Learning With Guarantee Via Constrained Multi-Armed Bandit: Theory and Network Applications

Kechao Cai, *Member, IEEE*, Xutong Liu, Yu-Zhen Janice Chen, and John C.S. Lui, *Fellow, IEEE*

Abstract—There have been studies that consider optimizing network applications in an online learning context using multi-armed bandit models. However, existing frameworks are problematic as they only consider finding the optimal decisions to minimize the regret, but neglect the *constraints*(or guarantee) requirements that may be excessively violated. In this paper, we formulate the stochastic constrained multi-armed bandit model with *either "time-varying" or "stochastic" multi-level rewards* for network application optimizations with guarantee by taking both *regret* and *violation* into consideration. Alongside this model, we design two constrained multi-armed bandit policies, Learning with Guarantee with *time-Varying* rewards (LG-V) and Learning with Guarantee with *Stochastic* rewards (LG-S), with provable *sub-linear* regret and violation bounds. Moreover, we illustrate *how our policies can be applied to several emerging network application optimizations*, namely, (1) opportunistic multichannel selection, (2) data-guaranteed mobile crowdsensing, and (3) stability-guaranteed crowdsourced transcoding. To show the effectiveness of LG-V and LG-S in optimizing these applications with different requirements, we also conduct extensive simulations by comparing both LG-V and LG-S with existing state-of-the-art policies. We also show the impact of parameter variations, namely, the variations of the guarantee threshold and the number of selected arms, on the regrets and violations of LG-V and LG-S.

Index Terms—Multi-armed bandit, network application optimization, online learning, performance guarantee, stochastic rewards, time-varying rewards

1 Introduction

Due to the ubiquitous deployment of network applications, network application optimization has become increasingly important so as to differentiate one application from others but also to provide the best possible experience for its users. When considering performance optimization for different network applications, a common feature is to make judicious decisions to perform an optimization task. For example, in an opportunistic multichannel access network [1], a secondary user needs to choose appropriate channels to use to increase his/her throughput for faster transmissions. In mobile crowdsensing [2], a task organizer needs to select proper participants to improve the quality of crowdsensed data. In crowdsourced live stream transcoding [3], a platform needs to schedule transcoding assignments to suitable viewers so to speed up live stream transcoding process.

- *Kechao Cai is with the School of Electronics and Communication Engineering, Sun Yat-Sen University, Shenzhen, Guangdong 510275, China. E-mail: caikch3@mail.sysu.edu.cn.*
- *Xutong Liu, Yu-Zhen Janice Chen, and John C.S. Lui are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: {liuxt, yzchen5, cslui}@cse.cuhk.edu.hk.*

Manuscript received 22 May 2021; revised 11 Mar. 2022; accepted 26 Apr. 2022. Date of publication 10 May 2022; date of current version 4 Aug. 2023.
The work of Kechao Cai was supported in part by the National Key R&D Program of China under Grant 2021YFB2900200, in part by NSF China under Grant 62171159, in part by Guangdong Science and Technology Planning Project under Grant 2018B030322004, and in part by the Fundamental Research Funds for the Central Universities, Sun Yat-sen University. The work of John C.S. Lui was supported by SRFS2122-4S02 Grant.
(Corresponding author: Kechao Cai.)
Digital Object Identifier no. 10.1109/TMC.2022.3173792

Setting the right parametric values is an essential prerequisite in making optimization decisions for the network applications. Many previous works in the literature assume that the parameters are known as a priori knowledge. However, the parameters of an application are not fixed, and can even *vary with the decisions or even change over time*. For instance, in the opportunistic multichannel access network, it is assumed in [4] that the statistical information of the channels, such as the probabilities that channels are free and the throughput of the channels are fully available to the channel users. In practice, however, due to the uncertainty in channel utilization or environmental noise, these statistics are not fixed as a priori and must be estimated by the users.

Therefore, to optimize a network application when its parameters are *unknown*, it is natural to consider the optimization in an *online learning context*, where the decision maker is not required to possess prior knowledge but will continuously estimate the parameters as observations are made. Within this context, the multi-armed bandit (MAB) [5], [6] setup has become an attractive modeling framework for many network applications as it allows a user to estimate the parameters and perform optimization throughout an online learning process. For this reason, there have been studies on this learning framework for optimizing different network applications, e.g., [7], [8] proposed MAB based policies to estimate the throughput of different channels in an opportunistic network and select the best channel to maximize the throughput and [9] designed budget limited crowdsensing policies based on MAB paradigms to maximize the revenue of a crowdsensing task.

However, all these learning policies are limited as they neglect the *constraints or guarantee requirements* in network application optimizations and only consider finding the optimal decision that maximizes the cumulative reward, e.g., (the channel throughput, the sensing revenue, etc.). The performance of these policies is merely measured by *regret*, the difference of the cumulative reward between a learning policy and the Oracle policy (or an optimal yet unrealizable policy) which always makes the optimal decision.

In fact, many real-world network application optimization tasks have some guarantee requirements (or constraints) in addition to the goal of seeking the optimal decision. Therefore, a decision maker may *violate these guarantee requirements* if he/she solely learns the parameters to optimize the applications. As an example, a secondary user in an opportunistic multichannel access network would select channels to not only maximize throughput but also try to ensure *at least one free channel to use at each time slot* to keep data transmission stable. In such a channel selection procedure, it is likely that the user cannot use any free channel in the selected channels at some time slots and thereby violate the guarantee requirement. Hence, there is a need to have an additional performance measure for a learning policy, which we call the *violation*, and it measures the cumulative violations of the guarantee requirement that a learning policy has made in the online learning process.

In this paper, we formulate the stochastic constrained MAB model with *either time-varying or stochastic multi-level* rewards to tackle the optimization task with a guarantee requirement in various network applications. Specifically, each arm is associated with *multi-level rewards:* a level-1 reward and a level-2 reward (can be further extended to level-n reward, $n \geq 2$). The rewards at different levels can have different characteristics. In particular, the level-1 reward of an arm is stochastic (stationary) and the level-2 reward of an arm can be either *time-varying or stochastic*, and together they produce a compound reward. At each time slot, a decision maker selects m arms (actions) from M arms ($1 \leq m \leq M$), observes the level-1 and level-2 rewards from each of the m selected arms, and receives the compound rewards from the m selected arms. Moreover, to satisfy the guarantee requirement, there is a *guarantee threshold* ρ such that the total level-1 reward of the selected m arms should be at least ρ. Our objective is to find learning policies to *maximize the total compound reward as well as to keep the average total level-1 reward above the guarantee threshold* ρ. The key issue is to balance between maximizing the total compound reward (i.e., minimizing the regret) and satisfying the guarantee threshold constraint (i.e., keeping low violation). To achieve this, for the model with *time-varying* reward, we propose a constrained MAB policy, *Learning with Guarantee with time-Varying rewards* (LG-V), and for the model with *stochastic* reward, we propose another constrained MAB policy, *Learning with Guarantee with* Stochastic*rewards* (LG-S). Both policies have the attractive property in achieving *sub-linear* regret and violation bounds. To show the flexibility and applicability of our policies, we illustrate how LG-V and LG-S can be applied to several network applications, namely,(1) opportunistic multichannel selection, (2) data-guaranteed mobile crowdsensing, and (3) stability-guaranteed crowdsourced transcoding.

Contributions. (i) We formulate the stochastic constrained MAB model with either time-varying or stochastic multi-level rewards for network application optimizations with guarantee requirements. (ii) We design two new constrained MAB policies, LG-V for *time-varying rewards* and LG-S for *stochastic rewards*, by taking the guarantee requirement ρ into consideration with provable *sub-linear* regret and violation bounds. (iii) We show how LG-V and LG-S can be applied to three different network applications. (iv) We demonstrate the effectiveness of our LG-V and LG-S policies by comparing them with existing state-of-the-art policies and show that our policies achieve better application optimizations in terms of maximizing the cumulative compound reward while meeting the guarantee requirements. (v) We also show the impact of the variations of the parameters of LG-V and LG-S on the regrets and violations.

The rest of the paper is organized as follows. The stochastic constrained MAB model with either *time-varying* or *stochastic* multi-level rewards is presented in Section 2. The detailed LG-V and LG-S policy designs and the proof of sub-linear regret and violation bounds are elaborated in Section 3. The applications of LG-V and LG-S and the comparison results are presented in Section 4. The impact of parameter variations of LG-V and LG-S on regret and violation are discussed in Section 5. Related work is given in Section 6. Finally, conclusion is given in Section 7.

2 Constrained Multi-Armed Bandit Model

In this section, we present the details of our stochastic constrained multi-armed bandit model with either *time-varying* or *stochastic* multi-level rewards.

Let $\mathcal{M} = \{1, \ldots, M\}$ denote the set of M arms. Each arm $i \in \mathcal{M}$ is associated with two *unknown* random processes, $U_i(t)$ and $V_i(t)$, $t = 1, \ldots, T$. Specifically, $U_i(t)$ characterizes the arm i's *level-1 reward* and $V_i(t)$ characterizes arm i's *level-2 reward*.[1] We assume that $U_i(t), i \in \mathcal{M}$ are stationary and independent across i, and the probability distribution of $U_i(t)$ for $i \in \mathcal{M}$ has a finite support. As for $V_i(t), i \in \mathcal{M}$, they are bounded across i and are modeled differently in different scenarios as follows,

$$V_i(t): \begin{cases} \text{time-varying}, & \text{if arm } i\text{'s level-2 reward is } time-varying, \\ \text{stochastic}, & \text{if arm } i\text{'s level-2 reward is } stochastic. \end{cases}$$

Without loss of generality, we normalize $U_i(t) \in [0, 1]$ and $V_i(t) \in [0, 1]$. We also assume that $U_i(t)$ is independent of $V_i(t)$ for $i \in \mathcal{M}$ and $t = 1, \ldots, T$.

The random process $U_i(t)$, is assumed to have *unknown* mean $u_i = \mathbb{E}[U_i(t)]$. Similarly, $V_i(t)$, if it is modeled as a *stationary stochastic process*, is assumed to have *unknown* mean $v_i = \mathbb{E}[V_i(t)]$. Otherwise, $V_i(t)$ can vary over time without following any probability distribution. u_i^t and v_i^t are the realizations of $U_i(t)$ and $V_i(t)$ at time t, respectively, $1 \leq i \leq M$. Let $\boldsymbol{u} = (u_1, \ldots, u_M)$ and $\boldsymbol{v} = (v_1, \ldots, v_M)$.[2] Let $\boldsymbol{u}_t = (u_1^t, \ldots, u_M^t)$ and $\boldsymbol{v}_t = (v_1^t, \ldots, v_M^t)$ denote the realization vectors at time t for the random processes $U_i(t)$ and $V_i(t)$, respectively for $1 \leq i \leq M$. Let $\boldsymbol{p}_t = (p_1^t, \ldots, p_M^t)$ be the *probabilistic selection vector*

1. One can easily extend two level rewards to n level ($n \geq 2$) rewards by associating each arm with n random processes.
2. All vectors defined in this paper are column vectors.

of the M arms at time t, where $p_i^t \in [0,1]$ is the probability of selecting arm i at time t.

At time t, a set of m ($1 \leq m \leq M$) arms $\mathcal{I}_t \in \mathcal{M}$ ($|\mathcal{I}_t| = m$) is selected via a dependent rounding procedure [10], which guarantees the probability that arm i is selected with probability p_i^t at time t (see Section 3). Thus, the number of selected arms is exactly m at each time t, i.e., $\mathbf{1}^\intercal \boldsymbol{p}_t = m$, where $\mathbf{1} = (1, \ldots, 1)$. For each arm $i \in \mathcal{I}_t$, the arm selection policy observes a *level-1 reward* u_i^t generated by $U_i(t)$, as well as a *level-2 reward* v_i^t generated by $V_i(t)$, and receives a *compound reward*. Specifically, the compound reward, g_i^t, of arm i at time t is generated by the random process $G_i(t) = U_i(t)V_i(t)$. Let $g_i^t = u_i^t v_i^t, 1 \leq i \leq M$, and $\boldsymbol{g}_t = (g_1^t, \ldots, g_M^t)$. Let N_i^t be the number of times that arm i is played before time t. The empirical average level-1 reward and the empirical average compound reward for arm i are defined as $\bar{u}_i^t = \frac{\sum_{\tau<t, i\in\mathcal{I}_\tau} u_i^\tau}{N_i^t+1}$ and $\bar{g}_i^t = \frac{\sum_{\tau<t, i\in\mathcal{I}_\tau} g_i^\tau}{N_i^t+1}$, respectively. In addition, there is a preset *guarantee threshold* $\rho > 0$ such that the sum of the level-1 rewards needs to be above this threshold.

2.1 Regret

Our objective is to design learning policies to choose the selection vectors $\boldsymbol{p}_t$ for $t = 1, \ldots, T$ for the stochastic constrained MAB model with either time-varying rewards or stochastic rewards such that the *regret*, which is also referred to as loss compared with the Oracle, is as small as possible.

For the constrained MAB model with *time-varying rewards*, in order to select the optimal arms, the Oracle should know the model parameters: $\boldsymbol{u}$ and the time-varying $\boldsymbol{v}_t$ at each time t. Thus it can select arms to maximize the cumulative compound rewards $\sum_t \boldsymbol{g}_t^\intercal \boldsymbol{p}_t$ while satisfying the constraint $\boldsymbol{u}^\intercal \boldsymbol{p}_t \geq \rho$ at each time t. The regret of a learning policy π_v that chooses the selection vector $\boldsymbol{p}_t^{\pi_v}$ at each time t is defined as

$$\mathrm{R}_{\pi_v}(T) = \max_{\boldsymbol{u}^\intercal \boldsymbol{p}_t \geq \rho} \sum_{t=1}^{T} \boldsymbol{g}_t^\intercal \boldsymbol{p}_t - \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{g}_t^\intercal \boldsymbol{p}_t^{\pi_v}\right], \tag{1}$$

where the expectation is taken over the randomness of the policy π_v in selecting the arms.

For the constrained MAB model with *stochastic rewards*, the Oracle can select the optimal arms with the optimal selection vector $\boldsymbol{p}^* = \arg\max_{\boldsymbol{u}^\intercal \boldsymbol{p} \geq \rho} \boldsymbol{p}^\intercal \boldsymbol{g}$ with the knowledge of the model parameters: $\boldsymbol{u}$ and $\boldsymbol{v}$. In this case, the regret of a learning policy π_s that chooses the selection vector $\boldsymbol{p}_t^{\pi_s}$ at each time t is defined as

$$\mathrm{R}_{\pi_s}(T) = T\boldsymbol{g}^\intercal \boldsymbol{p}^* - \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{g}_t^\intercal \boldsymbol{p}_t^{\pi_s}\right], \tag{2}$$

where the Oracle always selects the optimal arms with $\boldsymbol{p}^*$ throughout T time slots and the expectation is taken over the randomness of the policy π_s in selecting the arms.

2.2 Violation

A policy may violate the constraint when learning the model parameters to minimize its regret. Especially when a policy has little information about the arms at the early stage, it may choose selection vectors $\boldsymbol{p}_t$ that violate the constraint $\boldsymbol{u}^\intercal \boldsymbol{p}_t \geq \rho$.

To measure the overall violations of the constraint of the policy π_v for the constrained MAB model with *time-varying rewards* at time T, the *violation* of π_v is defined as

$$\mathrm{V}_{\pi_v}(T) = \mathbb{E}\left[\sum_{t=1}^{T} (\rho - \boldsymbol{u}^\intercal \boldsymbol{p}_t^{\pi_v})\right]^+, \tag{3}$$

where $[\cdot]^+ = \max(\cdot, 0)$ and only the non-negative part in the cumulative violation is taken into account.

Similarly, the *violation* of the policy π_s for the constrained MAB model with *stochastic rewards* is defined as

$$\mathrm{V}_{\pi_s}(T) = \mathbb{E}\left[\sum_{t=1}^{T} (\rho - \boldsymbol{u}^\intercal \boldsymbol{p}_t^{\pi_s})\right]^+. \tag{4}$$

As our MAB model is stochastic constrained by the stochastic level-1 rewards, the violation in (4) is defined in the same way with the violation in (3). Note that the violations are defined by the clipped cumulative constraint in (3) and (4) because we concern about the *long-term constraint satisfaction* where a policy is allowed to violate the constraint for some rounds but satisfies the constraint in the long term for all T rounds.

Regret and violation are two important metrics to measure the performance of an arm selection policy. In particular, a policy with a lower regret means that the policy can get closer to the Oracle, and a policy with a smaller violation means that the policy can satisfy the constraint better as time t increases.

3 Policy Design

In this section, we explain in detail the design of our policies, *Learning with Guarantee with time-Varying rewards* (LG-V) and *Learning with Guarantee with Stochastic rewards* (LG-S), for the constrained bandit model described in Section 2. The implementations of the two policies are open source and available at [11]. The key idea of our policies is to balance between maximizing the total compound reward (or minimizing the regret in Eqs. (1) and (2)) and, at the same time, satisfying the guarantee threshold ρ (or maintaining low violation in Eqs. (3) and (4)).

3.1 LG With Time-Varying Rewards (LG-V)

The time-varying property of the rewards makes the policy design challenging. To make a well tradeoff between minimizing the regret and maintaining small violation, we consider the optimization problem

$$\min_{\boldsymbol{p}_1^{\pi_v}, \boldsymbol{p}_2^{\pi_v}, \ldots, \boldsymbol{p}_T^{\pi_v}} R_{\pi_v}(T) \text{ subject to } (V_{\pi_v}(T))^2 \leq 0,$$

where $\boldsymbol{p}_1^{\pi_v}, \boldsymbol{p}_2^{\pi_v}, \ldots, \boldsymbol{p}_T^{\pi_v}$ are the selection vectors at time $1, 2, \ldots, T$, respectively. To solve the problem above, we incorporate the theory of Lagrange method in constrained optimization into our LG-V design. We consider minimizing a Lagrange function that includes the regret and the violation with an *adjustable* penalty coefficient that increases when there is any non-zero violation. More specifically, our policy introduces a sub-linear bound for the Lagrange function of $\mathrm{R}_{\pi_v}(T)$ and $\mathrm{V}_{\pi_v}(T)$ in the following structure

$$\mathrm{R}_{\pi_v}(T) + \lambda(T)(\mathrm{V}_{\pi_v}(T))^2 \leq T^{1-\theta}, 0 < \theta \leq 1, \tag{5}$$

where $\lambda(T)$ plays the role of a Lagrange multiplier. From (5), we derive a bound for $\mathrm{R}_{\pi}(T)$ in (6) and a bound for $\mathrm{V}_{\pi_v}(T)$ in (7) as

$$\mathrm{R}_{\pi_v}(T) \leq O(T^{1-\theta}), \tag{6}$$

$$\mathrm{V}_{\pi_v}(T) \leq \sqrt{O(T^{1-\theta} + mT)/\lambda(T)}, \tag{7}$$

where the bound for $\mathrm{V}_{\pi_v}(T)$ in (7) is for the fact that $-\mathrm{R}_{\pi_v}(T) \leq O(mT)$ for any policy π. Here $O(mT)$ is the loss of reward throughout T time slots in the worst case. With properly adjusted $\lambda(T)$ and θ, both the regret and violation can be bounded by sub-linear functions of T.

Algorithm 1. LG With Time-Varying Rewards (LG-V)

Input: $M, m, \rho, T; \gamma, \eta$.
Output: a set of selected arms at each round.
Init: $\boldsymbol{w}_1 = \mathbf{1}, \lambda_1 = 0, \rho > 0, \beta = (1/m - \gamma/M)/(1-\gamma)$, $\zeta = \gamma\eta m/(\eta + m)M$.
1: **for** $t = 1, \ldots, T$ **do**
2: Set $\mathcal{S}_t = \emptyset$, $\mathcal{I}_t = \emptyset$.
3: **if** $\max_{i\in\mathcal{M}} w_i^t \geq \beta \sum_{i=1}^{M} w_i^t$ **then**
4: Find α_t such that

$$\alpha_t / \left(\sum_{i=1, w_i^t \geq \alpha_t}^{M} \alpha_t + \sum_{i=1, w_i^t < \alpha_t}^{M} w_i^t \right) = \beta.$$

5: Set $\mathcal{S}_t = \{i : w_i^t \geq \alpha_t\}$.
6: **for** $i = 1, \ldots, M$ **do**

$$\tilde{w}_i^t = \alpha_t \text{ if } i \in \mathcal{S}_t; \text{ otherwise, } \tilde{w}_i^t = w_i^t.$$

7: **for** $i = 1, \ldots, M$ **do** ▷Compute $\boldsymbol{p}_t^{\pi_v}$.

$$\tilde{p}_i^t = m[(1-\gamma)\tilde{w}_i^t / \textstyle\sum_{i=1}^{M} \tilde{w}_i^t + \gamma/M].$$

8: Set $\mathcal{I}_t = \mathrm{DependentRounding}(m, \tilde{\boldsymbol{p}}_t)$.
9: **for** $i \in \mathcal{I}_t$ **do** receive u_i^t and v_i^t.
10: **for** $i = 1, \ldots, M$ **do**

$$\hat{u}_i^t = u_i^t / \tilde{p}_i^t \mathbb{1}(i \in \mathcal{I}_t), \ \hat{g}_i^t = u_i^t v_i^t / \tilde{p}_i^t \mathbb{1}(i \in \mathcal{I}_t).$$

11: **for** $i = 1, \ldots, M$ **do**

$$w_i^{t+1} = \begin{cases} w_i^t & \text{if } i \in \mathcal{S}_t; \\ w_i^t \exp[\zeta(\hat{g}_i^t + \lambda_t \hat{u}_i^t)] & \text{if } i \notin \mathcal{S}_t. \end{cases}$$

12: $\lambda_{t+1} = [(1 - \eta\zeta)\lambda_t - \zeta(\frac{\hat{\boldsymbol{u}}_t^{\intercal} \tilde{\boldsymbol{p}}_t}{1-\gamma} - \rho)]^+$.
13: **function** DependentRounding$m, \boldsymbol{p}$
14: **while** exist $i \wedge p_i \in (0,1)$ **do**
15: Find $i, j, i \neq j$, such that $p_{i,j} \in (0,1)$
16: $a = \min\{1 - p_i, p_j\}, b = \min\{p_i, 1 - p_j\}$.
17: $(p_i, p_j) = \begin{cases} (p_i + a, p_j - a) & \text{with prob. } \frac{b}{a+b}; \\ (p_i - b, p_j + b) & \text{with prob. } \frac{a}{a+b}. \end{cases}$
return $\mathcal{I} = \{i \,|\, p_i = 1, 1 \leq i \leq M\}$

Another technical challenge is that the search space for the optimal set of arms for our bandit model is very large since the number of possible choices can be as large as $\binom{M}{m}$ at each time. Hence, the upper bounds of the regret and the violation with respect to M and m can be large. Thanks to the independence among the arms, we can judiciously keep M weights for the M arms instead of $\binom{M}{m}$ weights for each set of m arms to avoid combinatorial explosion. Essentially, LG-V updates the weights of the M arms and recomputes the Lagrange multiplier λ_t in each iteration. Then the policy calculates the probabilistic selection vector $\tilde{\boldsymbol{p}}_t$ for the M arms according to their weights. Specifically, an arm that is more likely to maximize the total compound reward under the threshold constraint is assigned with a higher probability of being selected. Next, the policy selects m arms from the M arms according to the probabilistic selection vector $\tilde{\boldsymbol{p}}_t$ using the dependent rounding algorithm, DependentRounding, in [10].

The LG-V policy is shown in Algorithm 1. In particular, LG-V maintains a weight vector $\boldsymbol{w}_t = \{w_1^t, \ldots, w_M^t\}$ for the M arms at time t, which is used to calculate the probabilistic selection vector $\tilde{\boldsymbol{p}}_t$ (line 3 to line 7). Line 3 to line 6 ensure that the probabilities in $\tilde{\boldsymbol{p}}_t$ are less than or equal to 1. Line 7 computes the arm selection vector $\boldsymbol{p}_t^{\pi_v}$ of LG-V at time t.

At line 12, we deploy the dependent rounding function, DependentRounding, (see details from line 20 to line 26) to select m arms using the calculated $\tilde{\boldsymbol{p}}_t$. Specifically, DependentRounding probabilistically updates $\tilde{\boldsymbol{p}}_t$ until $\tilde{p}_i^t$ is either 0 or 1 and maintains the condition that $\mathbf{1}^{\intercal}\tilde{\boldsymbol{p}}_t = m$. This rounding procedure guarantees that exactly M arms would be selected and arm i is selected with probability p_i^t. Such properties of DependentRounding are guaranteed by Lemma 1.

Lemma 1 (Theorems 2.3 and 2.4 in [10]). *The* DependentRounding *procedure runs in linear time and guarantees that probability of arm i being selected at time t is $\tilde{p}_i^t$ and $|\mathcal{I}_t| = M$.*

Lemma 1 indicates that the set of selected arms $\mathcal{I}_t$ is *optimal in expectation* as long as $\tilde{\boldsymbol{p}}_t$ approaches the optimal selection vector at time t.

At line 13, the LG-V policy receives the rewards u_i^t and v_i^t, and then learns the parameters u_i, v_i and g_i of arm i's multi-level rewards by giving *unbiased* estimates of $\hat{u}_i^t$ and $\hat{g}_i^t$ for each arm $i \in \mathcal{M}$ at line 15. Here $\mathbb{1}(E)$ is the indicator function, i.e., $\mathbb{1}(E) = 1$ if the event E occurs and $\mathbb{1}(E) = 0$ otherwise. Specifically, the level-1 reward $\hat{u}_i^t$ and the compound reward $\hat{g}_i^t$ are estimated by $u_i^t/\tilde{p}_i^t$ and $u_i^t v_i^t/\tilde{p}_i^t$, respectively, such that $\mathbb{E}[\hat{u}_i^t] = u_i^t$, and $\mathbb{E}[\hat{g}_i^t] = u_i^t v_i^t$. Finally, the weight vector $\boldsymbol{w}_t$ and the Lagrange multiplier λ_t are updated (line 17 and line 19) using the estimations at the end of each iteration.

Inside the main for loop of Algorithm 1, line 5 requires at most M sequential searches; all the inner for loops can be completed in M operations; and the dependent rounding procedure runs in $O(M)$ by Lemma 1. Therefore, the overall *time complexity* of Algorithm 1 is $O(MT)$.

3.2 LG With Stochastic Rewards (LG-S)

When the rewards are stochastic, it is possible to approach the optimal policy by using the reward history of the selected arms. Thus, we try to find the optimal selection vector in hindsight step by step in our LG-S design.

The LG-S policy is shown in Algorithm 2. Specifically, at line 4, LG-S first estimates the level-1 and compound reward using the upper confidence bounds (defined in [12]) with the helper function $R(\mu, n)$. $R(\mu, n)$ detailed at line 12 is a radius function of the empirical average reward μ as defined in [13].

The key step of LG-S is line 7, where LG-S solves a linear optimization problem (8) to get the probabilistic selection vector $\boldsymbol{p}_t$ using the upper confidence bounds. The optimal solution to (8) is the optimal selection vector $\boldsymbol{p}_t^{\pi_s}$ in hindsight based on the current level-1 and level-2 reward estimations. Then, at line 8, given $\boldsymbol{p}_t$, $\mathcal{I}_t$ is generated via the same DependentRounding rounding procedure with that in Algorithm 1. At line 10, the LG-S policy receives the two-level rewards u_i^t and v_i^t for arms in $\mathcal{I}_t$ and updates the empirical average rewards so as to get the upper confidence bounds in the next round.

In Algorithm 2, solving the linear optimization problem in (8) with M variables requires $(\text{poly}(M))$ operations [14]. All the inner for loops and the dependent rounding procedure can be completed in M operations. Thus, the *time complexity* of Algorithm 2 is $O(\text{poly}(M)T)$.

Algorithm 2. LG With Stochastic Rewards (LG-S)

Input: M, m, ρ, T.
Output: a set of selected arms at each round.
Init: $\kappa = 72\ln(MT^2)$, $\bar{\boldsymbol{g}}_1 = \boldsymbol{0}$, $\bar{\boldsymbol{u}}_1 = \boldsymbol{0}$, and $N_i^1 = 0, \forall i$.
1: **for** $t = 1, 2, \ldots, T$ **do**
2: **for** $i = 1, 2, \ldots, M$ **do**
3: $\hat{u}_i^t = \min\{1, \bar{u}_i^t + 2R(\bar{u}_i^t, N_i^t + 1)\}$,
$\hat{g}_i^t = \min\{1, \bar{g}_i^t + 2R(\bar{g}_i^t, N_i^t + 1)\}$.
4: $\hat{\boldsymbol{u}}_t = (\hat{u}_1^t, \ldots, \hat{u}_M^t)$, $\hat{\boldsymbol{g}}_t = (\hat{g}_1^t, \ldots, \hat{g}_M^t)$.
5: Solve the following linear optimization problem:

$$\boldsymbol{p}_t = \arg\max_{\boldsymbol{p}^\intercal \hat{\boldsymbol{u}}^t \geq \rho, \boldsymbol{p} \in [0,1]^M} \boldsymbol{1}^\intercal \boldsymbol{p} = m \boldsymbol{p}^\intercal \hat{\boldsymbol{g}}^t \quad \triangleright \text{Compute} \boldsymbol{p}_t^{\pi_s}. \tag{8}$$

Set $\boldsymbol{p}_t \in \mathcal{X}$ arbitrarily if (8) no feasible solution.

6: Set $\mathcal{I}_t = \text{DependentRounding}(m, \boldsymbol{p}_t)$.
7: **for** $i \in \mathcal{I}_t$ **do**
8: Receive u_i^t and v_i^t, and update:
$\bar{g}_i^{t+1} = [\bar{g}_i^t(N_i^t + 1) + g_i^t]/(N_i^{t+1} + 1)$,
$\bar{u}_i^{t+1} = [\bar{u}_i^t(N_i^t + 1) + v_i^t]/(N_i^{t+1} + 1)$,
$N_i^{t+1} = N_i^t + 1$.
9: **function** $R\mu, n$
return $\sqrt{\frac{\kappa\mu}{n}} + \frac{\kappa}{n}$

Remark. both the LG-V policy shown in Algorithm 1 and the LG-S policy shown in Algorithm 2 can run in a distributed fashion in a similar way to that in [15]. Each learning agent running LG-V or LG-S can gather the multi-level rewards by selecting different arms and share the estimations with other learning agents. In this way, the learning agents can accomplish optimization tasks collaboratively in network applications.

3.3 Regret and Violation Analysis

For our policy LG-V with time varying rewards shown in Algorithm 1, we have the following attractive property,

Theorem 1. *Let* $\gamma = \min(\frac{1}{2}, \sqrt{\frac{2(e-2)M+Mm}{m\ln(M/m)T^{2/3}}})$ *and* $\eta = \frac{4(e-2)\gamma m}{1-\gamma}$. *By running the LG-V policy* π_v *shown in Algorithm 1, we achieve sub-linear bounds for both the regret and violation as follows:*

$$\mathrm{R}_{\pi_v}(T) \leq O(mM\ln(M)T^{\frac{2}{3}}),$$
$$\mathrm{V}_{\pi_v}(T) \leq O(m^{\frac{1}{2}}M^{\frac{1}{2}}T^{\frac{5}{6}}).$$

For our policy LG-S shown in Algorithm 2 with stochastic rewards, we can achieve much tighter upper bounds as follows:

Theorem 2. *For all* $T > 0$, *let* $\kappa = 72\ln(MT^2)$. *By running the LG-S policy* π_s *shown in Algorithm 2, we have sub-linear bounds for both the regret and violation as follows:*

$$\mathrm{R}_{\pi_s}(T) = O(M\sqrt{MT\ln T}),$$
$$\mathrm{V}_{\pi_s}(T) = O(M\sqrt{MT\ln T}).$$

We refer the interested readers to the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2022.3173792, for the details of the proofs of the above theorems.

Remark. A *good* arm selection policy should have both *sub-linear* regret and *sub-linear* violation with respect to T. If these two metrics are linear, it implies that the policy π is *not learning* from the history rewards of the selected arms. A simple example of such policies is the uniform arm selection policy, where any m arms are selected with equal probability. In this case, there would be a constant gap between such a random policy and the optimal policy. Thus, this uniform random policy would result in both linear regret and linear violation.

4 Applications

To illustrate the flexibility and applicability of LG-V and LG-S, we describe several network applications with different guarantee requirements where LG-V and LG-S can be applied. For each application, we first describe the problem and map the problem to our stochastic constrained MAB model. Then we carry out simulations and compare LG-V and LG-S with existing state-of-the-art online learning policies to show the effectiveness of LG-V and LG-S in different application settings.

4.1 Opportunistic Multichannel Selection

4.1.1 Problem Description

We study the opportunistic multichannel selection problem in opportunistic channel access networks [1] where primary users are licensed and have priority to use the channels, while secondary users are unlicensed and should sense and use the available channels to avoid taking up the channels of primary users [16]. Consider a network consisting of M independent channels which are licensed to primary users who communicate according to a synchronous time slot structure. At each time slot, channels can be primary-free (unoccupied by primary users) but with *unknown* probabilities. These M channels also have *unknown* throughput/bandwidths at different time slots. Consider now a secondary user seeking opportunities of transmitting in the free slots of these M channels. With a limited sensing capability, a secondary user can only access a subset of m $(1 \leq m \leq M)$ channels and *observe the occupancies and throughput of the accessed channels*. To use as many primary-free channels as possible and also maximize one's throughput, a sensible secondary user should take both the primary-free probabilities and the throughput of different channels into account in selecting the channels. On the one

TABLE 1
Primary-Free Prob. And Throughput of 10 Channels

Channel No.	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Primary-free Prob.	0.15	0.2	0.2	0.1	0.6	0.55	0.25	0.4	0.35	0.7
Channel Throughput	0.65	0.55	0.6	0.7	0.2	0.25	0.5	0.3	0.4	0.1

hand, selecting channels that have higher throughput facilitates faster data transmission. On the other hand, selecting channels that have higher primary-free probabilities helps to establish a multichannel connection with a higher success rate. Therefore, to strike a balance between high throughput and high primary-free probability, a worthy problem for the secondary user to consider is to *select channels that maximize the throughput and keep the average number of accessed primary-free channels above the guarantee threshold* ρ_1.[3]

For example, in an opportunistic channel access network with $M = 10$ channels, let us suppose that the primary-free probabilities and the throughput of different channels are known as shown respectively in row 2 and row 3 of Table 1. (These are typical parameter settings for an opportunistic channel access network.) Note that the primary-free probabilities do not sum up to 1 as the channels are independent, and the throughput of each channel is normalized to [0,1]. At a time slot, a secondary user can only access $m = 3$ channels (e.g., channel #5, #6, and #9), identify the primary-free channels therein, transmit over and estimate the throughput of the accessed primary-free channels. In addition, selecting channel #5, #6, and #9 in Table 1 satisfies the guarantee exactly supposing that $\rho_1 = 1.5$.

This problem fits the general framework of our stochastic constrained MAB model regarding the M channels as M arms, the channel access probabilities as u_i, and the throughput of different channels as v_i, $1 \leq i \leq M$, for $t = 1, \ldots, T$. Here, $\boldsymbol{g}_t$ is the *compound throughput* of the M channels and $\boldsymbol{p}_t$ is the probabilistic selection vector of the M channels.

The constraint for the opportunistic multichannel selection problem is $\boldsymbol{u}^{\intercal}\boldsymbol{p}_t \geq \rho_1$, where $\boldsymbol{u}^{\intercal}\boldsymbol{p}_t$ represents the average number of primary-free channels at time t and ρ_1 is the guarantee threshold.

4.1.2 Performance Evaluation

Now consider the opportunistic multichannel selection problem in the example above in the online learning context where a policy has to estimate all the parameters. The primary-free probabilities u_i are taking from row 2 of Table 1 for $1 \leq i \leq 10$. In particular, channel i is primary-free if a Bernoulli trial with success probability u_i is successful (i.e., $u_i^t = \text{Bernoulli}(u_i) = 1$). Otherwise, channel i is taken up by primary users if the trial is failed (i.e., $u_i^t = \text{Bernoulli}(u_i) = 0$) for $1 \leq i \leq 10$.

In the case that the channel throughput is *time-varying*, the throughput v_i^t of channel i is time-varying in an adversary fashion defined as follows:

$$v_i^{t+1} = \begin{cases} \max\{v_i^t + 0.01, v_i\}, & \text{if } u_i^t = 0, \\ \max\{v_i^t - 0.01, 0\}, & \text{if } u_i^t = 1, \end{cases}$$

for $t > 1$ and $v_i^1 = \text{Uniform}(0, v_i)$, where v_i is taken in row 3 of Table 1, and $\text{Uniform}(0, v_i)$ generates a uniformly random sample in $[0, v_i]$. This time-varying pattern can model the competitions among the secondary users in selecting the channels. Specifically, if a channel is primary-free at a time slot, a secondary user can use the channel at this time slot, and the throughput of this channel would decrease for other secondary users in the next time slot. If a channel is taken up by primary users, then the channel's potential throughput would be large if the channel becomes primary-free in the next time slot.

In the case that the channel throughput is *stochastic*, the throughput v_i of channel i is modeled as a [0,1]-truncated normal distributed variable with standard deviation 0.5 and mean v_i taken in row 3 of Table 1 for $1 \leq i \leq 10$. This stochastic modeling of the channel throughput can be adopted in a stationary multichannel network where the primary users have fixed access patterns.

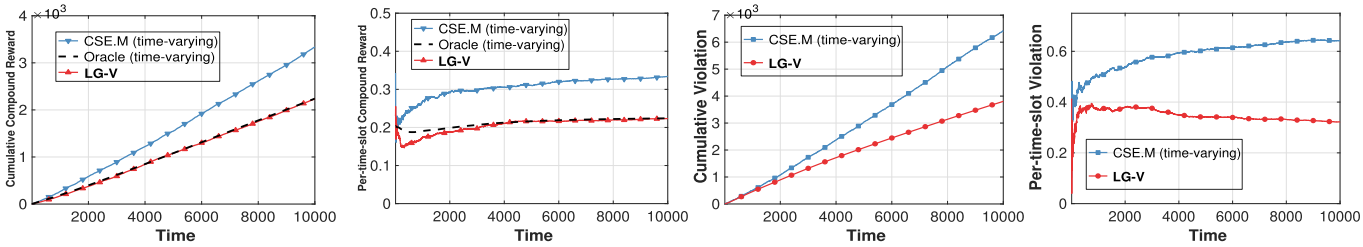
We compare our LG-V policy and LG-S policy with a modified version of the state-of-the-art online channel selection policy in [7]. It is worth noting that there are many salient works besides [7], such as [8], [17], [18] that also considered online dynamic selection for secondary users. However, these works mainly focus on maximizing the throughput but neglect the importance of the guarantee threshold on the primary-free probabilities. Furthermore, they are limited to single-channel selection scenarios. Therefore, they cannot be generalized to much broader multichannel selection scenarios. For comparison purposes, we adapt the *Continuous Sampling and Exploration* (CSE) policy proposed in [7], which is a single-channel selection policy using the highest UCB (upper confidence bound [6]) index, to select multiple channels with the m highest UCB indices of channel throughput estimations. Thus, we name the modified policy as CSE.M, where the letter M stands for multiple.

In our experiments, LG-V selects channels in the *time-varying* channel throughput case using Algorithm 1, and LG-S selects channels in the *stochastic* channel throughput case using Algorithm 2. For CSE.M, it always selects the top-3 channels with the highest UCB (upper confidence bound) indices $\hat{g}_i^t + \sqrt{3 \ln t/(2N_i^t)}$. As for the Oracle in the *time-varying* channel throughput case, it knows $\boldsymbol{u}$ and $\boldsymbol{v}_t$ and thus can calculate the optimal selection vector $\boldsymbol{p}_t^*$ by solving $\max_{\boldsymbol{u}^{\intercal}\boldsymbol{p} \geq \rho_1} (\boldsymbol{u} \circ \boldsymbol{v}_t)^{\intercal}\boldsymbol{p}$ at each time slot t.[4] For the Oracle in the *stochastic* channel throughput case, it can compute the optimal $\boldsymbol{p}^*$ by solving $\max_{\boldsymbol{u}^{\intercal}\boldsymbol{p} \geq \rho_1} (\boldsymbol{u} \circ \boldsymbol{v})^{\intercal}\boldsymbol{p}$ and select channels with at every time slot. Therefore, the Oracles in both cases can select channels that provide the maximum throughput and guarantee that the average total primary-free probability is at least ρ_1 by selecting the channels with the optimal selection vector at every time slot t.

We run the simulation for $T = 10,000$ rounds and compare the cumulative/per-time-slot compound rewards (throughput) and violations of LG-V, LG-S and CSE.M averaged over 200 runs. Note that we do not compare their regrets because the regrets of LG-V, LG-S, and CSE.M are defined differently: the regret of LG-V in (1) and the regret of LG-S in (2) compare the reward with the *constrained* optimal

3. We use ρ with different indices for different applications.

4. $\circ$ is the element-wise product of two vectors.



(a) Cumulative compound reward (b) Per-time-slot compound reward (c) Cumulative violation (d) Per-time-slot violation

Fig. 1. Compound rewards (throughput) and violations of CSE.M, Oracle, and LG-V. $M = 10$, $m = 3$, $\rho_1 = 1.5$, and $T = 10,000$ ($\gamma = 0.16$, $\eta = 1.67$) in opportunistic multichannel selection with *time-varying channel throughput*.

strategy while the regret of CSE.M compares the reward with *unconstrained* optimal strategy. In particular, for LG-V, the cumulative compound reward at t is calculated by $\sum_{\tau=1}^{t}\sum_{i\in\mathcal{I}_\tau} g_i^\tau$; the cumulative compound reward of the Oracle is calculated by $\sum_{\tau=1}^{t}(\boldsymbol{u}\circ\boldsymbol{v}_\tau)^\intercal \boldsymbol{p}_\tau^*$. For LG-S, the cumulative compound reward at t is calculated in the same way with LG-V; the cumulative compound reward of the Oracle at t is calculated by $t(\boldsymbol{u}\circ\boldsymbol{v})^\intercal\boldsymbol{p}^*$. The per-time-slot compound reward at t is the ratio between the cumulative compound reward and t. Besides, the long-term cumulative violation at t is calculated by $(\sum_{\tau=1}^{t}(\rho_1 - \sum_{i\in\mathcal{I}_\tau} u_i^\tau))^+$ and the per-time-slot violation at t is the ratio between the cumulative violation and t.

Fig. 1 shows the cumulative/per-time-slot compound rewards (throughput) and cumulative/per-time-slot violations of LG-V, and CSE.M in the *time-varying* channel throughput case. As shown in Fig. 1a, the cumulative compound reward of LG-V almost coincides with the cumulative compound reward of the Oracle at each time slot. To gain more insight, as shown in Fig. 1b, the per-time-slot compound reward of LG-V is decreasing and smaller than that of the Oracle in the first few time slots ($t \leq 237$). This is because before this time slot, LG-V does not have enough knowledge about the primary-free probabilities and throughput of the channels thereby selecting the channels that have low compound rewards/throughput but are less likely to violate the guarantee threshold ρ_1. After $t = 237$, the per-time-slot compound reward of LG-V keeps increasing and gets closer to the Oracle. This means LG-V is getting more accurate in estimating the channel parameters and selecting channels that have larger compound rewards/throughput after $t = 237$. CSE.M, however, its cumulative compound reward and per-time-slot compound reward keep increasing and are larger than the Oracle. This is because CSE.M merely selects channels that maximize the compound throughput without considering the guarantee threshold of the total primary-free probability in multichannel selection.

For the cumulative violation, as shown in Fig. 1c, LG-V has a lower cumulative violation than that of CSE.M in each time slot. Moreover, the cumulative violations of LG-V and CSE.M become increasingly separated from each other. This means that LG-V can select channels that are less likely to violate the guarantee threshold. This is also verified in Fig. 1d where the per-time-slot violation of LG-V keeps decreasing and the per-time-slot violation of CSE.M keeps increasing. As for the Oracle, it can select the channels without any violation thus the violation is always 0 (not drawn in Figs. 1c and 1d) at each time slot.

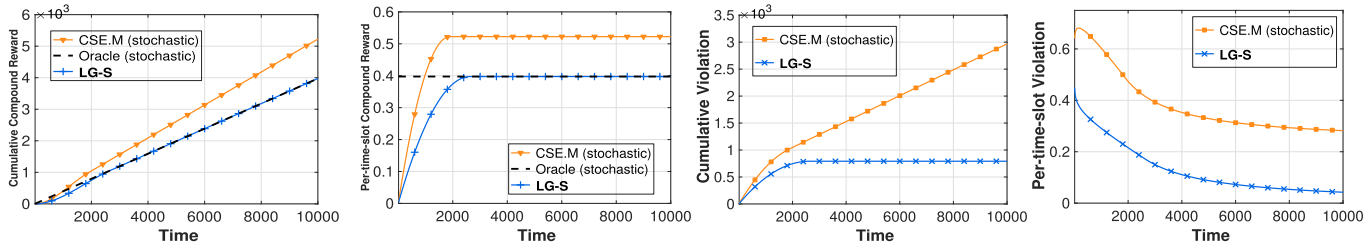
Fig. 2 shows the cumulative/per-time-slot compound rewards (throughput) and cumulative/per-time-slot violations of LG-S, and CSE.M in the *stochastic* channel throughput case. As shown in Fig. 2a, the cumulative compound reward of LG-S is closer to that of the Oracle than that of CSE.M. In Fig. 2b, the per-time-slot compound reward of LG-S is smaller than that of the Oracle when LG-S is learning about the channel parameters ($t \leq 2641$). After $t = 2641$, the per-time-slot compound reward of LG-S stays the same with that of the Oracle as it finds the optimal channel selection vector. Figs. 2c and 2d also demonstrate that LG-S achieves much smaller cumulative violations and per-time-slot violations than CSE.M by obeying the threshold constraint on the primary-free probabilities. In particular, the cumulative violation of LG-S stops increasing as LG-S can find the optimal channel selection vector and no longer violates the constraint after $t = 2641$.

Therefore, our simulation shows that LG-V is effective in selecting channels with *time-varying* channel throughput and LG-S is effective in selecting channels with *stochastic* channel throughput for the opportunistic multichannel selection problem with the guarantee threshold ρ_1.

4.2 Data-Guaranteed Mobile Crowdsensing

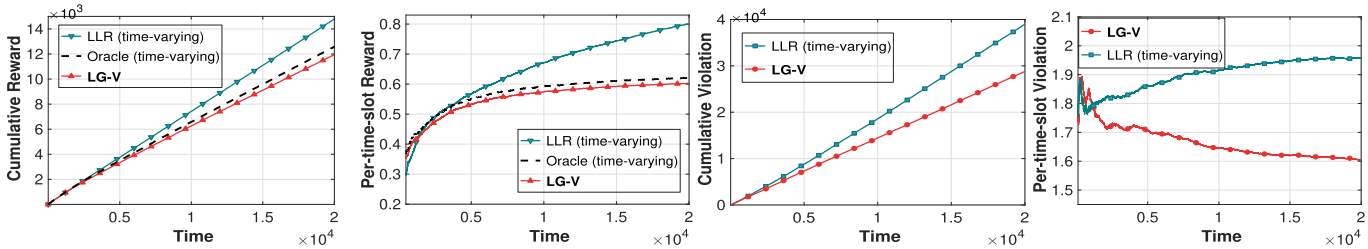
4.2.1 Problem Description

In the paradigm of mobile crowdsensing, different individuals/participants with sensing and computing devices are organized to collect data for a specific sensing task [2]. For example, smart phone users have been organized to use the equipped gravity sensors on their phones to detect earthquakes in [19]. Due to the heterogeneity of the mobile crowdsensing participants and the variability of manufacturing quality of the sensing devices, both the amount and the quality of collected data can vary *randomly* from different participants and different mobile sensing devices [20]. Thus, the amount of *qualified data* (or useful data), which is the product of the amount of data and the quality of data, also varies across different participants. Hence, a task organizer has to carefully choose participants in order to collect both massive and high quality (i.e., low-level of corruption or noise) sensing data. In many mobile crowdsensing tasks, task organizers have to choose multiple participants. For example, for dust level sensing in a large city, the task organizer needs to select several participants to cover a sensing area [9]. In addition, the participant selection procedure should be performed for multiple times instead of one time due to the randomness in collecting the sensing data. More importantly, when choosing a participant, the task organizer must consider both the



(a) Cumulative compound reward (b) Per-time-slot compound reward (c) Cumulative violation (d) Per-time-slot violation

Fig. 2. Compound rewards (throughput) and violations of CSE.M, Oracle, and LG-S. $M = 10$, $m = 3$, $\rho_1 = 1.5$, and $T = 10,000$ ($\kappa = 1492.1$) in opportunistic multichannel selection with *stochastic channel throughput*.



(a) Cumulative compound reward (b) Per-time-slot compound reward (c) Cumulative violation (d) Per-time-slot violation

Fig. 3. Compound rewards (the amount of qualified data) and violations of LLR, Oracle, and LG-V. $M = 100$, $m = 10$, $\rho_2 = 6$, and $T = 20,000$ ($\gamma = 0.25$, $\eta = 10.1$) in data-guaranteed mobile crowdsensing with *time-varying quality of data*.

amount of sensing data and the quality of sensing data that the participant can gather for two reasons: first, a larger amount of data makes the data more statistically significant; second, higher quality of data means the data is less corrupted or noisy. Therefore, the task organizer should consider *selecting participants that maximize the total amount of qualified data and keep the average total amount of collected data above the guarantee requirement* ρ_2.

Now consider selecting m participants from M candidate participants for a crowdsensing task in T time slots. Our LG-V policy can be applied by taking the M candidate participants as M arms, the amount of data participant i can gather as u_i, the quality of data participant i can provide as *the time-varying* v_i^t, for $1 \leq i \leq M$ and $t = 1, \ldots, T$. Our LG-S policy can be applied similarly by taking the the quality of data participant i as a *stochastic* random variable v_i for $1 \leq i \leq M$. Here, $\boldsymbol{g}_t$ represents the amount of qualified data that the M candidates can gather and $\boldsymbol{p}_t$ is the probability selection vector of the M candidates.

The constraint for the data-guaranteed mobile crowdsensing problem is $\boldsymbol{u}^\intercal \boldsymbol{p}_t \geq \rho_2$, where $\boldsymbol{u}^\intercal \boldsymbol{p}_t$ represents the amount of total collected data at time slot t and ρ_2 is the guarantee threshold.

4.2.2 Performance Evaluation

Consider selecting $m = 10$ participants from $M = 100$ candidate participants with $\rho_2 = 6$, i.e., no less than 6 units of data is collected at each time slot, for $T = 20,000$ slots. The amount of data that participant i can gather is modeled as a Bernoulli random variable with mean u_i uniformly random generated in $[0.1, 0.8]$, i.e., $u_i = \text{Uniform}(0.1, 0.8)$ and $u_i^t = \text{Bernoulli}(u_i)$, for $t \geq 1, 1 \leq i \leq 100$.

In the case that quality of data is time-varying, the quality of data for each participant i, v_i^t, is time-varying as follows:

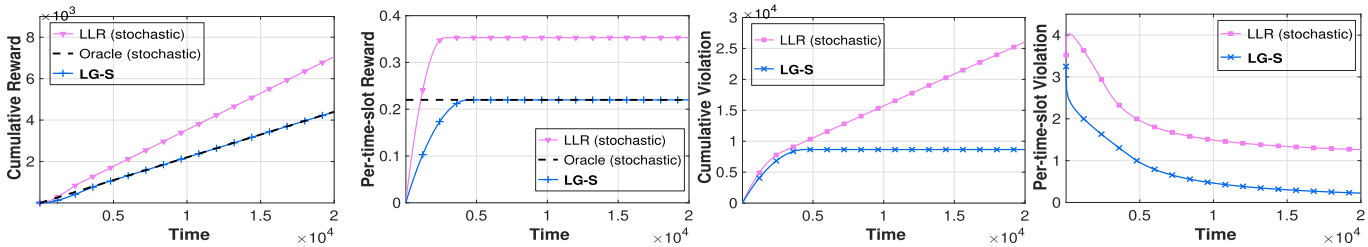
$$v_i^{t+1} = \begin{cases} v_i^t, & \text{if } u_i^t = 0, \\ v_i^t - 50/T, & \text{if } u_i^t = 1 \text{ and } v_i^t \geq 50/T, \\ v_i^1, & \text{if } u_i^t = 1 \text{ and } v_i^t < 50/T, \end{cases}$$

for $1 \leq i \leq 100, t > 1$ and $v_i^1 = \text{Uniform}(0, 1)$. This time-varying fashion can well model the sensing quality change due to the sensing device degradation when a device is used in multiple-round sensing [21].

In the case that the quality of data is stochastic, the quality of data for each participant i, v_i, is modeled as a [0,1]-truncated normal distributed random variable with standard deviation 0.5 and mean $v_i = v_i^1$ for $1 \leq i \leq 100$. This modeling can be used in a stochastic sensing environment where the mobile sensing participants are in designate sensing areas.

For both cases, we compare LG-V and LG-S with one of the state-of-the art policies, *Learning with Linear Rewards* (LLR), proposed in [22] which is an effective multi-armed bandit policy for selecting multiple arms using UCB indices. Specifically, LLR selects the 10 highest UCB indices $\hat{g}_i^t + \sqrt{(m+1)\ln t / N_i^t}$. The Oracles of LG-V and LG-S are calculated in the same way as in Section 4.1, and we use the methods described in Section 4.1 to calculate the cumulative/per-time-slot compound rewards and violations. We would like to point out that there is a novel online budget limited policy which stops learning once the budget is exceeded proposed in [9]. LG-V and LG-S, on the contrary, are non-stopping policies with no budget limitation.

The results averaged over 200 runs are shown in Figs. 3 and 4. Both the cumulative (Figs. 3a and 4a) and the per-time-slot (Figs. 3b and 4b) compound rewards of LG-V and LG-S are getting closer the Oracle after some time slots. This means LG-V for the *time-varying* data quality case and LG-S for the *stochastic* data quality case can both learn the amount of data and the quality of data accurately. Thus LG-V and LG-S both can make judicious participant selections after



(a) Cumulative compound reward (b) Per-time-slot compound reward (c) Cumulative violation (d) Per-time-slot violation

Fig. 4. Compound rewards (the amount of qualified data) and violations of LLR, Oracle, and LG-S. $M = 100$, $m = 10$, $\rho_2 = 6$, and $T = 20,000$ ($\kappa = 1757.7$) in data-guaranteed mobile crowdsensing with *stochastic quality of data*.

some time. The cumulative/per-time-slot rewards of LLR are larger than the Oracle as LLR only maximizes the amount of qualified data and ignores the data-guarantee requirement. Therefore, LLR has larger cumulative/per-time-slot violations than LG-V (LG-S) as shown in Fig. 3c (Fig. 4c) and Fig. 3d (Fig. 4d). Thus, this simulation shows the effectiveness of LG-V and LG-S in selecting participants for the data-guaranteed mobile crowdsensing problem with the guarantee threshold ρ_2.

4.3 Stability-Guaranteed Crowdsourced Transcoding

4.3.1 Problem Description

Attracted by the increasing popularity of live video streaming, many amateur broadcasters join in the live streaming platforms and produce video contents with different qualities (720p, 1080p, etc.) and codecs (H.264, H.265, etc.). To better serve their viewers with real-time live streaming, live streaming platforms have to transcode the video contents into industrial standard representations [3]. Due to the computation-intensive nature of video transcoding, these platforms start to resort to the crowdsourced transcoding model, where a platform can offload video transcoding assignments to viewers with computing devices [3]. To assign a transcoding assignment to viewers, a platform has to check whether the viewers are available for the assignment and examine the viewers' (devices') computing power. On the one hand, as the online durations of viewers are different, the availability probabilities of viewers are also different. In addition, the computing power of different viewers can be affected by the viewers' behaviors and different hardware configurations. Therefore, the *effective computing power* of a viewer, which can be represented by the product of the viewer's availability probability and the viewer's computing power, varies from viewer to viewer. On the other hand, the platform has to assign the task to a limited number of viewers as it has a limited online viewer capacity. Hence, the platform has to schedule the transcoding assignment to viewers in a multi-round fashion since the assignment cannot be completed in one time due to the variability of viewers' availability probabilities and computing power. Moreover, the platform has to consider both the viewers' availability probabilities and computing power in scheduling the transcoding assignment. A higher total availability probability means more viewers can participate in transcoding at each round to keep live stream transcoding stable without any interruption. Alternatively, larger total effective computing power means the platform can transcode more live streaming videos. Therefore, the platform should consider *selecting viewers that would maximize the total effective computing power while guaranteeing the average number of available viewers above* ρ_3.

Formally, let us consider selecting m viewers from M regular viewers for a crowdsourced transcoding assignment in T time slots. Thus, LG-V can be applied by considering M viewers as M arms, the viewers' availability probabilities as u_i, and the viewers' computing power as the *time-varying* v_i^t at t, $1 \le i \le M$ and $t = 1, \ldots, T$. LG-S can also be applied except that the viewer i's computing power is taken as a *stochastic* random variable v_i for $1 \le i \le M$. Here, $\boldsymbol{g}_t$ is the effective computing power of the M viewers and $\boldsymbol{p}_t$ is the probability selection vector of the M viewers.

The constraint for the stability-guaranteed crowdsourced transcoding problem is $\boldsymbol{u}^\intercal \boldsymbol{p}_t \ge \rho_3$, where $\boldsymbol{u}^\intercal \boldsymbol{p}_t$ represents the number of available viewers at time slot t and ρ_3 is the guarantee threshold.

4.3.2 Performance Evaluation

Consider selecting $m = 4$ viewers from $M = 15$ regular viewers and setting $\rho_3 = 2.5$, i.e., at least 2.5 viewers are available at each time slot, for $T = 12,000$ slots. In particular, viewer i's availability probability is modeled as a Bernoulli random variable with mean u_i generated with a uniform distribution with in $[0.2, 0.9]$, i.e., $u_i = \text{Uniform}(0.2, 0.9)$, and $u_i^t = \text{Bernoulli}(u_i)$.

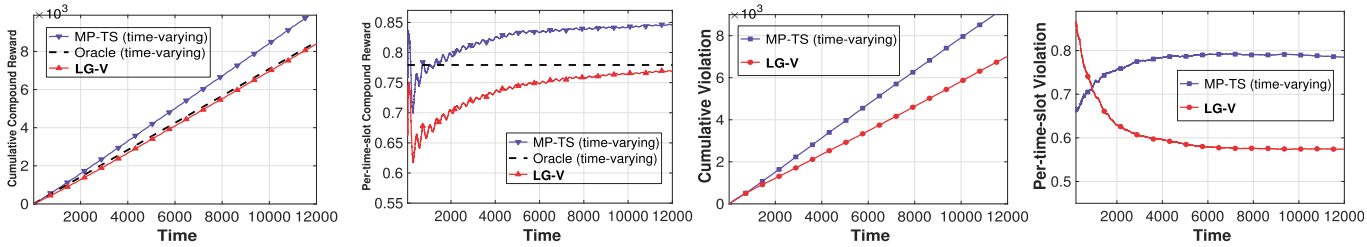
In the case that the viewers' computing power are time-varying, viewer i's computing power is modeled as a time-varying variable v_i^t as follows:

$$v_i^{t+1} = v_i^t[1 + \sin(2\pi t/24)]/2,$$

for $t > 1$ and $v_i^1 = \text{Uniform}(0, 1)$, $1 \le i \le 15$. This sinusoidal-varying pattern [23] can capture the periodic (hourly or daily) change of a viewer's computing power.

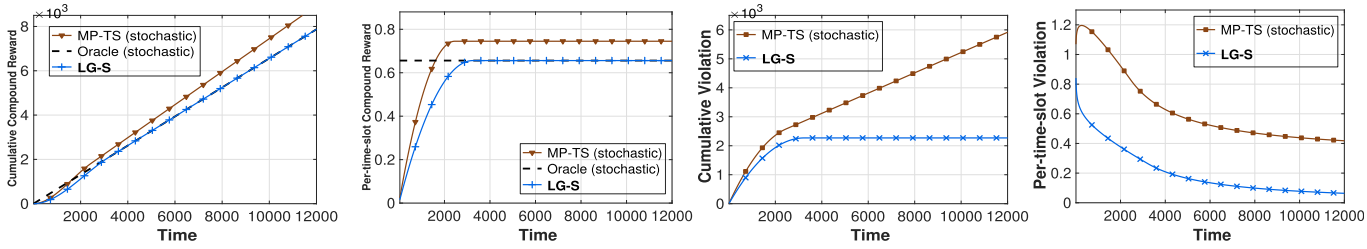
In the case that the viewers' computing power are stochastic, viewer's computing power v_i is modeled as a [0,1]-truncated normal distributed variable with standard deviation 0.5 and mean $v_i = v_i^1$ for $1 \le i \le 15$. This modeling can be employed to capture the variations of the viewer's computing power in the long run.

We notice that [3] proposes a smart scheduling scheme to select viewers for crowdsourced transcoding. However, it requires prior knowledge of viewers' availability probabilities and therefore does not fit into the online learning framework. For evaluation purpose, we compare another state-of-the-art online learning policy, *multi-play Thompson sampling* (MP-TS) [24], with LG-V and LG-S. In our simulation, MP-TS selects the top-4 arms ranked by the posterior samples sampled from the Beta distributions, $\text{Beta}(A_i, B_i)$,



(a) Cumulative compound reward (b) Per-time-slot compound reward (c) Cumulative violation (d) Per-time-slot violation

Fig. 5. Compound rewards (effective computing power) and violations of MP-TS, Oracle, and LG-V. $M = 15$, $m = 4$, $\rho_3 = 2.5$, and $T = 12,000$ ($\gamma = 0.17$, $\eta = 2.3$) in stability-guaranteed crowdsourced transcoding with *time-varying computing power*.



(a) Cumulative compound reward (b) Per-time-slot compound reward (c) Cumulative violation (d) Per-time-slot violation

Fig. 6. Compound rewards (effective computing power) and violations of MP-TS, Oracle, and LG-S. $M = 15$, $m = 4$, $\rho_3 = 2.5$, and $T = 12,000$ ($\kappa = 1547.5$) in stability-guaranteed crowdsourced transcoding with *stochastic computing power*.

$1 \leq i \leq 15$. Particularly, A_i=B_i=1 at t=1. At time $t > 1$, by performing a Bernoulli trial with success probability g_i^t for each arm i that is selected at t, A_i increases by 1 if the result is 1, otherwise, B_i increases by 1. Similarly, we calculate the Oracle, the cumulative/per-time-slot compound rewards, and violations in the same way as described in Section 4.1.

Figs. 5 and 6 show the results of our simulation averaged over 200 runs. From Figs. 5a and 5b in the *time-varying* computing power case, we see both the cumulative reward and the per-time-slot reward of LG-V are approaching the Oracle gradually. This indicates that LG-V is as superior in maximizing effective computing power as the Oracle does while learning the availability probabilities and computing power of viewers. Figs. 6a and 6b demonstrate the same results of LG-S in the *stochastic* computing power case. MP-TS focuses on maximizing the effective computing power so that it even has a higher compound reward than the Oracle. But by doing so, it violates the guarantee threshold as shown in Figs. 5c, 5d, 6c and 6d. On the contrary, in both cases, LG-V and LG-S have much lower cumulative/per-time-slot violations and their per-time-slot violations keep decreasing to zero. Thus, this simulation shows the effectiveness of LG-V and LG-S in selecting viewers for the stability-guaranteed crowdsourced transcoding problem with the guarantee threshold ρ_3.

5 Discussions

In this section, we first demonstrate the tradeoff between the reward and violation of LG-V and LG-S and other algorithms in the three applications. We also show the impact of the parameter variations, namely, the guarantee threshold ρ and the number of selected arms m on the regrets and violations of LG-V and LG-S.

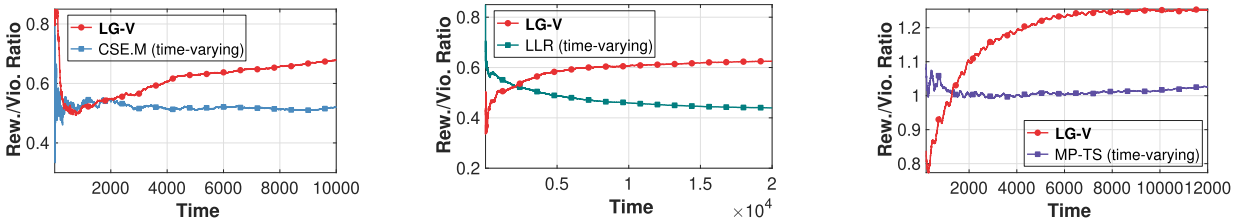
5.1 Reward and Violation Tradeoff

To put things into perspective, we compare the ratios between the cumulative compound rewards and the cumulative violations of LG-V, LG-S and other algorithms for the three applications. Such ratios show how much reward an algorithm can gain for each unit violation it has made. The experiment results are averaged over 200 runs of each algorithm and illustrated in Figure 7.

Figs. 7a, 7b, and 7c show that LG-V achieves a higher reward/violation ratio than the other algorithms in the three different network applications. Similarly, LG-S can also achieve a higher reward/violation ratio than the other algorithms as shown in Figs. 8a, 8b and 8c. This means that LG-V and LG-S both achieve a better tradeoff between rewards and violations and accumulates most reward for each unit violation they incur.

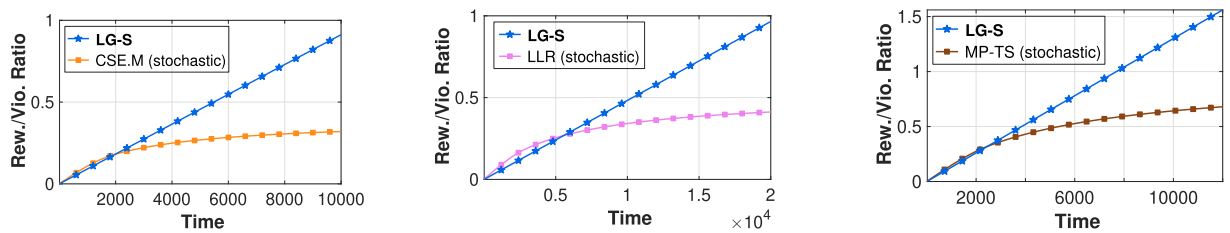
5.2 Impact of Parameter Variations

In this subsection, we study the impact of variations of the guarantee threshold ρ_1 and the number of arms m selected at each time slot on the regrets and violations of LG-V and LG-S by applying it to the opportunistic multi-channel selection application where the parameter M is fixed ($M = 10$). Similar results can be found in the other two applications and therefore these results are omitted for conciseness.• *Impact of variations of* ρ. To study the impact of variations of ρ_1 on the regret and violation, we run LG-V and LG-S using the parameter setting $m = 5$ and vary ρ_1 from 0.5 to 2.5 with the step size 0.5. (Here we have ensured that there exist channels that can satisfy the threshold constraint $\rho_1 = 2.5$.) Specifically, the regret of LG-V is calculated by $\sum_{\tau=1}^{t}[\boldsymbol{g}^\intercal \boldsymbol{p}_\tau^* - \sum_{i\in\mathcal{I}_\tau} g_i^\tau]$ and the regret of LG-S is calculated by $t\boldsymbol{g}^\intercal \boldsymbol{p}^* - \sum_{\tau=1}^{t}\sum_{i\in\mathcal{I}_\tau} g_i^\tau$. For each of LG-V and LG-V, we obtain the cumulative regret and cumulative violation at the end of $T = 50,000$. The regrets of LG-V and LG-S in Fig. 9a are decreasing as ρ_1 increases because the number of possible choices



(a) Opportunistic multichannel selection (b) Data-guaranteed mobile crowdsensing (c) Stability-guaranteed crowdsourced transcoding

Fig. 7. Ratio between the cumulative compound reward and the cumulative violation of LG-V and other algorithms in different applications with *time-varying* rewards.



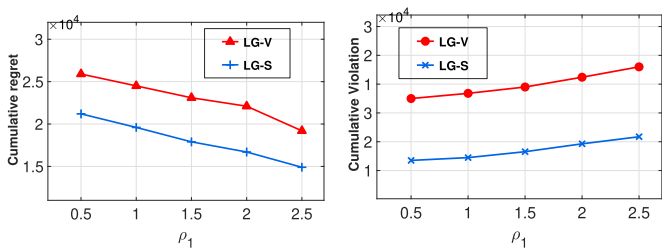
(a) Opportunistic multichannel selection (b) Data-guaranteed mobile crowdsensing (c) Stability-guaranteed crowdsourced transcoding

Fig. 8. Ratio between the cumulative compound reward and the cumulative violation of LG-S and other algorithms in different applications with *stochastic* rewards.

(search space) for the m channels decreases as the number of channels satisfying the guarantee threshold decreases. As shown in Fig. 9b, the violations of LG-V and LG-S increase but the regrets decrease as ρ_1 increases. The violations increase because the threshold constraint ρ_1 is getting larger and more difficult to satisfy.• *Impact of variations of* m. To study the impact of variations of m on the regret and violation, we run LG-V and LG-S using the parameter setting $\rho_1 = 1.2$ and vary m from 2 to 6 with the step size 1. We obtain the cumulative regrets and cumulative violations of LG-V and LG-S at the end of $T = 50,000$. The result is shown in Fig. 10. The regrets of both LG-V and LG-S shown in Fig. 10a increase as m increases because the number of possible choices, i.e., the size of the search space, for the m channels increases; the violations of both LG-V and LG-S shown in Fig. 10b decrease as m increases because the constraint ($\rho_1 = 1.2$) on the total number of available channels is easier to satisfy when the number of selected channels increases.

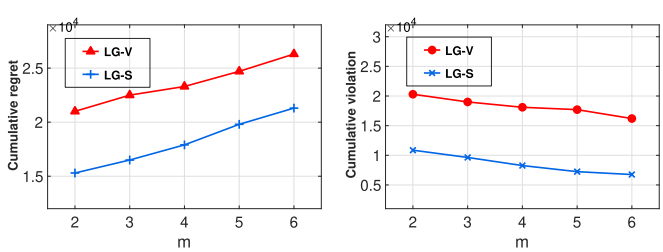
6 Related Work

There have been extensive studies regarding the online learning framework using multi-armed bandit model since the pioneering works [5], [6]. In this literature, our formulation of the stochastic multi-armed bandit model is related to the multi-armed bandit models with multiple plays, where multiple arms are selected at each time slot. [25] presents the Exp3.M algorithm to select multiple arms using exponential weights. [26] proposes the CUCB algorithm that selects multiple arms with the highest upper confidence bound (UCB) indices. [24] presents the multi-play Thompson Sampling algorithm (MP-TS) for arms with binary rewards. Our model differs from these multi-armed bandit models as we further consider the stochastic constraint or the guarantee requirement in selecting the multiple arms. Note that the constraint in our model is very different from that in the bandit with budgets [2], [27], [28], [29] and the bandit with knapsacks [30]. For these works, the optimal stopping time is considered since no arm can be selected/played if the budget/knapsack constraints are violated. However, the constraint in our model does not pose such restrictions and the arm selection procedure can continue without stopping. Our constraints, are in fact, related to the QoS guarantees of applications. This makes our problem more challenging as we need to consider the violations of the guarantee requirement introduced by such a non-stopping arm selection procedure. Our model also differs from the thresholding bandit in [31] which finds the best set of arms whose means are all above a given threshold through pure



(a) Cumulative regret (b) Cumulative violation

Fig. 9. Cumulative regret and cumulative violation of LG-V and LG-S versus ρ_1 in opportunistic multichannel selection. $M = 10$, $m = 5$, $T = 50,000$.



(a) Cumulative regret (b) Cumulative violation

Fig. 10. Cumulative regret and cumulative violation of LG-V and LG-S versus m in opportunistic multichannel selection. $M = 10$, $\rho_1 = 1.2$, $T = 50,000$.

explorations. Instead, the objective of our model is to select arms that maximize the cumulative compound rewards while satisfying the guarantee requirement. Our constrained MAB model is related to but different from the bandit model considered in [32] which tries to balance regret and violation. They only consider selecting a single arm without any time-varying multi-level rewards. Some other studies [33], [34] consider MAB models with non-stochastic rewards but neglect the possibility that rewards can be stochastic. While in our work, we consider how to select multiple arms and each arm is associated with *either time-varying or stochastic* multi-level rewards, making our model more flexible and better fit different network optimization problems.

There is another line of research on online learning framework using reinforcement learning techniques such as Markov decision process (MDP) to learn unknown parameters in time-varying or stochastic environments [35], [36], [37], [38], [39]. These works consider multiple states of actions (arms) and learn the parameters in different states to cope with different environments. However, such techniques are not suitable in time-varying and possibly adversarial environments where the states of actions cannot be well defined. Therefore, we consider stateless arms and design an exponential weight update policy (LG-V) and an upper confidence bound policy (LG-S) to adapt to the time-varying environments and stochastic environments, respectively. Thanks to such a stateless model, our two policies can find the optimal arms by learning the arms' stateless rewards efficiently. In contrast, other reinforcement learning techniques usually require a large amount of data and time to explore large state-action space to find the optimal strategy.

Due to the effectiveness of the online learning policies in finding the optimal decision and learning the unknown application parameters, a lot of works have incorporated them into network application optimizations. In [7], [8], [17], the authors consider selecting the single-best channel to maximize the throughput in opportunistic multichannel access networks, and [18] designs an online dynamic channel access policy. [22] proposes a combinatorial optimization framework based on UCB for graphs with unknown weights. [40] proposes a bandit learning algorithm with predicted context for virtual machine scheduling on cloud platforms. [41] formulates a multi-armed bandit based offload path selection scheme in edge unmanned aerial vehicle networks. In [42], we present preliminary results on the CMAB model with time-varying rewards for network application optimizations, and we extend the work here to capture broader network applications by considering a CMAB model with stochastic rewards. [43] designs an MAB-based learning algorithm for computation offloading in heterogeneous vehicular edge networks. [44] presents a CMAB-based approach for stochastic network utility maximization. [2] consider a CMAB model for worker recruitment in heterogeneous mobile crowdsensing. These works have been shown to be effective in different applications. However, they neglect the guarantee requirements in real-world network applications, where our policies can be applied and achieve better performance in terms of maximizing the cumulative compound reward while meeting the constraint.

7 Conclusion and Future Work

In this paper, we formulate the stochastic constrained MAB model with either *time-varying or stochastic multi-level rewards* for network application optimizations. We take into account the guarantee requirement and design two constrained MAB policies, LG-V for time-varying rewards and LG-S for stochastic rewards. We prove that both policies can achieve *sub-linear* regret and violation bounds. We apply LG-V and LG-S to three real-world network applications and show that our policies achieve better performance in terms of larger cumulative rewards (*close to the Oracle*) and smaller violations (*close to zero*) compared with the state-of-the-art algorithms.

There are several potential limitations in our work. First, our constrained MAB model only considers linear constraints, which may limit its applicability in network applications with nonlinear constraints. Second, our policies rely on the proper (i.e., time-varying or stochastic) assumptions on the rewards. For example, LG-S can achieve performance in accord with our theoretical results only if the rewards are stochastic instead of time-varying. Future work should address the limitations. In particular, future research should consider convex or concave constraints in real-world applications. It would also be interesting to design a single policy that has sub-linear regret and violation in both the time-varying and the stochastic environments. In addition, future research should explore other reinforcement learning techniques and develop dedicated distributed policies for distributed systems.

References

[1] Y. C. Liang, K. C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[2] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 179–188.

[3] Q. He, C. Zhang, and J. Liu, "CrowdTranscoding: Online video transcoding with massive viewers," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1365–1375, Jun. 2017.

[4] T. Bansal, D. Li, and P. Sinha, "Opportunistic channel sharing in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 852–865, Apr. 2014.

[5] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2/3, pp. 235–256, 2002.

[7] W. Dai, Y. Gai, and B. Krishnamachari, "Online learning for multi-channel opportunistic access over unknown markovian channels," in *Proc. 11th Annu. IEEE Int. Conf. Sens., Commun., Netw.*, 2014, pp. 64–71.

[8] P. Yang *et al.*, "Online sequential channel accessing control: A double exploration vs. exploitation problem," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4654–4666, Aug. 2015.

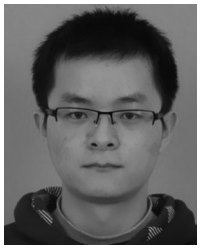
[9] K. Han, C. Zhang, and J. Luo, "Taming the uncertainty: Budget limited robust crowdsensing through online learning," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1462–1475, Jun. 2016.

[10] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *J. ACM*, vol. 53, no. 3, pp. 324–360, 2006.

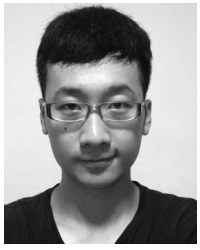
[11] K. Cai, "Algorithm implementations of learning with guarantee via constrained multi-armed bandit," 2021. [Online]. Available: https://codeocean.com/capsule/3948685/tree

[12] S. Agrawal and N. R. Devanur, "Bandits with concave rewards and convex knapsacks," in *Proc. 15th ACM Conf. Econ. Comput.*, 2014, pp. 989–1006.

[13] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proc. 14th Annu. ACM Symp. Theory Comp.*, 2008, pp. 681–690.
[14] N. Megiddo, "On the complexity of linear programming," in *Advances in Economic Theory: Fifth World Congress*, Cambridge, U.K.: Cambridge Univ. Press, 1987, pp. 225–268.
[15] E. Boursier and V. Perchet, "SIC-MMAB: Synchronisation involves communication in multiplayer multi-armed bandits," in *Proc. 33rd Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12 071–12 080.
[16] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio : State-of-the-art and recent advances," *IEEE Signal Process. Mag.*, vol. 29, no. 3, pp. 101–116, May 2012.
[17] C. Tekin and M. Liu, "Approximately optimal adaptive learning in opportunistic spectrum access," in *Proc. IEEE Comput. Commun. Soc.*, 2012, pp. 1548–1556.
[18] Y. Liu and M. Liu, "An online approach to dynamic channel access and transmission scheduling," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2015, pp. 187–196.
[19] A. Visuri, Z. Zhu, D. Ferreira, S. Konomi, and V. Kostakos, "Smartphone detection of collapsed buildings during earthquakes," in *Proc. ACM UbiComp*, 2017, pp. 557–562.
[20] C. Zhao, S. Yang, and J. A. McCann, "On the data quality in privacy-preserving mobile crowdsensing systems with untruthful reporting," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 647–661, Feb. 2021.
[21] O. Yurur and C. H. Liu, *Generic and Energy-Efficient Context-Aware Mobile Sensing*, Boca Raton, FL, USA: CRC Press, 2015.
[22] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1466–1478, Oct. 2012.
[23] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 199–207.
[24] J. Komiyama, J. Hondaand, and H. Nakagawa, "Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1152–1161.
[25] T. Uchiya, A. Nakamura, and M. Kudo, "Algorithms for adversarial bandit problems with multiple plays," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2010, pp. 375–389.
[26] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework, results and applications," in *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, 2013, pp. 151–159.
[27] W. Ding, T. Qiny, X.-D. Zhang, and T.-Y. Liu, "Multi-armed bandit with budget constraint and variable costs," in *Proc. 27th AAAI Conf. Artif. Intell.*, 2013, pp. 232–238.
[28] R. Combes, C. Jiang, and R. Srikant, "Bandits with budgets: Regret lower bounds and optimal algorithms," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 1, pp. 245–257, 2015.
[29] Y. Xia, T. Qin, W. Ma, N. Yu, and T.-Y. Liu, "Budgeted multi-armed bandits with multiple plays," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2210–2216.
[30] S. Agrawal, N. R. Devanur, L. Li, and N. Rangarajan, "An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives," in *Proc. 29th Conf. Learn. Theory*, 2016, pp. 4–18.
[31] A. Locatelli, M. Gutzeit, and A. Carpentier, "An optimal algorithm for the thresholding bandit problem," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1690–1698.
[32] M. Mahdavi, T. Yang, and R. Jin, "Online decision making under stochastic constraints," in *Proc. NIPS Workshop Discrete Optim. Mach. Learn.*, 2012, pp. 1–8.
[33] Y. Gur, A. J. Zeevi, and O. Besbes, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 199–207.
[34] P. Zhao, L. Zhang, Y. Jiang, and Z.-H. Zhou, "A simple approach for non-stationary linear bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 746–755.
[35] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–10.
[36] C. Szepesvári, "Algorithms for reinforcement learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 4, no. 1, pp. 1–103, 2010.
[37] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Hoboken, NJ, USA: Wiley, 2014.
[38] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. Cambridge, MA, USA: MIT press, 2015.
[39] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT press, 2018.
[40] J. Yang and S. Ren, "Bandit learning with predicted context: Regret analysis and selective context query," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
[41] A. Mukherjee, S. Misra, V. S. P. Chandra, and M. S. Obaidat, "Resource-optimized multiarmed bandit-based offload path selection in edge UAV swarms," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4889–4896, Jun. 2019.
[42] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "An online learning approach to network application optimization with guarantee," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 2006–2014.
[43] A. Bozorgchenani, S. Maghsudi, D. Tarchi, and E. Hossain, "Computation offloading in heterogeneous vehicular edge networks: On-line and off-policy bandit solutions," *IEEE Trans. Mobile Comput.*, early access, May 25, 2021, doi: 10.1109/TMC.2021.3082927.
[44] A. Verma and M. K. Hanawal, "Stochastic network utility maximization with unknown utilities: Multi-armed bandits approach," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 189–198.

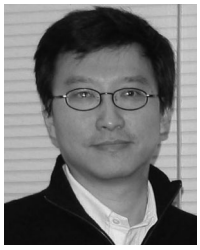


Kechao Cai (Member, IEEE) received the PhD degree from The Chinese University of Hong Kong, in 2019. He is currently an Assistant Professor with the School of Electronics and Communication Engineering, Sun Yat-Sen University in China. His current research area includes distributed protocol design, data analytics of online social networks, and online learning algorithms.



Xutong Liu received the BS degree from the University of Science and Technology of China, in 2017. He is currently working toward the PhD degree with the Chinese University of Hong Kong under the supervision of prof. John C.S. Lui. His current research interests include network science, online learning and, deep learning with graphs.

Yu-Zhen Janice Chen received the BS degree from The Chinese University of Hong Kong, in 2019. She is currently working toward the PhD degree with the University of Massachusetts, Amherst. Her current research interests include network science and online learning.



John C.S. Lui (Fellow, IEEE) received the PhD in computer science from UCLA. He is currently the Choh-Ming Li chair professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His current research interests include communication networks, system security (e.g., cloud security, mobile security, etc.), network economics, network sciences, large-scale distributed systems and performance evaluation theory. He serves in the editorial board of *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, *Journal of Performance Evaluation* and *International Journal of Network Security*. He was the chairman of the CSE Department from 2005-2011. He received various departmental teaching awards and the CUHK Vice-Chancellors Exemplary Teaching Award. He is also a corecipient of the IFIP WG 7.3 Performance 2005 and IEEEIFIP NOMS 2006 best student paper awards. He is an elected member of the IFIP WG 7.3, fellow of the ACM, and croucher senior research fellow. His personal interests include films and general reading.