An Online Learning Approach to Network Application Optimization with Guarantee

Kechao Cai, Xutong Liu, Yu-Zhen Janice Chen, John C.S. Lui
Department of Computer Science & Engineering, The Chinese University of Hong Kong
Email: {kccai, liuxt, yzchen5, cslui}@cse.cuhk.edu.hk

***Abstract*—Network application optimization is essential for improving the performance of the application as well as its user experience. The network application parameters are crucial in making proper decisions for network application optimizations. However, many works are impractical by assuming a priori knowledge of the parameters which are usually unknown and need to be estimated. There have been studies that consider optimizing network application in an online learning context using multi-armed bandit models. However, existing frameworks are problematic as they only consider to find the optimal decisions to minimize the regret, but neglect the *constraints* (or guarantee) requirements which may be excessively violated. In this paper, we propose a novel online learning framework for network application optimizations with guarantee. To the best of our knowledge, we are the first to formulate the stochastic constrained multi-armed bandit model with time-varying "*multi-level rewards*" by taking both "*regret*" and "*violation*" into consideration. We are also the first to design a constrained bandit policy, Learning with Minimum Guarantee (LMG), with provable *sub-linear* regret and violation bounds. We illustrate how our framework can be applied to several emerging network application optimizations, namely, (1) opportunistic multichannel selection, (2) data-guaranteed crowdsensing, and (3) stability-guaranteed crowdsourced transcoding. To show the effectiveness of LMG in optimizing these applications with different minimum requirements, we also conduct extensive simulations by comparing LMG with existing state-of-the-art policies.**

I. Introduction

In a world where network applications are becoming ubiquitous and competitive, it is important for a network application to be optimized not only to differentiate itself from other applications but also to provide the best possible experience for its users. When considering performance optimization for different network applications, a common feature emerges: one has 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 to speed up live stream transcoding.

The parameter settings are essential prerequisites in making optimization decisions for the network applications. Many works in the literature assume that the parameters are known as a priori knowledge. However, the parameters of a network application are not fixed and can *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 estimate the network application parameters and to relax the impractical assumption, it is natural to consider optimizing network application in an *online learning context*, where the decision maker is not required to possess prior statistics of the parameters but will try to learn 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 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 only consider finding the optimal decision that maximizes the cumulative reward, e.g., (the channel throughput, the sensing revenue, etc.) while neglecting the *constraints or guarantee* requirements in network application optimizations. It follows that the performance of these policies is measured by *regret*, the difference between a learning policy and the Oracle policy which always makes the optimal decision.

In fact, in addition to seeking the optimal decision, many real-world network application optimization tasks have some minimum guarantee requirements (or constraints), and therefore a decision maker may *violate these requirements* while learning the parameters to optimize the network 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 minimum guarantee requirement.

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

Hence, there is a need to have an additional performance measure for a learning policy, *violation*, which measures the cumulative violations of the minimum guarantee requirement that a learning policy has made in the online learning process.

To the best of our knowledge, we are the first to propose an online learning framework and formulate the stochastic constrained MAB model with time-varying multi-level rewards to tackle the optimization task with a minimum guarantee requirement in various network applications. Specifically, each arm is associated with a stochastic level-1 reward and a time-varying level-2 reward (can be further extended to level-n reward), and together they produce a compound reward. At each time slot, a decision maker selects m arms from M arms ($1 \leq m \leq M$), observes the level-1 and level-2 rewards of the m selected arms, and receives the compound rewards from the m selected arms. Moreover, to satisfy the minimum guarantee requirement, there is a *minimum guarantee threshold* ρ such that the total level-1 reward of the selected m arms should be at least ρ.

Our goal is to find a learning policy to *maximize the total compound reward as well as to keep the average total level-1 reward above the minimum guarantee threshold* ρ. The key issue is to balance between maximizing the total compound reward (i.e., minimizing the regret) and satisfying the minimum guarantee threshold constraint (i.e., keeping low violation). To achieve this, we are the first work to design a constrained bandit policy, *Learning with Minimum Guarantee* (LMG), which has the attractive property that achieves *sub-linear* regret and violation bounds. To show the general applicability of our framework, we illustrate how our LMG can be applied to several network applications: (1) opportunistic multichannel selection, (2) data-guaranteed crowdsensing, and (3) stability-guaranteed crowdsourced transcoding. We also compare LMG with existing state-of-the-art algorithms and conduct extensive simulations to show the effectiveness of our LMG policy.

Contributions: Our contributions are as follows: (i) To our best knowledge, we are the first to propose an online learning framework using the stochastic constrained MAB model with time-varying multi-level rewards for network application optimizations with the minimum guarantee requirements. (ii) We design a new constrained bandit policy, LMG, by taking the minimum guarantee requirement ρ into consideration with provable *sub-linear* regret and violation bounds. (iii) We show how LMG can be applied to three different network applications. (iv) We demonstrate the effectiveness of our LMG policy by comparing it with existing state-of-the-art policies and show that LMG achieves better network application optimizations in terms of maximizing the cumulative compound reward while meeting the minimum guarantee requirements.

The rest of the paper is organized as follows. The framework using stochastic constrained MAB model with time-varying multi-level rewards is presented in Sec. II. The detailed LMG policy design and the proof of sub-linear regret and violation bounds are elaborated in Sec. III. The applications of LMG and the comparison results are presented in Sec. IV. Related work is given in Sec. V. Finally, Sec. VI concludes the paper.

II. Constrained Multi-armed Bandit Model

In this section, we present the details of our online learning framework using stochastic constrained multi-armed bandit model with time-varying multi-level rewards.

Formally, 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 $U_i(t)$ are stationary and independent across i, and the probability distribution of $U_i(t)$ has a finite support. As for $V_i(t)$, they are not necessarily stationary but are bounded across i. 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 stationary random process $U_i(t)$, is assumed to have *unknown* mean $u_i = \mathbb{E}[U_i(t)]$. 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 $\boldsymbol{u} = (u_1, \ldots, u_M)$.[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 $\boldsymbol{p}_t = (p_1^t, \ldots, p_i^t, \ldots, p_M^t)$ be the *probabilistic selection vector* 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 $i \in \mathcal{I}_t$ is p_i^t at time t (see Sec. III). Thus, the expected 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 an 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 $\boldsymbol{g}_t = (g_1^t, \ldots, g_i^t, \ldots, g_M^t)$. In addition, there is a preset *minimum guarantee threshold* $\rho > 0$ such that the average of the sum of the level-1 rewards needs to be above this threshold, i.e., $\mathbb{E}[\boldsymbol{u}^\intercal \boldsymbol{p}_t] \geq \rho$. At time t, the expected total compound reward is $\mathbb{E}[\sum_t \boldsymbol{g}_t^\intercal \boldsymbol{p}_t]$.

Our objective is to design a learning policy π to choose the selection vectors $\boldsymbol{p}_t^\pi$ for $t = 1, \ldots, T$ such that the *regret*, which is also referred to as loss compared with the Oracle, is as small as possible. Specifically, the Oracle is a policy that knows all the parameters: $\boldsymbol{u}$ and $\boldsymbol{v}_t$ at each time t. Thus it can select the optimal arms such that $\sum_t \boldsymbol{g}_t^\intercal \boldsymbol{p}_t$ is maximized and $\boldsymbol{u}^\intercal \boldsymbol{p}_t \geq \rho$ is satisfied at each time t. Regret for a policy π is defined as,

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

Note that computing the Oracle requires full knowledge of the parameters: $\boldsymbol{u}$ and $\boldsymbol{v}_t$ at each time t. However, these parameters can only be estimated when arms are selected. Hence, designing a policy π to maximize the total compound

[1] One can easily extend two level rewards to model n level ($n \geq 2$) rewards by associating each arm with n random processes.

[2] All vectors defined in this paper are column vectors.

reward (or equivalently, minimize the regret) without full knowledge is very challenging. To approach the Oracle, a policy π should learn the parameters by leveraging the observations from the selected arms at each time t. Note that $\boldsymbol{p}_t^\pi$ may initially violate the constraint especially when it has little information about the arms. To measure the overall violations of the constraint at time T, the *violation* of the policy π is defined as,

$$\mathrm{V}_\pi(T) = \mathbb{E}\left[\sum_{t=1}^{T}(\rho - \boldsymbol{u}^\intercal \boldsymbol{p}_t^\pi)\right]^+, \tag{2}$$

where $[\cdot]^+ = \max(\cdot, 0)$. Regret and violation are two important metrics to measure the performance of an arm selection policy π. In particular, for a policy π, a lower regret means that π gets closer to the Oracle and a smaller violation means that π becomes better in satisfying the constraint as time t increases.

III. Policy Design

In this section, we elaborate the design of our policy, "*Learning with Minimum Guarantee* (LMG)", for the stochastic constraint bandit model described in Sec. II. The main technical challenge is to balance between maximizing the total compound reward (or minimizing the regret in Eq. (1)) and, at the same time, satisfying the minimum guarantee threshold ρ (or maintaining low violation in Eq. (2)). To address this challenge, we incorporate the theory of Lagrange method in constrained optimization into our policy design. We consider minimizing a modified regret function that includes the violation with an *adjustable* penalty coefficient that increases the regret when there is any non-zero violation. Specifically, our policy introduces a sub-linear bound for the Lagrange function of $\mathrm{R}_\pi(T)$ and $\mathrm{V}_\pi(T)$ in the following structure,

$$\mathrm{R}_\pi(T) + \lambda(T)(\mathrm{V}_\pi(T))^2 \leq T^{1-\theta}, 0 < \theta \leq 1, \tag{3}$$

where $\lambda(T)$ plays the role of a Lagrange multiplier. From (3), we derive a bound for $\mathrm{R}_\pi(T)$ and a bound for $\mathrm{V}_\pi(T)$ as:

$$\mathrm{R}_\pi(T) \leq O(T^{1-\theta}), \mathrm{V}_\pi(T) \leq \sqrt{O(T^{1-\theta} + mT)/\lambda(T)}, \tag{4}$$

where the bound for $\mathrm{V}_\pi(T)$ in (4) is for the fact that $-\mathrm{R}_\pi(T) \leq O(mT)$ for any policy π. With properly adjusted $\lambda(T)$ and θ, both the regret and violation can be bounded by sub-linear functions of T.

Another technical challenge is that the search space for the optimal set of arms for our bandit problem 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. To avoid the combinatorial explosion, we judiciously keep M weights for the M arms instead of $\binom{M}{m}$ weights for each set of m arms. Essentially, LMG 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, DepRound, in [10].

Algorithm 1: Learning with Minimum Guarantee (LMG)

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: $\quad \mathcal{S}_t = \emptyset,\ \mathcal{I}_t = \emptyset.$
3: $\quad$ **if** $\max_{i\in\mathcal{M}} w_i^t \geq \beta \sum_{i=1}^M w_i^t$ **then**
4: $\quad\quad$ 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: $\quad\quad \mathcal{S}_t = \{i : w_i^t \geq \alpha_t\}$
6: $\quad$ **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: $\quad$ **for** $i = 1, \ldots, M$ **do**
$$\tilde{p}_i^t = m[(1-\gamma)\tilde{w}_i^t / \textstyle\sum_{i=1}^M \tilde{w}_i^t + \gamma/M]$$
8: $\quad \mathcal{I}_t = \text{DepRound}(m, \tilde{\boldsymbol{p}}_t)$
9: $\quad$ **for** $i \in \mathcal{I}_t$ **do** receive u_i^t and v_i^t
10: $\quad$ **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: $\quad$ **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: $\quad \lambda_{t+1} = [(1 - \eta\zeta)\lambda_t - \zeta(\frac{\hat{\boldsymbol{u}}_t^\intercal \tilde{\boldsymbol{p}}_t}{1-\gamma} - \rho)]^+$
13: **function** DepRound$(m, \boldsymbol{p})$
14: $\quad$ **while** exist $i \wedge p_i \in (0,1)$ **do**
15: $\quad\quad$ Find $i, j, i \neq j$, such that $p_{i,j} \in (0,1)$
16: $\quad\quad a = \min\{1 - p_i, p_j\},\ b = \min\{p_i, 1 - p_j\}$
17: $\quad\quad (p_i, p_j) = \begin{cases} (p_i + a, p_j - a) \text{ with prob. } \frac{b}{a+b}; \\ (p_i - b, p_j + b) \text{ with prob. } \frac{a}{a+b}. \end{cases}$
$\quad$ **return** $\mathcal{I} = \{i \,|\, p_i = 1, 1 \leq i \leq M\}$

The details of LMG policy are shown in Algorithm 1. In particular, Algorithm 1 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. At line 8, we deploy the dependent rounding function, DepRound, (see details from line 13 to line 17) to select m arms using the calculated $\tilde{\boldsymbol{p}}_t$. Specifically, DepRound probabilistically updates $\tilde{\boldsymbol{p}}_t$ until $\tilde{p}_i^t$ is either 0 or 1 and maintains the condition that $\mathbf{1}^\intercal \tilde{\boldsymbol{p}}_t = m$. At line 9, the LMG policy receives the rewards u_i^t and v_i^t, and then performs online learning on 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 10. 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 $\mathbb{E}[\hat{g}_i^t] = u_i^t v_i^t$. 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. Finally, the weight vector $\boldsymbol{w}_t$ and the Lagrange multiplier λ_t are updated (line 11 and line 12) using the estimations at the end of each iteration.

A. Regret and Violation Analysis

For our policy LMG shown in Algorithm 1, we have the following attractive property,

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

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

We refer the interested readers to Appendix VII for the details of the proof.

Remark: Note that a *good* arm selection policy π should have both *sub-linear* $\mathrm{R}_\pi(T)$ and *sub-linear* $\mathrm{V}_\pi(T)$ with respect to T. If these two metrics are linear, it means 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. Compared with the optimal policy, such a random policy would result in both linear regret and linear violation.

IV. Network Applications

In this section, we describe several network applications with different minimum guarantee requirements where LMG can be applied. For each application, we first describe the problem and map the problem to our stochastic constrained bandit model. Then we carry out simulations and compare LMG with existing state-of-the-art online learning policies to show the effectiveness of LMG in different application settings.

A. Opportunistic Multichannel Selection

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 [11]. 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* varying 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 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 minimum 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 I. (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 I satisfies the minimum guarantee exactly supposing that $\rho_1 = 1.5$.

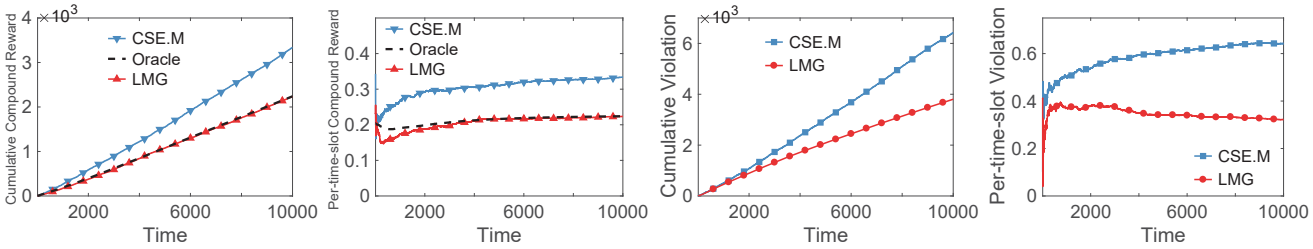
TABLE I: Primary-free Prob. and Throughput of 10 Channels

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

This problem fits the general framework of our stochastic constrained bandit model regarding the M channels as M arms, the channel access probabilities as u_i, and the throughput of different channels as the time-varying v_i^t, $1 \leq i \leq M$, and $t = 1, \ldots, T$. 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 average number of primary-free channels is above the minimum guarantee threshold ρ_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 I for $1 \leq i \leq 10$. The throughput v_i^t of channel i is time varying in an adversary fashion: the throughput starts from a random value drawn uniformly at random from $[0, v_i]$ where v_i takes from row 3 in Table I; then in each time slot, it decreases by $10/T$ until reaching 0 if the channel is primary-free, or increases by $10/T$ until reaching v_i if the channel has been taken up by primary users, $1 \leq i \leq 10$. 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

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



(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 LMG. $M = 10$, $m = 3$, $\rho_1 = 1.5$, and $T = 10,000$ in opportunistic multichannel selection.

up by primary users, then the channel's potential throughput would be large if the channel becomes primary-free in the next time slot.

We compare our LMG 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], [12], [13] that also considered online dynamic selection for secondary users. However, these works mainly focus on maximizing the throughput but neglect the importance of the minimum 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, LMG selects channels using Algorithm 1. For CSE.M, it always selects the top-3 channels with the highest UCB (upper confidence bound) indices $\hat{g}_i^t + \sqrt{3 \ln t/(2N_i(t))}$ where $N_i(t)$ is the number of times that channel i has been selected by the time slot t. As for the Oracle, 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] Therefore, the Oracle 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 $\boldsymbol{p}_t^*$ 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 LMG and CSE.M. Note that we do not compare the regret because the regret of the two policies are defined differently: the regret of LMG is defined as in Eq. (1) which compares the reward with the *constrained* optimal strategy while the regret of CSE.M compares the reward with *unconstrained* optimal strategy. In particular, the cumulative compound reward at t is calculated by $\sum_{t'=1}^{t} \sum_{i \in \mathcal{I}_{t'}} g_i^{t'}$. The cumulative compound reward for the Oracle is calculated by $\sum_{t'=1}^{t} (\boldsymbol{u} \circ \boldsymbol{v}_{t'})^\intercal \boldsymbol{p}_{t'}^*$. The per-time-slot compound reward at t is the ratio between the cumulative compound reward and t.

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

Besides, the long-term cumulative violation at t is calculated by $(\sum_{t'=1}^{t} (\rho_1 - \sum_{i \in \mathcal{I}_{t'}} u_i^{t'}))^+$ and the per-time-slot violation at t is the ratio between the cumulative violation and t.

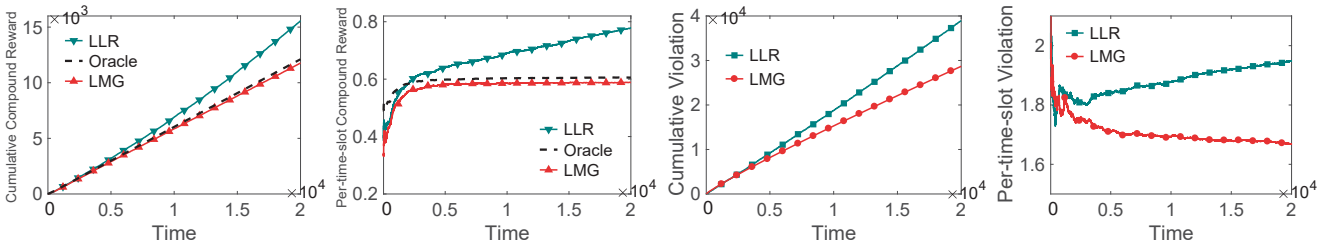
As shown in Fig. 1(a), the cumulative compound reward of LMG almost coincides with the cumulative compound reward of the Oracle at each time slot. To gain more insight, as shown in Fig. 1(b), the per-time-slot compound reward of LMG is decreasing and smaller than that of the Oracle in the first few time slots ($t \leq 237$). This is because before this time slot, LMG 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 minimum guarantee threshold ρ_1. After $t = 237$, the per-time-slot compound reward of LMG keeps increasing and gets closer to the Oracle. This means LMG 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 minimum guarantee threshold of the total primary-free probability in multichannel selection.

For the cumulative violation, as shown in Fig. 1(c), LMG has a lower cumulative violation than that of CSE.M in each time slot. Moreover, the cumulative violations of LMG and CSE.M become increasingly separated from each other. This means that LMG can select channels that are less likely to violate the minimum guarantee threshold. This is also verified in Fig. 1(d) where the per-time-slot violation of LMG 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 Fig. 1(c) and Fig. 1(d)) at each time slot.

Therefore, our simulation shows that LMG is effective in selecting channels for the opportunistic multichannel selection problem with the minimum guarantee threshold ρ_1.

B. Data-guaranteed Crowdsensing

1) **Problem Description:** In the paradigm of 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



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

Fig. 2: Compound rewards (the amount of qualified data) and violations of LLR, Oracle, and LMG. $M = 100$, $m = 10$, $\rho_2 = 6$, and $T = 20,000$ in data-guaranteed crowdsensing.

detect earthquakes in [14]. Due to the heterogeneity of the 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 sensing devices [15]. 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 crowdsensing tasks, task organizers have to choose multiple participants. For example, for dust level sensing for a large city, the task organizer needs to select a number of 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 has to consider both the 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 minimum guarantee threshold* ρ_2.

Now consider selecting m participants from M candidate participants for a crowdsensing task in T time slots. Our LMG 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 $t = 1, \ldots, T$. 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 amount of total collected data at each time slot should be above the minimum guarantee threshold ρ_2.

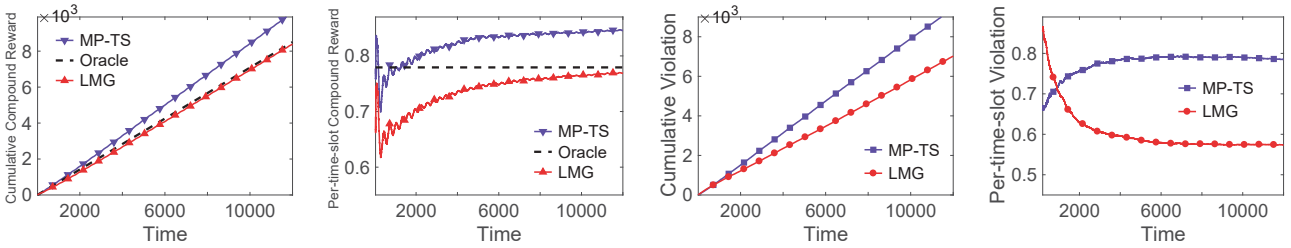
2) **Performance Evaluation:** Consider selecting $m = 10$ participants from $M = 100$ candidate participants with $\rho_2 = 6$, i.e., no less than 6 units of data is collected at each time slot, for $T = 20,000$ slots. The amount of data that participant i can gather is modeled as a Bernoulli random variable with mean u_i uniformly random generated in $[0.1, 0.8]$ for $1 \leq i \leq M$. For each participant i, the quality of data v_i^t is time-varying: the initial value of v_i^t is uniform random generated in $[0, 1]$; if participant i is chosen for the sensing task, v_i^t decreases by $50/T$; if v_i^t becomes 0, it restores to the initial value. 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 [16].

We compare LMG with one of the state-of-the art policies, *Learning with Linear Rewards* (LLR), proposed in [17] which is an effective multi-armed bandit policy for selecting multiple arms using UCB indices. Specifically, LLR selects the 10 highest UCB indices $\hat{g}_i^t + \sqrt{(m+1)\ln t / N_i(t)}$ where $N_i(t)$ is the number of times that participant i has been selected by the time slot t. The Oracle is calculated in the same way as in Sec. IV-A, and we use the methods described in Sec. IV-A 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]. LMG, on the contrary, is a non-stopping policy with no budget limitation.

The results are shown in Fig. 2. Both the cumulative (Fig. 2(a)) and the per-time-slot (Fig. 2(b)) compound rewards of LMG are getting closer the Oracle after $t = 3697$. This means LMG can learn the amount of data and the quality of data that each participant can collect accurately after this time to make judicious participant selections. The cumulative reward of LLR is larger than the Oracle as it only maximizes the amount of qualified data and ignores the minimum data-guarantee requirement. Therefore, LLR has larger cumulative/per-time-slot violations than LMG as shown in Fig. 2(a) and Fig. 2(b). Thus, this simulation shows the effectiveness of LMG in selecting participants for the data-guaranteed crowdsensing problem with the minimum guarantee threshold ρ_2.

C. Stability-guaranteed Crowdsourced Transcoding

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 [18]. 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



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

Fig. 3: Compound rewards (effective computing power) and violations of MP-TS, Oracle, and LMG. $M = 15$, $m = 4$, $\rho_3 = 2.5$, and $T = 12,000$ in stability-guaranteed crowdsourced transcoding.

viewers' (devices') computing power. 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. 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 LMG 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, $1 \leq i \leq M$ and $t = 1, \ldots, T$. 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 number of available viewers at each time slot should be at least ρ_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 power-law distribution with scale 0.45, shape 6.2, and normalized to $[0, 1]$; viewer i's computing power is modeled as a time-varying variable v_i^t: the initial value v_i is generated uniform randomly in $[0, 1]$, and $v_i^t = v_i[1 + \sin(2\pi t/24)]/2$, $1 \leq i \leq M$. This sinusoidal-varying pattern [19] can capture the periodic (hourly or daily) change of a viewer's computing power.

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) [20], with LMG. In our simulation, MP-TS selects the top-4 arms ranked by the posterior samples sampled from the Beta distributions, Beta(A_i, B_i), $1 \leq i \leq 15$. Particularly, A_i=B_i=1 at t=1. At time t>1, by performing a Bernoulli trial with success probability $\hat{g}_i^t$ 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 Sec. IV-A.

Fig. 3 shows the results of our simulation. From Fig. 3(a) and Fig. 3(b), we see both the cumulative reward and the per-time-slot reward of LMG are approaching the Oracle gradually. This indicates that LMG is as superior in maximizing effective computing power as the Oracle does while learning the availability probabilities and computing power of viewers. 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 minimum guarantee threshold as shown in Fig. 3(c) and Fig. 3(d). On the contrary, LMG has much lower cumulative/per-time-slot violations and its per-time-slot violation keeps decreasing. Thus, this simulation shows the effectiveness of LMG in selecting viewers for the stability-guaranteed crowdsourced transcoding problem with the minimum guarantee threshold ρ_3.

V. 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 bandit models with multiple plays, where multiple arms are selected at each time slot. [21] presents the Exp3.M algorithm to select multiple arms using exponential weights. [22] proposes the CUCB algorithm that selects multiple arms with the highest upper confidence bound (UCB) indices. [20] presents the multi-play Thompson Sampling algorithm (MP-TS) for arms with binary rewards. Our model differs from these bandit models as we further consider the stochastic constraint or the minimum guarantee requirement in selecting the multiple

arms. Note that the constraint in our model is very different from that in the bandit with budgets [23], [24] and the bandit with knapsacks [25]. 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. This makes our problem more challenging as we need to consider the violations of the minimum guarantee requirement introduced by this non-stopping arm selection procedure. Finally, our constrained bandit model is related to but different from the bandit model considered in [26] which tries to balance regret and violation. They only consider selecting a single arm without any time-varying multi-level rewards. While in our work, we consider how to select multiple arms and each arm is associated with time-varying multi-level rewards, making our model more flexible and better fit different network optimization problems.

Due to the effectiveness of the online learning framework in finding the optimal decision and learning the unknown application parameters, a lot of works have incorporated it into network application optimizations. In [7], [8], [12], the authors consider selecting the single-best channel to maximize the throughput in opportunistic multichannel access networks, and [13] designs an online dynamic channel access policy. [17] proposes a combinatorial network optimization framework based on UCB for graphs with unknown weights. [9] adopts an online greedy strategy to employ participants for budget limited crowdsensing. These works have been shown to be effective in different network applications. However, they neglect the minimum guarantee requirements in real-world network applications, where our framework can be applied and achieves better performance in terms of maximizing the cumulative compound reward while meeting the constraint.

VI. Conclusion

In this paper, we first point out that existing works neglect the minimum guarantee requirements in real-world network application optimizations. To our best knowledge, we are the first to propose an online learning framework using stochastic constrained bandit model with time-varying multi-level rewards for this kind of optimizations, and we are the first to design a constrained bandit policy, LMG, by taking the minimum guarantee requirement ρ into consideration with provable *sub-linear* regret and violation bounds. We also show how to apply LMG to three network applications: opportunistic multichannel selection, data-guaranteed crowdsensing, and stability-guaranteed crowdsourced transcoding. To gain more insight into the policy, we demonstrate the effectiveness of LMG by comparing it with existing state-of-the-art policies and show that LMG achieves better network application optimizations with larger cumulative rewards and smaller violations.

Acknowledgment

The work of John C.S. Lui is supported in part by GRF 14208816 and Huawei's Research Grant.

VII. Appendix

Proof of Theorem 1: From line 12 of the algorithm, we have: $\lambda_{t+1}=\left[(1-\eta\zeta)\lambda_t-\zeta(\frac{\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t}{1-\gamma}-\rho)\right]^+\leq\left[(1-\eta\zeta)\lambda_t+\zeta\rho)\right]^+$, where $\tilde{\boldsymbol{p}}_t$ is the probabilistic selection vector of the policy $\tilde{\pi}$ at time t. By induction on λ_t, we can obtain $\lambda_t\leq\frac{\rho}{\eta}$. Let $\Phi_t=\sum_{i=1}^M w_i^t$ and $\tilde{\Phi}_t=\sum_{i=1}^M\tilde{w}_i^t$. Define $\boldsymbol{r}_t=\boldsymbol{g}_t+\lambda_t\boldsymbol{u}_t$ and $\hat{\boldsymbol{r}}_t=\hat{\boldsymbol{g}}_t+\lambda_t\hat{\boldsymbol{u}}_t$. Let $\boldsymbol{p}_t$ be an arbitrary probabilistic selection vector which satisfies $p_i^t\in[0,1]$, $\mathbf{1}^\intercal\boldsymbol{p}_t=m$ and $\boldsymbol{u}^\intercal\boldsymbol{p}_t\geq\rho$. For the sequence of selected $\mathcal{I}_t$ at $t=1,\ldots,T$,

$$\begin{aligned}
&\sum_{t=1}^T\ln\frac{\Phi_{t+1}}{\Phi_t}=\ln\frac{\Phi_{T+1}}{\Phi_1}=\ln(\sum_{i=1}^M w_i^{T+1})-\ln M\\
&\geq\ln(\sum_{i=1}^M p_i^t w_i^{T+1})-\ln M\geq\sum_{i=1}^M\frac{p_i^t}{m}\sum_{t:i\notin\mathcal{S}_t}\zeta\hat{r}_i^t-\ln\frac{M}{m}\\
&=\frac{\zeta}{m}\sum\nolimits_{i=1}^M p_i^t\sum_{t:i\notin\mathcal{S}_t}\hat{r}_i^t-\ln\frac{M}{m}. \qquad (5)
\end{aligned}$$

As $\zeta=\frac{\gamma\eta m}{(\eta+m)M}$ and $\lambda_t\leq\frac{\rho}{\eta}$, we have $\zeta\hat{r}_i^t\leq 1$. Therefore,

$$\begin{aligned}
&\frac{\Phi_{t+1}}{\Phi_t}=\sum_{i\in\mathcal{M}/\mathcal{S}_t}\frac{w_i^{t+1}}{\Phi_t}+\sum_{i\in\mathcal{S}_t}\frac{w_i^{t+1}}{\Phi_t}=\sum_{i\in\mathcal{M}/\mathcal{S}_t}\frac{w_i^t}{\Phi_t}\exp(\zeta\hat{r}_i^t)+\sum_{i\in\mathcal{S}_t}\frac{w_i^t}{\Phi_t}\\
&\leq\sum_{i\in\mathcal{M}/\mathcal{S}_t}\frac{w_i^t}{\Phi_t}[1+\zeta\hat{r}_i^t+(e-2)\zeta^2(\hat{r}_i^t)^2]+\sum_{i\in\mathcal{S}_t}\frac{w_i^t}{\Phi_t} \qquad (6)\\
&=1+\frac{\tilde{\Phi}_t}{\Phi_t}\sum_{i\in\mathcal{M}/\mathcal{S}_t}\frac{w_i^t}{\tilde{\Phi}_t}\left[\zeta\hat{r}_i^t+(e-2)\zeta^2(\hat{r}_i^t)^2\right]\\
&\leq 1+\frac{\zeta}{m(1-\gamma)}\sum_{i\in\mathcal{M}/\mathcal{S}_t}\tilde{p}_i^t\hat{r}_i^t+\frac{(e-2)\zeta^2}{m(1-\gamma)}\sum_{i\in\mathcal{M}/\mathcal{S}_t}\tilde{p}_i^t(\hat{r}_i^t)^2\\
&\leq 1+\frac{\zeta}{m(1-\gamma)}\sum_{i\in\mathcal{M}/\mathcal{S}_t}\tilde{p}_i^t\hat{r}_i^t+\frac{(e-2)\zeta^2}{m(1-\gamma)}\sum_{i=1}^M(1+\lambda_t)\hat{r}_i^t. \qquad (7)
\end{aligned}$$

Inequality (6) holds because $e^y\leq 1+y+(e-2)y^2$ for $y\leq 1$, and inequality (7) uses the fact that $\tilde{p}_i^t\hat{r}_i^t=r_i^t\leq 1+\lambda_t$ for $i\in I_t$ and $\tilde{p}_i^t\hat{r}_i^t=0$ for $i\notin I_t$. Since $\ln(1+y)\leq y$ for $y\geq 0$, we can get

$$\ln\frac{\Phi_{t+1}}{\Phi_t}\leq\frac{\zeta}{m(1-\gamma)}\sum_{i\in\mathcal{M}/\mathcal{S}_t}\tilde{p}_i^t\hat{r}_i^t+\frac{(e-2)\zeta^2}{m(1-\gamma)}\sum_{i=1}^M(1+\lambda_t)\hat{r}_i^t.$$

Then using (5), it follows that

$$\begin{aligned}
\frac{\zeta}{m}\sum_{i=1}^M p_i^t\sum_{t:i\notin\mathcal{S}_t}\hat{r}_i^t-\ln\frac{M}{m}\leq\frac{\zeta}{m(1-\gamma)}\sum_{t=1}^T\sum_{i\in\mathcal{M}/\mathcal{S}_t}\tilde{p}_i^t\hat{r}_i^t\\
+\frac{(e-2)\zeta^2}{m(1-\gamma)}\sum_{t=1}^T\sum_{i=1}^M(1+\lambda_t)\hat{r}_i^t.
\end{aligned}$$

As $\tilde{p}_i^t=1$ for $i\in\mathcal{S}_t$, and $\sum_{i=1}^M p_i^t\sum_{t:i\in\mathcal{S}_t}\hat{r}_i^t\leq\frac{1}{1-\gamma}\sum_{t=1}^T\sum_{i\in\mathcal{S}_t}\hat{r}_i^t$ trivially holds, we have

$$\sum_{t=1}^T\hat{\boldsymbol{r}}_t^\intercal\boldsymbol{p}_t-\frac{m}{\zeta}\ln\frac{M}{m}\leq\frac{\sum_{t=1}^T\hat{\boldsymbol{r}}_t^\intercal\tilde{\boldsymbol{p}}_t}{1-\gamma}+\frac{(e-2)\zeta}{1-\gamma}\sum_{t=1}^T\sum_{i=1}^M(1+\lambda_t)\hat{r}_i^t.$$

Taking expectation on both sides, we have

$$\mathbb{E}\left[\sum\nolimits_{t=1}^T\hat{\boldsymbol{r}}_t^\intercal\boldsymbol{p}_t-\frac{1}{1-\gamma}\sum\nolimits_{t=1}^T\hat{\boldsymbol{r}}_t^\intercal\tilde{\boldsymbol{p}}_t\right]$$

$$\leq \frac{m}{\zeta}\ln\frac{M}{m}+\frac{(e-2)\zeta}{1-\gamma}\sum_{t=1}^{T}\mathbb{E}\Big[\sum_{i=1}^{M}(1+\lambda_t)\hat{r}_i^t\Big]$$
$$\leq \frac{m}{\zeta}\ln\frac{M}{m}+\frac{2(e-2)\zeta M}{1-\gamma}T+\frac{2(e-2)\zeta M}{1-\gamma}\sum_{t=1}^{T}\lambda_t^2, \quad (8)$$

where (8) is from the inequality $\mathbb{E}[\sum_{i=1}^{M}(1+\lambda_t)\hat{r}_i^t]=\sum_{i=1}^{M}(1+\lambda_t)(g_i^t+\lambda_t u_i^t)\leq 2M+2M\lambda_t^2$. Next, we define a series of functions $f_t(\lambda)=\frac{\eta}{2}\lambda^2+\lambda(\frac{1}{1-\gamma}\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t-\rho), t=1,\ldots,T$, and we have $\lambda_{t+1}=[\lambda_t-\zeta\nabla f_t(\lambda_t)]_+$. It is clear that $f_t(\cdot)$ is a convex function for all t. Thus, for an arbitrary λ, we have

$$(\lambda_{t+1}-\lambda)^2=([\lambda_t-\zeta\nabla f_t(\lambda_t)]_+-\lambda)^2$$
$$\leq(\lambda_t-\lambda)^2+2\zeta^2\rho^2+2\zeta^2\frac{(\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t)^2}{(1-\gamma)^2}+2\zeta[f_t(\lambda)-f_t(\lambda_t)].$$

Let $\Delta=[(\lambda_t-\lambda)^2-(\lambda_{t+1}-\lambda)^2]/(2\zeta)+\zeta m^2$. We have,

$$f_t(\lambda_t)-f_t(\lambda)\leq\Delta+\frac{\zeta(\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t)^2}{(1-\gamma)^2}=\Delta+\frac{\zeta m^2(\frac{1}{m}\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t)^2}{(1-\gamma)^2}$$
$$\leq\Delta+\frac{\zeta m^2}{(1-\gamma)^2 m}\sum_{i=1}^{M}(\tilde{p}_i^t\hat{u}_i^t)^2\leq\Delta+\frac{\zeta m}{(1-\gamma)^2}\sum_{i=1}^{M}u_i^t.$$

Taking expectation over $\sum_{t=1}^{T}[f_t(\lambda_t)-f_t(\lambda)]$, we have

$$\mathbb{E}\Big[\frac{\eta}{2}\sum_{t=1}^{T}\lambda_t^2-\frac{\eta}{2}\lambda^2T+\sum_{t=1}^{T}\lambda_t(\frac{\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t}{1-\gamma}-\rho)$$
$$-\lambda\sum_{t=1}^{T}(\frac{\hat{\boldsymbol{u}}_t^\intercal\tilde{\boldsymbol{p}}_t}{1-\gamma}-\rho)\Big]\leq\frac{\lambda^2}{2\zeta}+\zeta m^2T+\frac{\zeta mM}{(1-\gamma)^2}T. \quad (9)$$

Combining (8) and (9), we have,

$$\sum_{t=1}^{T}\boldsymbol{g}_t^\intercal\boldsymbol{p}_t-\frac{\mathbb{E}[\sum_{t=1}^{T}\boldsymbol{g}_t^\intercal\tilde{\boldsymbol{p}}_t]}{1-\gamma}+\mathbb{E}\Big[-(\frac{\eta T}{2}+\frac{1}{2\zeta})\lambda^2$$
$$+\lambda\sum_{t=1}^{T}(\rho-\frac{\boldsymbol{u}^\intercal\tilde{\boldsymbol{p}}_t}{1-\gamma})\Big]\leq\frac{m}{\zeta}\ln\frac{M}{m}+\frac{2(e-2)\zeta MT}{1-\gamma}$$
$$+\zeta m^2T+\frac{\zeta mMT}{(1-\gamma)^2}+(\frac{2(e-2)\zeta M}{1-\gamma}-\frac{\eta}{2})\sum_{t=1}^{T}\lambda_t^2$$
$$+\mathbb{E}[\sum_{t=1}^{T}\lambda_t(\rho-\boldsymbol{u}^\intercal\boldsymbol{p}_t)].$$

Since $\zeta=\frac{\gamma\eta m}{(\eta+m)M}$ and $\eta\geq\frac{4(e-2)\gamma m}{1-\gamma}-m$, we have $\frac{2(e-2)\zeta M}{1-\gamma}\leq\frac{\eta}{2}$. As $\boldsymbol{u}^\intercal\boldsymbol{p}_t\geq\rho$, we have

$$(1-\gamma)\sum_{t=1}^{T}\boldsymbol{g}_t^\intercal\boldsymbol{p}_t-\mathbb{E}\Big[\sum_{t=1}^{T}\boldsymbol{g}_t^\intercal\tilde{\boldsymbol{p}}_t\Big]$$
$$+\mathbb{E}\Big[\lambda\sum_{t=1}^{T}((1-\gamma)\rho-\boldsymbol{u}^\intercal\tilde{\boldsymbol{p}}_t)-(\frac{\eta T}{2}+\frac{1}{2\zeta})\lambda^2\Big]$$
$$\leq\frac{m}{\zeta}\ln\frac{M}{m}+2(e-2)\zeta MT+\zeta m^2T+\frac{\zeta mMT}{1-\gamma}.$$

Let $\lambda=\frac{\sum_{t=1}^{T}((1-\gamma)\rho-\boldsymbol{u}^\intercal\tilde{\boldsymbol{p}}_t)}{\eta T+1/\zeta}$. Maximize over $\boldsymbol{p}_t$ and we have,

$$\max_{\boldsymbol{u}^\intercal\boldsymbol{p}_t\geq\rho}\sum_{t=1}^{T}\boldsymbol{g}_t^\intercal\boldsymbol{p}_t-\mathbb{E}\Big[\sum_{t=1}^{T}\boldsymbol{g}_t^\intercal\tilde{\boldsymbol{p}}_t\Big]$$
$$+\mathbb{E}\Big\{\frac{\big[\sum_{t=1}^{T}((1-\gamma)\rho-\boldsymbol{u}^\intercal\tilde{\boldsymbol{p}}_t)\big]^{+2}}{2(\eta T+1/\zeta)}\Big\}\leq F(T),$$

where $F(T)=\frac{m}{\zeta}\ln\frac{M}{m}+2(e-2)\zeta MT+\zeta m^2T+\frac{\zeta mMT}{1-\gamma}+\gamma mT$. Then we have results in the form of Eq. (4): $\mathrm{R}_{\tilde{\pi}}(T)\leq F(T)$, and $\mathrm{V}_{\tilde{\pi}}(T)\leq\sqrt{2\,(F(T)+mT)\,(\eta T+1/\zeta)}$. As $\gamma=\min(1,\sqrt{\frac{2(e-2)M+Mm}{m\ln(M/m)T^{2/3}}})=\Theta(T^{-\frac{1}{3}})$ and $\eta=\frac{4(e-2)\gamma m}{1-\gamma}=\Theta(T^{-\frac{1}{3}})$, we have $\zeta=\Theta(\frac{1}{M}T^{-\frac{2}{3}})$. Finally, we have $\mathrm{R}_{\tilde{\pi}}(T)\leq O(mM\ln(M)T^{\frac{2}{3}})$ and $\mathrm{V}_{\tilde{\pi}}(T)\leq O(m^{\frac{1}{2}}M^{\frac{1}{2}}T^{\frac{5}{6}})$. This finishes the proof. ∎

REFERENCES

[1] Y. C. Liang, K. C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: an overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386–3407, 2011.

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Comm. Magazine*, vol. 49, no. 11, 2011.

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

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

[5] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, 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, 2002.

[7] W. Dai, Y. Gai, and B. Krishnamachari, "Online learning for multi-channel opportunistic access over unknown markovian channels," in *Proc. of IEEE SECON*, 2014.

[8] P. Yang, B. Li, J. Wang, X. Y. Li, Z. Du, Y. Yan, and Y. Xiong, "Online sequential channel accessing control: A double exploration vs. exploitation problem," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4654–4666, 2015.

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

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

[11] 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 Processing Magazine*, vol. 29, no. 3, pp. 101–116, 2012.

[12] C. Tekin and M. Liu, "Approximately optimal adaptive learning in opportunistic spectrum access," in *Proceedings IEEE INFOCOM*, 2012.

[13] Y. Liu and M. Liu, "An online approach to dynamic channel access and transmission scheduling," in *Proc. of MobiHoc'15*, 2015.

[14] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, "The next big one: Detecting earthquakes and other rare events from community-based sensors," in *Proc. of IPSN*, 2011.

[15] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proc. of ACM UbiComp*, 2012.

[16] O. Yurur and C. H. Liu, *Generic and Energy-Efficient Context-Aware Mobile Sensing*. CRC Press, Inc., 2015.

[17] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1466–1478, 2012.

[18] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 870–883, 2013.

[19] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *Proc. of NIPS*, 2014.

[20] J. Komiyama, J. Hondaand, and H. Nakagawa, "Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays," in *ICML*, 2015.

[21] T. Uchiya, A. Nakamura, and M. Kudo, "Algorithms for adversarial bandit problems with multiple plays," in *Proc. of ACL'10*, 2010.

[22] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework, results and applications," in *Proc. of ICML*, 2013.

[23] R. Combes, C. Jiang, and R. Srikant, "Bandits with budgets: Regret lower bounds and optimal algorithms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 1, pp. 245–257, 2015.

[24] Y. Xia, T. Qin, W. Ma, N. Yu, and T.-Y. Liu, "Budgeted multi-armed bandits with multiple plays," in *Proc. of IJCAI*, 2016.

[25] 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. of COLT*, 2016.

[26] M. Mahdavi, T. Yang, and R. Jin, "Online decision making under stochastic constraints," in *NIPS workshop on Discrete Optimization in Machine Learning*, 2012.