Minimax Optimal Algorithms for Adversarial Bandit Problem With Multiple Plays

Nuri Mert Vural, Hakan Gokcesu, Kaan Gokcesu, and Suleyman S. Kozat, *Senior Member, IEEE*

***Abstract*—We investigate the adversarial bandit problem with multiple plays under semi-bandit feedback. We introduce a highly efficient algorithm that asymptotically achieves the performance of the best switching m-arm strategy with minimax optimal regret bounds. To construct our algorithm, we introduce a new expert advice algorithm for the multiple-play setting. By using our expert advice algorithm, we additionally improve the best-known high-probability bound for the multi-play setting by $O(\sqrt{m})$. Our results are guaranteed to hold in an individual sequence manner since we have no statistical assumption on the bandit arm gains. Through an extensive set of experiments involving synthetic and real data, we demonstrate significant performance gains achieved by the proposed algorithm with respect to the state-of-the-art algorithms.**

***Index Terms*—Adversarial multi-armed bandit, multiple plays, switching bandit, minimax optimal, individual sequence manner.**

I. Introduction

A. Preliminaries

MULTI-ARMED bandit problem is extensively investigated in the online learning [1]–[6] and signal processing [7]–[11] literatures, especially for the applications where feedback is limited, and exploration-exploitation must be balanced optimally. In the classical framework, the multi-armed bandit problem deals with choosing a single arm out of K arms at each round so as to maximize the total reward. We study the multiple-play version of this problem, where we choose an m sized subset of K arms at each round. We assume that

- The size m is constant throughout the game and known a priori by the learner.
- The order of arm selections does not have an effect on the arm gains.
- The total gain of the selected m arms is the sum of the gains of the selected individual arms.
- We can observe the gain of *each one of the selected* m *arms* at the end of each round. Since we can observe the gains of the individual arms in the selected subset, we also obtain partial information about the other possible subset selections with common individual arms (semi-bandit feedback).

We point out that this framework is extensively used to model several real-life problems such as online shortest path and online advertisement placement [12], [13].

We investigate the multi-armed bandit problem with multiple plays (henceforth *the MAB-MP problem*) in an individual sequence framework where we make no statistical assumptions on the data in order to model chaotic, non-stationary or even adversarial environments [5]. To this end, we evaluate our algorithms from a competitive perspective and define our performance with respect to a competing class of strategies. As the competition class, we use the switching m-arm strategies, where the term m-arm is used to denote any distinct m arms. We define the class of the switching m-arm strategies as the set of all deterministic m-arm selection sequences, where there are a total of $\binom{K}{m}^T$ sequences in a T length game. We evaluate our performance with respect to the best strategy (maximum gain) in this class. We note that similar competing classes are widely used in the control theory [14], [15], neural networks [16], [17], universal source coding theory [18]–[20] and computational learning theory [21]–[23], due to their modelling power to construct competitive algorithms that also work under practical conditions.

In the class of the switching m-arm strategies, the optimal strategy is, by definition, the one whose m-arm selection yields the maximum gain at each round of the game. If the optimal strategy changes its m-arm selection $S-1$ times, i.e., $S-1$ switches, we say the optimal strategy has S segments. Each such segment constitutes a part of the game (with possibly different lengths) where the optimum m-arm selection stays the same. For this setting, which we will refer as *the tracking the best* m*-arm setting*, we introduce a highly efficient algorithm that asymptotically achieves the performance of the best switching m-arm strategy with minimax optimal regret bounds.

To construct our tracking algorithm, we follow the derandomizing approach [23]. We consider each m-arm strategy as an expert with a predetermined m-arm selection sequence, where the number of experts grows with $\binom{K}{m}^T$, and combine them in an expert advice algorithm under semi-bandit feedback. Although we have exponentially many experts, we derive an optimal regret bound with respect to the best m-arm strategy with a specific choice of initial weights. We then efficiently implement this algorithm with a weight-sharing network, which requires $O(K \log K)$ time and $O(K)$ space. We note that our

Manuscript received December 17, 2018; revised March 30, 2019 and May 23, 2019; accepted June 16, 2019. Date of publication July 26, 2019; date of current version August 1, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wee Peng Tay. This work was supported by Turkish Academy of Sciences Outstanding Researcher Programme. *(Corresponding author: Nuri Mert Vural.)*

N. M. Vural, H. Gokcesu, and S. S. Kozat are with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: vural@ee.bilkent.edu.tr; hgokcesu@ee.bilkent.edu.tr; kozat@ee.bilkent.edu.tr).

K. Gokcesu is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: gokcesu@mit.edu).

Digital Object Identifier 10.1109/TSP.2019.2928952

algorithm requires prior knowledge of the number of segments in the optimal strategy, i.e., S. However, it can be extended to a truly online form, i.e., without any knowledge S, by using the analysis in [3] with an additional $O(\log T)$ time complexity cost.

We point out that the state-of-the-art expert advice algorithms [5], [24] cannot combine m-arm sequences optimally due to the additional $O(\sqrt{m})$ term in their regret bounds. Therefore, to construct an optimal algorithm, we introduce an optimal expert advice algorithm for the MAB-MP setting. In our expert advice algorithm, we utilize the structure of the expert set in order to improve the regret bounds of the existing expert advice algorithms [5], [24] up to $O(\sqrt{m})$. We then combine m-arm sequences optimally in this algorithm and obtain the minimax optimal regret bound. By using our expert advice algorithm, we additionally improve the best-known high-probability bound [25] by $O(\sqrt{m})$, hence, close the gap between high-probability bounds [25] and the expected regret bounds [24], [26]. In the end, we also demonstrate significant performance gains achieved by our algorithms with respect to the state-of-the-art algorithms [5], [24]–[27] through an extensive set of experiments involving synthetic and real data.

B. Prior Art and Comparison

The MAB-MP problem is mainly studied under three types of feedback: The full-information [28], where the gains of all arms are revealed to the learner, the semi-bandit feedback [24]–[27], [29], where the gains of the selected m arms are revealed, and the full bandit feedback [30]–[32], where only the total gain of the selected m-arm is revealed. Since our study lies in the semi-bandit scenario, we focus on the relevant studies for the comparison.

The adversarial MAB-MP problem where the player competes against the best fixed m-arm under semi-bandit feedback has a regret lower bound of $O(\sqrt{mKT})$[1] for K arms in a T round game [29]. On the other hand, a direct application of *Exp3* [5], i.e., the state-of-the-art for $m = 1$, achieves a regret bound $O(m^{3/2}\sqrt{K^m T \ln K})$ with $O(K^m)$ time and space complexity. One of the earliest studies to close this performance gap with an efficient algorithm is by György *et al.* [27]. They derived a regret bound $O(m^{3/2}\sqrt{KT \ln K})$ with respect to the best fixed m-arm in hindsight with $O(K)$ time complexity. This result is improved by Kale *et al.* [24] and Uchiya *et al.* [26] whose algorithms guarantee a regret bound $O(\sqrt{mKT \ln(K/m)})$ with $O(K^2)$ and $O(K \log K)$ time complexities respectively. Later, Audibert *et al.* [29] achieved the minimax optimal regret bound by the *Online Stochastic Mirror Descent (OSMD)* algorithm. The efficient implementation of *OSMD* is studied by Suehiro *et al.* [33] whose algorithm has $O(K^6)$ time complexity.

We emphasize that although minimax optimal bound has been achieved, all of these results have been proven to hold only in expectation. In practical applications, these algorithms suffer from the large variance of the unbiased estimator, which leads $O(T^{3/4})$ regret in the worst case [5], [34]. This problem is addressed by Györgi *et al.* [27] and Neu *et al.* [25] for the MAB-MP problem. They respectively derived $O(m^{3/2}\sqrt{KT \log(K/\delta)})$ and $O(m\sqrt{KT \log(K/m\delta)})$ regret bounds holding with probability $1 - \delta$.

In this paper, we introduce algorithms that achieve *minimax optimal regret* (up to logarithmic terms) with high probability for both the vanilla MAB-MP and the tracking the best m-arm settings. In order to generalize both settings in an optimal manner, we first introduce an optimal expert-mixture algorithm for the MAB-MP problem in Section III. In our expert-mixture algorithm, differing from the state-of-the-art [24], we exploit the structure of the expert set, and introduce the notion of underlying experts. By exploiting the structure of the expert set, we improve the regret bound of the state-of-the-art expert mixture algorithm for the MAB-MP setting [24] up to $O(\sqrt{m})$ and obtain the optimal regret bound against to the best expert, which can follow *any arbitrary strategy*. We then consider the set of the deterministic m-arm in our expert-mixture algorithm in Remark 3.1 and close the gap between high-probability bounds [25], [27] and the expected regret bounds [24], [26] for the vanilla MAB-MP setting.

In addition to our improvement in high-probability bound, we use our optimal expert mixture algorithm to develop a tracking algorithm for the MAB-MP setting. We note that when competing against the best switching m-arm strategy (as opposed to the best-fixed m-arm), the minimax lower bound can be derived as $O(mSKT)$[2]. However, similar to the case of *Exp3*, the direct implementation of the traditional multi-armed bandit algorithms into this problem suffers poor performance guarantees. To the best of our knowledge, only György *et al.* [27] studied competing against the switching m-arm sequences and derived $\tilde{O}(m^{3/2}\sqrt{SKT})$ regret bound holding with probability $1 - \delta$. In Section IV, by mixing the sets of switching m-arm sequences optimally in our expert mixture algorithm, we improve this result to $\tilde{O}(\sqrt{mSKT})$ regret bound holding with probability $1 - \frac{S-1}{e(T-1)}\delta$. We note that the computational complexity of our final algorithm is $O(K \log K)$, whereas György *et al.*'s algorithm [27, Section 6] requires $O(\min(KT, \binom{K}{m}))$ per round. Therefore, we also provide a highly efficient counterpart of the state-of-the-art.

C. Contributions

Our main contributions are as follows:

- As the first time in the literature, we introduce an online algorithm, i.e., *Exp3.MSP*, that truly achieves (with minimax optimal regret bounds) the performance of the best multiple-arm selection strategy.
- We achieve this performance with computational complexity only log-linear in the arm number, which is significantly

[1]We use big-O notation, i.e., $O(f(x))$, to ignore constant factors and use soft-O notation, i.e., $\tilde{O}(f(x))$, to ignore the logarithmic factors as well.

[2]When competing against the best switching bandit arm strategy (as opposed to the best fixed arm strategy), we can apply $O(\sqrt{mKT})$ bound separately to each one of S segment (if we know the switching instants). Hence, maximization of the total regret bound yields a minimax bound of $O(\sqrt{mSKT})$ since the square-root function is concave and the bound is maximum when each segment is of equal length T/S.

smaller than the computational complexity of the state-of-the-art [27].
- In order to obtain the minimax optimal regret bound with *Exp3.MSP*, we introduce an optimal expert mixture algorithm for the MAB-MP setting, i.e., *Exp4.MP*. We derive a lower bound for the MAB-MP with expert advice setting and mathematically show the optimality of the *Exp4.MP* algorithm.
- By using *Exp4.MP*, we additionally improve the best-known high-probability bound for the multiple-play setting by $O(\sqrt{m})$, hence, close the gap between high-probability bounds [25], [27] and the expected regret bounds [24], [26].

D. Organization of the Paper

The organization of this paper is as follows: In Section II we formally define the adversarial multi-armed bandit problem with multiple plays. In Section III, we introduce an optimal expert mixture algorithm for the MAB-MP setting. In Section IV, by using our expert mixture algorithm, we construct an algorithm that competes with the best switching m-arm strategy in a computationally efficient way. In Section V, we demonstrate the performance of our algorithms via an extensive set of experiments. We conclude with final remarks in Section VI.

II. Problem Description

We use bracket notation $[n]$ to denote the set of the first n positive integers, i.e., $[n]=\{1,\dots,n\}$. We use $\mathbf{C}([n],m)$ to denote the m-sized combinations of the set $[n]$. We use $[K]$ to denote the set of arms and $\mathbf{C}([K],m)$ to denote the set of all possible m-arm selections. We use $\mathbf{1}_A$ to denote the column vector, whose j^{th} component is 1 if $j\in A$, and 0 otherwise.

We study the MAB-MP problem, where we have K arms, and randomly select an m-arm at each round t. Based on our m-arm selection $U(t)\in\mathbf{C}([K],m)$, we observe only the gain of the selected arms, i.e., $x_i(t)\in[0,1]$ for $i\in U(t)$, and receive their sum as the gain of our selection $U(t)$. We assume $x_i(t)\in[0,1]$ for notational simplicity; however, our derivations hold for any bounded gain after shifting and scaling in magnitude. We work in the adversarial bandit setting such that we do not assume any statistical model for the arm gains $x_i(t)$. The output $U(t)$ of our algorithm at each round t is strictly online and randomized. It is a function of only the past selections and observed gains.

In a T round game, we define the variable $\mathbf{M}_T$, which represents a deterministic m-arm selection sequence of length T, i.e., $\mathbf{M}_T(t)\in\mathbf{C}([K],m)$ for $t=1,\dots,T$. In the rest of the paper, we refer to each such deterministic m-arm selection sequence, $\mathbf{M}_T$, as an m-arm strategy. The total gain of an m-arm strategy and the total gain of our algorithm (for this section, say the name of our algorithm is *ALG*) are respectively defined as

$$G_{\mathbf{M}_T}\triangleq\sum_{t=1}^{T}\sum_{i\in\mathbf{M}_T(t)}x_i(t),\ \text{and}\ G_{ALG}\triangleq\sum_{t=1}^{T}\sum_{i\in U(t)}x_i(t).$$

Since we assume no statistical assumptions on the gain sequence, we define our performance with respect to the optimum strategy $\mathbf{M}_T^*$, which is given as $\mathbf{M}_T^*=\arg\max_{\mathbf{M}_T}G_{\mathbf{M}_T}$. In order to measure the performance of our algorithm, we use the notion of *regret* such that

$$R(T)\triangleq G_{\mathbf{M}_T^*}-G_{ALG}.$$

There are two different regret definitions for the randomized algorithms: the expected regret and high-probability regret. Since the algorithms that guarantee high-probability regret yield more reliable performance [5], [34], we provide high-probability regret with our algorithms. High-probability regret is defined as

$$\mathbf{Pr}\Big[R(T)\geq\epsilon\Big]\leq\delta,$$

which means that the total gain of our selections up to T is not much smaller than the total gain of the best strategy $\mathbf{M}_T^*$ with probability at least $1-\delta$.

The regret $R(T)$ depends on how hard it is to learn the optimum m-arm strategy $\mathbf{M}_T^*$. Since at every switch we need to learn the optimal m-arm from scratch, we quantify the hardness of learning the optimum strategy by the number of segments it has. We define the number of segments as $S=1+\sum_{t=2}^{T}\mathbb{1}_{\mathbf{M}_T^*(t-1)\neq\mathbf{M}_T^*(t)}$. Our goal is to achieve that minimax optimal regret up to logarithmic factors with high probability, i.e.,

$$\mathbf{Pr}\Big[R(T)\geq\tilde{O}(\sqrt{mSKT})\Big]\leq\delta.$$

III. MAB-MP with Expert Advice

In this section, we consider selecting an m-arm with expert advice and introduce an optimal expert-mixture algorithm for the MAB-MP setting. We note that the primary aim of this section is to provide an optimal expert advice framework for the MAB-MP setting, on which we develop our optimal tracking algorithm in Section IV. By using our expert mixture algorithm, we additionally improve the best-known high-probability regret bound for the MAB-MP setting by $O(\sqrt{m})$ in the last remark of this section.

For this section, we define the phrase "expert advice" as the reference policies (or vectors) of the algorithm. The setting is as follows: At each round, each expert presents its m-arm selection advice as a K-dimensional vector, whose entries represent the marginal probabilities for the individual arms. The algorithm uses those vectors, along with the past performance of the experts, to choose an m-arm. The goal is to asymptotically achieve the performance of the best expert with high probability. For this setting, we introduce an optimal algorithm *Exp4.MP*, which is shown in Algorithm 1. In *Exp4.MP*, instead of directly using the expert set, we use an *underlying expert set* to utilize the possible structure of the expert set. An underlying expert set is defined as a non-negative vector set, whose sum of m-combinations constitute a set containing the expert advices (see Fig. 1). By using an underlying expert set, we replace the dependence of the regret on the size of the expert set N with the size of the underlying expert set N_r, thus, obtain the minimax lower bound in the soft-Oh sense (proven in the following). In the rest of the paper, we use the term *underlying experts* to denote the elements of the underlying expert set, and the term *actual*

Algorithm 1: Exp4.MP.

1: **Parameters:** $\eta, \gamma \in [0,1]$ and $c \in R^+$
2: **Initialization:** $w_i(1) \in R^+$ for $i \in [N_r]$
3: **for** $t = 1$ **to** T **do**
4: Get the actual advice vectors $\boldsymbol{\xi}^1(t), \ldots, \boldsymbol{\xi}^N(t)$
5: Find the underlying experts $\boldsymbol{\zeta}^1(t), \ldots, \boldsymbol{\zeta}^{N_r}(t)$
6: $v_j(t) = \sum_{i=1}^{N_r} \frac{w_i(t)\zeta_j^i(t)}{\sum_{l=1}^{N_r} w_l(t)}$ for $j \in [K]$
7: **if** $\arg\max_{j\in[K]} v_j(t) \geq \frac{(1/m)-(\gamma/K)}{(1-\gamma)}$ **then**
8: Decide α_t as $\frac{\alpha_t}{\sum_{v_j(t)\geq\alpha_t} \alpha_t + \sum_{v_j(t)<\alpha_t} v_j(t)} = \frac{(1/m)-(\gamma/K)}{(1-\gamma)}$
9: Set $U_0(t) = \{j : v_j(t) \geq \alpha_t\}$
10: $v'_j(t) = \alpha_t$ for $j \in U_0(t)$
11: **else**
12: Set $U_0(t) = \emptyset$
13: **end if**
14: Set $v'_j(t) = v_j(t)$ for $j \in [K] - U_0(t)$
15: $p_j(t) = m\left((1-\gamma)\frac{v'_j(t)}{\sum_{l=1}^{K} v'_l(t)} + \frac{\gamma}{K}\right)$ for $j \in [K]$
16: Set $U(t) = \text{DepRound}(m, (p_1(t), \ldots, p_K(t)))$
17: Observe and receive $x_j(t) \in [0,1]$ for each $j \in U(t)$
18: $\hat{x}_j(t) = x_j(t)/p_j(t)$ for $j \in U(t)$
19: $\hat{x}_j(t) = 0$ for $j \in [K] - U(t)$
20: **for** $i = 1$ **to** N_r **do**
21: $\hat{y}_i(t) = \sum_{j\in[K]-U_0(t)} \zeta_j^i(t)\hat{x}_j(t)$
22: $\hat{u}_i(t) = \sum_{j\in[K]-U_0(t)} \zeta_j^i(t)/p_j(t)$
23: $w_i(t+1) = w_i(t)\exp\left(\eta(\hat{y}_i(t) + \frac{c}{\sqrt{KT}}\hat{u}_i(t))\right)$
24: **end for**
25: **end for**

experts (respectively *actual advice vectors*) to denote the experts (respectively expert advices) presented to the algorithm.

In *Exp4.MP*, we first get the actual advice vectors $\boldsymbol{\xi}^k(t)$ for $k \in [N]$ in line 4. Since the entries of the actual advice vectors represent the marginal probabilities for the individual arms, they satisfy

$$\sum_{j=1}^{K} \xi_j^k(t) = m, \quad \max_{1\leq j\leq K} \xi_j^k(t) \leq 1, \quad \min_{1\leq j\leq K} \xi_j^k(t) \geq 0.$$

Then we find the underlying experts, i.e., $\boldsymbol{\zeta}^i(t)$ for $i \in [N_r]$, in line 5. We note that for the algorithms presented in this paper, we derive the underlying expert sets a priori. Therefore, our algorithms do not explicitly compute the underlying experts at each round.

In the algorithm, we keep a weight for each underlying expert, i.e., $w_i(t)$ for $i \in [N_r]$. We use those weights as confidence measure to find the arm weights, i.e., $v_j(t)$, in line 6 as follows:

$$v_j(t) = \sum_{i=1}^{N_r} \frac{w_i(t)\zeta_j^i(t)}{\sum_{l=1}^{N_r} w_l(t)} \quad \text{for } j \in [K]. \tag{1}$$

In order to select an m-arm, the expected total number of selection should be m, i.e., $\sum_{j=1}^{K} p_j(t) = m$. To satisfy this, we cap the arm weights so that the arm probabilities are kept in the range [0, 1]. For the arm capping, we first check if there is an arm weight larger than $\frac{(1/m)-(\gamma/K)}{(1-\gamma)}$ in line 7. If there is, we find the threshold α_t, and define the set $U_0(t)$ that includes the indices of the weights larger than α_t, i.e., $U_0(t) = \{j : v_j(t) \geq \alpha_t\}$. We set the temporal weights of the arms in $U_0(t)$ to α_t, i.e., $v'_j(t) = \alpha_t$ for $j \in U_0(t)$, and leave the other weights unchanged, i.e., $v'_j(t) = v_j(t)$ for $j \in [K] - U_0(t)$ (The implementation of this procedure is detailed in Appendix A). We then calculate the arm probabilities with the capped arm weights by

$$p_j(t) = m\left((1-\gamma)\frac{v'_j(t)}{\sum_{l=1}^{K} v'_l(t)} + \frac{\gamma}{K}\right) \quad \text{for } j \in [K]. \tag{2}$$

In order to efficiently select m distinct arms with the marginal probabilities $p_j(t)$, we employ Dependent Rounding (DepRound) algorithm [35] in line 16 (For the description of DepRound, see Appendix A). After selecting an m-arm, we observe the gain of each one of the selected m arms and receive their sum as the reward of the round.

To update the weights of the underlying experts, i.e., $w_i(t)$ for $i \in [N_r]$, we first find the estimated arm gains in lines 18–19:

$$\hat{x}_j(t) = \begin{cases} \frac{x_j(t)}{p_j(t)} & \text{if } j \in U(t) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Then by using the estimated arm gains $\hat{x}_j(t)$ for $j \in [K] - U_0(t)$, we calculate the estimated expected gain of the underlying experts by

$$\hat{y}_i(t) = \sum_{j\in[K]-U_0(t)} \zeta_j^i(t)\hat{x}_j(t) \quad \text{for } i \in [N_r]. \tag{4}$$

In order to obtain high-probability bound, we use upper confidence bounds. However, we note that we cannot directly use the upper confidence bound of the single arm setting [34] since *Exp4.MP* includes an additional non-linear weight capping in

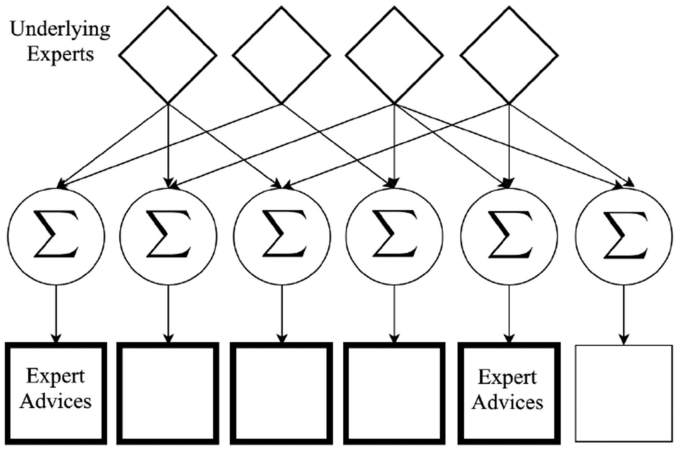


Fig. 1. In *Exp4.MP*, instead of directly using the expert set, we use an underlying expert set, whose sum of m-combinations constitute a set containing the expert advices. In the figure, the diamonds represent the underlying experts. The squares represent the sum of m-combinations. The bold squares are the expert advices presented to the algorithm. We note that in this figure, $m = 2$, $N_r = 4$, $N = 5$.

lines 7–14. In the following, we show that we can use a similar upper bounding technique by not including the capped arm weights $U_0(t)$, i.e.,

$$\hat{u}_i(t) = \sum_{j \in [K]-U_0(t)} \frac{\zeta_j^i(t)}{p_j(t)}. \tag{5}$$

Then by using $\hat{y}_i(t)$ and $\hat{u}_i(t)$, we update the weights $w_i(t)$ in line 23 by

$$w_i(t+1) = w_i(t) \exp\left(\eta(\hat{y}_i(t) + \frac{c}{\sqrt{KT}}\hat{u}_i(t))\right), \tag{6}$$

where η is the learning rate and $c/\sqrt{KT}$ is the scaling factor, which determines the range of the confidence bound.

For the following theorems, we respectively define the total gain of the underlying expert with index i, and its estimation as

$$G_i \triangleq \sum_{t=1}^{T} \boldsymbol{\zeta}^i(t) \cdot \boldsymbol{x}(t) \text{ and } \hat{G}_i \triangleq \sum_{t=1}^{T} \boldsymbol{\zeta}^i(t) \cdot \hat{\boldsymbol{x}}(t), \tag{7}$$

where $\boldsymbol{x}(t)$ and $\hat{\boldsymbol{x}}(t)$ are the column vectors containing the real and the estimated arm gains, i.e., $\boldsymbol{x}(t) = [x_1(t), \ldots, x_K(t)]^T$ and $\hat{\boldsymbol{x}}(t) = [\hat{x}_1(t), \ldots, \hat{x}_K(t)]^T$. Let us define a set A that includes m arbitrary underlying experts, i.e., $A \in \mathbf{C}([N_r], m)$. Then, by using the total gain of the underlying experts in the best A (in terms of the total gain), the total gain of the best actual expert can be written as

$$G_{max} = \max_{A \in \mathbf{C}([N_r],m)} \sum_{i \in A} G_i. \tag{8}$$

We also define the upper bounded estimated gain of a set A, i.e., $\hat{\Gamma}_A$, and the set with the maximum upper bounded estimated gain, i.e., A^*, as follows:

$$\hat{\Gamma}_A \triangleq \sum_{i \in A} \hat{G}_i + \frac{c}{\sqrt{KT}} \sum_{t=1}^{T} \sum_{i \in A} \hat{u}_i(t) \text{ and } A^* = \underset{A \in \mathbf{C}([N_r],m)}{\arg\max} \hat{\Gamma}_A. \tag{9}$$

In the following theorem, we provide a useful inequality that relates $\hat{\Gamma}_{A^*}$, $G_{Exp4.MP}$ and the initial weights of the underlying experts in A^*, i.e., $w_i(1)$ for $i \in A^*$ under a certain assumption. This inequality will be used to derive regret bounds for our algorithms in Corollary 3.1 and Theorem 4.1, where we ensure that the assumption in Theorem 3.1 holds.

Theorem 3.1: Let W_1 denote $\sum_{i=1}^{N_r} w_i(1)$. Assuming

$$\eta\left(\hat{y}_i(t) + \frac{c\hat{u}_i(t)}{\sqrt{KT}}\right) \leq 1, \quad \forall i \in [N_r] \text{ and } \forall t \in [T]$$

Exp4.MP ensures that

$$\left(1 - \gamma - 2\eta\frac{K}{m}\right)\hat{\Gamma}_{A^*} + \frac{(1-\gamma)}{\eta}\left(\sum_{i \in A^*} \ln(w_i(1)) - m \ln \frac{W_1}{m}\right) \leq G_{Exp4.MP} + c\sqrt{KT} + \frac{\eta c^2 2K}{\gamma m} \tag{10}$$

holds for any $K, T > 0$.

Proof: See Appendix B. ■

In the following corollary, we derive the regret bound of *Exp4.MP* with uniform initialization.

Corollary 3.1: If *Exp4.MP* is initialized with $w_i(1) = 1$ $\forall i \in [N_r]$, and run with the parameters

$$\eta = \frac{m\gamma}{2K} \qquad \gamma = \sqrt{\frac{K \ln \frac{N_r}{m}}{mT}} \qquad c = \sqrt{m \ln \frac{N_r}{\delta}},$$

for any $\frac{m \ln(N_r/\delta)}{K(e-2)} \leq T$ and $\delta \in [0, 1]$, it ensures that

$$G_{max} - G_{Exp4.MP} \leq 2\sqrt{mKT \ln \frac{N_r}{\delta}} + 4\sqrt{mKT \ln \frac{N_r}{m}} + m \ln \frac{N_r}{\delta} \tag{11}$$

holds with probability at least $1 - \delta$.

Proof: See Appendix B. ■

In the next theorem, we show that in the MAB-MP with expert-advice setting, no strategy can enjoy smaller regret guarantee than $O(\sqrt{mKT \ln N_r / \ln K})$ in the minimax sense. In its following, we also show that the derived lower bound is tight and it matches the regret bound of *Exp4.MP* given in Corollary 3.1.

Theorem 3.2: Assume that $N_r = K^n$ for an integer n and that T is a multiple of n. Let us define the regret of an arbitrary forecasting strategy ALG in a game length of T as

$$R_{ALG}(T) = G_{max} - G_{ALG}. \tag{12}$$

Then there exists a distribution for gain assignments such that

$$\inf_{\text{ALG}} \sup_{\boldsymbol{\xi}} R_{ALG}(T) \geq O\left(\sqrt{\frac{mKT \ln N_r}{\ln K}}\right), \tag{13}$$

where $\inf_{\text{ALG}}$ is an infimum over all possible forecasting strategies, $\sup_{\boldsymbol{\xi}}$ is a supremum over all possible expert advice sequences.

Proof: The presented proof is a modification of [36, Theorem 1] for the MAB-MP with expert advice setting. To derive a lower bound for the MAB-MP with expert advice setting, we split the interval $\{1, \ldots, T\}$ into n non-overlapping subintervals of length T/n, where each subinterval is assumed independent and indexed by $k \in \{1, \ldots, n\}$. For each subinterval, we design a MAB-MP game, where the optimal policy is some different $A_k \in \mathbf{C}([K], m)$. We also design $N_r = K^n$ sequences of underlying expert advice, such that for every possible every possible sequence of arms $j_1, \ldots, j_n \in \{1, \ldots, K\}^n$, there is an underlying expert that recommends the arms from the sequence throughout the corresponding subintervals. By using the lower bound $O(\sqrt{mKT})$ for the vanilla MAB-MP setting [29], for each subinterval k, we have

$$\inf_{\text{ALG}} R_{ALG}^k(T/n) \geq O\left(\sqrt{\frac{mKT}{n}}\right).$$

where $R_{ALG}^k(T/n)$ is the regret bound corresponding to the subinterval k. By summing all the regret components in each subinterval, and noting $R_{ALG}(T) \geq \sum_{k=1}^{m} R_{ALG}^k(T/n)$ and

$n = \ln N_r / \ln K$, we obtain

$$\inf_{\text{ALG}} \sup_{\boldsymbol{\xi}} R_{ALG}(T) \geq O\left(\sqrt{\frac{mKT \ln N_r}{\ln K}}\right).$$

■

We note that for $N_r = K$, MAB-MP with expert advice can be reduced to the vanilla MAB-MP setting (by considering underlying experts as arms) and in this case our regret lower bound matches the lower and upper bounds for the MAB-MP shown in [29]. Therefore, we maintain that our lower bound is tight. Furthermore, we note that the regret bound of *Exp4.MP* in Corollary 3.1 matches the presented lower bound with an additional $\ln K$ term, while the state-of-art [24] provides a suboptimal regret bound $O(\sqrt{mKT \ln N})$ (notably for $N_r << N$ as in the vanilla MAB-MP and the tracking the best m-arm settings). Therefore, we state that *Exp4.MP* is an optimal algorithm and it is required to obtain the improvements presented in this paper.

In the following remark, we improve the best-known high-probability bound [25] for the vanilla K-arm multi-play setting by $O(\sqrt{m})$. We note that the resulting bound matches with the minimax lower bound in the soft-Oh sense. Therefore, it cannot be improved in the practical sense.

Remark 3.1: If we use constant and deterministic actual advice vectors in *Exp4.MP*, i.e., $\boldsymbol{\xi}^k(t) = \mathbf{1}_A \in \mathbb{R}^K$ where $A \in \mathbf{C}([K], m)$, the algorithm becomes a vanilla K-armed MAB-MP algorithm. We note that in this scenario, we can directly operate with $\boldsymbol{\zeta}^i(t) = \mathbf{1}_i \in \mathbb{R}^K$, where $N_r = K$. By Corollary 3.1, if we use $\gamma = \sqrt{\frac{K \ln(K/m)}{mT}}$ and $c = \sqrt{m \ln(K/\delta)}$, *Exp4.MP* guarantees the regret bound $O(\sqrt{mKT \ln(K/\delta)})$ with probability at least $1 - \delta$. Since the most expensive operation of this scenario is arm capping, our algorithm achieves this performance with $O(K \log K)$ time and $O(K)$ space.

IV. Competing Against the Switching Strategies

In this section, we consider competing against the switching m-arm strategies. We present *Exp3.MSP*, shown in Algorithm 2, which guarantees to achieve the performance of the best switching m-arm strategy with the minimax optimal regret bound.

We construct *Exp3.MSP* algorithm by using *Exp4.MP* algorithm. For this, we first consider a hypothetical scenario, where we mix each possible m-arm selection strategy as an actual expert in *Exp4.MP*. We point out that the actual advice vectors will be a repeated permutation of the vectors $\{\mathbf{1}_A \in R^K : A \in \mathbf{C}([K], m)\}$ at each round, which we can write as the sum of m sized subsets of the set $\{\mathbf{1}_i \in R^K : i \in [K]\}$. Therefore, in this hypothetical scenario, we can directly combine all possible single arm sequences as the underlying experts, where $N_r = K^T$. However, since the regret bound of *Exp4.MP* is $O(\sqrt{mKT \ln(N_r/\delta)})$, a straightforward combination of K^T underlying experts produces a non-vanishing regret bound $O(T)$. To overcome this problem, we will assign a different prior weight for each one of K^T strategies based on its complexity cost, i.e., the number of segments S (more detail will be given later on).

Algorithm 2: Exp3.MSP.

Parameters: $\eta, \gamma, \beta \in [0, 1]$ and $c \in R^+$
Init: $v_1(j) = 1/K$ for $j \in [K]$
for $t = 1$ **to** T **do**
  if $\arg\max_{j \in [K]} v_j(t) \geq \frac{(1/m)-(\gamma/K)}{(1-\gamma)}$ **then**
    Decide α_t as $\frac{\alpha_t}{\sum_{v_j(t) \geq \alpha_t} \alpha_t + \sum_{v_j(t) < \alpha_t} v_j(t)} = \frac{(1/m)-(\gamma/K)}{(1-\gamma)}$
    Set $U_0(t) = \{j : v_j(t) \geq \alpha_t\}$
    $v'_j(t) = \alpha_t$ for $j \in U_0(t)$
  else
    Set $U_0(t) = \emptyset$
  end if
  Set $v'_j(t) = v_j(t)$ for $j \in [K] - U_0(t)$
  $p_j(t) = m\left((1-\gamma)\frac{v'_j(t)}{\sum_{l=1}^K v'_l(t)} + \frac{\gamma}{K}\right)$ for $j \in [K]$
  Set $U(t) = \text{DepRound}(m, (p_1(t), \ldots, p_K(t)))$
  Observe and receive rewards $x_j(t) \in [0, 1]$ for each $j \in U(t)$
  $\hat{x}_j(t) = x_j(t)/p_j(t)$ for $j \in U(t)$
  $\hat{x}_j(t) = 0$ for $j \in [K] - U(t)$
  for $j = 1$ **to** K **do**
    if $j \in [K] - U_0(t)$ **then**
      $\tilde{v}_j(t) = v_j(t) \exp\left(\eta(\hat{x}_j(t) + \frac{c}{p_j(t)\sqrt{KT}})\right)$
    else
      $\tilde{v}_j(t) = v_j(t)$
    end if
  end for
  $v_j(t+1) = \frac{(1-\beta)\tilde{v}_j(t) + \frac{\beta}{K-1}\sum_{i \neq j} \tilde{v}_i(t)}{\sum_{l=1}^K \tilde{v}_l(t)}$ for $j \in [K]$
end for

Let $\mathbf{s}_t$ be a sequence of single arm selections, $\mathbf{s}_t = \{s_1, s_2, \ldots, s_t\}$ where $\mathbf{s}_t(t) = s_t \in [K]$, and $w_{\mathbf{s}_t}$ be its corresponding weight. For ease of notation, we define

$$\hat{n}_{\mathbf{s}_t(t)} = \left(\hat{x}_{\mathbf{s}_t(t)} + \frac{c}{p_{\mathbf{s}_t(t)}\sqrt{KT}}\right) \mathbb{1}_{\mathbf{s}_t(t) \notin U_0(t)} \tag{14}$$

where $p_{\mathbf{s}_t(t)}$ is the probability of choosing $\mathbf{s}_t(t)$ at round t, and

$$\hat{N}_{\mathbf{s}_t(1:t-1)} = \sum_{\tau=1}^{t-1} \hat{n}_{\mathbf{s}_t(\tau)} \tag{15}$$

where $\mathbf{s}_t(i:j)$ denotes the i^{th} through j^{th} elements of the sequence $\mathbf{s}_t$. Then, the weight of the sequence $\mathbf{s}_t$ is given by

$$w_{\mathbf{s}_t} = \pi_{\mathbf{s}_t} \exp(\eta \hat{N}_{\mathbf{s}_t(1:t-1)}), \tag{16}$$

where $\pi_{\mathbf{s}_t}$ is the prior weight assigned to the sequence $\mathbf{s}_t$.

We point out that using non-uniform prior weights, i.e., $\pi_{\mathbf{s}_t}$, is required to have a vanishing regret bound since the number of single arm sequences grows exponentially with T. As noted earlier, by Corollary 3.1, the regret of *Exp4.MP* with uniform initialization is dependent on the logarithm of N_r. Therefore, combining K^T strategies with uniform initialization results in a linear regret bound, which is undesirable (the average regret does not diminish). In order to overcome this problem, similar to complexity penalty of AIC [37] and MDL [38], we assign

different prior weights $\pi_{\mathbf{s}_t}$ for each strategy $\mathbf{s}_t$ based on its the number of segments S. To get a truly online algorithm, we use a sequentially calculable prior assignment scheme that only depends on the last arm selection such that

$$\pi(\mathbf{s}_t|\mathbf{s}_t(t-1)) = \begin{cases} \frac{1}{K} & \text{if } t=1 \\ 1-\beta & \text{if } \mathbf{s}_t(t) = \mathbf{s}_t(t-1) \text{ (no switch)} \\ \frac{\beta}{K-1} & \text{if } \mathbf{s}_t(t) \neq \mathbf{s}_t(t-1) \text{ (switch).} \end{cases} \tag{17}$$

With the assignment scheme in (17), the prior weights are sequentially calculable as

$$\pi_{\mathbf{s}_t} = \pi(\mathbf{s}_t|\mathbf{s}_t(t-1))\pi_{\mathbf{s}_t(1:t-1)}$$

and the weights of the arm selection strategies are given by

$$w_{\mathbf{s}_t} = \pi(\mathbf{s}_t|\mathbf{s}_t(t-1))w_{\mathbf{s}_t(1:t-1)} \exp(\eta \hat{n}_{\mathbf{s}_t(t-1)}). \tag{18}$$

In the following theorem, we show that *Exp4.MP* algorithm running with K^T underlying experts with the prior weighting scheme given in (17) and (18) guarantees the minimax optimal regret bound up to logarithmic factors with probability at least $1 - \frac{S-1}{e(T-1)}\delta$.

Theorem 4.1: If *Exp4.MP* uses the prior weighting scheme given in (17) and (18), and the parameters

$$\eta = \frac{m\gamma}{2K} \qquad \beta = \frac{S-1}{T-1}$$

$$\gamma = \sqrt{\frac{K \ln\left(\frac{eK(T-1)}{S-1}\right)}{mT}} \qquad c = \sqrt{mS \ln\left(\frac{eK(T-1)}{(S-1)\delta}\right)}$$

to combine all possible single arms sequences as the underlying experts, for any $\frac{mS}{(e-2)K} \ln\left(\frac{eK(T-1)}{(S-1)\delta}\right) \leq T$ and $\delta \in [0,1]$

$$G_{\mathbf{M}_T^*} - G_{Exp4.MP} \leq 6\sqrt{mSKT \ln\left(\frac{eK(T-1)}{(S-1)\delta}\right)} + mS \ln\left(\frac{eK(T-1)}{(S-1)\delta}\right) \tag{19}$$

holds with probability at least $1 - \frac{S-1}{e(T-1)}\delta$.

Proof: See Appendix C. ■

Although we achieved minimax performance, we still suffer from exponential time and space requirements. In the following theorem, we show that by keeping K weights and updating the weights as

$$v_j(t+1) = \frac{(1-\beta)\tilde{v}_j(t) + \frac{\beta}{K-1}\sum_{i \neq j} \tilde{v}_i(t)}{\sum_{l=1}^{K} \tilde{v}_l(t)}, \tag{20}$$

where

$$\tilde{v}_j(t) = \begin{cases} v_j(t) \exp\left(\eta\left(\hat{x}_j(t) + \frac{c}{p_j(t)\sqrt{KT}}\right)\right) & \text{if } j \in [K] - U_0(t) \\ v_j(t) & \text{otherwise,} \end{cases} \tag{21}$$

we can efficiently compute the same weights in the hypothetical *Exp4.MP* run with a computational complexity linear in K. To show this, we extend [1, Theorem 5.1] for the MAB-MP problem:

Theorem 4.2: For any $\beta, \gamma, \eta \in [0,1]$, and for any $c, T > 0$, *Exp4.MP* algorithm that mixes all possible single arm sequences as underlying experts with the weighting scheme given in (17) and (18) has equal arm weights with *Exp3.MSP* algorithm, which updates its weights according to formulas in (20) and (21). ■

Proof: See Appendix C.

Theorem 4.2 proves that for any parameter selection *Exp3.MSP* is equivalent algorithm to the hypothetical *Exp4.MP* run. Therefore, Theorem 4.1 is valid for *Exp3.MSP*, which shows that *Exp3.MSP* has a $\tilde{O}(\sqrt{mSKT})$ regret bound holding with at probability at least $1 - \frac{S-1}{e(T-1)}$ with respect to the optimal m-arm strategy. Since the most expensive operation in the algorithm is capping, *Exp3.MSP* requires $O(K \log K)$ time complexity per round.

V. Experiments

In this section, we demonstrate the performance of our algorithms with simulations on real and synthetic data. These simulations are mainly meant to provide a visualization of how our algorithms perform in comparison to the state-of-the-art techniques and should not be seen as verification of the mathematical results in the previous sections. We note that simulations only show the loss/gain of an algorithm for a typical sequence of examples; however, the mathematical results of our paper are the worst-case bounds that hold even for adversarially-generated sequences of examples.

For the following simulations, we use four synthesized datasets and one real dataset. We compare *Exp4.MP* with *Exp3.P* [5], *Exp3-IX* [39], *Exp3.M* [26], *FPL+GR.P* [25], [27, Figure 2], and [24, Figure 4]. We compare *Exp3.MSP* with [27, Figure 4], and *Exp3.S* [5]. We note that all the simulated algorithms are constructed as instructed in their original publications. The parameters of the individual algorithms are set as instructed by their respective publications. The information of the game length T and the number of the segments in the best strategy S have been given a priori to all algorithms. In each subsection, all the compared algorithms are presented to the identical games.

A. Robustness of the Performances

We conduct an experiment to demonstrate the robustness of our high-probability algorithms. For this, we run algorithms several times and compare the distributions of their total gains. For the comparison, we use *Exp4.MP* with the deterministic and constant advice vectors. We compare *Exp4.MP* with *Exp3.P* [5], *Exp3-IX* [39], *Exp3.M* [26], *FPL+GR.P* [25] and the high-probability algorithm introduced by György *et al.* in [27]. Since György *et al.* did not name their algorithms, we use *GYA-P* to denote their high-probability algorithm. We highlight that all the algorithms except *Exp3.M* guarantee a regret bound with high probability, whereas *Exp3.M* guarantees an expected regret bound.

For this experiment, we construct a 10-arm bandit game where we choose 5 bandit arms at each round. All gains are generated by independent draws of Bernoulli random variables. In the first half of the game, the mean gains of the first 5 arms are $0.5 + \epsilon$, and the mean gains of others' are $0.5 - \epsilon$. In the second half of

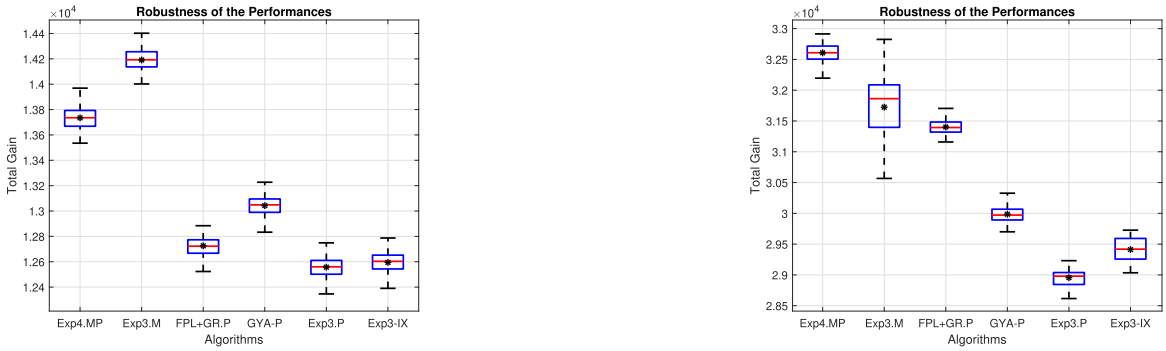


Fig. 2. Comparison of the distributions of the total gains received by the algorithms (a) up to $T/2$ (b) up to T.

the game, the mean gains of the first 5 arms have been reduced to $0.5-\epsilon$, while the mean gains of others' have been increased to $0.5+4\epsilon$. We point out that based on these selections, the m-arm consisting of the last 5 arms performs better than the others in the full game length.

We set the parameters $T=10^4, \epsilon=0.1$ for all the algorithms, and $\delta=0.01$ for the high-probability algorithms. Our experiments are repeated 100 times to obtain statistically significant results. Since the game environments are the same for all the algorithms, we directly compare the total gains. We study the total gain up to two interesting rounds in the game: up to $T/2$, where the losses are independent and identically distributed, and up to T, where the algorithms have to notice the shift in the gain distribution.

We have constructed box plots by using the resulting total gains of the algorithms. In the box plots, the lines extending from the boxes (the whiskers) illustrate the minimum and the maximum of the data. The boxes extend from the first quartile to the third quartile. The horizontal lines and the stars stand for the median and the mean of the distributions. Fig. 2a illustrates the distributions of the total gains up to $T/2$. We observe that the variance of all the total gains are comparable. The mean total gain received by *Exp4.MP* is only comparable with that of *Exp3.M* while outperforms the rest. On the other hand, when the change occurred in the game, *Exp4.MP* outperforms the rest in the overall performance (Fig. 2b). As expected, *Exp3.M* and *Exp4.MP* receive relatively higher gains than the other algorithms. However, since we give a special care for bounding the variance, *Exp4.MP* has a more robust performance. From the results, we can conclude that *Exp4.MP* yields the superior performance of the algorithms with an expected regret guarantee and the robustness of the high-probability algorithms at the same time.

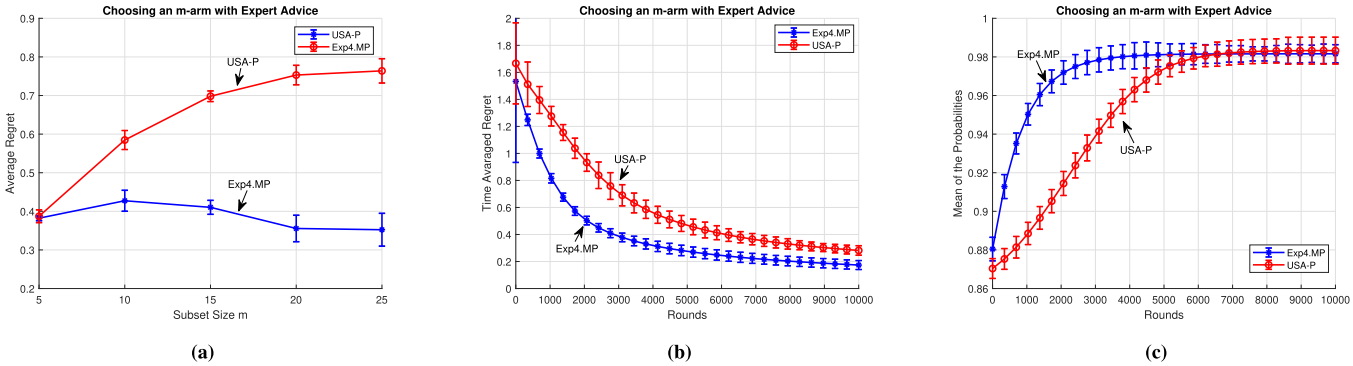
B. Choosing an m-arm With Expert Advice

In this part, we demonstrate the performance of *Exp4.MP* when the advice vectors of the actual advice vectors are not necessarily constant nor deterministic. We compare our algorithm with the only known algorithm that is capable of choosing m-arm with expert advice, i.e., *Unordered Slate Algorithm with policies (USA-P)* introduced in [24]. Since our main point is to improve the regret bound by $O(\sqrt{m})$, we compare algorithms under different subset sizes. For this, we construct five 30-armed games, where we choose $m\in\{5,10,15,20,25\}$ arms respectively. In each game, the gains of the first m arms are 1, and the gains of the others are 0 throughout the game. In order to satisfy the condition $N_r=O(N^{\frac{1}{m}})$, we first generate the underlying expert set where $N_r=m+2$. The generation process is as follows: The first m underlying experts are chosen constant and deterministic where $\boldsymbol{\zeta^i}=\mathbf{1}_i$ for $i\in\{1,\dots,m\}$. The first m entries of the last two underlying experts are chosen 0, i.e., $\zeta_j^{m+1}(t)=\zeta_j^{m+2}(t)=0$ for $j\in\{1,\dots,m\}$, while the other entries are determined randomly at each round under the constraint that their sum is 1. The actual vectors are generated by summing each m sized subset of the underlying expert set at each round, where we have a total of $\binom{m+2}{m}$ experts. We note that based on our arm gains selection and the advice vector generating process, the actual expert which is the sum of the constant underlying experts is the best expert.

In the experiment, we set the parameters $T=10^4$ for both algorithms and $\delta=0.01$ for *Exp4.MP*. We have repeated all the games 100 times and plotted the ensemble distributions in Fig. 3a, Fig. 3b, and Fig. 3c. Fig. 3a illustrates the time averaged regret incurred by the algorithms at the end of the games with increasing m. As can be seen, the regret incurred by our algorithm remains almost constant while the regret of *USA-P* increases as m increases. In order to observe the temporal performances, we have plotted Fig. 3b, which illustrates the time averaged regret performances of the algorithms when $K=30$, and $m=15$. We observe that our algorithm suffers a lower regret value at each round. To analyze this difference in the performances, we have also plotted the mean of the probability values assigned to the optimum m arms by the algorithms at each round when $m=15$ (Fig. 3c). We observe that *USA-P* saturates at the same probability value as *Exp4.MP*, its convergence rate is slower. Therefore, since our algorithm can explore the optimum m-arm more rapidly, it is able to achieve better performance, especially in high m values.

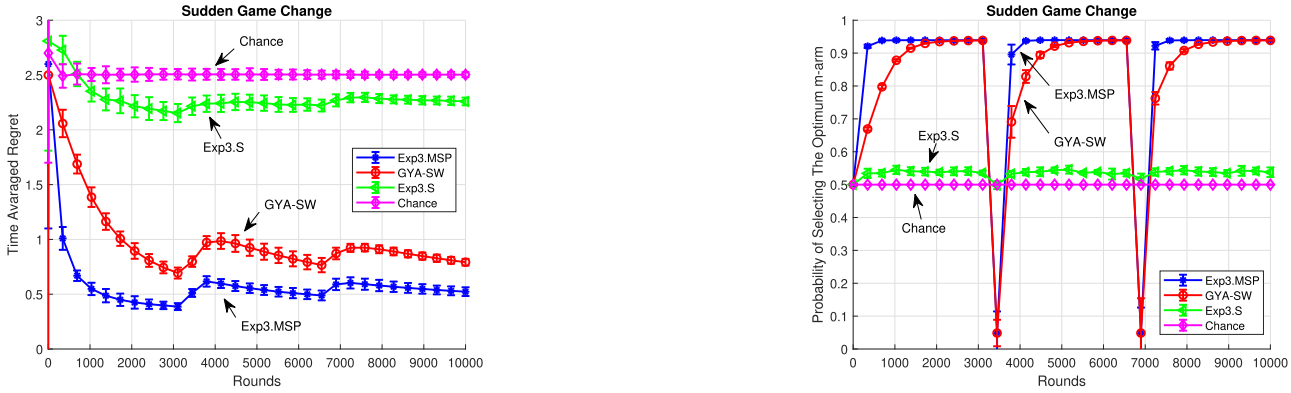
C. Sudden Game Change

In this section, we demonstrate the performance of *Exp3.MSP* in a synthesized game. We compare our algorithm with Exp3.S



(a) **(b)** **(c)**

Fig. 3. (a) Per round regret performances of the algorithms with increasing m. (b) Time-averaged regret performances of the algorithms in a game where $K = 30$, $m = 15$. (c) Mean of the probabilities assigned to the optimum m arms by both algorithms when $m = 15$.



(a) **(b)**

Fig. 4. (a) Time averaged regrets of the algorithms in a game where $K = 10$, $m = 5$ and the optimum m-arm changes at every 3333 rounds. (b) Probabilities of selecting the optimum m-arm for all of the algorithms in sudden game change setting.

and the algorithm introduced by György *et al.* in [27, Section 6]. Since György *et al.* did not name their algorithms, we use *GYA-SW* to denote their algorithm. We also compare each algorithm against the trivial algorithm, Chance (i.e., random guess) for a baseline comparison.

For this experiment, we construct a game of length $T = 10^4$, where we need to choose 5 arms out of 10 bandit arms. The gains of the arms are deterministically selected as follows: Up to round 3333, the gains of the first 5 arms are 1, while the gains of the rest are 0. Between rounds 3334 and 6666, the gains of the last 5 arms are 1, while the gains of the rest 0. In the rest of the game, the gain distribution is the same as in the first 3333 rounds. The optimum m-arms at consecutive segments are intentionally selected mutually exclusive in order to simulate sudden changes effectively. We point out that based on our arm gains selections, the number of segments in the optimum m-arm sequence is 3, i.e., $S = 3$.

For *Exp3.MSP* and *GYA-SW*, we set $\delta = 0.01$. We have repeated the games 100 times and plotted the ensemble distributions in Fig. 4a and Fig. 4b. Fig. 4a illustrates the time-averaged regret performance of the algorithms. We observe that our algorithm has a lower regret value at any time instance. To analyse this, we have plotted the mean probability values assigned to the optimum m arms by the algorithms at each round in Fig. 4b. We observe that *Exp3.S* cannot reach high values of probability due to the exponential size of its action set. We also see that although *GYA-SW* saturates at the same probability value as *Exp3.MSP*, its convergence rate is slower. Therefore, Fig. 4b shows that since our algorithm adapts faster to the changes in the environment, it achieves a better performance throughout the game.

D. Random Game Change

In this part, we demonstrate the performance of *Exp3.MSP* on random data sequences. We compare our algorithm with two state-of-the-art techniques: *Exp3.S* [5] and *GYA-SW* [27]. We also compare each algorithm against the trivial algorithm, Chance (i.e., random guess) for a baseline comparison. For this experiment, we construct a game whose behavior is completely random with the only regularization condition being an m-arm should be optimum throughout a segment. We start to synthesize the dataset by randomly selecting gains in [0, 1] for all arms for all rounds. We predetermine the optimum m-arms in each segment and then switch the maximum gains with the gains of the optimum m-arm at each round. This synthesized dataset creates a game with randomly determined gains while maintaining that

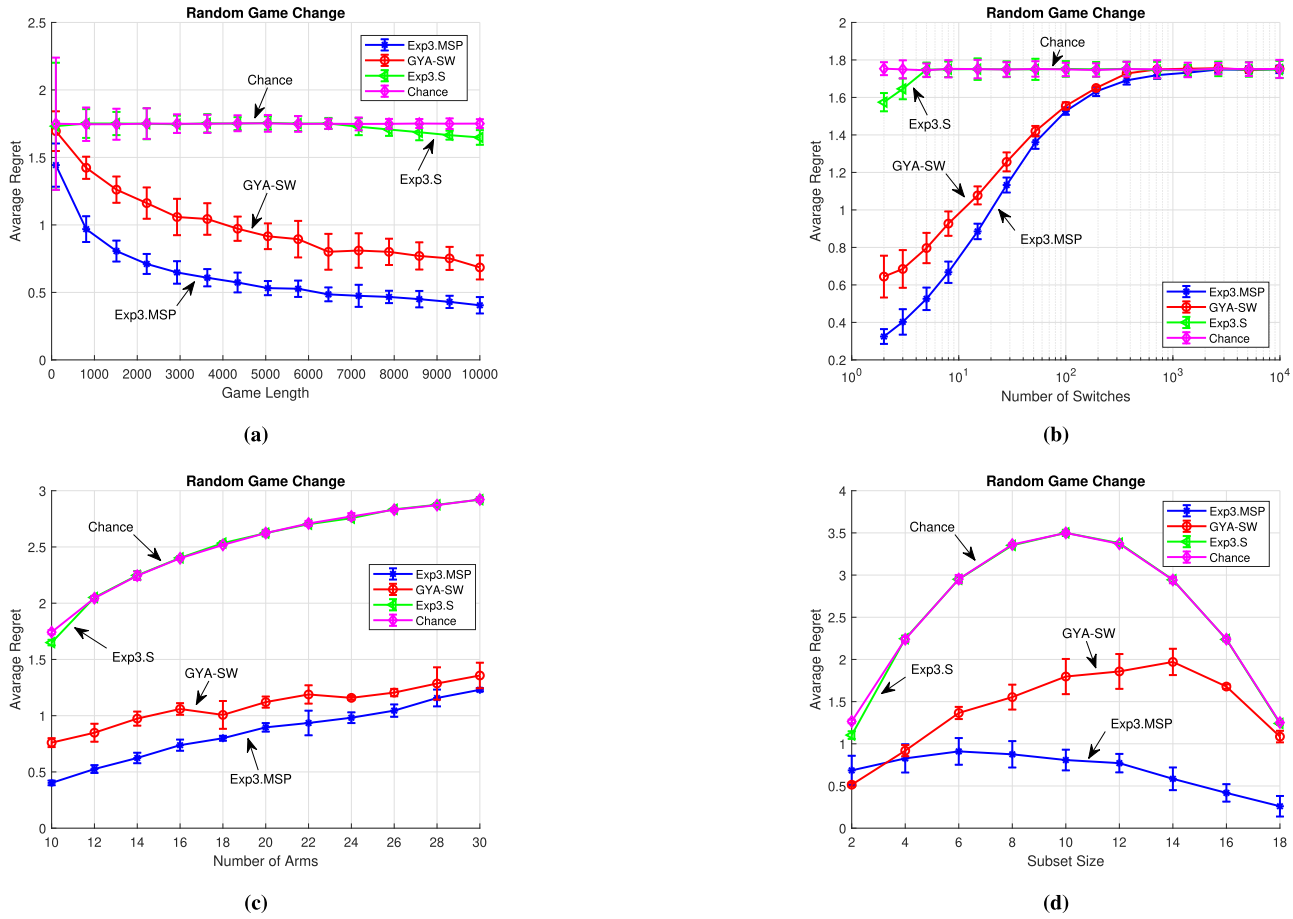


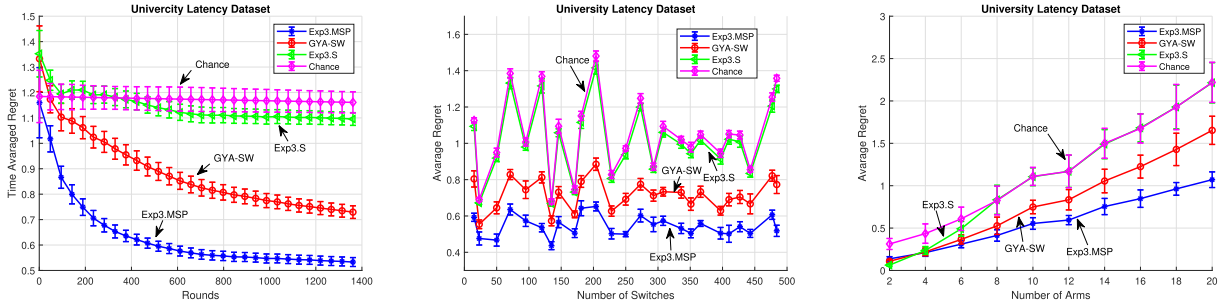
Fig. 5. (a) Per round regrets of the algorithms with increasing game length. (b) Per round regrets of the algorithms with increasing number of switches. (c) Per round regrets of the algorithms with increasing number of bandit arms. (d) Per round regrets of the algorithms with increasing number of subset size.

one m-arm is uniformly optimum throughout each segment. We synthesize multiple datasets to analyze the effects of the parameters of the game individually, where we compare the algorithms performances for varying game length (T), number of switches (S), number of arms (K) and subset size (m). We start with the control group of $T = 10^4$, $K = 10$, $m = 5$, $S = 3$ and both T and S is known a priori. Then, for each case, we vary one of the above four parameters. Differing from before, the time instances of switches are not fixed to 3333 and 6666 but instead selected randomly to be in anywhere in the game. Thus, we create completely random games with three segments.

To observe the effect of game length, we selected the 15 different game lengths, which are linearly spaced between 10^2 and 10^4. We provided the algorithms with the prior information of both the game length and the number of switches. In Fig. 5a, we have plotted the average regret incurred at the end of the game, i.e., $R(T)/T$, by the algorithms at different values of game length while fixing the other parameters. We note that the error bars in Fig. 5a illustrate the maximum and the minimum average regret incurred at a fixed value of game length. For any set of parameters, we have simulated the setting for 25 times with recreating the game each time to obtain statistically significant results. We observe that the algorithm *Exp3.S* performs close to random guess up to approximately game length of 7000 rounds, which is expected since it assumes each action as a separate arm. We also observe that *Exp3.MSP* and *GYA-SW* perform better than the chance for all values of game length. However, there is a significant performance difference in favor of *Exp3.MSP*.

To observe the effect of the number of switches on the performances, we created random change games with 10 arms, subset size $m = 5$ and game length of 10^4. We provided the algorithms with the prior information of both the game length and the number of switches. We selected 15 different switch values, which are logarithmically spaced between 2 and 10^4. In Fig. 5b, we have plotted the average regret incurred at the end of the game by the algorithms at different values of number of switches while fixing the other parameters. For any set of parameters, we have simulated the setting for 25 times with recreating the game each time. The algorithm *Exp3.S* performs similar to random guess after approximately 5 switches, i.e., $S = 5$. Both *Exp3.MSP* and *GYA-SW* catch random guess at $S = 1000$, which is comparable to the value of game length, i.e., $T = 10^4$. However, *Exp3.MSP* manages to provide better performance than the other algorithms for all number of switches.

To observe the effect of the number of bandit arms on the performances, we created random change games with 3 segments, subset size of 5 and game length of 10^4. We provided the algorithms with the prior information of both the game length



(a) (b) (c)

Fig. 6. (a) Time averaged regrets in "univ-latencies" dataset when $K = 10$, $m = 5$ and $S = 3$. (b) Per round regret performances of the algorithms at the end of the games with increasing number of switches. (c) Per round regret performances of the algorithms at the end of the games with increasing number of bandit arms.

and the number of switches. We selected the number of bandit arms to be even numbers between 10 and 30. In Fig. 5c, we have plotted the average regret incurred at the end of the game by the algorithms at different values of the number of bandit arms while fixing the other parameters. For any set of parameters, we have simulated the setting for 25 times with recreating the game each time. The algorithm *Exp3.S* performs similar to random guess after approximately 12 bandit arms. *Exp3.MSP* and *GYA-SW* outperform random guess for all values of bandit arms. On the other hand, *Exp3.MSP* outperforms all algorithms for all values of bandit arms uniformly.

To observe the effect of the subset size on the performances, we created random change games with 3 segments, 20 bandit arms and game length of 10^4. We provided the algorithms with the prior information of both the game length and the number of switches. We selected the subset size to be even numbers between 2 and 18. In Fig. 5d, we have plotted the average regret incurred at the end of the game by the algorithms at different values of number of subset sizes while fixing the other parameters. For any set of parameters, we have simulated the setting for 25 times with recreating the game each time. We observe that the algorithm *Exp3.S* performs similar to random guess after subset size is equal to 4. *Exp3.MSP* and *GYA-SW* outperform random guess for all values of subset sizes. On the other hand, *Exp3.MSP* yields better performance, especially in high values of subset size.

E. Online Shortest Path

In this subsection, we use a real-world networking dataset that corresponds to the retrieval latencies of the homepages of 760 universities. The pages were probed every 10 min for about 10 days in May 2004 from an internet connection located in New York, NY, USA [40]. The resulting data includes 760 URLs and 1361 latencies (in millisecond) per URL. For the setting, we consider an agent that must retrieve data through a network with several redundant sources available. For each retrieval, the agent is assumed to select m sources and wait until the data is retrieved. The objective of the agent is to minimize the sum of the delays for the successive retrievals. Intuitively, each page is associated with a bandit arm and each latency with a loss.

Before experiments, we have preprocessed the dataset. We observed that the dataset includes too high latencies. Therefore, we have truncated the latencies at 1000 ms, which can be thought of as timeout for a more realistic real-world setting. We have normalized the truncated latencies into [0, 1], and converted them to the gains by subtracting from 1. Since there are 1361 latencies for each URL, we set $T = 1361$. The other parameters are selected as $\delta = 0.01$ and $S = 3$. We note that the value of S we chose might not correspond to the actual segment number in the optimum m-arm sequence. Nonetheless, it is selected arbitrarily to simulate the practical cases where the value of S is not available a priori.

Using the universities as the bandit arms, we have extracted 100 different games with 10 bandit arms. For each game, we assumed that the agent chose 5 arms, i.e., $K = 10$ and $m = 5$. We have repeated each game 20 times and plotted the time-averaged regrets in Fig. 6a. Similar to all of the tests before, we observe that our algorithm achieves a better performance in all time instances. Additionally, we do benchmarks on the number of segments and the number of arms. For the number of segments benchmark, we have extracted 20 10-arm games, where the actual segment number in the optimum strategy is in the interval $[20i, 20i + 19]$ for each game indexed by $i \in \{0, 1, \ldots, 19\}$. We have repeated each game 20 times and plotted the ensemble distributions in Fig 6b. Interestingly, there is no direct correlation between the segment numbers and the average regret values. Furthermore, as can be observed, irrespective of the number of segments, our algorithm outperforms the other algorithms for all number of switches. For the number of bandit arms, we have used the even numbers from 2 to 20 and chose the half of the arms in every game. For each number of arms, we have extracted 20 different games. We have run each game 20 times plotted the ensemble distributions in Fig. 6c. As expected, the regret values of the algorithms increase with the number of arms. Moreover, the difference in performances becomes more apparent as m increases, which is consistent with our theoretical results.

VI. Concluding Remarks

We studied the adversarial bandit problem with multiple plays, which is a widely used framework to model online shortest

Algorithm 3: DepRound.

Inputs: The subset size $m(< K)$, $(p_1, p_2, \ldots, p_K)$ with $\sum_{i=1}^{K} p_i = m$
Output: Subset of $[K]$ with m elements
while there is an i with $0 < p_i < 1$ **do**
  Choose distinct i and j with $0 < p_i < 1$ and $0 < p_j < 1$
  Set $\alpha = \min(1 - p_i, p_j)$ and $\beta = \min(p_i, 1 - p_j)$
  Update p_i and p_j as

$$(p_i, p_j) = \begin{cases} (p_i + \alpha, p_j - \alpha) \text{ with probability } \frac{\beta}{\alpha+\beta} \\ (p_i - \beta, p_j + \beta) \text{ with probability } \frac{\alpha}{\alpha+\beta} \end{cases}$$

end while
return $\{i : p_i = 1, 1 \geq i \geq K\}$

path and online advertisement placement problems [12], [13]. In this context, as the first time in the literature, we have introduced an online algorithm that truly achieves (with minimax optimal regret bounds) the performance of the best multiple-arm selection strategy. Moreover, we achieved this performance with computational complexity only log-linear in the arm number, which is significantly smaller than the computational complexity of the state-of-the-art [27]. We also improved the best-known high-probability bound for the multi-play setting by $O(\sqrt{m})$, thus, close the gap between high-probability bounds [25], [27] and the expected regret bounds [24], [26]. We achieved these results by first introducing a MAB-MP with expert advice algorithm that is capable of utilizing the structure of the expert set. Based on this algorithm, we designed an online algorithm that sequentially combines the selections of all possible m-arm selection strategies with carefully constructed weights. We show that this algorithm achieves minimax regret bound with respect to the best switching m-arm sequence and it can be efficiently implementable with a weight-sharing network applied on the individual arm weights. Through an extensive set of experiments involving synthetic and real data, we demonstrated significant performance gains achieved by our algorithms with respect to the state-of-the-art adversarial MAB-MP algorithms [5], [24]–[27].

APPENDIX A

A. Dependent Rounding (DepRound)

To efficiently select a set of m distinct arms from $[K]$, we use a nice technique called dependent rounding (DepRound) [35], see Algorithm 3. DepRound takes as input the subset size m and the arm probabilities $(p_1, p_2, \ldots, p_K)$ with $\sum_{i=1}^{K} p_i = m$. It updates the probabilities until all the components are 0 or 1 while keeping the sum of probabilities unchanged, i.e., m. The while-loop is executed at most K times since at least one of p_i and p_j becomes 0 or 1 in each time of the execution. The algorithm updates the probabilities in a randomized manner such that it keeps the expectation values of p_i the same, namely, $E[p_i^{t+1}] = E[p_i^t]$ for every $i \in [K]$, where p_i^t denotes p_i after the t^{th} execution of the inside of the while-loop. This follows

Algorithm 4: Capping Algorithm.

1: **Input:** The subset size $m(< K)$, $(v_1, v_2, \ldots, v_K)$ with $\sum_{j=1}^{K} v_i = 1$
2: $\boldsymbol{v}^{\downarrow} \leftarrow$ Sort $(v_1, v_2, \ldots, v_K)$ in a descending order
3: **indices**$^{\downarrow} \leftarrow$ Keep the original indices of the sorted weights
4: upper_bound $= \frac{(1/m)-(\gamma/K)}{(1-\gamma)}$
5: $i \leftarrow 1$
6: $\mathbf{temp} \leftarrow \boldsymbol{v}^{\downarrow}$
7: **repeat**
8: (Set first i largest components to upper_bound and normalize the rest to $(1 - i *$ upper_bound$)$)
9: $\mathbf{temp} \leftarrow \boldsymbol{v}^{\downarrow}$
10: temp$(j) =$ upper_bound for $j = 1, \ldots, i$
11: temp$(j) = (1 - i *$ upper_bound$)\frac{\text{temp}(j)}{\sum_{l=i+1}^{K} \text{temp}(l)}$ for $j = i+1, \ldots, K$
12: $i \leftarrow i + 1$
13: **until** $\max(\mathbf{temp}) \leq$ upper_bound
14: $(v_1, v_2, \ldots, v_K) \leftarrow$ Replace the entries of **temp** by using **indices**$^{\downarrow}$
15: **return**

from

$$(p_i + \alpha)\frac{\beta}{\alpha+\beta} + (p_i - \beta)\frac{\alpha}{\alpha+\beta} = (p_i - \alpha)\frac{\beta}{\alpha+\beta} + (p_i + \beta)\frac{\alpha}{\alpha+\beta} = p_i, \quad (22)$$

which indicates that each arm in the output is selected by its marginal probability p_i. Since the while-loop is executed at most K times, DepRound runs in $O(K)$ time and $O(K)$ space.

B. Arm Capping

In this section, we describe how we find the threshold α_t and cap the weights, i.e the lines 7–14 in Algorithm 1, and the lines 4–11 in Algorithm 2. The presented algorithm in Algorithm 4 simultaneously finds the threshold α_t and caps the weights.

In the algorithm, we start with sorting the arm weights in a descending order (line 2). Then, we set the largest i arm weights to $\frac{(1/m)-(\gamma/K)}{(1-\gamma)}$ (line 10) and normalize the other weights (line 11) so that the sum of the weights stays 1. By the lines 10, 11, and 13, we aim to satisfy

$$\frac{\alpha_t}{\sum_{v_j(t) \geq \alpha_t} \alpha_t + \sum_{v_j(t) < \alpha_t} v_j(t)} = \frac{(1/m) - (\gamma/K)}{(1-\gamma)} \quad (23)$$

subject to $\max(v_j(t)) = \alpha_t$. Since $\frac{(1/m)-(\gamma/K)}{(1-\gamma)} > \frac{1}{m}$, there is always an $i < m$ that satisfies Eq. (23) and it can be found in $O(m)$ step. After finding i, the algorithm replaces the capped arm-weights (line 14) and returns. Since the most expensive operation in the algorithm is sorting, the algorithm requires $O(K \log K)$ time complexity.

APPENDIX B

Proof of Theorem 3.1: Define $q_i(t) = w_i(t)/W_t$, where $W_t = \sum_{i=1}^{N_r} w_i(t)$, and $\tilde{y}_i(t) = \hat{y}_i(t) + c\hat{u}_i(t)/\sqrt{KT}$. By following the first steps of the proof of *Exp3.M* (up to inequality (4) in [26]), we can write

$$\ln\left(\frac{W_T}{W_1}\right) \le \eta \sum_{t=1}^{T}\sum_{i=1}^{N_r} q_i(t)\tilde{y}_i(t) + \eta^2 \sum_{t=1}^{T}\sum_{i=1}^{N_r} q_i(t)\tilde{y}_i(t)^2 \tag{24}$$

with the assumption of $\eta\tilde{y}_i(t) \le 1$. By using the AM-GM inequality, we get:

$$\begin{aligned}\ln\left(\frac{W_T}{W_1}\right) &\ge \sum_{r\in A^*} \frac{\ln(w_r(T+1))}{m} - \ln\frac{W_1}{m} \\ &= \frac{\eta}{m}\sum_{t=1}^{T}\sum_{r\in A^*}\tilde{y}_r(t) + \frac{1}{m}\sum_{r\in A^*}\ln(w_r(1)) - \ln\frac{W_1}{m}.\end{aligned} \tag{25}$$

where A^* is the set defined in (9). To bound the terms with $\tilde{y}_i(t)$, we give two useful facts:

$$\begin{aligned}\sum_{i=1}^{N_r}\frac{w_i(t)\zeta_j^i(t)}{W_t} = v_j(t) &\le \frac{v_j'(t)}{\sum_{l=1}^{K} v_l'(t)} \\ &\le \frac{p_j(t)}{m(1-\gamma)}, \quad j \in [K]-U_0(t)\end{aligned} \tag{26}$$

where we use $\sum_{l=1}^{K} v_l'(t) \le \sum_{l=1}^{K} v_l(t) = 1$. For the second fact, let us say $d_i(t) = \sum_{j\in[K]-U_0(t)} \zeta_j^i(t)$. Then,

$$\begin{aligned}\sum_{i=1}^{N_r}\frac{w_i(t)}{W_t}\hat{y}_i(t)^2 &= \sum_{i=1}^{N_r}\frac{w_i(t)}{W_t}d_i(t)^2\left(\sum_{j\in[K]-U_0(t)}\frac{\zeta_j^i(t)}{d_i(t)}\hat{x}_j(t)\right)^2 \\ &\le \sum_{i=1}^{N_r}\frac{w_i(t)}{W_t}\left(\sum_{j\in[K]-U_0(t)}\zeta_j^i(t)\hat{x}_j(t)^2\right) \qquad (27) \\ &\le \frac{1}{m(1-\gamma)}\sum_{j=1}^{K}\hat{x}_j(t) \qquad (28)\end{aligned}$$

where we use $E[X]^2 \le E[X^2]$ and $d_i(t) \le 1$ in (27), then (26) and $p_j(t)\hat{x}_j(t) \le 1$ in (28). Next, we bound terms with $\tilde{y}_i(t)$:

$$\begin{aligned}\sum_{i=1}^{N_r} q_i(t)\tilde{y}_i(t) &= \sum_{i=1}^{N_r}\frac{w_i(t)}{W_t} \\ &\quad\times\left(\sum_{j\in[K]-U_0(t)}\zeta_j^i(t)\hat{x}_j(t) + \frac{c}{\sqrt{KT}}\frac{\zeta_j^i(t)}{p_j(t)}\right) \\ &= \sum_{j\in[K]-U_0(t)}\sum_{i=1}^{N_r}\frac{w_i(t)\zeta_j^i(t)}{W_t}\left(\hat{x}_j(t) + \frac{c}{p_j(t)\sqrt{KT}}\right) \\ &\le \frac{1}{m(1-\gamma)}\left(\sum_{j\in[K]-U_0(t)} p_j(t)\hat{x}_j(t)\right) + \frac{c}{m(1-\gamma)}\sqrt{\frac{K}{T}}.\end{aligned} \tag{29}$$

$$\begin{aligned}\sum_{i=1}^{N_r} q_i(t)\tilde{y}_i(t)^2 &\le \sum_{i=1}^{N_r}\frac{w_i(t)}{W_t}\left(\hat{y}_i(t) + \frac{c\hat{u}_i(t)}{\sqrt{KT}}\right)^2 \\ &\le \frac{2}{m(1-\gamma)}\left(\sum_{j=1}^{K}\hat{x}_j(t)\right) \\ &\quad + \frac{2c^2}{KT}\frac{K}{m\gamma}\sum_{i=1}^{N_r}\frac{w_i(t)}{W_t}\sum_{j\in[K]-U_0(t)}\frac{\zeta_j^i(t)}{p_j(t)} \qquad (30) \\ &\le \frac{2}{m(1-\gamma)}\left(\sum_{j=1}^{K}\hat{x}_j(t)\right) + \frac{2c^2K}{Tm^2\gamma(1-\gamma)} \qquad (31)\end{aligned}$$

where we use $(a+b)^2 \le 2(a^2+b^2)$ and $\hat{u}_i(t) \le K/(\gamma m)$ in (30). By using (25), (29) and (31) and by noting that $p_j(t) = 1$ for $j \in U_0(t)$, we get:

$$\begin{aligned}&\frac{\eta}{m}\sum_{t=1}^{T}\sum_{r\in A^*}\boldsymbol{\zeta}^r(t)\cdot\hat{\boldsymbol{x}}(t) + \frac{\eta}{m}\frac{c}{\sqrt{KT}}\sum_{t=1}^{T}\sum_{r\in A^*}\hat{v}_r(t) - \ln\frac{W_1}{m} \\ &\quad + \frac{1}{m}\sum_{r\in A^*}\ln(w_r(1)) \le \frac{\eta}{m(1-\gamma)}G_{Exp4.MP} + \frac{\eta c\sqrt{KT}}{m(1-\gamma)} \\ &\quad + \frac{2\eta^2c^2K}{\gamma m^2(1-\gamma)} + \frac{2\eta^2}{m(1-\gamma)}\sum_{t=1}^{T}\sum_{j=1}^{K}\hat{x}_j(t)\end{aligned} \tag{32}$$

where $\boldsymbol{\zeta}^r(t)\cdot\hat{\boldsymbol{x}}(t) = \sum_{j=1}^{K}\zeta_j^r(t)\hat{x}_j(t)$. By dividing both sides with $\eta/(m(1-\gamma))$, and noting that $\sum_{t=1}^{T}\sum_{j=1}^{K}\hat{x}_j(t) \le (K/m)\hat{\Gamma}_{A^*}$, the statement in the theorem can be obtained. ■

Proof of Corollary 3.1: The proof consists of two steps. First, we prove an auxiliary result to help us to derive high-probability bound. Second, by using the auxiliary result and Theorem 3.1 we prove the statement in the corollary. In the first step, we use the beautiful martingale property given in Theorem 1 in [34]. Let us say, $Y_i(t) = y_i(t) - \hat{y}_i(t)$ for any fixed $i \in [N_r]$, where $y_i(t) = \sum_{j\in[K]-U_0(t)}\zeta_j^i(t)x_j(t)$ and $\hat{y}_i(t) = \sum_{j\in[K]-U_0(t)}\zeta_j^i(t)\hat{x}_j(t)$. We point out that

$$E[Y_i(t)] = 0, \qquad Y_i(t) \le 1, \qquad E[Y_i(t)^2] \le \hat{u}_i(t).$$

Let us define

$$V' \triangleq \frac{KT}{m} \text{ and } \sigma_i \triangleq \sqrt{\frac{m}{KT}}\sum_{t=1}^{T}\hat{u}_i(t) + \sqrt{\frac{KT}{m}}. \tag{33}$$

With the assumption of $\ln(N_r/\delta) \le (e-2)KT/m$, by Theorem 1 in [34], we can write

$$\mathbf{Pr}\left[\sum_{t=1}^{T}Y_i(t) \ge \sqrt{(e-2)\ln\frac{N_r}{\delta}}\sigma_i\right] \le \frac{\delta}{N_r} \tag{34}$$

for any $i \in [N_r]$. By applying union of events over the set $[N_r]$, and noting $(e-2) < 1$, we get

$$\mathbf{Pr}\left[\forall i \in [N_r] : \sum_{t=1}^{T} Y_i(t) \leq \sqrt{\ln \frac{N_r}{\delta}}\sigma_i\right] \geq 1-\delta \tag{35}$$

Since the event in (35) includes every $i \in [N_r]$, we can sum any m of them without changing the bound. Then we get

$$\mathbf{Pr}\left[\forall A \in \mathbf{C}([N_r], m) : \sum_{i\in A}\sum_{t=1}^{T} Y_i(t) \leq \sqrt{\ln \frac{N_r}{\delta}}\sum_{i\in A}\sigma_i\right] \geq 1-\delta. \tag{36}$$

Note that $\sum_{t=1}^{T} Y_i(t) = \sum_{t=1}^{T}\sum_{j\in[K]-U_0(t)} \zeta_j^i(t)(x_j(t) - \hat{x}_j(t))$. Since $x_j(t) = \hat{x}_j(t)$ for $j \in U_0(t)$, we can equivalently write

$$\mathbf{Pr}\left[\forall A \in \mathbf{C}([N_r], m) : \sum_{i\in A} G_i - \hat{G}_i \leq \sqrt{\ln \frac{N_r}{\delta}}\sum_{i\in A}\sigma_i\right] \geq 1-\delta. \tag{37}$$

Lastly, we point out that, according to (33),

$$\sqrt{\ln \frac{N_r}{\delta}}\sum_{i\in A}\sigma_i = \sqrt{m\ln \frac{N_r}{\delta}}\left(\frac{1}{\sqrt{KT}}\sum_{t=1}^{T}\sum_{i\in A}\hat{u}_i(t) + \sqrt{KT}\right).$$

In the second step, we first observe that our parameter selection satisfies the assumption (2) in Theorem 3.1. Then, we restate the result of the theorem when $w_i(1) = 1\ \forall i \in [N_r]$, and $\eta = m\gamma/(2K)$:

$$(1-2\gamma)\Gamma_{A^*} - (1-\gamma)\frac{2K}{\gamma} \times \ln \frac{N_r}{m} \leq G_{Exp4.MP} + c\sqrt{KT} + c^2. \tag{38}$$

By using (37), Γ_{A^*} from (9), and noting $c = \sqrt{m\ln(N_r/\delta)}$, we get $\mathbf{Pr}[G_{max} \leq \Gamma_{A^*} + c\sqrt{KT}] \geq 1-\delta$. Then,

$$G_{max} - G_{Exp4.MP} \leq \frac{2K}{\gamma}\ln\frac{N_r}{m} + 2\gamma G_{max} + 2c\sqrt{KT} + c^2 \tag{39}$$

holds with probability at least $1-\delta$. We point out that since γ has a small value, we ignore $(1-\gamma)$ and $(1-2\gamma)$ terms at the right hand side. In the end, by noting that $G_{max} \leq mT$ and using the given parameters in the corollary, the statement can be obtained. ■

APPENDIX C

Before the proof, we give one technical lemma:

Lemma C.1: For any $S > 1$ and $T > S$,

$$e^{1-S} \leq \left(1 - \frac{S-1}{T-1}\right)^{T-S}.$$

Proof: By taking the natural logarithms of both sides, we get $S - 1 \geq \ln(1 + \frac{S-1}{T-S})^{T-S}$. The fact that $\ln(1+\frac{\alpha}{x})^x \leq \alpha$ for $x \geq 0$ completes the proof. ■

Proof of Theorem 4.1: In this proof, we again begin with proving an auxiliary statement to derive high-probability regret bound. Fix $\mathbf{s}_T$ and say $X_{\mathbf{s}_T(t)} = x_{\mathbf{s}_T(t)} - \hat{x}_{\mathbf{s}_T(t)}$. Then,

$$E[X_{\mathbf{s}_T(t)}] = 0, \quad X_{\mathbf{s}_T(t)} \leq 1, \quad E[X^2_{\mathbf{s}_T(t)}] \leq 1/p_{\mathbf{s}_T(t)}.$$

We define

$$\Delta = \frac{\delta'}{K}\left(\frac{\beta}{K-1}\right)^{S-1}(1-\beta)^{T-S} \tag{40}$$

$$\Delta' = \frac{\delta'}{K}\left(\frac{\beta}{K}\right)^{S-1}(1-\beta)^{T-S} \tag{41}$$

where $\delta' \in [0,1]$ and $\beta = \frac{S-1}{T-1}$. We note that since S is the same for all the elements of the set Z, Δ and Δ' are arbitrary constants in [0, 1]. We use the same V' in (33) and define

$$\sigma_{\mathbf{s}_T} \triangleq \sqrt{\frac{m}{KT}}\sum_{t=1}^{T}\frac{1}{p_{\mathbf{s}_T(t)}} + \sqrt{\frac{KT}{m}}.$$

With the assumption of $\ln(1/\Delta') \leq (e-2)KT/m$, by Theorem 1 in [34], we can write

$$\mathbf{Pr}\left[\sum_{t=1}^{T} X_{\mathbf{s}_T(t)} \geq \sqrt{(e-2)\ln\frac{1}{\Delta'}}\sigma_{\mathbf{s}_T}\right] \leq \Delta. \tag{42}$$

Applying a union bound over Z and noting $\sum_Z \Delta \leq \delta$,

$$\mathbf{Pr}\left[\forall \mathbf{s}_T \in Z : \sum_{t=1}^{T} X_{\mathbf{s}_T(t)} \leq \sqrt{\ln\frac{1}{\Delta'}}\sigma_{\mathbf{s}_T}\right] \geq 1-\delta. \tag{43}$$

In order to get a clear expression, we aim to write $\ln\frac{1}{\Delta'}$ in terms of δ. Therefore, we aim to satisfy

$$\delta^{S-1} \leq \beta^{S-1}(1-\beta)^{T-S}. \tag{44}$$

By Lemma C.1, $\delta' \leq \frac{S-1}{e(T-1)}$ satisfies inequality (44). Then, by writing $\delta' = \frac{S-1}{e(T-1)}\delta$, where $\delta \in [0,1]$, and summing any m $\mathbf{s}_T$ in (43), we get

$$\mathbf{Pr}\left[\forall A \in C(Z,m) : \sum_{i\in A}\sum_{t=1}^{T} X_{\mathbf{s}_T(t)} \leq c\sigma_A\right] \geq 1 - \frac{S-1}{e(T-1)}\delta'. \tag{45}$$

where

$$\sigma_A = \left(\frac{1}{\sqrt{KT}}\sum_{t=1}^{T}\sum_{i\in A}\frac{1}{p_{\mathbf{s}_T(t)}} + \sqrt{KT}\right)$$

and c is one of the given parameters in the theorem.

In the second step, we first observe that our parameter selection satisfies the assumption (2) in Theorem 3.1. Second, we introduce a new notation $\{\mathbf{s}_T^1, \ldots, \mathbf{s}_T^m\} \in \mathbf{M}_T$, which means that $\mathbf{M}_T$ can be written as the combination of single arm sequences $\mathbf{s}_T^1, \ldots, \mathbf{s}_T^m$. We point out that the single arm sequences $\{\mathbf{s}_T^1, \ldots, \mathbf{s}_T^m\} \in \mathbf{M}_T^*$ can be selected the ones with the same

switching instants, and the same segment number. Then, their prior weights become

$$w_{\mathbf{s}_T}(1) = \frac{1}{K}\left(\frac{\beta}{K-1}\right)^{S-1}(1-\beta)^{T-S} \tag{46}$$

for $\{\mathbf{s}_T^1, \ldots, \mathbf{s}_T^m\} \in \mathbf{M}_T^*$. To have a bound w.r.t. $\mathbf{M}_T^*$, we define

$$\hat{\Gamma}_{\mathbf{M}_T^*} \triangleq \hat{G}_{\mathbf{M}_T^*} + \sum_{t=1}^{T}\sum_{\mathbf{s}_T \in \mathbf{M}_T^*} \frac{c}{p_{\mathbf{s}_T(t)}\sqrt{KT}}. \tag{47}$$

We note that $\hat{\Gamma}_{\mathbf{M}_T^*} \leq \hat{\Gamma}_{A^*}$. We also point out that $W_1 = 1$ by our prior scheme. Then by using $\eta = \frac{m\gamma}{2K}$, (47), and (46) in inequality (10), we can write

$$(1-2\gamma)\left(\hat{G}_{\mathbf{M}_T^*} + \sum_{t=1}^{T}\sum_{\mathbf{s}_T \in \mathbf{M}_T^*} \frac{c}{p_{\mathbf{s}_T(t)}\sqrt{KT}} + c\sqrt{KT}\right) - 2c\sqrt{KT} - \frac{2K}{\gamma}\ln\frac{K^S}{\beta^{S-1}(1-\beta)^{T-S}} + c^2 \leq G_{Exp4.MP} \tag{48}$$

where we use $\gamma < 0.5$, $w_{\mathbf{s}_T}(1) \leq 1$, and

$$w_{\mathbf{s}_T}(1) \geq \frac{1}{K}\left(\frac{\beta}{K}\right)^{S-1}(1-\beta)^{T-S} \text{ for } \{\mathbf{s}_T^1, \ldots, \mathbf{s}_T^m\} \in \mathbf{M}_T^*. \tag{49}$$

By (45) and Lemma C.1, if we use the given c and β values,

$$(1-2\gamma)G_{\mathbf{M}_T^*} - G_{Exp4.MP} \leq \frac{2KS}{\gamma}\ln\frac{eK(T-1)}{S-1} + 2\sqrt{mKST\ln\left(\frac{eK(T-1)}{(S-1)\delta}\right)} + mS\ln\left(\frac{eK(T-1)}{(S-1)\delta}\right) \tag{50}$$

holds with probability at least $1 - \frac{S-1}{e(T-1)}\delta$. By noting $G_{\mathbf{M}_T^*} \leq mT$ and using the given γ value, the statement in the theorem can be obtained. ■

Proof of Theorem 4.2: We use $\hat{n}_{\mathbf{s}_t(t)}$, $\hat{N}_{\mathbf{s}_t(1:t-1)}$ defined in Section IV, and

$$\hat{n}_j(t) = \left(\hat{x}_j(t) + \frac{c}{p_j(t)\sqrt{KT}}\right)\mathbb{1}_{j\notin U_0(t)} \quad \text{for } j \in [K]. \tag{51}$$

Let $w_{\mathbf{s}_t}$ be the weight of an arbitrary sequence $\mathbf{s}_t$, r be an arbitrary arm, and $\overline{v}_r(t)$ be the weight of the arm r in the hypothetical *Exp4.MP* run at round t, given by

$$\overline{v}_r(t) = \sum_{\mathbf{s}_t(t)=r} \frac{w_{\mathbf{s}_t}}{\sum\limits_{\mathbf{s}_t} w_{\mathbf{s}_t}}. \tag{52}$$

In the proof, we use mathematical induction to show $\overline{v}_r(t) = v_r(t)$ for any $r \in [K]$ and $t = 1, 2, \ldots, T$. The proof begins with noting $v_j(1) = w_{\mathbf{s}_1} = 1/K$. Then,

$$\begin{aligned}\overline{v}_r(t) &= \sum_{\mathbf{s}_t(t)=r} \frac{w_{\mathbf{s}_t}}{\sum\limits_{\mathbf{s}_t} w_{\mathbf{s}_t}} = \sum_{\mathbf{s}_t(t)=r} \frac{\pi_{\mathbf{s}_t}\exp(\eta\hat{N}_{\mathbf{s}_t(1:t-1)})}{\sum\limits_{\mathbf{s}_t} w_{\mathbf{s}_t}} \\ &= \sum_{\mathbf{s}_t(t)=r} \frac{\pi(\mathbf{s}_t|\mathbf{s}_t(t-1))\pi_{\mathbf{s}_t(1:t-1)}\exp(\eta\hat{N}_{\mathbf{s}_t(1:t-1)})}{\sum\limits_{\mathbf{s}_t} w_{\mathbf{s}_t}} \\ &= \sum_{\mathbf{s}_t(1:t-1)} \frac{w_{\mathbf{s}_t(1:t-1)}\exp(\eta\hat{n}_{\mathbf{s}_t(t-1)})\pi(\mathbf{s}_t|\mathbf{s}_t(t-1))}{\sum\limits_{\mathbf{s}_t} w_{\mathbf{s}_t}}\end{aligned}$$

$$\text{Assuming } \sum\nolimits_{\mathbf{s}_t(t-1)=j} w_{\mathbf{s}_t(1:t-1)} \propto v_j(t-1) \quad \text{for } j \in [K] \tag{53}$$

$$\begin{aligned}&= \sum_{j=1}^{K} \frac{v_j(t-1)\exp(\eta\hat{n}_j(t-1))\pi(\mathbf{s}_t|\mathbf{s}_t(t-1))}{\sum_{l\in[K]} v_l(t-1)\exp(\eta\hat{n}_l(t-1))} \\ &= \sum_{j=1}^{K} \frac{\tilde{v}_j(t-1)\left(\frac{\beta}{K-1}\mathbb{1}_{j\neq r} + (1-\beta)\mathbb{1}_{j=r}\right)}{\sum_{l\in[K]}\tilde{v}(t-1)} = v_r(t).\end{aligned} \tag{54}$$

Since our assumption in (53) holds for $t = 1$, by (54) it holds for all t. Then, the theorem holds for all t as well. ■

REFERENCES

[1] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. New York, NY, USA: Cambridge Univ. Press, 2006.

[2] H. S. Chang, J. Hu, M. C. Fu, and S. I. Marcus, "Adaptive adversarial multi-armed bandit approach to two-person zero-sum Markov games," *IEEE Trans. Autom. Control*, vol. 55, no. 2, pp. 463–468, Feb. 2010.

[3] K. Gokcesu and S. S. Kozat, "An online minimax optimal algorithm for adversarial multiarmed bandit problem," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5565–5580, Nov. 2018.

[4] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3700–3714, Jul. 2015.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Proc. IEEE 36th Annu. Found. Comput. Sci.*, Oct. 1995, pp. 322–331.

[6] C. Tekin and M. Liu, "Online learning in opportunistic spectrum access: A restless bandit approach," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2462–2470.

[7] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5667–5681, Nov. 2010.

[8] K. Wang and L. Chen, "On optimality of myopic policy for restless multi-armed bandit problem: An axiomatic approach," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 300–309, Jan. 2012.

[9] Y. Gai and B. Krishnamachari, "Distributed stochastic online learning policies for opportunistic spectrum access," *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6184–6193, Dec. 2014.

[10] S. Vakili, K. Liu, and Q. Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 759–767, Oct. 2013.

[11] K. Liu, Q. Zhao, and B. Krishnamachari, "Dynamic multichannel access with imperfect channel state detection," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2795–2808, May 2010.

[12] W. M. Koolen, M. K. Warmuth, and D. Adamskiy, "Open problem: Online sabotaged shortest path," in *Proc. 28th Conf. Learn. Theory*, 2015, pp. 1764–1766.

[13] A. Nakamura and N. Abe, "Improvements to the linear programming based scheduling of web advertisements," *Electron. Commerce Res.*, vol. 5, no. 1, pp. 75–98, Jan. 2005.

[14] A. Heydari and S. N. Balakrishnan, "Optimal switching and control of nonlinear switching systems using approximate dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1106–1117, Jun. 2014.

[15] X. Liu, H. Su, and M. Z. Q. Chen, "A switching approach to designing finite-time synchronization controllers of coupled neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 471–482, Feb. 2016.

[16] P. Lim, C. K. Goh, K. C. Tan, and P. Dutta, "Multimodal degradation prognostics based on switching Kalman filter ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 136–148, Jan. 2017.

[17] A. Heydari, "Feedback solution to optimal switching problems with switching cost," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2009–2019, Oct. 2016.

[18] F. M. J. Willems, "Coding for a binary independent piecewise-identically-distributed source," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2210–2217, Nov. 1996.

[19] N. Merhav, "On the minimum description length principle for sources with piecewise constant parameters," *IEEE Trans. Inf. Theory*, vol. 39, no. 6, pp. 1962–1967, Nov. 1993.

[20] G. I. Shamir and N. Merhav, "Low-complexity sequential lossless coding for piecewise-stationary memoryless sources," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1498–1519, Jul. 1999.

[21] P. Auer and M. K. Warmuth, "Tracking the best disjunction," *Mach. Learn.*, vol. 32, no. 2, pp. 127–150, Aug. 1998.

[22] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Mach. Learn.*, vol. 32, no. 2, pp. 151–178, Aug. 1998.

[23] V. Vovk, "Derandomizing stochastic prediction strategies," in *Proc. 10th Annu. Conf. Comput. Learn. Theory*, 1997, pp. 32–44.

[24] S. Kale, L. Reyzin, and R. E. Schapire, "Non-stochastic bandit slate problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1054–1062.

[25] G. Neu and G. Bartók, "Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 5355–5375, Jan. 2016.

[26] T. Uchiya, A. Nakamura, and M. Kudo, "Algorithms for adversarial bandit problems with multiple plays," in *Algorithmic Learning Theory*. Berlin, Germany: Springer, 2010, pp. 375–389.

[27] A. György, T. Linder, G. Lugosi, and G. Ottucsák, "The on-line shortest path problem under partial monitoring," *J. Mach. Learn. Res.*, vol. 8, pp. 2369–2403, Dec. 2007.

[28] E. Takimoto and M. K. Warmuth, "Path kernels and multiplicative updates," *J. Mach. Learn. Res.*, vol. 4, pp. 773–818, Dec. 2003.

[29] J. Audibert, S. Bubeck, and G. Lugosi, "Regret in online combinatorial optimization," *Math. Oper. Res.*, vol. 39, no. 1, pp. 31–45, 2014.

[30] V. Dani, T. P Hayes, and S. Kakade, "The price of bandit information for online optimization," in *Advances in Neural Information Processing Systems 20*, J. C. Platt and D. Koller and Y. Singer and S. T. Roweis, Eds. Red Hook, NY, USA: Curran Associates, 2008, pp. 345–352.

[31] J. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the dark: An efficient algorithm for bandit linear optimization," in *Proc. 21st Annu. Conf. Learn. Theory*, 2008, pp. 263–274.

[32] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1404–1422, Sep. 2012.

[33] D. Suehiro, K. Hatano, S. Kijima, E. Takimoto, and K. Nagano, "Online prediction under submodular constraints," in *Algorithmic Learning Theory*. Berlin, Germany: Springer, 2012, pp. 260–274.

[34] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. Schapire, "Contextual bandit algorithms with supervised learning guarantees," in *Proc 14th Int. Conf. Artif. Intell. Statist.*, Apr. 2011, vol. 15, pp. 19–26.

[35] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *J. ACM*, vol. 53, no. 3, pp. 324–360, May 2006. [Online]. Available: http://doi.acm.org/10.1145/1147954.1147956

[36] Y. Seldin and G. Lugosi, "A lower bound for multi-armed bandits with expert advice," in *Proc. 13th Eur. Workshop Reinforcement Learn.*, to be published.

[37] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.

[38] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[39] G. Neu, "Explore no more: Improved high-probability regret bounds for non-stochastic bandits," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3168–3176.

[40] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Proc. Eur. Conf. Mach. Learn. Learn.*, 2005, pp. 437–448.