Exploiting Side Information for Improved Online Learning Algorithms in Wireless Networks

Manjesh K. Hanawal, Senior Member, IEEE, and Sumit J. Darak, Senior Member, IEEE

Abstract—In wireless networks, the transmitter adapts its parameters based on the receiver’s feedback to achieve a high throughput. The throughput also depends on factors like interference level and channel gain, which can be measured at the transmitter. They provide useful information about the instantaneous throughput as well. For example, higher interference implies a lower throughput. This work treats any measurable quality with a non-zero correlation with the throughput as side information (SI). We also study how it can be exploited to quickly learn the channel that offers higher throughput (reward). When the mean value of the SI is known, using control variate theory, we develop online learning algorithms that require fewer samples to learn and can improve the learning rate compared to cases where SI is ignored. Specifically, we incorporated SI in the Upper Confidence Bound (UCB) algorithm and proposed the UCBwSI algorithm. We quantify the gain achieved in terms of the regret and show that the improvement in regret over state-of-the-art UCB is proportional to the correlation between the reward and SI. Simulations demonstrate a 5-10% improvement in the bit-error rate. Even when the mean of the SI is unknown, we demonstrate the superiority of the UCBwSI over UCB.

Index Terms—Multi-armed bandit, control variate, wireless communication, distributed learning, upper confidence bound.

I. INTRODUCTION

The evolution of software-defined radio and cognitive radio allows the wireless transmitter and receiver to choose the physical layer specifications such as bandwidth, carrier frequency, modulation scheme, and waveform on-the-fly. Such reconfigurable transceivers need intelligence to learn and adapt to unknown radio frequency (RF) environment. The multi-armed Bandit (MAB) based online learning framework is widely used for learning in various wireless settings, like millimeter wave communication [1], cognitive radio networks [2], massive multi-antenna system [3] and ad-hoc networks [4]. In the MAB framework, the goal is to learn the channels (i.e., carrier frequency), number of antennas (i.e., beam pattern), beam directions, or other physical layer parameters that give the highest throughput. The MAB-based online learning algorithms do not need prior knowledge of the RF environment and use the signal-to-noise ratio (SNR) or signal-to-interference-plus-noise ratio (SINR) at the receiver as feedback to identify the best parameters. The online nature of these algorithms means they do not need training datasets and can be deployed in any dynamic environment. Hence, they are preferred over dataset-dependant, compute and memory-intensive machine learning and deep learning-based approaches [5].

The emergence of new applications like AR and VR demands wireless networks to support high throughputs [6]. This necessitates speeding up the learning rate of the MAB algorithms so that the best channel is identified quickly and used for data transfer. It is well known that the learning rate can be improved if feedback is of high quality or additional feedback is available. In this work, we focus on exploiting any additional information that can be measured or readily available in wireless networks to improve the performance of the MAB algorithms.

The throughput in a wireless network is a function of the SINR given as $\frac{\eta I}{\eta + \eta}$, where $P$ is the transmission power in watts, $I$ is the instantaneous channel gain, $I_0$ is the total interference power in watts, and $\eta$ is the total noise power in watts at the receiver. Interference, channel gain, and noise can be random, and their realization may not be known before transmission. However, in many wireless systems, measuring instantaneous values of some of these quantities may be possible, which can be used as side information (SI) (see Sec. III). This SI has a non-zero correlation with the received throughput that can be exploited to improve the learning rate and performance of MAB algorithms to identify the best quality (rate/throughput/success) channels.

Wireless network protocols provide provision to measure interference and channel gain [7][Chap. 3]. A transmitter can sense the channel to measure the level of interference before the information is transmitted in a slot. If measured interference at the transmitter in a slot is higher, the receiver may also experience higher interference. The receiver may experience inferences from other sources that the transmitter may not see; still, the throughput at the receiver can be correlated with the interference level seen at the transmitter. Also, in the networks where uplink and downlink are used for data transmission
using time division duplexing [7][Chap. 4], the transmitter can measure the channel gain using pilots in the uplink. These channel gains will strongly correlate with the throughput received on the downlink and can be used as SI.

We show that when the mean values of the SI are known, the SI can be used to get a better estimate of the quality of channels with smaller variance and sharper confidence bounds that could be used to develop faster learning algorithms. To exploit the available SI, we combine the reward sample with the ‘centered’ side-information samples with an ‘optimal’ weight and use the new samples to estimate the channel rates. The optimal weight depends on the correlation coefficient between the throughput and the SI and may not be known but can be estimated (see Sec. IV).

The assumption that the mean value of the SI is known may appear strong and overly restrictive, but it is possible to get this information in wireless networks; the mean of SI could be available from the measurements during the network planning and deployment phase or could be obtained by offline measurements. Further, experiments demonstrate that the mean value of SI need not be known precisely. A good approximation is enough to achieve the gain from SI.

Using the SI-based new estimator, we build on the well-known Upper Confidence Bound (UCB) based MAB algorithm. We refer to the proposed algorithm as UCB with SI (UCBwSI). The SI samples used for the new estimator are correlated. This brings out significant challenges in analyzing the regret performance of the UCBwSI algorithm, as standard concentration bounds cannot be used. For the case where reward and SI are jointly Gaussian, we provide performance guarantees for the UCBwSI algorithm leveraging the results from control variate theory. The UCBwSI algorithm offers better performance than vanilla UCB and its extensions that do not exploit SI. The gain in the improvement depends on the correlation between the reward and SI for each channel. For the general distributions, we use one of the popular re-sampling techniques named Splitting to leverage the SI and improve the performance of the UCB algorithm. Our contributions can be summarized as follows:

- We develop a new framework that exploits the SI in wireless networks to improve the performance of online learning algorithms.
- For the case where the reward and SI are jointly Gaussian, we develop the UCBwSI algorithm. We show that its regret is better by a factor at least \((1 - \rho^2)\) in Theorem 1, where \(\rho = \min_i \rho_i\) is the minimum correlation coefficient between throughput and SI across all the channels.
- For general distributions, we give an algorithm named UBC with SI using Splitting (UCBwSI-Split). This algorithm is based on the popular re-sampling method named Splitting [8].
- We take cellular networks (LTE PHY), L-band digital aeronautical communication systems (LDACS), and cognitive radio networks as case studies to demonstrate various SI available in wireless networks. We simulate these networks over a wide range of realistic parameters and validate the performance improvement achieved by the proposed algorithms over state-of-the-art UCB and UCBV algorithms that do not use SI. On average, the proposed algorithms offer 5-10% improvement in bit successful transmission rate (BSTR) or bit error rate (BER) over UCB and its extensions.
- We demonstrate the significance of SI on the performance of proposed algorithms via various experiments. We also evaluate the performance of the proposed algorithms when the mean of the SI is not known exactly. For the illustrative LTE PHY case, we demonstrate that even when mean values are known approximately, performance improvement can still be achieved compared to the case when SI is not used.

This work is an extension of our initial work in [9]. In this work, we have included a theoretical analysis of the proposed algorithm along with the UCBwSI-Split algorithm. In experiments, we consider cognitive radio and LDACS setup in addition to LTE in [9]. Furthermore, we consider the extension of UCBwSI when the mean of the SI is not known.

The paper is organized as follows: In Section II, we review the recent works followed by the proposed framework in Section III. The proposed UCBwSI algorithm for the Gaussian reward distribution is presented in Section IV, and its adaptation to general rewards, named UCBwSI-Split, is presented in Section V. We evaluate the performance of our algorithms on Cognitive radio networks in Section VI, on LTE PHY in Section VII, and on LDACS in Section VIII. Section IX concludes the paper with a discussion on future directions.

II. RELATED WORK

Multi-armed Bandit (MAB) framework based online learning algorithms have been used extensively to develop learning algorithms for decision-making due to their analytical tractability, low complexity, and does not need prior training like machine learning and deep learning algorithms [10], [11], [12]. MAB based decision making is widely used in various wireless setups like cognitive radio networks [13], [14], [15], Energy harvesting networks [16], Millimeter Wave communications [17], massive multi-antenna system [18], and 5G cellular networks [19]. We focus on the stochastic setting where classical bandit algorithms like UCB [20], KL-UCB [21], and Thompson sampling [22] are applied to networks by suitably mapping actions (channels, modulations schemes, power level) to arms and quantity of interests (SNR, SINR, success rate, throughput) to rewards to obtain learning algorithms. As our focus is on better adapting the MAB setting to wireless networks, we discuss the works that incorporate properties of the wireless networks or any specific network information to improve the performance of existing online learning algorithms. For details on various MAB setups in wireless networks, we refer to [10] and [11]. Additionally, various reinforcement learning (RL) and deep RL algorithms have been used for wireless applications [23], [24]. Since we focus on improving the performance of state-of-the-art MAB algorithms, the discussion on RL algorithms is skipped to maintain the conciseness of the discussion.

Dummy probe packets can be used in networks to get additional information about the success probability of a transmission rate/arm. Such probes result in throughput loss but provide additional information about throughputs from
different rates/arms. Reference [25] demonstrate that exploitation of such additional information offers better rate selection strategies in 802.11-like wireless systems. When the interference graph of a network is known, the authors in [26] exploit the graph structure to develop efficient MAB algorithms to maximize the network throughput. A transmitter can have contextual information like received signal strength, the cell area, user location, and the average number of devices in the channel. Authors in [27] exploit such contextual information to improve link selection in cellular networks. Reference [28] assumes the availability of Side Observation (SO). The side observations obtained by playing an arm provide information about the arms that are not played in that round. In our setting, we do not get any information about the arms that are not played in a given round. We only get to observe the SI of the selected arm, and such information is correlated only with the reward obtained on the selected arm. Note that to make use of SO, the correlation matrix needs to be known beforehand. However, such information may not be available. SI is more readily available in wireless networks as the receiver can measure multiple factors like SNR, Success Rate, and throughput, which are correlated and can be used as SI. Reference [29] considers the setting where the learner can pay to observe-before-play a subset of arms and decide which arm to play. In our setting, we cannot observe a reward before play. SI and reward are revealed only after the arm is played.

Our work differs from the previous works as we do not require any structure in the reward, seek side-observations, or consider payment for observing rewards before arms are played. We note that in contextual bandits, context is available before an arm is selected. In our case, SI is observed only when the arm is played. Moreover, contextual bandits assume a parametric relation between the reward and the contextual (e.g., linear), whereas we considered a non-parametric setting. Hence, we cannot apply the contextual bandit machinery in our setup. We utilize readily available SI to improve the performance of MAB algorithms. Specifically, we use SI to obtain unbiased point estimates with smaller variance and confidence intervals using control variate theory [8], [30]. The new estimators are used in the UCB algorithm to improve performance. To the best of our knowledge, leveraging control variate theory to improve performance online algorithms for wireless networks has not been explored in the literature yet.

### III. Problem Setup

Consider a heterogeneous network consisting of macrocells and picocells as shown in Fig. 1. The user equipment (UE) in a picocell communicates with its picocell base station through uplink and downlink channels and with a macrocell base station through a supplementary downlink channel. A UE gets interference (both in-band and out-of-band) from neighboring picocell base stations, macrocell base stations, and other sources like Wi-Fi networks. Consider a communication between picocell base station 1 and UE. Conventionally, the base station uses SNR or CQI feedback shared by UE over uplink for channel selection in the subsequent time slot.

![Fig. 1. An heterogeneous network consisting of macro and picocells. UEs see interference from other base stations and networks. Some of these could be measurable and can be used SI.](image)

We assume that picocell base station 1 can measure interference level at UE due to macrocell base station 1 and picocell base station 2 (shared through fiber back-haul or sensing), and it is referred to as measurable interference. However, other inferences experienced at the UE, such as interference from WiFi, picocell base station 3, and other adjacent picocell base stations (not shown in Figure), are not measurable or unknown. In this work, we discuss the efficient use of measurable interference as SI to quickly identify optimal channel for communication. Other network setups like cognitive radio and LDACS are discussed in detail in Sections VI and VIII.

In the following, we abstract the network settings and set it up as a learning problem where the aim of the learner (i.e., transmitter) is to identify the channel offering the highest throughput for the given receiver. If the learner has to learn the best modulation and coding scheme (MCS), the same setup can be used with the set of MCS as actions.

Let the set of channels be denoted by $[K] = \{1, 2, \ldots, K\}$. The throughput on a channel $i \in [K]$ depends on various stochastic quantities denoted as $\{W_i, W_{-i}\}$ where $W_i$ denotes the quantity that the learner can observe and $W_{-i}$ denote the set of all other that are not observed. We refer to $W_i$ as SI for channel $i$. The throughput on channel $i$ is denoted as $X_i$ and is an unknown function of $\{W_i, W_{-i}\}$, denoted as $X_i = f_i(W_i, W_{-i})$. For example, consider that the transmitter can observe the interference on the channel $i$, and other factors like channel gains ($h_i$), receiver noise ($\eta_i$), and interference at the receiver ($I_i$) are not observed. Then $X_i$ is the interference at the transmitter and other quantities constitute $W_{-i} = (h_i, \eta_i, I_i)$ and the throughput on channel $i$ as a function of SINR can be given by

$$X_i = f_i(W_i, W_{-i}) = f\left(\frac{|h_i|^2P}{|h_i|^2W_i + I_i + \eta_i}\right),$$

(1)

where $P$ is the power transmitted. If channel gain is observed at the transmitter, but total interference at the receiver ($I_i$) and a receiver noise ($\eta_i$) are not observed, then $W_{-i} = (I_i, \eta_i)$ and throughput can be expressed as

$$X_i = f_i(W_i, W_{-i}) = f\left(\frac{W_iP}{I_i + \eta_i}\right).$$

(2)

We assume the availability of one SI for each arm.\(^1\) We refer to the channels as arms and the throughput as a reward. In round

\(^1\)When multiple SI are available for the arms, we can extend the analysis straightforwardly. See Remark 1 later.
where \( \beta^* \) is a constant. Let \( \hat{\mu}_{c,i} \) denote the point-estimator using \( s \) new samples given as

\[
\hat{\mu}_{c,i} = \frac{1}{s} \sum_{r=1}^{s} \hat{X}_{r,i} = \hat{\mu}_{s,i} + \beta^*_i (\omega_i - \hat{\omega}_{s,i}).
\] (5)

Notice that \( \hat{\mu}_{c,i} \) is an unbiased estimator. Further, when \( \beta^*_i \) is set appropriately, the variance of \( \hat{\mu}_{c,i} \) is smaller than that of \( \hat{\mu}_{s,i} \) as given by the following lemma.

**Lemma 1** [31]: Let \( \text{Cov}(X_i, W_i) \) denotes the covariance between reward and SI pair \((X_i, W_i)\) and \( \text{Var}(W_i) \) denotes the variance of SI from the i-th arm. Set \( \beta^*_i = \text{Cov}(X_i, W_i) / \text{Var}(W_i) \). Then,

\[
\text{Var}(\hat{\mu}_{c,i}) = (1 - \rho^2) \text{Var}(\hat{\mu}_{s,i}) = (1 - \rho^2) \sigma^2 / s.
\]

To realize the improvement in the variance of \( \hat{\mu}_{c,i} \), we need to know the covariance between reward and SI and the variance of the SI, both of which may not be known a priori. But they can be estimated, and we replace \( \beta^*_i \) by its estimate given as follows

\[
\hat{\beta}^*_i = \frac{1}{s} \sum_{r=1}^{s} (X_{r,i} - \hat{\mu}_{s,i})(W_{r,i} - \omega_i)^2 / \hat{\sigma}^2,
\] (6)

We added subscript \( s \) to the estimate of \( \beta^*_i \) to make dependency on the number of samples explicit. Replacing \( \beta^*_i \) by its estimate in (5), the new point-estimate for mean reward of arm \( i \) is given as

\[
\hat{\mu}_{c,i} = \hat{\mu}_{s,i} + \hat{\beta}^*_i (\omega_i - \hat{\omega}_{s,i}).
\] (7)

The new estimator \( \hat{\mu}_{c,i} \) is a mean of non i.i.d. samples as \( \hat{\beta}^*_i \) depends on the past \( s \) samples and we cannot apply Lemma 1 to estimate variance of \( \hat{\mu}_{c,i} \). It is generally hard to calculate it for an arbitrary distribution due to the dependency introduced by \( \hat{\beta}^*_i \). However, for the special case where reward and SI are jointly Gaussian, one can estimate the variance of \( \hat{\mu}_{c,i} \) as shown in the following Proposition.

**Proposition 1**: Let the reward and SI pair for each arm be jointly Gaussian. For \( s \) reward and SI sample pairs of arm \( i \), define

\[
S^2 = \frac{1}{s} \sum_{r=1}^{s} (X_{r,i} - \hat{\mu}_{s,i})^2 / s - 2.
\]

\[
\hat{\sigma}_{s,i} = S^2 \left( \frac{1}{s} - \frac{(\sum_{r=1}^{s} (W_{r,i} - \omega_i)^2) / \hat{\sigma}^2}{\sum_{r=1}^{s} (X_{r,i} - \hat{\mu}_{s,i})^2 / \hat{\sigma}^2} \right)^{-1}
\]

is an unbiased variance estimator of \( \hat{\mu}_{c,i} \), i.e., \( \text{E}[\hat{\sigma}_{s,i}] = \text{Var}(\hat{\mu}_{c,i}) \).

For Gaussian distributed reward with variance \( \sigma^2 \), Eq. (4) can be rewritten as \( X_{t,i} = \mu_i + \beta^*_i (\omega_i - W_{t,i}) + \epsilon_{t,i} \), where \( \epsilon_{t,i} \) is a zero Gaussian random variable with variance \( (1 - \rho^2) \sigma^2 \). The proof exploits the fundamental properties of linear regression. Detailed proof of the Proposition is given in [8][Thm. 1]. With an unbiased variance estimator for \( \text{Var}(\hat{\mu}_{c,i}) \), we can construct the tight confidence intervals on the reward mean estimator as given by our next result.

**Proposition 2**: For each arm, let reward and SI be jointly Gaussian. Then for all \( i \in [K] \) we have

\[
P \left\{ |\hat{\mu}_{c,i} - \mu_i| \geq V_{t,i}^{(c)} \sqrt{\hat{\sigma}_{s,i}} \right\} \leq 2 / t^c,
\]

where \( V_{t,i}^{(c)} \) denote 100(1 - 1/t^c)^{th} percentile value of the \( t \)-distribution with \( s - 2 \) degrees of freedom and \( \hat{\sigma}_{s,i} \) is an unbiased estimator for variance of \( \hat{\mu}_{c,i} \).

The proof follows along similar lines in [8][Thm. 1] after setting appropriate percentile values for the \( t \)-distribution.
The above concentration bound is analogous to Hoeffding inequality, which shows that the deviation of the estimate of mean reward obtained by samples \( \{X_{t,i}\} \) around the true mean decays very fast. Note that the deviation factor \( V_{t,s}^{(\alpha)}/\sqrt{\nu_{s,i}} \) depends on the variance of the estimator \( \hat{\mu}_{s,i} \) which guarantees sharper confidence intervals. As we will see, these confidence terms are smaller by a factor \((1-\rho_i^2)\) compared to the case of no SI use. We note that this improvement in confidence bound is achieved without requiring additional samples. Hence, algorithms using the new confidence bounds are expected to learn the optimal arm faster.

A. Algorithm: UCBwSI

Let \( N_t(t) \) denote the number of times \( i \)-th arm is selected till round \( t \) and \( \nu_{N_t(t),i} \) be the variance of mean reward estimator \( \hat{\mu}_{N_t(t),i} \) for arm \( i \). Following Proposition 2, we define an optimistic upper confidence bound for the estimate of the mean reward of arm \( i \) as follows:

\[
UCB_{t,i} = \hat{\mu}_{N_t(t),i} + V_{t,N_t(t),i}^{(\alpha)} \sqrt{\nu_{N_t(t),i}}. \tag{8}
\]

Treating the above value as indices of arms, we develop an algorithm named UCB with SI (UCBwSI). Its pseudo-code is given below. The algorithm works as follows: It takes \( K \) and \( \alpha > 1 \) as inputs. \( \alpha \) determines the trade-off between exploration and exploitation. At the start, each arm is played \( 3 \) times to ensure that the sample variance \( \nu_{s,i} \) is well defined (see Prop. 1). In round \( t > 3K \), UCBwSI computes the UCB index of each arm as in Eq. (8) and selects the arm with the highest UCB index. The arm selected is denoted as \( I_t \). After playing arm \( I_t \), the reward \( X_{t,I_t} \) and SI \( W_{t,I_t} \) are observed. Then, the value of \( N_t(t) \) is updated and \( \hat{\mu}_{N_t(t),i} \), \( \nu_{N_t(t),i} \), and \( \nu_{t,N_t(t)} \) are re-estimated. The same process is repeated till the end. We note that Algorithm UCBwSI uses the estimator’s variance to define the UCB index, whereas other variance estimation-based algorithms like UCBV [32] user variance of the reward. In UCB, each arm is explored once in the beginning compared to three times in UCBwSI. However, the increase in the exploration is compensated by improved estimation and faster identification of the optimal arm, resulting in lower regret, as shown using simulation results in Sections VI, VII, and VIII.

B. Computational Complexity of UCBwSI

We next discuss the computation complexity of finding UCB indices in Eq. (8) in each round. To obtain the first term \( \hat{\mu}_{s,i} \), UCBwSI uses mean estimates obtained by reweighing the samples (Eq. 7) with the correlation factor (Eq. 6). This factor can be expressed as

\[
\hat{\mu}_{s,i} = \sum_{r=1}^{s} X_{r,i}(W_{r,i} - \omega_i) - \mu_{s,i} \sum_{r=1}^{s} (W_{r,i} - \omega_i)
\]

By maintaining the running sums of \( \sum_{r=1}^{s} X_{r,i} \) and \( \sum_{r=1}^{s} (W_{r,i} - \omega_i)^2 \), \( \hat{\mu}_{s,i} \) can be recomputed when a new sample is observed with the new mean value \( \mu_{s,i} \). Thus \( \hat{\mu}_{s,i} \) can be computed efficiently in an iterative fashion.

The confidence term in Eq. (8) can also be efficiently computed. Note that \( S^2 \) in \( \nu_{s,i} \) and \( \nu_{t,i} \) can be iteratively computed by additionally maintaining the running sums of \( \sum_{r=1}^{s} X_{r,i}^2 \) and \( \sum_{r=1}^{s} (W_{r,i} - \hat{\omega}_s)^2 \). The values of \( V_{t,s}^{(\alpha)} \) can be obtained by a look-up table. Hence, compared to the UCB, UCBwSI requires more parameters and storage to perform the computation, but its computation complexity is of the same order as that of the UCB.

C. Analysis of UCBwSI

Our regret bound is based on the following classical result from the control variate theory.

Proposition 3 [8][Thm. 1]: Let reward and SI are jointly Gaussian for all the arms. The estimate in (7) obtained with \( s > 3 \) reward and side-observations samples satisfies the following properties:

\[
\mathbb{E}[\hat{\mu}_{s,i}] = \mu_i \quad \text{and} \quad \text{Var}(\hat{\mu}_{s,i}) = \frac{s^2 - 2}{s - 3} (1 - \rho_i^2) \text{Var}(\hat{\mu}_{s,i}).
\]

Following is our main result, which is the upper bounds regret of UCBwSI. Its proof is given in Appendix C.

Theorem 1: Let the UCBwSI is run with \( \alpha = 2 \). Then the regret of UCBwSI in \( T \) rounds is upper bounded by

\[
\mathbb{E}[\hat{\mu}_{s,i}] = \mu_i \quad \text{and} \quad \text{Var}(\hat{\mu}_{s,i}) = \frac{s^2 - 2}{s - 3} (1 - \rho_i^2) \text{Var}(\hat{\mu}_{s,i}).
\]

Unfortunately, we cannot directly compare our bound with that of UCB1, UCB1-NORMAL [20], and UCBV [32] as \( V_{T,T}^{(2)} \) do not have closed-form expression. \( V_{T,T}^{(2)} \) denotes 100(1-\( T^2 \))th percentile of student’s \( t \)-distribution with \( T \) degrees of freedom. The value of \( V_{T,T}^{(2)} \) can be numerically shown to be upper bounded by 3.726 \( \log(T) \). \( C_{T,i} \) also do not have close form expression, but \( C_{T,i} \) \( \to 1 \) as \( T \to \infty \). When each arm is explored sufficiently (\( N_{T,i} \sim 40 \), one can numerically verify that \( C_{T,i} \leq 1.5 \). Hence the regret is of the order \( \mathcal{O}( \sum_{i \neq \star} (1 - \rho_i^2) \log(T) \). This shows that the proposed algorithm reduces regret by a factor of at least \((1 - \rho_i^2)\), where \( \rho_i^2 = \min \rho_i^2 \) compared to the algorithm that does not use side information. The factor of improvement \((1 - \rho_i^2)\) is significant when the correlation between the SI and the reward is high.

Remark 1: The above results can be slightly generalized to the case where rewards are Gaussian but need not be jointly

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Gaussian with SI as shown in [8][Thm. 2]. Also, the analysis can be extended when the arms have more than one SI. More details are in Appendix B.

Remark 2: If the mean SI is unknown, the algorithm can use an approximate mean or estimate the mean from samples. A brief discussion is given in Appendix C.

V. GENERAL DISTRIBUTION

When reward and SI have arbitrary distribution, \( \hat{\beta}^*_j \) is no longer guaranteed to be a Gaussian and unbiased estimator. Therefore, we cannot obtain confidence intervals using properties of \( t \)-distributions as done in Proposition 2. However, we can use re-sampling methods such as Jackknifing and Splitting to reduce the bias of the estimators and develop confidence intervals that hold approximately or asymptotically [8]. Below, we present an algorithm based on the Splitting method.

A. Algorithm With Splitting Approach

Splitting is a re-sampling technique that splits the correlated observations into two or more groups, computes an estimate of \( \beta^*_j \) from each group, and then exchanges the estimates among the groups. We consider the form of splitting discussed in [8] [Sec. VI] where \( s \) groups are formed. The \( j \)-th group, \( j \in [s] \), is obtained by dropping the \( j \)-th reward and side-observation pair. From each group \( j \in [s] \), \( \hat{\beta}^*_j \) is estimated and denoted as \( \hat{\beta}^{(s-1),i}_j \). The new samples are then calculated as

\[
\hat{X}^S_{j,i} = X_{j,i} + \hat{\beta}^{(s-1),i}_j (\omega - W_{j,i}), \quad \forall j \in [s],
\]

The point estimator for splitting method is \( \hat{\mu}^{c,S}_{s,i} = \frac{1}{s} \sum_{j=1}^{s} \hat{X}^S_{j,i} \) and its sample variance is \( \hat{\nu}^{c,S}_{s,i} = (s(s-1))^{-1} \sum_{j=1}^{s} (\hat{X}^S_{j,i} - \hat{\mu}^{c,S}_{s,i})^2 \). Then \( \hat{\mu}^{c,S}_{s,i} \pm t_{\alpha/2}(n-1)\hat{\nu}^{c,S}_{s,i} \) gives an approximate confidence interval [8]. Using this, we define the optimistic upper confidence bound for mean reward as

\[
\text{UCB}^b_{s,i} = \hat{\mu}^{c,S}_{s_i(t),i} + \sqrt{\frac{V^{G,\alpha}_{s_i(t)}}{t_i} \hat{\nu}^{c,S}_{s_i(t),i}},
\]

where \( V^{G,\alpha}_{s_i,g} \) is the 100(1 − 1/t^{\alpha})th percentile value of the \( t \)-distribution with \( (s-1) \) degrees of freedom. We can use the above UCB indices to get an algorithm for the general distribution case. We refer to the algorithm as UCB with SI and Splitting (UCBwSI-Split). Its pseudo-code is given in Algorithm UCBwSI-Split.

**Algorithm UCBwSI-Split**

1: **Input:** \( K, \alpha > 1 \)
2: Play each arm \( i \in [K] \) 3 times
3: for \( t = 3K + 1, 3K + 2, \ldots \) do
4:   \( \forall i \in [K] \): compute \( \text{UCB}^b_{s-1,i} \) as given in Eq. (9)
5:   Play \( I_t = \arg \max_{i \in [K]} \text{UCB}^b_{s-1,i} \)
6:   Observe \( X_{I_t,t} \) and \( W_{I_t,t} \). Increment the value of \( N_{I_t}(t) \)
7:   Compute \( \hat{\beta}^{(s-1),j}_i (N_{I_t}(t) - 1), I_t \) and \( \bar{X}^S_{j,I_t} \) for all \( j \in [N_{I_t}] \)
8:   Compute \( \hat{\mu}^{c,S}_{s_i(t),i} \) and \( \hat{\nu}^{c,S}_{s_i(t),i} \)
9: end for

**TABLE I**

| Parameters | Value |
|------------|-------|
| No. of Subcarriers (NSC) | \{128, 1024, 4096\} |
| No. of Data Subcarriers | \{116, 1012, 4088\} |
| Sub-carrier Spacing (SCS) | \{30 KHz\} |
| OFDM Symbol Duration (\( \mu s \)) | 35.68 |
| OFDM Symbols/Time Slot | 14 |
| Modulation Scheme | QPSK, 64-QAM |
| (Bipolar) ADC Bits | \{6, 16\} |
| Wireless Channel | AWGN with 5 KHz offset |
| Wireless Channel Gain | As per the distribution |
| In-band Interference | Narrowband spread over few unknown subcarriers |
| Out-of-band Interference | Wideband with non/partially overlapping frequency range |
| Interference Power Level | As per the distribution |

The optimistic upper bound defined in Eq. (9) holds asymptotically [8], [33], but they good approximated for finite number of samples. Due to this, we cannot use them to provide any finite time guarantees as done in Prop. 2 and Thm. 1 for the Gaussian case. We experimentally validate its performance in the next sections for various wireless network setups.

VI. APPLICATION TO COGNITIVE RADIO NETWORKS

We consider the cognitive radio (CR) network deployed in the unlicensed spectrum. A CR user (learner) aims to identify the wireless channel that offers a lower bit-error rate or higher Bit Successful Transmission Rate (BSTR) for given physical layer parameters such as coding rate, modulation scheme, and bandwidth. The power level of measurable interference at the transmitter is used as SI on each channel. The performance of UCBwSI and UCBwSI-Split are compared against UCB1-Normal [20] (referred to simply as UCB in the following) and UCBV [32] both of which use variance estimate. For all the algorithms, we compare measured BSTR performance instead of regret as expressions for computation of optimal BSTR (as in Eqs. 1 & 2) are not available for our realistic setup. All the experiments in this section assume \( K = 8 \) wireless channels and orthogonal frequency division multiplexing (OFDM) based physical layer for CR users. Various parameters of the physical layers are summarized in Table I. The parameters such as the number of subcarriers (NSC) of 4096, subcarrier spacing (SCS) of 30 KHz, and 64-QAM modulation scheme are similar to the 5G cellular physical layer.

Two types of interference, In-band and Out-of-band, are considered. For in-band interference, the transmission bandwidth of the interferer overlaps with that of CR users. We consider high-power narrowband interference, which affects only a few subcarriers of the CR users. For out-of-band interference, interferers and CR users transmit over non-overlapping but adjacent frequency bands. In both cases, the interferers’ transmit power can significantly impact the BSTR of the CR users. The wireless channel is modeled as

\[2\] We did not compare UCBwSI and UCBwSI-Split against KL-UCB and Thompson sampling as KL-UCB does not incorporate variance estimation and it is unclear how to incorporate side-information in Thompson sampling.
a basic additive white Gaussian noise (AWGN) channel. This excludes performance degradation due to multi-path, Doppler effects, and non-ideal channel estimation methods. This means the achieved BSTR will mainly depend on the channel SNR, interference, and impairments in the analog-front-end (AFE). Realistic wireless channel effects are considered later in Sections VII and VIII.

We model frequency division duplexing (FDD) system based CR communication in which the SNRs of downlink channels are modeled as a stochastic parameter and unknown to the transmitter. The mean SNR distribution of wireless channels is set as \( \mu^1 = \{5, -1, 3, -9, 7, -2, 18, -7\} \) with two types of variances: 1) Low variance, \( \sigma^1_L = \{0.5, 0.5, 1, 0.8, 0.1, 0.3, 0.2, 0.4\} \), and 2) High variance, \( \sigma^1_H = \{2, 2, 2, 2, 2, 2, 2, 2\} \). Among various measurable, non-measurable, and unknown interferences, we consider the measurable interference as the SI. The mean of the measurable interference is known, and its instantaneous value can be measured. The mean received power distribution of the SI at the CR transmitter as \( w_1 = \{1, 7, 0.2, -3, -0.9, -0.4, 1, -0.6, 1\} \) with two types of variances: 1) Low variance, \( \sigma^1_L = \{0.2, 0.4, 0.3, 0.2, 0.3, 0.1, 0.4, 0.7\} \), and 2) High variance, \( \sigma^1_H = \{2, 2, 2, 2, 2, 2, 2, 2\} \). Impairments due to AFE are fixed but unknown. The MAB algorithm learns and identifies the channel with higher SNR and lower interference to maximize the BSTR. For all the experiments discussed in this section, results are averaged over 15 experiments, and each experiment consists of the transmission of 5000 time slots. Each time slot is 0.5 ms, comprising a data frame of 14 OFDM symbols.

A. Effect of Variance on BSTR

In Fig. 2, we compare the performance of the four candidate algorithms in terms of BSTR at different time instants of the horizon. We consider four combinations:

- **LL:** Low SNR and low SI variance, \( \{\mu^1, \sigma^1_L, w_1, \sigma^1_L\} \)
- **HL:** High SNR and low SI variance, \( \{\mu^1, \sigma^1_H, w_1, \sigma^1_L\} \)
- **LH:** Low SNR and high SI variance, \( \{\mu^1, \sigma^1_L, w_1, \sigma^1_H\} \)
- **HH:** High SNR and high SI variance, \( \{\mu^1, \sigma^1_H, w_1, \sigma^1_H\} \)

In all cases, UCBwSI and UCBwSI-Split algorithms outperform the UCB and UCBV algorithms. Furthermore, Fig. 2 (a) and (b) show that the low SNR variance leads to improved BSTR for low and high SI variances, respectively, due to faster learning of the channel statistics. Furthermore, for a given SNR variance, low SI variance offers higher BSTR, highlighting the usefulness of SI. On average, the BSTR of UCBwSI is higher by 0.8 and 0.5 than UCB and UCBV, respectively. This corresponds to 0.5 and 0.3 megabits per second (MBPS) improvement in throughput in each time slot over UCB and UCBV, respectively.

B. Effect of Modulation Scheme and NCS on BSTR

We consider two modulation schemes: 1) Quadrature phase shift keying (QPSK), which maps two bits in one data symbol, and 2) 64- Quadrature amplitude modulation (64-QAM), which maps six bits in one data symbol. In addition, we consider three NCS: \{128, 1024, 4096\}. In Fig. 3, it can be observed that the achieved BSTR is higher when the channel SNR variance is low. For a given NCS, the achieved BSTR using the QPSK is higher than the BSTR using the 64-QAM. This is expected since QPSK offers a higher tolerance to channel distortions compared to 64-QAM. Similarly, with the increase in NCS from 128 to 4096, there is a slight degradation in the BSTR due to the fewer guard and pilot subcarriers. The performance can be improved using a higher number of guard and pilot subcarriers.

C. Effect of Modulation Scheme and NCS on Throughput

We consider the achieved throughput, which considers the BSTR along with the transmission time. For instance, higher BSTR in QPSK may not correspond to higher throughput. This is because, in a symbol time duration, QPSK transmits only two bits compared to six bits in 64-QAM. The throughput results for different combinations of modulation scheme and NCS are given in Table II. Note that the throughput is calculated by incorporating the re-transmissions needed to transmit all the message bits successfully. As expected, throughput in the case of channels with low SNR variance is higher than those with high SNR variance. Furthermore, the throughput of 64-QAM is higher than QPSK, and throughput increases significantly with the increase in the NCS. In all cases, UCBwSI offers significantly higher throughput than the UCB and UCBV algorithms. In an ideal noise-free distortionless channel, the expected throughput using the 64-QAM is 3 times the throughput using QPSK. However, due to
realistic channel conditions and interference effects considered in this simulation, the improvement factor is 1.2 to 2.5. When compared to UCBwSI, UCBwSI-Split does not make any assumptions on the distribution of rewards and SI. In the experimental setup, the distribution of the average SNR and SI is arbitrary. However, due to the large number of samples considered in experiments, the resulting distribution is nearly Gaussian due to the Central Limit Theorem, which leads to 2% or smaller error between BSTR and throughput of UCBwSI and UCBwSI-Split. We believe such a small error is an acceptable statistical variation.

VII. APPLICATION TO LTE PHY AND EFFECT OF SI: ERRONEOUS, NON-MEASURABLE AND INSIGNIFICANT

In this section, we extend OFDM based transceivers in Section VI for LTE and consider the scenarios where SI may be erroneous, non-measurable, and insignificant.

As shown in Fig. 4, the base station selects the channel and configures the downlink PHY using the decision-making block comprising a learning algorithm and scheduler. The data is transmitted over the selected channel. The UE receives the data using receiver PHY and estimates signal-to-interference and noise ratio (SINR). It then calculates the channel quality indicator (CQI), precoder matrix indicator (PMI), and rank indicator (RI), which is then sent to the base station over the uplink. The base station receives the data sent by the UE and also measures the SI. The learning algorithm uses SINR as a reward and SI to select the channel for transmitting the data frame in the subsequent time slot.

The transmitter and receiver PHY of the base station and UE are designed as per the LTE standard as given in [34]. Various parameters of the physical layers are summarized in Table III. The data packets are first processed by a channel encoding block, which performs cyclic-redundancy check (CRC) addition, code block segmentation, and channel encoding using a Turbo encoder. Then, rate matching is done to meet available bandwidth constraints, followed by scrambling and data modulation. The physical downlink control channels (PDCCH) and physical downlink shared channel (PDSCH) are processed separately. The next step is resource mapping according to the LTE resource grid, which consists of a ten millisecond (ms) frame with ten sub-frames, and each sub-frame consists of 14 OFDM symbols. Along with PDCCH and PDSCH, other signals, such as synchronization

---

**TABLE II**

| No. of Subcarriers | 128 | 1024 | 4096 |
|--------------------|-----|------|------|
| Modulation Scheme  | QPSK| 64-QAM | QPSK| 64-QAM | QPSK| 64-QAM |
| SNR Variance       | H   | L     | H   | L     | H   | L     | H   | L     | H   | L     |
| UCB                | 2.45| 2.93 | 5.90| 7.35 | 4.66| 20.7 | 25  | 45.1 | 53  | 35.93| 77.5| 93  |
| UCBwSI            | 2.68| 3.20 | 6.82| 8.66 | 5.54| 23.5 | 27.9| 50.8 | 62  | 41   | 86.4| 103.8|
| UCBwSI-Split      | 2.91| 3.42 | 7.30| 9.10 | 5.68| 25.1 | 29.7| 54.8 | 66  | 43.9 | 92.9| 103.9|

---
and reference signals, are mapped to the appropriate subcarriers. This is followed by OFDM modulation. In MIMO OFDM, additional precoding and layer mapping are done based on the chosen MIMO mode. The transmitter output is passed through various channel models such as flat-fading low mobility channel (FLM), extended pedestrian A (EPA), and extended Typical Urban (ETU) channels [34]. The receiver performs operations similar to the transmitter in reverse order. We consider non-ideal synchronization and channel estimation. For all the experiments discussed in this section, results are averaged over 15 experiments, and each experiment consists of the transmission of 1500 time slots. Each time slot is 1 ms, comprising a LTE sub-frame of 14 OFDM symbols.

In Fig. 5, we validate the functional correctness of three algorithms in terms of average BSTR and throughput. Since the performance of UCBwSI and UCBwSI-Split is nearly identical, we have skipped UCBwSI-Split to avoid cluttering in plots. We begin with FLM channels and three cases considering various combinations of channel SNR, interference power, and location.

- **LSNR_4**: We consider 4 channel with SNR in dB as {-2, 4, 1, 6}, and corresponding interference levels in dB as {1, 0, 5, 5}. The first and last channels have in-band interference, while the second and third have out-of-band interference. Thus, the second channel is the preferred choice.

- **HSNR_4**: Same as LSNR_4 except the channel SNRs are bit higher and equal to {5, 10, 15, 20}. Thus, the third channel is the preferred choice.

- **Rand_8**: We consider eight channels with channel and interference SNRs randomly chosen at the beginning of each experiment.

In all three cases, UCBwSI offers higher BSTR and throughput. As expected, the throughput of LSNR_4 is lower than HSNR_4 due to the low SNR of the preferred channel.

**A. Effect of Erroneous SI on BSTR**

Next, we consider three wireless channels. Both EPA and ETU are multi-path medium-to-high mobility channels. Among them, ETU degrades the signal more severely due to low SNR, high Doppler frequency, and a wide range of multipath gains and delays. In Fig. 6 and Fig. 7, we compare the average BSTR and throughput performances of the UCB and UCBwSI algorithm for HSNR_4 and Rand_8, respectively. In all three channels, UCBwSI outperforms the UCB algorithm.

In Fig. 6 and Fig. 7, we also consider the UCBwSIE algorithm, which has erroneous information about the SI mean. In the UCBwSIE algorithm, we have incorporated an additional learning stage to learn the mean of the SI as accurately as possible. It can be observed that there is a slight degradation in the performance of the UCBwSIE compared to UCBwSI since it takes more time to identify the optimal arm due to erroneous SI. However, in all cases, UCBwSIE identified the optimal arm as indicated by constant average BSTR after initial exploration.

**B. Effect of Insignificant SI on BSTR**

Next, we analyze the performance of the UCB and UCBwSI algorithms when interference is insignificant to impact the BSTR and throughput. This is similar to the conventional MAB setup, and the aim is to select the channel with the highest SNR. Surprisingly, UCBwSI performs slightly better than the UCB algorithm, as shown in Fig. 8. Similar results are also observed in other cases. This can be attributed to the better exploration approach in UCBwSI, which uses tighter confidence intervals to define the UCB indices.

**C. Effect of Non-measurable SI on BSTR**

The achieved BSTR of the UCBwSIE algorithm may not be significantly better than that of the UCB and UCBv algorithms...
Fig. 6. Average BER and Throughput in MBPS for HSNR_4 with (a) FLM, (b) EPA, and (c) ETU channels for different combinations of channel SNR, interference power, and location.

Fig. 7. Average BER and Throughput in MBPS for Rand_8 with (a) FLM, (b) EPA, and (c) ETU channels for different combinations of channel SNR, interference power, and location.

Fig. 8. Average BER and Throughput in MBPS without interference for FLM channels for HSNR_4.

Fig. 9. Effect of non-measurable interference on the achieved BSTR.

if the unknown or non-measurable SI is stronger than the measurable SI. To understand this behavior, we consider out-of-band interference. Specifically, we consider two interfering transmissions on the channels adjacent to the channel selected by the LTE user. One of the interference power levels is measurable at the transmitter, while the other is not measurable or even unknown. We consider four cases:

- **WNMI**: Non-measurable interference weaker than measurable interference.
- **SNMI_A**: Non-measurable interference is slightly stronger than measurable interference, but its carrier frequency is away from the LTE user.
- **SNMI_C**: Non-measurable interference is slightly stronger than measurable interference, and its carrier frequency is close to the LTE user with no overlap.
- **SNMI_O**: Non-measurable interference is slightly stronger than measurable interference, and its carrier frequency is close to the LTE user with overlap over a few sub-carriers.

As shown in Fig. 9, achieved BSTR degrades with an increased impact of the non-measurable interference. Furthermore, the gap between the BSTR of the proposed algorithms and UCB and UCBV algorithms increases as the non-measurable interference becomes weaker than the measurable interference.

VIII. APPLICATION TO LDACS

We consider $L-$band (960-1164 MHz) digital aeronautical communication system (LDACS), which is proposed as an alternative to the existing very high frequency (VHF) band (118-137 MHz) based air-to-ground communication system [35], [36]. It aims to utilize 1 MHz vacant frequency bands between incumbent Distance Measuring Equipment (DME) signals in the $L$-band. Though the average transmit power levels of the DME signals are specified in the standards, the instantaneous power levels of the DME may vary dynamically due to moving aircraft [37]. In practical deployment, dynamically changing DME power levels significantly impact the BSTR, and hence, the LDACS users need intelligence to
identify the 1 MHz band with minimum DME interference and high SNR to maximize the BSTR and throughput. In this section, we demonstrate the application of UCBwSI at the LDACS ground terminal to quickly identify the optimal frequency band for communication with the aircraft. Here, the DME power level is used as SI. We realized the end-to-end physical layer as per the LDACS standard \cite{35}, \cite{38} and physical layer parameters are summarized in Table IV.

For all the experiments discussed in this section, results are averaged over 15 experiments, and each experiment consists of the transmission of 5000 time slots. Each time slot is 6.48 ms, comprising a data frame of 54 OFDM symbols.

### A. Effect of Wireless Channels on BSTR

In the LDACS environment, three types of wireless channels are experienced by users: 1) Airport (APT): During aircraft taxiing, 2) Terminal Maneuvering Area (TMA): During landing and take-off, and 3) En-routing (ENR): During the flying phase. They are modeled as wide sense stationary uncorrelated scattering channels and characterized as shown in Table V.

In Fig. 10, the performance of UCB, UCBV, and UCBwSI for three wireless channels is compared. For all three channels, UCBwSI outperforms UCB and UCBV. Also, the achieved BSTR is highest for the ENR channel, followed by TMA and APT. This is due to a strong line-of-sight (LoS), weak LoS, and no LoS paths in the ENR, TMA, and APT channels, respectively.

### B. Effect of Interference Mitigation on BSTR

Various simulation results have shown slight degradation in the BSTR when the DME signal power level in the adjacent channel is high. The pulse banking approach has been incorporated recently in the LDACS standard to mitigate interference. In Fig. 11, we analyzed the effect of DME interference mitigation in LDACS on achieved BSTR.
interference mitigation techniques in existing LDACS on the achieved BSTR. Fig. 11 (a) and (b) consider the LDACS ENR channel with high and low SNR variance, respectively. As expected, the achieved BSTR is higher when the variance is low. Furthermore, UCBwSI offers significantly higher BSTR than UCB and UCBV. The improvement after enabling the interference mitigation at the receiver is significant for UCB and UCBV compared to UCBwSI, as UCBwSI already exploits the DME SI to select channels. Thus, our approach can eliminate additional interference mitigation at the receiver, thereby reducing its complexity and latency.

IX. CONCLUSION AND FUTURE DIRECTIONS

We developed improved online learning algorithms that exploited side-information (SI) in wireless networks. We developed two UCB-based algorithms named UCBwSI and UCBwSI-Split that incorporate SI and achieve better regret performance. We showed that the higher the correlation, the smaller the regret. The analysis used control variate theory to derive tight concentration bounds, which in turn allowed us to obtain tight estimates of mean rewards. The improvement came without requiring any additional samples, thus exploiting the learning rate.

We showed that SI is available in multiple wireless networks like cognitive radios, LTE, and L-band digital aeronautical communication systems. Further, we demonstrated how this SI information can be exploited to improve network throughput. Specifically, our experiments demonstrated that the correlation between SI and reward is often strong in networks, and exploiting it improves the bit-error rate by 5-10%. We have also demonstrated the efficacy of the proposed algorithms when the mean of the SI is not known accurately.

Another potential wireless application of the proposed algorithm is an integrated sensing and communication (ISAC) system in the mmWave spectrum where the radar signal processing enables the communication transceiver to identify the optimum beam by quickly localizing the dynamically moving mobile targets [39]. Here, the range, azimuth, and Doppler velocity from radar signal processing can be used as SI and SNR as a reward. The non-wireless application of the proposed algorithm includes task allocation to servers in data centers where task size can be used as SI.

The extension of the proposed work includes efficient realization of the proposed algorithms on hardware and analysis of area, latency, and power consumption with respect to conventional MAB algorithms. The proposed approach of exploiting SI to improve the performance of MAB algorithms can be extended to other works such as [26], [27], [28], [40], and [29]. Corresponding regret analysis and performance verification via simulations is interesting future direction.

APPENDIX A

CONTROL VARIATES

Let $\mu$ be the unknown quantity that needs to be estimated, and $X$ be an unbiased estimator of $\mu$, i.e., $\mathbb{E}[X] = \mu$. A random variable $W$ is a CV if its expectation $\omega$ is known and correlated with $X$. Linear control methods use errors in estimates of known random variables to reduce errors in estimating an unknown random variable. For any choice of a coefficient $\beta$, define a new estimator as $\hat{X} = X + \beta(\omega - W)$. It is straightforward to verify that its variance is given by

$$\text{Var}(\hat{X}) = \text{Var}(X) + \beta^2 \text{Var}(W) - 2\beta \text{Cov}(X, W).$$

and is minimized by setting $\beta$ to $\beta^\star = \text{Cov}(X, W)/\text{Var}(W)$. The minimum value of the variance is given by $\text{Var}(\hat{X}) = (1 - \rho^2)\text{Var}(X)$, where $\rho$ is the correlation coefficient of $X$ and $W$.

The larger the correlation, the greater the variance reduction the CV achieves. In practice, the values of $\text{Cov}(X, W)$ and $\text{Var}(W)$ are unknown and need to be estimated to compute the best approximation for $\beta^\star$.

APPENDIX B

INCORPORATING MULTIPLE SIDE-INFORMATION

In some applications, more than one side-information could be available. We denote the number of side-information with each arm as $q$. Let $W_{i,j}$ be the $j$th side-information of arm $i$ that is observed in round $t$. Then the unbiased mean reward estimator for arm $i$ with associated side-information is given by $\hat{\mu}_{s,i} = \hat{\mu}_{s,i} + \hat{\beta}_i^j (\omega_i - \hat{\omega}_{s,i})$ where $\hat{\beta}_i^j = (\hat{\beta}_{i,1}, \ldots, \hat{\beta}_{i,q})$, $\omega_i = (\omega_{i,1}, \ldots, \omega_{i,q})^\top$, and $\hat{\omega}_{s,i} = (\hat{\omega}_{s,i,1}, \ldots, \hat{\omega}_{s,i,q})^\top$. Let $s$ be the number of rewards and associated side-information samples for arm $i$, $W_i$ be the $s \times q$ matrix whose $r$th row is $(W_{r,i,1}, W_{r,i,2}, \ldots, W_{r,i,q})$, and $X_i = (X_{i,1}, \ldots, X_{i,s})^\top$. By extending the arguments used in Eq. (6) to $q$ side-information, the estimated coefficient vector is given by $\hat{\beta}_i = (W_i^\top W_i - s\hat{\omega}_{i}\hat{\omega}_{i}^{-1})^{-1}(W_i^\top X_i - s\hat{\omega}_{i}\hat{\mu}_{s,i})$. We can generalize this for MAB problems with $q$ side-informations and use UCBwSI with $Q = q + 2$ and appropriate optimistic upper bound for multiple side-informations.

APPENDIX C

UNKNOWN MEAN OF CONTROL VARIATE

When the mean of the control variate for $i$th arm is unknown, an approximate value $\hat{\omega}_i$ can be found using two approaches: 1) Estimate it by collecting additional samples at the start of the algorithm and replace $\omega_i$ by $\hat{\omega}_i$. 2) Use the approximate value, i.e., assume $\hat{\omega}_i \in [\omega_i - \varepsilon, \omega_i + \varepsilon]$, where $\varepsilon$ is the known error range. Both methods will degrade the performance, which is quantified in [41] and [42].

APPENDIX D

PROOF OF THEOREM 1

Following the standard UCB arguments, if $i \neq i^\star$ is selected in round $t$, at least one of the following events happens.

$$E_1. \, \hat{\mu}_{N(t),i^\star}^\alpha + V_{N(t)}(t)^\alpha \leq \mu_{i^\star},$$

$$E_2. \, \hat{\mu}_{N(t),i} - V_{N(t)}(t) \hat{\mu}_{N(t)}(t) > \mu_i,$$

$$E_3. \, N_i(t) < \frac{4(V_{N(t)}(T))^2 \hat{\mu}_{N_i(T)}(T)}{\Delta_i^2}.$$
Define $u = \left[ \frac{4\langle V_{T,N}(T) \rangle^2 \hat{P}_{N_i}(T), N_i(T) \rangle}{\Delta_i^2} \right]$. Write $N_i(T) = \sum_{t=1}^{T} 1 \{ t_i = 1 \}$. We have

$$E[N_i(T)]$$

$$\leq E[u] + E \left[ \sum_{t=1}^{T} 1 \{ E1 \text{ is true} \} \right] + E \left[ \sum_{t=1}^{T} 1 \{ E2 \text{ is true} \} \right].$$

$$E \left[ \sum_{t=1}^{T} 1 \{ E1 \text{ is true} \} \right] = \sum_{t=1}^{T} P \{ E1 \text{ is true} \}$$

$$= \sum_{t=1}^{T} P \{ \hat{P}_{C} \langle \rangle^{(2)}_{t,N_i(t),i}, t^{*}, q - \mu_{i} \leq -\langle V_{i}(2)_{t,N_i(t),q} \rangle \sqrt{P_{C} \langle \rangle^{(2)}_{t,N_i(t),i}, t^{*}, q} \}$$

$$\leq \sum_{t=1}^{T} \frac{1}{t^2} \leq \frac{\pi^2}{6},$$

where the first inequality follows from Proposition 2 by setting $\alpha = 2$. Following similar arguments, we get

$$E \left[ \sum_{t=1}^{T} 1 \{ E2 \text{ is true} \} \right] \leq \frac{\pi^2}{6}. $$

Notice that $u$ is random as its depends on $V_{T,N}(T)$ and $P_{N_i}(t)$, i. We bound $E[u]$ as

$$E[u] \leq \frac{4}{\Delta_i^2} E \left[ \langle V_{T,N}(T) \rangle^2 \hat{P}_{N_i}(T), N_i(T) \right] + 1.$$

$$= \frac{4\langle V_{T,N}(T) \rangle^2 \hat{P}_{N_i}(T), N_i(T) \rangle}{\Delta_i^2} + 1.$$

$$= \frac{4\langle V_{T,N}(T) \rangle^2 \hat{P}_{N_i}(T), N_i(T) \rangle}{\Delta_i^2} + 1.$$

$$= \frac{4\langle V_{T,N}(T) \rangle^2 \hat{P}_{N_i}(T), N_i(T) \rangle}{\Delta_i^2} + 1.$$

(From Proposition 1 and Proposition 3)

$$\frac{8\langle V_{T,N}(T) \rangle^2 (1-\rho^2_i)\sigma^2_i}{\Delta_i^2} \left[ \frac{\langle V_{T,N}(T) \rangle}{\langle V_{T,T} \rangle} \right]^2 + 1 \text{ as } N_i(T) \geq 4.$$
