizing the total reward. Other examples from diverse domains, such as wireless resource allocation, online advertising, etc., also take this form (see Section 2 for more discussion). In order to solve fundamental learning applications such as these, a fast and effective constrained bandit optimization framework is required.

This motivates us in this paper to formulate and solve the following budget-constrained bandit problem in a stochastic setting. Each arm pull takes a random and arm-dependent resource (e.g. time, energy, etc.) from the budget, and the decision-making process continues until the total consumed resource exceeds a given budget $B > 0$. At the end of each arm pull, the controller receives a random reward and a random penalty. The objective of the controller is to maximize the expected total reward subject to an inequality constraint on the average penalty per unit resource consumption.

1.1 Main Contributions

In this work, we tackle the aforementioned general constrained optimization problem with bandit feedback, and propose a novel Lyapunov-based design methodology to develop efficient algorithms that achieve sharp convergence results. Our main contributions can be summarized as follows:

- *General model:* We consider a generic constrained bandit optimization problem (in Section 2), which: (i) incorporates random costs for each action; (ii) is subject to a stringent budget (knapsack) constraint; and (iii) has stochastic feasibility constraints as required by many applications.
- *Lyapunov methodology for bandit optimization:* Based on a Lyapunov-drift minimization technique from stochastic control, we design novel low-complexity bandit algorithms with provably sharp convergence properties. This approach suggests a general design methodology (outlined in Section 3) that can be utilized in other constrained bandit optimization scenarios.
- *Analysis techniques:* We also employ new analysis techniques (in Section 5) for the reward maximization problem subject to stochastic and knapsack constraints based on a combination of renewal theory, stochastic control, and bandit optimization.

1.2 Related Work

Knapsack-constrained bandit problem was considered in (Badanidiyuru, Kleinberg, and Slivkins 2013), where the objective of the controller is to maximize the expected total reward under a stringent budget constraint. The authors proposed a learning algorithm with $O(\sqrt{B})$ problem-independent regret. Bandit algorithms with $O(\log(B))$

[*]These authors contributed equally.

problem-dependent regret bounds were proposed under various extensions of the knapsack-constrained bandit models (Tran-Thanh et al. 2012; Flajolet and Jaillet 2015; Xia et al. 2015, 2016; Cayci, Eryilmaz, and Srikant 2019, 2020). These works differ from ours in that, while the controller is constrained by stringent knapsack constraints, stochastic feasibility constraints are not accommodated.

As an extension of (Badanidiyuru, Kleinberg, and Slivkins 2013), finite-armed bandit problem with stochastic feasibility constraints was considered in (Agrawal and Devanur 2014), and a UCB-type algorithm with $O(\sqrt{B})$ regret and $O(1/\sqrt{B})$ constraint-violation was proposed. This setting was extended to episodic Markov decision processes in (Qiu et al. 2020), and similar order results for regret and constraint-violation were obtained. In these models, each action incurs a unit cost. Furthermore, the proposed algorithms in (Agrawal and Devanur 2014) require solving a convex optimization / linear programming problem at each stage. Our model accommodates random cost, which is subject to a knapsack constraint, in addition to the stochastic feasibility constraint. Based on Lyapunov optimization theory, we develop computationally-efficient iterative algorithms .

Lyapunov optimization methods have been widely used in stochastic network optimization and queueing systems (see (Neely 2012, 2010; Georgiadis, Neely, and Tassiulas 2006) and references therein). This methodology was later used in convex optimization problems (Yu and Neely 2020) with known gradients. In these approaches, a predominant assumption is that the random system state is known to the controller prior to its decision, therefore the existing methods do not work for online learning setting where the controller does not have the system state or the system statistics before making a decision.

Lyapunov optimization methods were first used in the context of online learning in (Cayci, Gupta, and Eryilmaz 2020), where the goal of the controller is to maximize the total utility as a function of each arm's time-average reward subject to a knapsack constraint under delayed experts feedback. Our work extends (Cayci, Gupta, and Eryilmaz 2020) in that we consider bandit feedback and also incorporate stochastic feasibility constraints in this paper. Some recent works (Liu et al. 2020, 2021), which utilized Lyapunov-drift methods for online learning, studied the online-dispatching and linear bandits with cumulative constraints. Our work substantially differs from these works in that we incorporate knapsack budget constraints and random costs per arm selection.

2 Constrained Reward Maximization Problem with Bandit Feedback

We consider a finite-armed bandit problem with $K > 1$ arms, and the set of arms denoted by $\mathbb{K} = \{1, 2, \dots, K\}$. If arm k is chosen at n^{th} epoch, it incurs a cost of $X_{n,k}$, yields a reward of $R_{n,k}$, and returns a penalty of $Y_{n,k}$, where the outcome of the joint random vector $(X_{n,k}, R_{n,k}, Y_{n,k})$ is learned via bandit feedback at the end of each arm decision. We assume that the random process $\{(X_{n,k}, R_{n,k}, Y_{n,k}) : n \geq 1\}$ is independent and identically distributed over n, and independent across different arms for all $k \in \mathbb{K}$. For simplicity, we assume that $X_{n,k}, R_{n,k}, Y_{n,k} \in [0,1]$ for all n, k, which can be easily extended to general sub-Gaussian random variables by using the same techniques used in this paper. The controller has a total budget $B > 0$ at the beginning of the process, and tries to maximize the expected cumulative reward under time-average constraints on the penalties by sampling the arms wisely under this budget constraint.

Note that stochastic constraints and budget constraints imply completely different system dynamics. Violation of a budget constraint immediately stops the decision-making process. On the other hand, the stochastic constraints are aimed to be satisfied asymptotically, while instantaneous violations do not stop the decision-making process.

First, we introduce the causal policy space.

Definition 1 (Causal Policy)**.** *Let π be a policy that yields a sequence of arm pulls $\{I_n^\pi \in \mathbb{K} : n \geq 1\}$. Under π, the history until epoch n is the following filtration:*

$$\mathcal{F}_n^\pi = \sigma(\{(I_j^\pi, X_{j,k}, R_{j,k}, Y_{j,k}) : I_j^\pi = k, 1 \leq j \leq n\}), \tag{1}$$

where $\sigma(Z)$ denotes the sigma-field of a random variable Z. We call an algorithm π causal if π is non-anticipating, i.e., $\{I_n^\pi = k\} \in \mathcal{F}_{n-1}^\pi$ for all k, n.

The set of all causal policies is denoted as Π. We denote the variables at epoch n under policy π as $X_n^\pi = X_{n,I_n^\pi}$, $R_n^\pi = R_{n,I_n^\pi}$ and $Y_n^\pi = Y_{n,I_n^\pi}$. The total cost incurred in n epochs under an causal policy $\pi \in \Pi$ is a controlled random walk which is defined as $S_n^\pi = \sum_{i=1}^n X_i^\pi$. The decision process under a policy π continues until the budget B is depleted. We assume that the reward corresponding to the final epoch during which the budget is depleted is gathered by the controller. Thus, the total number of pulls under π is a random variable that is defined as follows:

$$N^\pi(B) = \inf\Big\{n \geq 1 : S_n^\pi > B\Big\}. \tag{2}$$

Note that the total number of pulls $N^\pi(B)$ is a stopping time adapted to the filtration $\{(\mathcal{F}_n^\pi) : n \geq 0\}$. Accordingly, the cumulative reward under a policy π can be written as follows:

$$\mathtt{REW}^\pi(B) = \sum_{n=1}^{N^\pi(B)} R_n^\pi. \tag{3}$$

Then, we can write the generic problem formulation considered in this paper as follows:

$$\begin{aligned} \sup_{\pi \in \Pi} \quad & \mathbb{E}[\mathtt{REW}^\pi(B)], \\ \text{subject to:} \quad & \mathbb{E}\left[\frac{1}{B}\sum_{n=1}^{N^\pi(B)} Y_n^\pi\right] \leq c. \end{aligned} \tag{4}$$

Definition 2 ((Pseudo) Regret and constraint-violation)**.** *Let $\pi_{\mathtt{Opt}}$ be the solution of* (4) *and* $\mathtt{OPT}(B) = \mathbb{E}[\mathtt{REW}^{\pi_{\mathtt{Opt}}}(B)]$*. For any causal policy $\pi \in \Pi$ and a budget $B > 0$ level, the (pseudo) regret,* $\mathtt{REG}^\pi(B)$*, and constraint-violation,* $D^\pi(B)$*,* *are defined as follows:*

$$\mathtt{REG}^{\pi}(B) = \mathtt{OPT}(B) - \mathbb{E}[\mathtt{REW}^{\pi}(B)], \tag{5}$$

$$D^{\pi}(B) = \mathbb{E}\left[\frac{1}{B}\sum_{n=1}^{N^{\pi}(B)} Y_n^{\pi}\right] - c. \tag{6}$$

The objective of this paper is to design low-complexity bandit algorithms that are guaranteed to give a low regret and a vanishing constraint-violation level that decays rapidly with the budget level B, in the absence of any statistical knowledge on the costs, rewards, and penalties. The generic problem (4) has numerous applications in communications, control, operations research and management, whereby optimal decision-making under data scarcity and uncertainty is common. Next, we provide a detailed application in wireless scheduling in next generation networks, and refer the reader to Appendix A for other example applications for contractual hiring, and online advertising.

Application to Next Generation Wireless Scheduling with Quality-of-Service Guarantees: Next-generation wireless technologies are required to serve a highly dynamic population of users with stringent quality-of-service (QoS) guarantees (such as low delay (Khalek, Caramanis, and Heath 2014)) over ultra-high frequency bands with non-traditional statistical and temporal characteristics (Rappaport et al. 2015). As such, existing estimation and allocation techniques that rely strongly on persistent users and slowly-changing nature and known statistical models of channel conditions are no longer suitable for use in this new ultra-wideband communication paradigm. The controller is required to learn how to optimize the throughput subject to QoS guarantees by using the ARQ (bandit) feedback received after each transmission.

This calls for the design of time/energy-constrained point-to-point communication solutions over K parallel memoryless channels with unknown and diverse statistical characteristics. In particular, each connection starts with a total time or energy budget of B units. The transmission of n^{th} packet over the k^{th} channel consumes a random amount of $X_{n,k}$ resource (e.g., transmission time or energy), yields a reward (e.g., throughput) $R_{n,k}$ and incurs a penalty $Y_{n,k}$ upon completion of transmission. In this context, $Y_{n,k}$ is a generic penalty that will be used in modeling time-average quality-of-service guarantees. As an example, consider delay-constrained communication, where the arriving packets should be transmitted in a timely manner. Then, for a given deadline level $d \in [0,1]$, we let $Y_{n,k} = \mathbb{I}\{X_{n,k} > d\}$, which counts the number of packets that are delayed for more than d time units. For a given time or energy budget B and a quality-of-service constraint c, the optimization problem (4) leads to throughput maximization subject to a guarantee on the time-average number of delayed packets. Note that many other QoS criteria, such as the fraction of dropped packets, can be modeled in a similar manner, which implies the generality of this approach.

3 Outline of the Lyapunov-Based Design Methodology and Main Results

In this work, we develop a low-complexity online algorithm for solving the generic constrained reward maximization problem (4) by employing a Lyapunov-drift minimization methodology. Since this methodology may be of independent value, in this section we provide an outline of its main steps along with an informal discussion of the key results we obtained under them.

(i) Characterization of the Asymptotically-Optimal Stationary Randomized Oracle: The optimization problem described in Section 2 is a variant of the unbounded knapsack problem, and it is known that similar stochastic control problems are PSPACE-hard (Badanidiyuru, Kleinberg, and Slivkins 2013; Papadimitriou and Tsitsiklis 1999). In Section 4 we propose a *stationary randomized* policy π^* in Definition 4 that achieves (see Proposition 1) $O(1)$ regret and $O(1/B)$ constraint-violation gap. This proves that the stationary policy is asymptotically optimal as the budget B goes to infinity.

Our Lyapunov-based policy design, developed in Section 5, are broken into the following two steps:

(ii) Offline Lyapunov-Drift-Minimizing Policy Design: We first consider in Section 5.1 the 'offline' setting with known reward, cost, and penalty statistics. There, we introduce a virtual queue $\{Q_n^{\pi}\}$ that is updated as: $Q_{n+1}^{\pi} = \max\{0, Q_n^{\pi} + Y_n^{\pi} - (c-\delta)X_n^{\pi}\}$, with a design choice $\delta \in [0, c)$, which keeps track of the constraint-violation level under policy π over decisions $n \geq 1$. Then, under this queue dynamics, we propose a quadratic Lyapunov drift-minimizing policy $\pi_{\mathtt{LyOff}}$ in Definition 6 that achieves (cf. Proposition 2) $O((\delta + 1/V)B)$ regret and $O(V/B - \delta)$ constraint-violation gap, where $V > 0$ is a design parameter. With the particular selection of $V = \Theta(\sqrt{B})$ and $\delta = \Theta(1/\sqrt{B})$, we can guarantee $O(\sqrt{B})$ regret and zero constraint-violation for $\pi_{\mathtt{LyOff}}$ with sufficiently large B.

(iii) Online Lyapunov-Drift-Minimizing Policy Design: Then, in Section 5.2, we return to the original 'online' setting with unknown statistics, and develop a low-complexity empirical Lyapunov-drift minimizing policy $\pi_{\mathtt{LyOn}}$ that integrates confidence bounds of proposed empirical estimator with the queueing dynamics from the offline case. Then, the main result of the paper (cf. Theorem 1) establishes that $\pi_{\mathtt{LyOn}}$ achieves $O(\sqrt{KB\log B} + B(1+\delta\log B)/V + B\delta + K\log B)$ regret and $O(K\log B/B + V/B - \delta)$ constraint-violation level. With the particular selection of design parameters as $V = \Theta(\sqrt{B\log B})$ and $\delta = \Theta(\sqrt{\log B/B})$, we guarantee $O(\sqrt{KB\log B})$ regret and zero constraint-violation for $\pi_{\mathtt{LyOn}}$ with sufficiently large B.

The online analysis is especially complicated by the fact that the cumulative reward and penalty processes form stopped and controlled random walks. To address the associated challenge, we combine techniques from renewal theory and Martingale concentration inequalities (Wainwright 2019) to find a high probability upper bound for $N^{\pi}(B)$. Additionally, for the online policy with unknown statistics, we carefully integrate empirical concentration inequalities (Cayci, Eryilmaz, and Srikant 2020) with hitting time analysis for Martingales (Hajek 1982) as well as Lyapunov drift analysis (Neely 2012) to bound $\mathtt{REW}^{\pi}(B)$ and $D^{\pi}(B)$.

4 Asymptotically-Optimal Stationary Randomized Oracle Design

As a tractable benchmark, in this section, we consider approximation algorithms with provably good performance.

Definition 3 (Reward Rate and Penalty Rate)**.** *Consider a stationary randomized policy $\pi = \pi(\mathbf{p})$ for a given probability mass function $\mathbf{p} = (p_1, p_2, \ldots, p_K)$, which takes action k with probability p_k independent from the history. Then, under $\pi(\mathbf{p})$, the reward rate and penalty rate are defined as:*

$$r(\mathbf{p}) = \frac{\sum_{k\in\mathbb{K}} p_k \mathbb{E}[R_{1,k}]}{\sum_{k\in\mathbb{K}} p_k \mathbb{E}[X_{1,k}]}, \quad y(\mathbf{p}) = \frac{\sum_{k\in\mathbb{K}} p_k \mathbb{E}[Y_{1,k}]}{\sum_{k\in\mathbb{K}} p_k \mathbb{E}[X_{1,k}]}, \tag{7}$$

Intuitively, if an arm is chosen persistently according to the stationary randomized policy $\pi(\mathbf{p})$ until the budget $B > 0$ is depleted, the cumulative reward becomes $r(\mathbf{p})B + o(B)$ and cumulative penalty becomes $y(\mathbf{p})B + o(B)$. Moreover, whenever $\mathbb{E}[R_{1,k}^2] < \infty$ and $\mathbb{E}[Y_{1,k}^2] < \infty$ (trivially true for bounded random variables), the additive $o(B)$ term is $O(1)$ in both cases by Lorden's inequality (Asmussen 2008).

In the following, we prove that a stationary randomized policy achieves $O(1)$ optimality gap with constraint-violation vanishing at a rate of $O(1/B)$.

Definition 4 (Optimal Stationary Randomized Policy, π^*)**.** *Let $\mathbf{p}^*$ be the solution to the following optimization problem:*

$$\max_{\mathbf{p}\in\Delta_K} \left\{ r(\mathbf{p}), \quad \textit{subject to:} \quad y(\mathbf{p}) \le c \right\}.$$

where Δ_K is the K-dimensional probability simplex. The optimal stationary randomized policy*, denoted by π^*, pulls arm k with probability p_k^* independently at each epoch until the budget is depleted: $\mathbb{P}(I_n^{\pi^*} = k) = p_k^*$, for all $n \le N^{\pi^*}(B)$.*

The main result of this section is the following proposition, which implies that π^* is a good approximation algorithm for $\pi_{\mathtt{Opt}}$ for $B > 0$.

Proposition 1 (Optimality Gap for π^*)**.** *For the optimal static policy π^* for any given $B > 0$, the following regret and constraint-violation gap results hold:*

$$\mathtt{REG}^{\pi^*}(B) = O(1), \quad D^{\pi^*}(B) = O\Big(\frac{1}{B}\Big). \tag{8}$$

Therefore, π^* is asymptotically optimal, i.e. $\lim_{B\to\infty} \mathtt{REG}^{\pi^*}(B)/B = 0$ and $\lim_{B\to\infty} D^{\pi^*}(B) = 0$.

The proof of Proposition 1 can be found in Appendix C. In the next section, we will introduce a learning algorithm to achieve the performance of the optimal stationary randomized policy with low regret and constraint-violation.

5 Algorithm Design Based on Empirical Lyapunov Drift Minimization

In the previous section, we proved that the stationary randomized policy π^* achieves the optimality in offline setting with small optimality gap and constraint-violation, which implies it can be used as a benchmark for the design and analysis of learning algorithms. By using this, we will develop a dynamic learning algorithm based on the Lyapunov-drift-minimization approach. For details about this dynamic optimization approach in offline setting, see (Neely 2010, 2012).We refer to Section 1.2 for the detailed discussion of the differences from related works in this space.

We make two mild assumptions that are needed for the development and analysis of our design:

Assumption 1 (ϵ-Slater Condition)**.** *There exists an arm $k \in \mathbb{K}$ such that $\mathbb{E}[Y_{n,k} - cX_{n,k}] \le -\epsilon$ for some $\epsilon > 0$. We only need ϵ to be a positive lower-bound of the actual value.*

Assumption 1 is reasonable because for feasibility, either all arms should satisfy $\mathbb{E}[Y_{n,k} - cX_{n,k}] = 0$ for all k or Assumption 1 should hold, otherwise the constraint can never be satisfied once it is violated. Since $\mathbb{E}[Y_{n,k} - cX_{n,k}] = 0,\ \forall k \in \mathbb{K}$ is a trivial case, Assumption 1 is satisfied in almost all applications.

Assumption 2 (Bounded Moments)**.** *For all arms $k \in \mathbb{K}$, assume $\max_k \frac{\mathbb{E}[R_{1,k}]}{\mathbb{E}[X_{1,k}]} \le r_{\mathtt{max}} < \infty$ and $\max_k \frac{\mathbb{E}[Y_{1,k}]}{\mathbb{E}[X_{1,k}]} \le y_{\mathtt{max}} < \infty$. In addition, assume $\sigma^2 = \max_{k\in\mathbb{K}} \mathbb{E}\big[\big(Y_{1,k} - cX_{1,k}\big)^2\big] < 1$, and $\min_k \mathbb{E}[X_{1,k}] \ge \mu_{\mathtt{min}} > 0$. We only need $\mu_{\mathtt{min}}$ to be a lower bound and $r_{\mathtt{max}}$, $y_{\mathtt{max}}$ to be upper bounds of the actual values.*

This assumption is reasonable because otherwise the optimization problem in (4) would become either trivial or unsolvable. For bounded rewards and penalty between $[0, 1]$, $r_{\mathtt{max}}$ and $y_{\mathtt{max}}$ can be upper bounded by $1/\mu_{\mathtt{min}}$.

5.1 Offline Lyapunov-Drift Minimizing Policy `LyOff` Design

First, we consider the Lyapunov optimization methods in the offline setting with known first-order statistics by closely following (Neely 2012), while improving the results for finite-time performance by using the drift results in (Hajek 1982). As a measure of constraint-violation under a causal policy $\pi \in \Pi$, we define the variables Q_n^π recursively as follows:

$$Q_{n+1}^\pi = \max\Big\{0, Q_n^\pi + Y_n^\pi - (c - \delta)X_n^\pi\Big\}, \tag{9}$$

where $Q_0^\pi = 0$ and $\delta \in [0, c)$ is a fixed parameter that controls the tightness of the constraint. Note that $Q_{n+1}^\pi \in \mathcal{F}_n^\pi$ for all n since π is causal. Intuitively, the stability of $\{Q_n^\pi\}_n$ implies that the constraint is satisfied. The key metric for decision-making is the Lyapunov drift-plus-penalty ratio, which is defined in the following definition.

Definition 5 (Lyapunov Drift-plus-Penalty Ratio)**.** *For any given $V > 0$, under a causal policy π, the Lyapunov drift-plus-penalty ratio is defined as follows:*

$$\Psi_n(Q_n^\pi) = -V\frac{\mathbb{E}[R_n^\pi|\mathcal{F}_{n-1}^\pi]}{\mathbb{E}[X_n^\pi|\mathcal{F}_{n-1}^\pi]} + Q_n^\pi\frac{\mathbb{E}[Y_n^\pi|\mathcal{F}_{n-1}^\pi]}{\mathbb{E}[X_n^\pi|\mathcal{F}_{n-1}^\pi]}. \tag{10}$$

For any stationary randomized policy $\pi(\mathbf{p}_n)$, with $\mathbf{p}_n \in \mathcal{F}_{n-1}$, the Lyapunov drift-plus-penalty ratio becomes:

$$\Psi_n(Q_n^{\pi(\mathbf{p}_n)}) = -V\frac{\sum_{k=1}^K p_{n,k}\mathbb{E}[R_{n,k}]}{\sum_{k=1}^K p_{n,k}\mathbb{E}[X_{n,k}]} + Q_n^{\pi(\mathbf{p}_n)}\frac{\sum_{k=1}^K p_{n,k}\mathbb{E}[Y_{n,k}]}{\sum_{k=1}^K p_{n,k}\mathbb{E}[X_{n,k}]}. \tag{11}$$

Intuitively, in the offline setting where all first-order moments are known, a stationary randomized policy $\pi(\mathbf{p}_n)$ that minimizes (11) over all probability distributions in every epoch n, achieves a near-optimal trade-off between the cumulative reward and constraint-violation determined by the parameter $V > 0$ (Neely 2012). In the following, we outline this result in the offline setting, which will guide us in developing the online algorithm in Section 5.2.

Definition 6 (Offline Lyapunov-Drift-Minimizing Distribution)**.** *For any n, let $\mathbf{q}_n^*$ be defined as follows:*

$$\mathbf{q}_n^* \in \arg\min_{\mathbf{p}\in\Delta_K} \Psi_n(Q_n^{\pi(\mathbf{p})}). \tag{12}$$

The problem in (12) is an optimization problem over Δ_K, the K-dimensional probability simplex, which is computationally complex and can be solved by using algorithmic techniques in (Neely 2012). However, as it is shown in Proposition 5 in Appendix D, the optimal solution in our K-armed bandit setting is deterministic given the history $\mathcal{F}_{n-1}$. This allows us to define the offline Lyapunov-Drift Minimizing Policy $\pi_{\texttt{LyOff}}$ as in Algorithm 1.

Intuition: The policy $\pi_{\texttt{LyOff}}$ makes a balanced choice between the reward maximization and satisfying the constraints. For small Q_n, the controller selects the arm I_n with the highest drift-plus-penalty ratio so as to maximize the expected total reward under the budget constraints. If Q_n is large, then it means the constraint has been violated considerably, thus I_n is selected so as to reduce the penalty rate and hence violation level. Next, we prove finite-time performance bounds for $\pi_{\texttt{LyOff}}$.

Proposition 2 (Performance Bounds for $\pi_{\texttt{LyOff}}$)**.** *Suppose that Assumption 1 and Assumption 2 hold with positive ϵ, σ^2, $r_{\texttt{max}}$ and $\mu_{\texttt{min}}$. Then, given the budget B, for any $V > 0$ and $\delta \in [0, c)$, the regret and constraint-violation levels under $\pi_{\texttt{LyOff}}$ satisfy:*

$$\texttt{REG}^{\pi_{\texttt{LyOff}}}(B) = O\left(\frac{\sigma^2 B}{V\mu_{\texttt{min}}^2} + \frac{\delta r_{\texttt{max}} B}{\epsilon\mu_{\texttt{min}}^2}\right), \tag{13}$$

$$D^{\pi_{\texttt{LyOff}}}(B) = O\Big(\frac{1}{B} + \frac{V r_{\texttt{max}}}{B\mu_{\texttt{min}}\epsilon} - \frac{\delta}{\mu_{\texttt{min}}}\Big). \tag{14}$$

Specifically, let $V = v_0\sqrt{B}$ and $\delta = \delta_0/\sqrt{B}$ with some design parameters $\delta_0 > 0, v_0 > 0$. We can select $\delta_0 \in (\frac{r_{\texttt{max}}}{\epsilon}v_0, c\sqrt{B})$ such that for sufficiently large B,

$$\texttt{REG}^{\pi_{\texttt{LyOff}}}(B) = O\Big(\sqrt{B}\Big), \quad D^{\pi_{\texttt{LyOff}}}(B) = O\Big(\frac{-1}{\sqrt{B}}\Big). \tag{15}$$

The proof of Proposition 2 can be found in Appendix D.1. Proposition 2 establishes the fact that $\pi_{\texttt{LyOff}}$ policy achieves $O(\sqrt{B})$ regret and zero constraint-violation for sufficiently large B given the first-order statistics $\mathbb{E}[X_{n,k}], \mathbb{E}[R_{n,k}], \mathbb{E}[Y_{n,k}]$ for all arms $k \in \mathbb{K}$. The $\pi_{\texttt{LyOff}}$ policy will serve as a guide for our online learning algorithm, introduced next.

Algorithm 1: `LyOff` Algorithm

```
Input: $B, K, c, V, \delta,$
       $\mathbb{E}[X_{1,k}], \mathbb{E}[R_{1,k}], \mathbb{E}[Y_{1,k}]$
Initialize $Q_0 = 0$, $\texttt{cost} = 0$, $n = 1$
while $\texttt{cost} \le \texttt{B}$ do
    $\Psi_n(k, Q_n) = -V\frac{\mathbb{E}[R_{1,k}]}{\mathbb{E}[X_{1,k}]} + Q_n\frac{\mathbb{E}[Y_{1,k}]}{\mathbb{E}[X_{1,k}]}$
    $k_n = \arg\min_{k\in\mathbb{K}} \Psi_n(k, Q_n)$
    Select arm $I_n = k_n$.
    Observe $X_n, R_n, Y_n$.
    $Q_{n+1} = \max\{0, Q_n + Y_n - (c-\delta)X_n\}$
    $\texttt{cost} = \texttt{cost} + X_n$.
    $n = n + 1$.
end while
```

5.2 Online Lyapunov-Drift Minimizing Policy `LyOn` Design

A strong assumption in $\pi_{\texttt{LyOff}}$ was the a priori knowledge of the first-order statistics for all variables. Recall that in the learning problem, we do not have this knowledge. Instead, we must work with estimations by using the observed outcomes from bandit type feedback to learn the optimal decision. Furthermore, like all exploration-exploitation problems, the *online exploration* is a crucial component of the learning problem here as well. Optimizing this trade-off with low regret and constraint-violation is particularly challenging in this setting due to the knapsack-type budget constraints from random costs, as well as the random penalties in the constraint. In this section, we will design and analyze the `LyOn` Algorithms by combining tools from renewal theory, stochastic control, as well as bandit optimization to address these challenges for optimal learning.

Strategy: Our strategy will be to approximate the Lyapunov drift-plus-penalty ratio Ψ_n in equation (10) by using the empirical estimates for the first-order statistics. In order to encourage online exploration, we will use confidence bounds so that the index at the end will be a high-probability lower bound for Ψ_n. The following definitions will be needed to define the online algorithm.

Definition 7 (Confidence Radius)**.** *For any $n \ge 1$ and arm $k \in \mathbb{K}$, let $\mathcal{I}_n^\pi(k) = \{t \in [1, n] : I_t^\pi = k\}$, $T_k^\pi(n) = |\mathcal{I}_n^\pi(k)| = \sum_{t=1}^n \mathbb{I}\{I_t^\pi = k\}$ be the number of pulls for arm k under a policy π in the first n epochs. For a given $\alpha > 0$, the confidence radius for arm k is defined as:* $\texttt{rad}_k(n, \alpha) = \sqrt{\frac{2\alpha\log(n)}{T_k(n)}}$.

To ensure the confidence radius is small enough, we have an initial exploration phase that is controlled by a parameter β_0 which depends on $\epsilon, y_{\texttt{max}}$, and $\mu_{\texttt{min}}$. Specifically, we set

$\beta_0 = \frac{32\alpha(1+y_{\mathtt{max}})^2}{\mu_{\mathtt{min}}^2\epsilon^2}$ to guarantee the concentration event in Lemma 6 of Appendix D.

For a subset of indices $S \subset \mathbb{N}$ and a stochastic process $\{Z_n : n \in \mathbb{N}\}$, let $\widehat{\mathbb{E}}_S[Z] = \min\left\{1, \frac{1}{|S|}\sum_{t\in S} Z_t\right\}$, be the empirical mean estimator. Then, the empirical reward rate and empirical penalty rate under policy π after n epochs are defined as:

$$\widehat{r}^{\pi}_{n,k} = \frac{\widehat{\mathbb{E}}_{\mathcal{I}^{\pi}_n(k)}[R_k]}{\widehat{\mathbb{E}}_{\mathcal{I}^{\pi}_n(k)}[X_k]}, \quad \widehat{y}^{\pi}_{n,k} = \frac{\widehat{\mathbb{E}}_{\mathcal{I}^{\pi}_n(k)}[Y_k]}{\widehat{\mathbb{E}}_{\mathcal{I}^{\pi}_n(k)}[X_k]}, \quad k \in \mathbb{K}. \tag{16}$$

Definition 8 (Empirical Lyapunov Drift-Plus-Penalty Ratio)**.** *Let Q^{π}_n be the variable evolving under π as in (9). Then, the empirical Lyapunov drift-plus-penalty ratio at epoch n is defined as follows:*

$$\widehat{\Psi}_n(k, Q^{\pi}_n) = -V \cdot \widehat{r}^{\pi}_{n-1,k} + Q^{\pi}_n \cdot \widehat{y}^{\pi}_{n-1,k}, \tag{17}$$

where $V > 0$ is a design parameter. Define the empirical lower confidence bound for $\widehat{\Psi}_n(k, Q^{\pi}_n)$ as

$$\widehat{\Gamma}_n(k, Q^{\pi}_n) = \widehat{\Psi}_n(k, Q^{\pi}_n) - \mathtt{rad}_k(n-1,\alpha)\frac{V(1+\widehat{r}^{\pi}_{n-1,k})}{\widehat{\mathbb{E}}_{\mathcal{I}^{\pi}_{n-1}(k)}[X_k]} + \mathtt{rad}_k(n-1,\alpha)\frac{Q^{\pi}_n(1+\widehat{y}^{\pi}_{n-1,k})}{\widehat{\mathbb{E}}_{\mathcal{I}^{\pi}_{n-1}(k)}[X_k]} \tag{18}$$

With these definitions, the online Lyapunov-Drift Minimizing Algorithm LyOn is defined in Algorithm 2.

Algorithm 2: LyOn Algorithm

```
Input: $B, K, c, \alpha, V, \delta, \beta_0, \mu_{\mathtt{min}}$
Initialize $Q_0 = 0$, $\mathtt{cost} = 0$, $n = 1$
Select each arm $\lceil \beta_0 \log\left(\frac{2B}{\mu_{\mathtt{min}}}\right)\rceil$ times.
Update $n$, $\mathtt{cost}$, $\widehat{\Gamma}_n(k, Q_n)$ (eq. (18)).
while $\mathtt{cost} \le \mathtt{B}$ do
    $k_n = \arg\min_{k\in\mathbb{K}}\{\widehat{\Gamma}_n(k, Q_n)\}$
    Select arm $I_n = k_n$. Observe $X_n, R_n, Y_n$.
    $Q_{n+1} = \max\{0, Q_n + Y_n - (c-\delta)X_n\}$
    $\mathtt{cost} = \mathtt{cost} + X_n$.
    Update $\widehat{\Gamma}_n(k, Q_n)$ (eq. (18)).
    $n = n + 1$.
end while
```

Remark 1. Before we analyze it, we make the following observations about the LyOn Algorithm.

1. Lyon is an extremely low-complexity, iterative algorithm, whereby in every step a simple update is performed.
2. The index to be minimized in (18) is a high-probability lower bound for $\Psi_n(k, Q^{\pi_{\mathtt{LyOn}}}_n)$. Thus, given the available data $\mathcal{F}^{\pi_{\mathtt{LyOn}}}_{n-1}$, the algorithm makes an optimistic drift-minimizing arm selection in the face of uncertainty.
3. If $I^{\pi_{\mathtt{LyOn}}}_n = k$, then at least one of the following must be true: a) High confidence for arm k, large r_k and small $Q^{\pi_{\mathtt{LyOn}}}_n$. b) High confidence for arm k, large $Q^{\pi_{\mathtt{LyOn}}}_n$ and small y_k. c) Low confidence for arm k. As such, the LyOn Algorithm incentivizes online exploration by choosing arms with very low confidence.
4. The LyOn Algorithm extends the UCB-BwI Algorithm proposed in (Cayci, Eryilmaz, and Srikant 2019) to the non-trivial and useful cases with stochastic feasibility constraints. Note that $Q^{\pi_{\mathtt{LyOn}}}_n = 0$ if there is no constraint, thus the LyOn Algorithm reduces to the UCB-BwI Algorithm.

Theorem 1 (Performance Bounds for $\pi_{\mathtt{LyOn}}$)**.** *Suppose that Assumption 1 and Assumption 2 hold with positive ϵ, σ^2, $r_{\mathtt{max}}$, $y_{\mathtt{max}}$, and $\mu_{\mathtt{min}}$. Then, for any $V > 0$ and $\delta \in [0, c)$, the regret and constraint-violation levels under $\pi_{\mathtt{LyOn}}$ satisfy:*

$$\mathtt{REG}^{\pi_{\mathtt{LyOn}}}(B) = O\Bigg(\frac{r_{\mathtt{max}}\sqrt{KB\log B}}{\mu^2_{\mathtt{min}}} + \frac{y^2_{\mathtt{max}}K\log B}{\epsilon^2\mu^2_{\mathtt{min}}} + \frac{\sigma^2 + y_{\mathtt{max}} + \delta\log B}{V\mu^2_{\mathtt{min}}}B + \frac{\delta r_{\mathtt{max}}}{\epsilon\mu^2_{\mathtt{min}}}B\Bigg), \tag{19}$$

$$D^{\pi_{\mathtt{LyOn}}}(B) = O\Bigg(\frac{y^2_{\mathtt{max}}K\log B}{\epsilon^2\mu^2_{\mathtt{min}}B} + \frac{V r_{\mathtt{max}}}{B\mu_{\mathtt{min}}\epsilon} - \frac{\delta}{\mu_{\mathtt{min}}}\Bigg), \tag{20}$$

Specifically, let $V = v_0\sqrt{B\log B}$ and $\delta = \delta_0\sqrt{\log B/B}$ with design parameters $\delta_0 > 0, v_0 > 0$. We can select $\delta_0 \in (\frac{r_{\mathtt{max}}}{\epsilon}v_0, c\sqrt{B})$ such that for sufficiently large B,

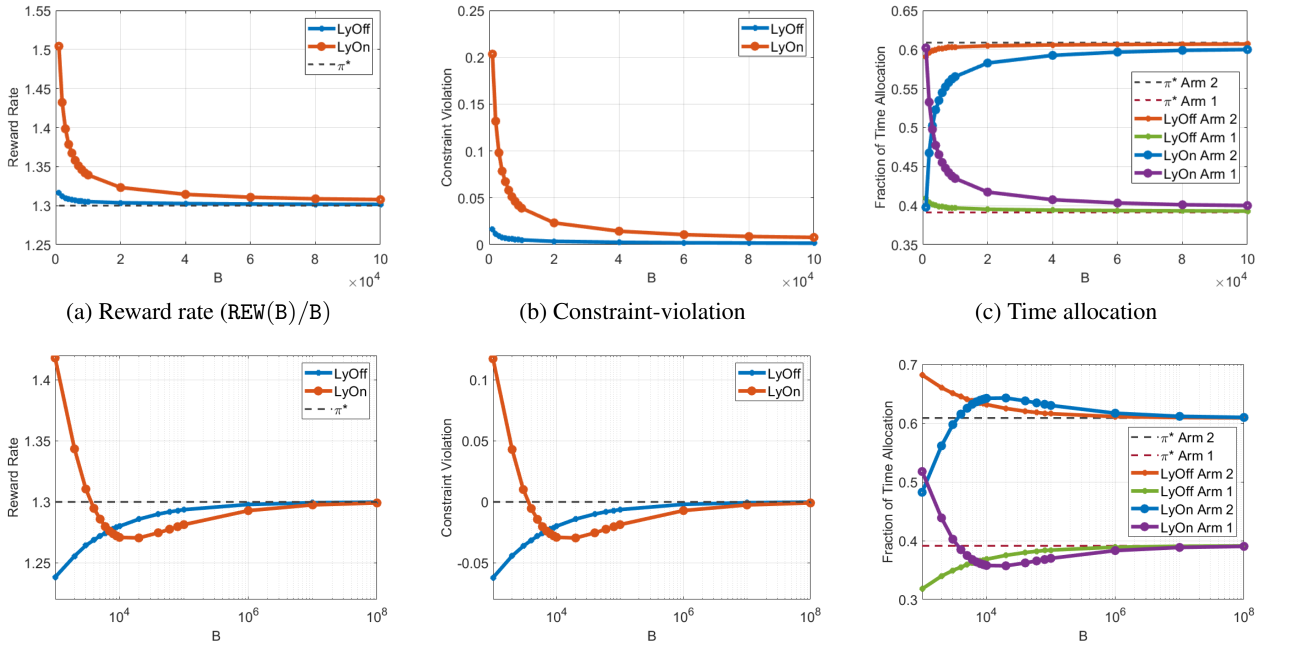
$$\mathtt{REG}^{\pi_{\mathtt{LyOn}}}(B) = O\Big(\sqrt{KB\log B}\Big), \; D^{\pi_{\mathtt{LyOn}}}(B) = O\Big(\frac{-1}{\sqrt{B}}\Big). \tag{21}$$

The proof of Theorem 1 can be found in Appendix E. Theorem 1 implies that $\pi_{\mathtt{LyOn}}$ achieves $O(\sqrt{KB\log B})$ regret and zero constraint-violation for sufficiently large B while learning the first order statistics under a bandit feedback.

In addition to the fact that cumulative reward and penalty processes form stopped and controlled random walks, the main challenge in analyzing the LyOn algorithm performance is that Q_n is correlated with the sample path. To address this, we prove a maximal inequality for Q_n under a concentration event (Lemma 7 in Appendix E), which can have its own value in other queuing systems. Also note that, compared with LyOff, the online algorithm has a very small increase on the regret bounds by a factor of $\sqrt{K\log B}$. This is a reasonable price to pay since we are not assuming any known statistics. To the best of our knowledge, these are the best results available on both regret and constraint-violation in the current setup. In the special case of unit cost scenario, our algorithm theoretically guarantees a similar regret performance to prior designs (Agrawal and Devanur 2014) while providing a stronger constraint-violation guarantee.

6 Simulations

We implement both LyOff and LyOn algorithms for $K = 2$ arms with Bernoulli distributed rewards, costs, and penalties. Assuming $c = 0.8$, arm 1 is selected to have a high reward rate and a high penalty rate with $\mathbb{E}[X_1] = 0.4, \mathbb{E}[Y_1] =$



(a) Reward rate $(\mathtt{REW}(\mathtt{B})/\mathtt{B})$

(b) Constraint-violation

(c) Time allocation

(d) Reward rate $(\mathtt{REW}(\mathtt{B})/\mathtt{B})$

(e) Constraint-violation

(f) Time allocation

Figure 1: Performance of `LyOff` and `LyOn` for different parameters. (a), (b), and (c) use $v_0 = 1$, $\delta_0 = 0.5$, (d), (e), and (f) use $v_0 = 1$, $\delta_0 = 15$.

0.6, and $\mathbb{E}[R_1] = 0.8$. Arm 2 is selected to have a low reward rate and a low penalty rate with $\mathbb{E}[X_2] = 0.6, \mathbb{E}[Y_2] = 0.3$, and $\mathbb{E}[R_2] = 0.6$. These values are interesting in that, an optimal controller will have to make a trade-off between the two arms, whereas any static policy selecting one of the arms will result in either linear regret or linear constraint-violation.

Figure 1a, 1b, and 1c show the simulation results (averaged over 10^4 runs) with $v_0 = 1$ and $\delta_0 = 0.5$ for `LyOff` and `LyOn` algorithms. To observe the reward rate behavior, in Figure 1a, we plot the reward rates $\mathtt{REW}^{\pi}(B)/B$ of $\pi_{\mathtt{LyOff}}$ and $\pi_{\mathtt{LyOn}}$ and the optimal randomized policy π^*, with varying budgets B. This figure shows that both the offline and the online designs reach the rate of the optimal design, as predicted by our analysis. Also, Figure 1b verifies the fast decaying of constraint-violation with rate $\tilde{O}(1/B)$ as B increases, which confirms the scaling behaviour revealed in our analyses. Figure 1c further confirms the convergence of `LyOff` and `LyOn` towards π^* by showing the proportion of time allocated to each arm. As predicted by Theorem 1, Figure 1d, 1e, and 1f show that we can indeed select specific v_0 and δ_0 values such that the constraint-violation becomes negative when B is sufficiently large. At the same time, the reward rate and proportion of time allocated to each arm still converge to the rate of the optimal design.

In Appendix F, to check the performance of our algorithms for larger K, we increase the number of arms by adding arms with the principle that high reward rate arm also has high penalty rate (otherwise the arms are not competitive). We also investigate the effect of design choices V and δ to capture the tradeoff between constraint-violation and regret under the `LyOff` and `LyOn` algorithms.

7 Conclusion

In this paper, we proposed a broadly applicable computationally efficient methodology based on Lyapunov-drift-minimization for solving a penalty-constrained reward maximization problem with a limited budget, random costs, and bandit feedback. Both offline and online algorithms are developed based on this design methodology, which are also proven to have sharp regret and constraint-violation performance. The approach and algorithms are applicable in diverse domains whereby knapsack budget constraints and stochastic feasibility constraints are required. An interesting future work that can benefit from the same methodology would be to extend our setting to the scenario of multiple constraints and infinitely many arms.

Acknowledgments

This work is supported in part by the NSF grants: CNS-NeTS-1717045, CNS-SpecEES-1824337, CNS-NeTS-2007231, CNS-NeTS-2106679, IIS-2112471, CCF-1934986; and the ONR Grant N00014-19-1-2621.

References

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

Asmussen, S. 2008. *Applied probability and queues*, volume 51. Springer Science & Business Media.

Badanidiyuru, A.; Kleinberg, R.; and Slivkins, A. 2013. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 207–216. IEEE.

Balseiro, S. R.; and Gur, Y. 2019. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9): 3952–3968.

Berry, D. A.; and Fristedt, B. 1985. Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). *London: Chapman and Hall*, 5: 71–87.

Bitran, G. R.; and Magnanti, T. L. 1976. Duality and sensitivity analysis for fractional programs. *Operations Research*, 24(4): 675–699.

Bonnans, J. F.; and Shapiro, A. 2000. Stability and Sensitivity Analysis. In *Perturbation Analysis of Optimization Problems*, 260–400. Springer.

Bubeck, S.; Cesa-Bianchi, N.; et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1): 1–122.

Cayci, S.; Eryilmaz, A.; and Srikant, R. 2019. Learning to Control Renewal Processes with Bandit Feedback. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2): 43.

Cayci, S.; Eryilmaz, A.; and Srikant, R. 2020. Budget-constrained bandits over general cost and reward distributions. In *International Conference on Artificial Intelligence and Statistics*, 4388–4398. PMLR.

Cayci, S.; Gupta, S.; and Eryilmaz, A. 2020. Group-Fair Online Allocation in Continuous Time. *Advances in Neural Information Processing Systems*, 33.

Flajolet, A.; and Jaillet, P. 2015. Logarithmic regret bounds for bandits with knapsacks. *arXiv preprint arXiv:1510.01800*.

Georgiadis, L.; Neely, M. J.; and Tassiulas, L. 2006. *Resource allocation and cross-layer control in wireless networks*. Now Publishers Inc.

Ghosh, A.; McAfee, P.; Papineni, K.; and Vassilvitskii, S. 2009. Bidding for representative allocations for display advertising. In *International workshop on internet and network economics*, 208–219. Springer.

Hajek, B. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, 502–525.

Hannák, A.; Wagner, C.; Garcia, D.; Mislove, A.; Strohmaier, M.; and Wilson, C. 2017. Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, 1914–1933.

Harchol-Balter, M. 2000. Task assignment with unknown duration. In *Proceedings 20th IEEE International Conference on Distributed Computing Systems*, 214–224. IEEE.

Ipeirotis, P. G. 2010. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2): 16–21.

Khalek, A. A.; Caramanis, C.; and Heath, R. W. 2014. Delay-constrained video transmission: Quality-driven resource allocation and scheduling. *IEEE Journal of Selected Topics in Signal Processing*, 9(1): 60–75.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1): 4–22.

Liu, X.; Li, B.; Shi, P.; and Ying, L. 2020. POND: Pessimistic-Optimistic oNline Dispatch. *arXiv preprint arXiv:2010.09995*.

Liu, X.; Li, B.; Shi, P.; and Ying, L. 2021. An Efficient Pessimistic-Optimistic Algorithm for Stochastic Linear Bandits with General Constraints. *arXiv preprint arXiv:2102.05295*.

Neely, M. J. 2010. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1): 1–211.

Neely, M. J. 2012. Dynamic optimization and learning for renewal systems. *IEEE Transactions on Automatic Control*, 58(1): 32–46.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1999. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2): 293–305.

Qiu, S.; Wei, X.; Yang, Z.; Ye, J.; and Wang, Z. 2020. Upper confidence primal-dual optimization: Stochastically constrained Markov decision processes with adversarial losses and unknown transitions. *arXiv preprint arXiv:2003.00660*.

Rappaport, T. S.; Heath Jr, R. W.; Daniels, R. C.; and Murdock, J. N. 2015. *Millimeter wave wireless communications*. Pearson Education.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5): 527–535.

Tran-Thanh, L.; Chapman, A.; Rogers, A.; and Jennings, N. R. 2012. Knapsack based optimal policies for budget–limited multi–armed bandits. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Wainwright, M. J. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Xia, Y.; Ding, W.; Zhang, X.-D.; Yu, N.; and Qin, T. 2016. Budgeted bandit problems with continuous random costs. In *Asian conference on machine learning*, 317–332.

Xia, Y.; Li, H.; Qin, T.; Yu, N.; and Liu, T.-Y. 2015. Thompson sampling for budgeted multi-armed bandits. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Yu, H.; and Neely, M. J. 2020. A Low Complexity Algorithm with $O(\sqrt{T})$ Regret and $O(1)$ Constraint Violations for Online Convex Optimization with Long Term Constraints. *Journal of Machine Learning Research*, 21(1): 1–24.