Staged Multi-armed Bandits

Cem Tekin, Member, IEEE, Kenneth Moon, Mihaela van der Schaar, Fellow, IEEE

Abstract—In this paper, we introduce a new class of reinforcement learning methods referred to as staged multi-armed bandits (S-MAB). In S-MAB the learner proceeds in rounds, each composed of several stages, in which it chooses an action and observes a feedback signal. Moreover, in each stage, it can take a special action, called the stop action, that ends the current round. After the stop action is taken, the learner collects a terminal reward, and observes the costs and terminal rewards associated with each stage of the round. The goal of the learner is to maximize its cumulative gain (i.e., the terminal reward minus costs) over all rounds by learning to choose the best sequence of actions based on the feedback it gets about these actions. First, we define an oracle benchmark, which sequentially selects the actions that maximize the expected immediate gain. This benchmark is known to be approximately optimal when the reward sequence associated with the selected actions is adaptive submodular. Then, we propose our online learning algorithm, named Feedback Adaptive Learning (FAL), and show (i) a problem-independent confidence bound on the performance of the selected actions, (ii) a finite regret bound that holds with high probability, (iii) a logarithmic bound on the expected regret. S-MAB can be used to model numerous applications, ranging from personalized medical screening to personalized web-based education, where the learner does not obtain rewards after each action, but only after sequences of actions are taken, intermediate feedbacks are observed, and a final decision is made based on which a terminal reward is obtained. Our illustrative results show that S-MAB can be used to model medical screening, and FAL outperforms the existing approaches in this field.

Index Terms—Staged decision making, online learning, multi-armed bandits, reinforcement learning, regret, greedy benchmark, submodularity, medical diagnosis.

I. INTRODUCTION

Many applications involving sequential decision making under uncertainty can be formalized as multi-armed bandits (MAB): clinical trials [1], dynamic spectrum access [2][4], cognitive compressive sensing [5], recommender systems [6] and web advertising [7], [8] etc. A common assumption in all these problems is that each decision step involves taking an action after which a reward is observed. (MAB extensions also allow for settings in which the rewards are missing, delayed or erroneous or multiple actions are taken simultaneously.) However, in numerous applications such as humanoid robot locomotion [9], online education [10], [11] and healthcare [12], each decision step involves taking multiple actions (e.g., selecting various educational materials to display or various tests to be performed) whose reward is only revealed after the entire action sequence is completed and a decision is made to stop the action sequence and (possibly) take a final action (e.g., take a final exam, perform a surgery or finalize a treatment decision).

For instance, in personalized online education, a sequence of materials can be used to teach or remind students the key concepts of a course subject. While the final exam is used as a benchmark to evaluate the overall effectiveness of the given sequence of teaching materials, a sequence of intermediate feedbacks like students’ performance on quizzes, homework grades, etc., can be used to guide the teaching examples online. Similarly, in personalized healthcare, a sequence of treatments is given to a patient over a period of time. The overall effectiveness of the treatment plan depends on the given treatments as well as their order [12]. Moreover, the patient can be monitored during the course of the treatment which yields a sequence of feedbacks about the selected treatments, while the final outcome is only available after the entire sequence of treatments is completed.

In conclusion, in such sequential decision making problems the order of the taken actions matters. Moreover, the feedback available after each taken action drives the action selection process. We call online learning problems exhibiting the aforementioned properties staged multi-armed bandits (S-MAB). An S-MAB problem proceeds in rounds \( r = 1, 2, \ldots \) composed of multiple stages, in which the learner selects actions sequentially in stages, one after another, with each action belonging to the action set \( \mathcal{A} \). After each taken action \( a \in \mathcal{A} \), a feedback \( f \in \mathcal{F} \) is observed about the taken action. Based on all its previous observations in that round, the learner either decides to continue to the next stage by selecting another action or selecting a stop action which ends the current round and starts the next round. Hence, the number of stages in each round is a decision variable. The rewards and losses in a round are observed only after the stop action is taken. The goal of the learner is to maximize its total expected gain (i.e., the terminal reward minus costs) over all rounds by learning to choose the best action sequence given the feedback. An illustration that shows the order of stages, gains and rounds is given in Fig. [1]

The contributions are summarized as follows:

- We propose a new online learning model called S-MAB. S-MAB covers other learning models including the online adaptive submodular maximization problem [13] as special cases.
- For the offline S-MAB problem, we propose a benchmark which selects the next action by maximizing the immediate gain.
- We propose the Feedback Adaptive Learning (FAL) algorithm as a solution to the S-MAB problem.
- We prove (i) a problem-independent confidence bound on the gains of the sequence of actions chosen by FAL, (ii) a finite regret bound that holds with high probability,
(iii) a logarithmic in the number of rounds bound on the expected regret. 

- We perform experiments on FAL by formulating a medical screening scenario as an S-MAB, and compare the performance of FAL with existing methods.

Rest of the paper is organized as follows. Problem formulation, and the definitions of the benchmark and the regret are given in Section II. The learning algorithm is introduced in Section III. Regret analysis of this algorithm is provided in Section IV. Illustrative results on a medical screening application are given in Section V. Related work is given in Section VI, followed by the concluding remarks given in Section VII.

II. PROBLEM FORMULATION

A. Notation

Unless noted otherwise, sets are denoted by calligraphic letters, vectors are denoted by boldface letters and random variables are denoted by capital letters. We use \(|\cdot|\) to denote the cardinality of a set or sequence. Cardinality of a set \(E\) is denoted by \(|E|\). The set of positive integers up to integer \(t\) is denoted by \([t]\). \(E_p[\cdot]\) denotes the expectation with respect to probability distribution \(p\). \(\Delta (E)\) denotes the indicator function of event \(E\) which is one if \(E\) is true and 0 otherwise. For a set \(E\), \(\Delta(E)\) denotes the set of probability distributions over \(E\). Unless noted otherwise, all inequalities that involve random variables hold with probability one.

B. Problem Description

The learner proceeds in discrete rounds indexed by \(t\). Each round is composed of multiple stages indexed by \(t\). Each stage corresponds to a decision epoch in which the learner can choose an action from a finite set of actions denoted by \(\mathcal{A}\). There are two types of actions in \(\mathcal{A}\): (i) continuation actions which move the learner to the next stage and allow it to acquire more information (feedback), (ii) a terminal action (also named stop) which moves the learner to the next stage and allows it to acquire information (feedback), and (iii) a terminal reward. The set of possible actions is denoted by \(\mathcal{A}\).

The maximum number of stages in a round is \(l_{\text{max}} < \infty\), which means that the stop action must be selected in at most \(l_{\text{max}}\) stages. After an action \(a\) is selected in a stage \(t\), the learner observes a feedback \(f \in \mathcal{F}\) before moving to the next stage, where \(\mathcal{F}\) denotes the set of all feedbacks.

Let \(a_{[t]} := (a_1, a_2, \ldots, a_t)\) denote a length \(t\) sequence of continuation actions and \(f_{[t]} := (f_1, f_2, \ldots, f_t)\) denote a length \(t\) sequence of feedbacks. Let \(\mathcal{A}^t := \prod_{i=1}^t \mathcal{A}\) denote the set of length \(t\) sequences of continuation actions and \(\mathcal{F}^t := \prod_{i=1}^t \mathcal{F}\) denote the set of length \(t\) sequences of feedbacks. Set of all continuation action sequences is denoted by \(\mathcal{A}^{\text{all}} := \bigcup_{t=1}^{l_{\text{max}}-1} \mathcal{A}^t\) and the set of all feedback sequences is denoted by \(\mathcal{F}^{\text{all}} := \bigcup_{t=1}^{l_{\text{max}}-1} \mathcal{F}^t\). At each stage, the system is in one of the finitely many states, where the set of states is denoted by \(\mathcal{X}\).

When action \(a\) is chosen in stage \(t\), the feedback it generates depends on the state of the system in that stage. Specifically, we assume that \(f_t \sim p_{t,x,a} \in \Delta(\mathcal{F})\), where \(p_{t,x,a}\) denotes the probability distribution of the feedback given the state-action triplet \((t, x, a)\). Let \(\phi_t : \mathcal{X} \times \mathcal{A} \times \mathcal{F} \to \mathcal{X}\) be the state mapping which encodes every state-action-feedback triplet to one of the states in \(\mathcal{X}\). Since the feedback is random, the next state is not a deterministic function of the previous state. Moreover, the state transition probabilities are stage dependent.

The expected cost of action \(a\) in stage \(t\) when the state is \(x\) is given by \(c_{t,x,a} \in [0, c_{\text{max}}]\). The expected terminal reward in stage \(t\) when the state is \(x\) is given by \(r_{t,x} \in [0, r_{\text{max}}]\). The ex-ante terminal reward of the triplet \((t, x, a)\) is defined as

\[
y_{t,x,a} := E_{f_t \sim p_{t,x,a}}[r_{t+1, \phi_t(x,a,f_t)}]
\]

which gives the expected terminal reward of stopping at stage \(t + 1\) after choosing action \(a\) in stage \(t\) and before observing the feedback \(f_t\). For the stop action the cost is always zero and

\[
y_{t,x,\text{stop}} = r_{t,x} \quad \forall t \in [l_{\text{max}}], \forall x \in \mathcal{X}.
\]

The gain of an action \(a \in \mathcal{A}\) in stage \(t\) when the state is \(x\) is defined as

\[
g_{t,x,a} := y_{t,x,a} - c_{t,x,a}.
\]

At each round \(r\), the learner chooses a sequence of actions \(a^r := (a^r_1, \ldots, a^r_{T_r})\), observes a sequence of feedbacks \(f^r := (f^r_1, \ldots, f^r_{T_r-1})\) and encounters a sequence of states \(x^r := (x^r_1, \ldots, x^r_{T_r})\), where \(T_r\) is the stage in which the stop action is taken. Since no feedback is present in the first stage, the state in the first stage is a constant denoted by \(x^r_1 = 0\).

Moreover, after the stop action is taken, the learner observes costs of the selected actions \(C_{t}^r = c_{t,x^r_t,a^r_t} + \eta_t^r\) for \(t \in [T_r - 1]\) and the terminal rewards \(R_1^r = r_{1,x^r_1} + \kappa^r_1\) for \(t \in [T_r]\), where \(\eta_t^r\) and \(\kappa^r_t\) are random variables that satisfy Assumption 1.

When the round is clear from the context, we will drop the superscripts from the notation.

Assumption 1. \(\eta_t^r\) and \(\kappa_t^r\) are zero mean \(\sigma\)-sub-Gaussian random variables (see Definition 2) that are independent of \((x^r_1, \ldots, x^r_{t-1}, f^r_{t-1}, \kappa^r_{t-1}, \eta^r_{t-1})\) and the random variables in all previous rounds.

Definition 1. Given \(\sigma > 0\), a random variable \(Y\) is \(\sigma\)-sub-Gaussian if

\[
\forall \lambda \in \mathbb{R} \quad E[e^{\lambda Y}] \leq \exp\left(\frac{\lambda^2 \sigma^2}{2}\right).
\]

We assume that the learner knows the state mapping and can compute the state of the system at any stage by using the actions taken and feedbacks observed in the previous stages. The learner does not know the feedback, cost and terminal reward distributions. The goal of the learner is to maximize its cumulative gain over the rounds by repeated interaction with the system.

1Hence, our definition of state is more general than the definition of state used in reinforcement learning in MDPs [14, 15], which is assumed to be time homogeneous.
Fig. 1: The S-MAB problem. $x_t^\rho$ is the state observed, $a_t^\rho$ is the action selected and $f_t^\rho$ is the feedback observed in the stage $t$ of round $\rho$. $C_t^\rho$ is the cost of selecting action $a_t^\rho$ and $R_t^\rho$ is the terminal reward in stage $t$ of round $\rho$. $T_\rho$ is the stage in which the learner selects the stop action after which the costs and terminal rewards are revealed.

C. The Benchmark

Since the number of possible action and feedback sequences is exponential in $l_{\text{max}}$, it is very inefficient to learn the best action sequence by separately estimating the expected gain of each action sequence $a \in A^{\text{all}}$. In this section we propose a benchmark (given in Algorithm 1) whose action selection strategy can be learned quickly.

![Algorithm 1 Benchmark](image)

The benchmark incrementally selects the next action based on the past sequence of feedbacks and actions. If the stop action is not taken up to stage $t$, the benchmark selects its action in stage $t$ according to the following rule: Assume that the state in stage $t$ is $x_t$. If $g_t{x_t,a} \geq g_t{x_t,\text{stop}}$ for all $a \in A$ (which implies that $r_t{x_t,a} \geq r_t{x_t,\text{stop}}$ for all $a \in A$), then the benchmark selects the stop action in stage $t$. Otherwise, it decides to continue for one more stage by selecting one of the actions $a \in A$ which maximizes $g_t{x_t,a}$.

Let $a^* := (a_1^*, \ldots, a_{T_\rho}^*)$ be the action sequence selected, $x^* := (x_1^*, \ldots, x_{T_\rho}^*)$ be the state sequence, $C^* := (C_1^*, \ldots, C_{T_\rho}^*)$ be the cost sequence and $R^* := (R_1^*, \ldots, R_{T_\rho}^*)$ be the terminal reward sequence observed, and $R_{T_\rho}^{\text{stop}}$ be the terminal reward collected by the benchmark in round $\rho$, where $T_\rho$ is the stage in which the stop action is selected. The cumulative expected gain, i.e., the expected terminal reward minus costs, of the benchmark in the first $n$ rounds is equal to

$$RW_B(n) := E \left[ \sum_{\rho=1}^{n} \left( R_{T_\rho}^{\text{stop}} - \sum_{t=1}^{T_\rho-1} C_t^{\rho} \right) \right].$$

D. Performance of the Benchmark for Special Cases

In this section we evaluate the performance of the benchmark under some special cases.

The benchmark vs. the best fixed sequence: Although the benchmark may not always select the optimal action sequence,
it can perform better than the best fixed action sequence that is not adapted based on the observed feedbacks. This is illustrated in the following example: Consider a medical application where \( A = \{a, b\} \) is the set of treatments, \( F = \{-1, 1\} \) is the outcome of an individual treatment, \( c_a = c_b = 1 \) are the costs of administering treatments \( a \) and \( b \), and \( l_{\max} = 3 \). Assume that the state in stage \( t \) is defined as the sequence of all actions chosen and feedbacks observed prior to stage \( t \). For the simplicity of notation, for length \( t - 1 \) sequence of actions and feedbacks \( a \) and \( f \), let \( r_{a,f} := r_{a}(a,f) \).

Assume that the expected patient recovery scores are given as follows: \( r_{a,a}(f_1,f_2) = 0 \), \( r_{b,b}(f_1,f_2) = 0 \) and \( r_{a,b}(f_1,f_2) = 0 \) for any \( f_1, f_2 \in F \); \( r_{a,-1} = 0 \), \( r_{b,-1} = 0 \), \( r_{a,1} = 12 \), \( r_{b,1} = 6 \), \( r_{a,b}(1,1) = 13 \), \( r_{a,b}(1,-1) = 12 \), \( r_{a,b}(-1,1) = 10 \), \( r_{a,b}(-1,-1) = 9 \). Let \( \Pr(f|a) \) denote the probability that feedback sequence \( f \) is observed for the action sequence \( a \). Assume that we have \( \Pr(1|a) = 0.5 \), \( \Pr(1|b) = 0.5 \), \( \Pr((-1,-1)|a,b) = 0.3 \), \( \Pr((-1,1)|a,b) = 0.2 \), \( \Pr((1,1)|a,b) = 0.4 \), \( \Pr((1,-1)|a,b) = 0.1 \).

The benchmark selects \( a \) in the first stage. Then, if feedback is \(-1\) it selects \( b \) in the second stage before selecting the \textit{stop} action. Else, it selects the \textit{stop} action after the first stage. Hence, the expected reward of the benchmark in a round is

\[
RW_B(1) = 0.5 \times (12 - 1) + 0.3 \times (9 - 2) + 0.2 \times (10 - 2) = 9.2
\]

The best fixed action sequence is \( (a, b) \) which gives an expected reward that is equal to \( 0.3 \times (9 - 2) + 0.2 \times (10 - 2) + 0.4 \times (13 - 2) + 0.1 \times (12 - 2) = 9.1 \).

\subsection*{Approximate optimality of the benchmark in adaptive monotone submodular S-MAB problems}

Consider a special case of the S-MAB problem in which: (i) action selection costs are set to zero, i.e., \( c_{t,x,a} = 0 \), (ii) \( l_{\max} \leq S_A \), (iii) if an action is selected in stage \( t \) it cannot be selected in the future stages, (iv) a state \( s_a \in \{-1, 1\} \) is associated with each action \( a \in A \), and the joint state vector \( s = \{s_a\}_{a \in A} \in \{-1, 1\}^{S_A} \) is sampled independently from some fixed distribution at the beginning of each round. These assumptions reduce our problem to the adaptive submodular maximization problem studied in [13, 17].

In [13], the function to be maximized is given as \( h : 2^A \times \{-1, 1\}^{S_A} \to \mathbb{R} \), where \( 2^A \) denotes the power set of \( A \). Space of observations are defined as \( \mathcal{Y} := \{-1, 0, 1\}^{S_A} \). For an observation vector \( y \in \mathcal{Y} \), \( y_a = 0 \) implies that action \( a \) is not selected, and \( y_a = i \), \( i \in \{-1, 1\} \) implies that action \( a \) is selected and \( y_a = s_a \) is observed. Let \( dom(y) \) denote the set and \( l(y) \) denote the number of actions selected according to observation vector \( y \). They define the greedy policy for maximizing \( h \) as \( \pi^g \), such that given an observation vector \( y \), it selects the action \( a \) that maximizes \( h(a, y) \).

The (random) regret of a learning algorithm which selects the action sequence \( a^\rho \) and observes the feedback sequence \( f^\rho \) in round \( \rho \) with respect to the benchmark in the first \( n \) rounds is given by

\[
R(n) := \left( \sum_{\rho=1}^{n} \left( \sum_{\rho=1}^{T_\rho} - \sum_{i=1}^{T_n-1} c_{t,x,a} \right) \right)

- \left( \sum_{\rho=1}^{n} \left( \sum_{\rho=1}^{T_\rho} - \sum_{i=1}^{T_n-1} c_{t,x,a} \right) \right).
\]

(4)

When we take expectation of (4) over all sources of randomness, we obtain the expected regret, which is equivalent to

\[
E[R(n)] = RW_B(n) - \mathbb{E} \left[ \sum_{\rho=1}^{n} \left( \sum_{i=1}^{T_n-1} C_i^\rho \right) \right].
\]

(5)

Any algorithm whose expected regret increases at most sublinearly, i.e., \( E[R(n)] = O(n^{1+ \gamma}) \), \( 0 < \gamma < 1 \), in the number of rounds will converge in terms of the average reward to
the average reward of the benchmark as \( n \to \infty \). In the next section we will propose an algorithm whose expected regret increases only logarithmically in the number of rounds and polynomially in the number of stages.

### III. A LEARNING ALGORITHM FOR THE S-MAB PROBLEM

In this section we propose Feedback Adaptive Learning (FAL) (pseudocode given in Fig. 2), which learns the sequence of actions to select based on the observed feedbacks to the actions taken in previous stages of a round (as shown in Fig. 1). In order to minimize the regret given in (5), FAL balances exploration and exploitation when selecting the actions.

FAL keeps the sample mean estimates \( \hat{g}^p_{t,x,a} \) of the gains \( g^p_{t,x,a} \) of the actions \( a \in A \) and the sample mean estimates \( \hat{r}^p_{t,x} \) of the terminal rewards \( r^p_{t,x} \) for all stage-state pairs \( (t,x) \).

Using the definition of the gain for the \( K \) of round \( t \) in (3), in order to minimize the regret given in (5), FAL balances polynomially in the number of stages.

The set of optimal actions for \( t \), \( x \), and \( a \) is given in (6) and (7) for the stage-state pair \( (t,x,a) \) which counts the number of times that it will take an action that is different from the best action.

After selecting the action in stage \( t \), FAL observes the feedback \( f^p_t \sim p_{t,x,a},a_t \), which is then used to calculate the next state as

\[ x^p_{t+1} = \phi_t(x^p_t, a_t, f^p_t). \]

This procedure repeats until FAL takes the \textit{stop} action, which will eventually happen since the number of stages is bounded by \( l_{\text{max}} \). This way the length of the sequence of selected actions is adapted based on the sequence of received feedbacks and costs of taking the actions. After round \( \rho \) ends, FAL observes the costs \( C^p_t \), \( t \in [T_\rho - 1] \) and the terminal rewards \( R^p_{t} \), \( t \in [T_\rho] \). Using these values, it updates the values of the sample mean terminal rewards, gains and the counters before round \( \rho + 1 \) starts (line 16 of FAL). FAL reaches its objective of maximizing the expected cumulative gain by capturing the tradeoff between the rewards and the costs of selecting actions. Performance of FAL is analysed in the following section.

#### IV. PERFORMANCE BOUNDS FOR FAL

We will bound the regret of FAL by bounding the number of times that it will take an action that is different from the action selected by the benchmark.

Let \( \hat{g}_{t,x,a} = \max_{a \in A} g_{t,x,a} \) be the gain of the best action and \( \Delta_{t,x,a} = g_{t,x,a} - \hat{g}_{t,x,a} \) be the suboptimality gap of action \( a \) for the stage-state pair \( (t,x) \). The set of optimal actions for

\[ \text{Algorithm 2 Feedback Adaptive Learning (FAL)} \]

\[ \text{Require: } A, X, l_{\text{max}}, \sigma, \delta \]

\[ \text{Initialize counters: } N_{t,x} = 0, N_{t,x,a} = 0, \forall t \in [l_{\text{max}}], \forall x \in X, \forall a \in A, \text{ and } \rho = 1. \]

\[ \text{Initialize estimates: } \hat{r}_{t,x,a} = 0, \forall t \in [l_{\text{max}}], \forall x \in X, \forall a \in A. \]

1: while \( \rho \geq 1 \) do

2: \( \rho = 1 \), \( x_1 = \emptyset \)

3: while \( t \in [l_{\text{max}}] \) do

4: Calculate UCBs: \( u_{t,x,a} = \hat{g}_{t,x,a} + \text{conf}_{t,x,a}, \forall a \in A \), where \( \text{conf}_{t,x,a} \) is given in (6) and (7)

5: if \( \{ \text{stop} \in \arg \max_{a \in A} u_{t,x,a} \} \cap \{ t = l_{\text{max}} \} \) then

6: \( a_t = \text{stop}, T_\rho = t // \text{BREAK} \)

7: else

8: Select \( a_t \) from \( \arg \max_{a \in A} u_{t,x,a} \)

9: end if

10: Observe feedback \( f_t \)

11: Set \( x_{t+1} = \phi_t(x_t, a_t, f_t) \)

12: \( t = t + 1 \)

13: end while

14: Observe the costs \( C^p_t \), \( t \in [T_\rho - 1] \) and the terminal rewards \( R^p_t \), \( t \in [T_\rho] \)

15: Collect terminal reward \( R^p_{T_\rho} \)

16: Update: \( \hat{r}_{t,x} = \frac{N_{t,x} \hat{r}_{t,x} + R^p_{t}I(x_t = x)}{N_{t,x} + I(x_t = x)} \)

\( N_{t,x} = N_{t,x} + I(x_t = x) \)

for \( t \in [T_\rho] \) and \( x \in X \), and

\[ \hat{g}_{t,x,a} = \frac{N_{t,x,a} \hat{g}_{t,x,a} + (R^p_{t+1} - C^p_t)I(x_t = x, a_t = a)}{N_{t,x,a} + I(x_t = x, a_t = a)} \]

\( N_{t,x,a} = N_{t,x,a} + I(x_t = x, a_t = a) \)

for \( t \in [T_\rho - 1] \), \( x \in X \) and \( a \in A \)

17: \( \rho = \rho + 1 \)

18: end while

\[ \text{end while} \]
stage-state pair \((t,x)\) is given by \(O_{t,x} := \{a \in \bar{A} : \Delta_{t,x,a} = 0\}\). We assume the following condition on the gains of the actions.

**Assumption 2.** For any stage-state pair \((t,x)\): (i) \(\text{stop} \in O_{t,x} \Rightarrow |O_{t,x}| = 1\), (ii) \(|O_{t,x}| > 1 \Rightarrow O_{t,x} \subset \bar{A}\).

Assumption 2 implies that \(O_{t,x}\) cannot include both the \text{stop} action and another action in \(A\). This assumption is required for our regret analysis. If \(O_{t,x}\) includes both the \text{stop} action and another action in \(A\), then any learning algorithm may incur linear regret. The reason for this is that the benchmark will always choose the \text{stop} action in this case, whereas the learner may take the other action more than it takes the \text{stop} action due to the fluctuations of the sample mean gains around their expected values. To circumvent this effect, the learner can add a small positive bias \(\epsilon > 0\) to the gain of the \text{stop} action. If this bias is small enough such that the \text{stop} action remains suboptimal for any stage-state pair \((t,x)\) in which the \text{stop} action was suboptimal, then our regret analysis can also be applied to the case when Assumption 2 is violated.

Let

\[
E_{\text{conf}} := \{|\hat{g}_{t,x,a}^\rho - g_{t,x,a}| \leq \epsilon_{t,x,a} \forall \rho \geq 2, \forall t \in [t_{\max}], \forall x \in \mathcal{X}, \forall a \in \bar{A}\}
\]

be the event that the sample mean gains are within \(\epsilon_{t,x,a}\) of the expected gains. The following lemma bounds the probability that \(E_{\text{conf}}\) happens.

**Lemma 1.** \(\text{Pr}(E_{\text{conf}}) \geq 1 - \delta\).

**Proof.** Fix any stage-state-action triplet \((t,x,a)\). Let

\[
E_{\text{conf}}(t,x,a) := \{|\hat{g}_{t,x,a}^\rho - g_{t,x,a}| \leq \epsilon_{t,x,a} \forall \rho \geq 1\}
\]

By replacing \(\delta\) term in (13) and (15) given in Appendix A with \(\delta/(\max_{t \in [t_{\max}]} \hat{S}_{t,x})\), we get \(\text{Pr}(E_{\text{conf}}(t,x,a)) \leq 1 - \delta/(\max_{t \in [t_{\max}]} \hat{S}_{t,x})\) (details can be found in Appendix A). This implies that \(\text{Pr}(\bigcup_{t \in [t_{\max}]} \bigcup_{x \in \mathcal{X}} \bigcup_{a \in \bar{A}} E_{\text{conf}}(t,x,a)) \leq \delta/(\max_{t \in [t_{\max}]} \hat{S}_{t,x})\) for all \(t \in [t_{\max}], x \in \mathcal{X}\) and \(a \in \bar{A}\). Using a union bound, we get

\[
\text{Pr}(E_{\text{conf}}) \leq \sum_{t \in [t_{\max}]} \sum_{x \in \mathcal{X}} \sum_{a \in \bar{A}} \text{Pr}(E_{\text{conf}}(t,x,a)) \leq \delta.
\]

The next lemma upper bounds the number of times each action can be selected.

**Lemma 2.** On event \(E_{\text{conf}}\) we have

\[
N_{t,x,a}^\rho \leq \frac{16\sigma^2}{\Delta_{t,x,a}} \log\left(\frac{16\sigma^2 K}{\Delta_{t,x,a}^2} \right) \forall \rho \geq 1, \forall t \in [t_{\max}], \forall x \in \mathcal{X}, \forall a \in \bar{A}.
\]

**Proof.** For \(\rho = 1\), the result is trivial. For \(\rho > 1\), the proof proceeds in a way that is similar to the proof of Lemma 6 in [18]. Firstly, assume that action \(a \in \bar{A}\) is selected in stage \(t \in [T_{\rho}]\) of round \(\rho\) when the state is \(x\). Since

\[
\hat{g}_{t,x,a}^\rho \in [g_{t,x,a} - \text{conf}_{t,x,a}^\rho, g_{t,x,a} + \text{conf}_{t,x,a}^\rho]
\]

on event \(E_{\text{conf}}\), using

\[
\hat{g}_{t,x,a}^\rho + \text{conf}_{t,x,a}^\rho \geq g_{t,x,a}^\rho
\]

and the definition of \(\Delta_{t,x,a}\), we obtain \(\text{conf}_{t,x,a}^\rho \geq \Delta_{t,x,a}/2\). Substituting the values in (6) and (7) into \(\text{conf}_{t,x,a}^\rho\) and using the fact that \((z^2 - 1)/(z + 1) \leq z^2/(z + 1)\) for positive integers \(z\), we get for \(a \in \bar{A}\)

\[
\frac{(N_{t,x,a}^\rho)^2 - 1}{N_{t,x,a}^\rho + 1} \leq \frac{4\sigma^2 \log\left(\frac{K(1 + N_{t,x,a}^\rho)^{1/2}}{\delta}\right)}{\Delta_{t,x,a}^2}
\]

Finally, assume that the \(s := \text{stop}\) action is selected in stage \(t = T_{\rho}\) of round \(\rho\) when the state is \(x\). Let

\[
\text{conf}_{t,x,s}^\rho = \sqrt{\frac{1 + N_{t,x,s}^\rho}{(N_{t,x,s}^\rho)^2}} \left(4\sigma^2 \log\left(\frac{K(1 + N_{t,x,s}^\rho)^{1/2}}{\delta}\right)\right).
\]

Since \(N_{t,x,a}^\rho \leq N_{t,x,s}^\rho\), we have \(\text{conf}_{t,x,s}^\rho \leq \text{conf}_{t,x,a}^\rho\), which implies that on event \(E_{\text{conf}}\)

\[
\hat{g}_{t,x,a}^\rho \leq g_{t,x,a} + \text{conf}_{t,x,a}^\rho
\]

and the definition of \(\Delta_{t,x,a}\), we obtain \(\text{conf}_{t,x,s}^\rho \geq \Delta_{t,x,a}/2\). This implies that (10) also holds for \(a = \text{stop}\).

Next, we use the a lemma from [19] to bound (10), which also given in Appendix B. From (10) we obtain

\[
N_{t,x,a}^\rho \leq 1 + \frac{16\sigma^2}{\Delta_{t,x,a}^2} \log\left(\frac{K}{\delta}\right) + \frac{8\sigma^2}{\Delta_{t,x,a}^2} \log(1 + N_{t,x,a}^\rho)
\]

Since \(1 + N_{t,x,a}^\rho \geq 1\), we substitute \(a = \Delta_{t,x,a}^2/(16\sigma^2)\) and \(b = \log(16\sigma^2)/\Delta_{t,x,a}^2\) in Appendix B to get the bound

\[
\log(1 + N_{t,x,a}^\rho) \leq a(1 + N_{t,x,a}^\rho) + b
\]

The result is obtained by substituting this into (11).

As a corollary of Lemma 2 we derive the following bound on the confidence of the actions selected by FAL.

**Corollary 1.** (Confidence bound on the actions chosen by FAL). With probability at least \(1 - \delta\)

\[
\forall \rho \geq 1, \forall t \in [t_{\max}], \forall x \in \mathcal{X}, \forall a \in \bar{A}, \forall \tilde{a} \in O_{t,x} \setminus \{\text{stop}\}, \text{conf}_{t,x,a}^\rho \leq 2\text{conf}_{t,x,a}^\rho\tilde{a}.
\]

**Proof.** The result follows by a simple application of (8) and (9) on event \(E_{\text{conf}}\).
Corollary 1 bounds the suboptimality of the action selected by FAL in any stage of any round by $2\text{conf}^t_{l,x,a}$, which only depends on quantities $\delta$, $K$, $\sigma^2$ and $N^0_{l,x,a}$, which are known by the learner at the time $a^t_l$ is selected.

Consider any algorithm that deviates from the benchmark for the first time in stage-state pair $(t, x)$ by choosing action $a$ that is different from the action that will be chosen by the benchmark at $(t, x)$. Let $\mu^*_{t,x}$ be the maximum expected gain that can be acquired by the benchmark starting from stage-state pair $(t, x)$. Let $\mu_{t,x,a}$ be the minimum expected gain that can be acquired by any algorithm by choosing the worst-sequence of actions starting from stage-state pair $(t, x)$ after choosing action $a$. We define the deviation gap in stage-state pair $(t, x)$ as $\Omega_{t,x,a} := \mu^*_{t,x} - \mu_{t,x,a}$. The following theorem show that the regret of FAL is bounded with probability at least $1 - \delta$.

**Theorem 1.** (High probability regret bound for FAL) With probability at least $1 - \delta$, the regret given in (4) is bounded by

$$R(n) \leq \sum_{t=1}^{l_{\max}} \sum_{x \in X} \sum_{a \notin O_{t,x}} \Omega_{t,x,a} \left(3 + \frac{16\sigma^2}{\Delta^2_{l,x,a}} \log\left(\frac{16\sigma^2Kn}{\Delta^2_{l,x,a}}\right)\right).$$

**Proof.** The proof directly follows by summing the result of Lemma 2 among all stage-state-action triplets $(t, x, a)$. \hfill \Box

The bound given in Theorem 1 does not depend on $n$. This bound can be easily converted to a bound on the expected regret by setting $\delta = 1/n$ in FAL.

**Theorem 2.** When FAL is run with $\delta = 1/n$, its expected regret given in (5) is bounded by

$$E[R(n)] \leq \Omega_{\max} + \sum_{t=1}^{l_{\max}} \sum_{x \in X} \sum_{a \notin O_{t,x}} \Omega_{t,x,a} \left(3 + \frac{16\sigma^2}{\Delta^2_{l,x,a}} \log\left(\frac{16\sigma^2Kn}{\Delta^2_{l,x,a}}\right)\right)$$

where $\Omega_{\max} = \max_{t,x,a} \Omega(t, x, a)$.

**Proof.** Consider Theorem 1 With probability $\delta$, the regret is bounded above by $n\Omega_{\max}$. With probability $1 - \delta$, the regret is bounded by this theorem’s main statement. The result is obtained by applying the law of total expectation. \hfill \Box

Theorem 2 shows that the expected regret of FAL is $O(\log n)$. Although the constant terms given in Theorems 1 and 2 depend on unknown parameters $\Delta_{l,x,a}$ and $\Omega_{t,x,a}$, FAL does not require the knowledge of these parameters to run and to calculate its confidence bounds. From the expressions in Theorems 1 and 2, it is observed that the regret scales linearly with $\Omega_{t,x,a}/\Delta^2_{l,x,a}$, which is a term that indicates the hardness of the problem. If the suboptimality gap $\Delta^2_{l,x,a}$ is small, FAL makes more errors by choosing $a \notin O_{l,x}$ when it tries to follow the benchmark. This results in a loss in the expected gain that is bounded by $\Omega_{t,x,a}$.

Next, we consider problems in which deviations from the benchmark in early stages cost more than deviations from the benchmark at later stages.

**Assumption 3.** $\Omega_{t,x,a} \leq (l_{\max} - t)\Delta_{l,x,a}$ for all $t \in [l_{\max}]$, $x \in X$, $a \notin O_{t,x}$.

Using this assumption, the following result is derived for the expected regret of FAL.

**Corollary 2.** If Assumption 1 holds, and FAL is run with $\delta = 1/n$, we have

$$E[R(n)] \leq \Omega_{\max} + \frac{l_{\max}^2 - l_{\max}}{2} \sum_{x \in X} \sum_{a \notin O_{t,x}} \left(3\Delta_{l,x,a} + \frac{16\sigma^2}{\Delta_{l,x,a}} \log\left(\frac{16\sigma^2Kn}{\Delta^2_{l,x,a}}\right)\right).$$

**Remark 1.** FAL adaptively learns the expected gains of action and feedback sequences that correspond to stopping at various stages. Although our model allows at most $l_{\max}$ actions to be taken in each round, the actual number of actions taken may be much lower than this value depending on the expected costs $c_{l,x,a}$. High costs imply a decrease in the marginal benefit of continuation, which implies that the benchmark may take the stop action earlier than the case when costs are low.

**Remark 2.** The state-space model we proposed is very general, and as we stated in Section III-B, it includes the adaptive monotone submodular problem [13], [17] as a special case. The state-space model of S-MAB generalizes these problems in a way that the distribution of feedback given the action also depends on the state of the system.

Although, the regret bound of FAL increases polynomially in the size of the state-space, for many interesting applications of the S-MAB, the state-space is small. For instance, consider the breast cancer treatment example in [20]. In this example, $X$ has only four states: no cancer, in situ cancer, invasive ductal carcinoma, dead. $F$ is the set of treatment options, and $\mathcal{F}$ is the feedback set, which can be the reduction in tumor size given a particular treatment in a particular state.

V. ILLUSTRATIVE EXAMPLE

In this section we model a medical screening application using S-MAB and compare FAL with other state-of-the-art algorithms.

A. Dataset Description

We use the Electronic Health Record (EHR) dataset from [21], which includes de-identified data of patients that took screening tests at the UCLA medical center. The patients received multiple breast cancer screening tests which consist of regular check-ups as well as follow-up tests insisted by clinicians. The dataset is composed of 24,484 patients out of which 660 (2.68%) is positively diagnosed with breast cancer.

For each patient, the dataset contains 8 patient features that are risk factors for the breast cancer: age, gender, breast density, family history of breast cancer in first-degree relatives, 4

4In calculating $\mu^*_{t,x}$, we assume that in stages in which the benchmark needs to randomize between at least two actions, the action that maximizes the expected reward of the benchmark is selected.

5The reward assigned to state “dead” can be 0, and to state “no cancer” can be 1.
age at first childbirth, age at menarche, the number of previous biopsies, and an indicator of multiple similar cancer findings. Moreover, each patient has a binary label indicating whether the patient is found out to have breast cancer or found out to be negative or benign status.

In this setting, there are three modalities of the breast screening tests, which correspond to the actions in S-MAB: mammogram, ultrasound and magnetic resonance imaging. Each screening test has associated breast imaging reporting and data system (BI-RADS) scores which range from 1 through 6. BI-RADS score assessments are given in Table I.

| BI-RADS | Assessment                          |
|---------|-------------------------------------|
| 1       | Negative                            |
| 2       | Benign finding                      |
| 3       | Probably benign                     |
| 4       | Suspicious or indeterminate abnormality |
| 5       | Highly suggestive of malignancy     |
| 6       | Known biopsy with proven malignancy |

### B. Simulation Setup

In this section, we describe how FAL and the competing benchmarks are simulated on the dataset described above.

1) **Actions, feedbacks, states, rewards and costs**: A round consists of multiple stages, where a screening test is applied at each stage. It is assumed that each modality can be used at most once for each patient (in each round). Thus, the set of actions at the beginning of a round are \( A = \{ \text{MG, US, MR} \} \), where MG, US, and MR stands for the mammogram, ultrasound, and MRI screening tests, respectively. Hence \( I_{\text{max}} = 4 \).

The feedback in a stage is set to be the BI-RADS score from the screening test administered at that stage.

For each patient, \( \mathcal{X} \) consists of four states: very unlikely to have cancer, unlikely to have cancer, likely to have cancer, and very likely to have cancer. For each round, the initial state is decided by how many criteria the patient’s features meet for the risk factor of the breast cancer. For example, if the patient has high breast density, multiple family histories of breast cancer, earlier age at menarche, and previous experience of biopsies, the patient’s initial state is set to very likely to have cancer. After each screening test, the state of the patient changes based on a simple rule which depends on the feedback BI-RADS score. If the feedback BI-RADS score is 1, 2, 3, or 4 or more, the patient’s next state becomes very unlikely, unlikely, likely, and very likely states, respectively.

The terminal rewards for detection, missed detection and false alarm are set in a way that finding the actual cancer is much more important than finding the actual negative or benign cases. The cost of MG is set to be lower than the costs of US and MR.

### C. Algorithm and Benchmarks

1) **FAL**: Since it is assumed that each modality can be used at most once for each patient, FAL is modified in a way that it cannot select the same action in a round more than once. When the stop action is selected, a decision tree predicts whether the patient has cancer or not using the BI-RADS scores observed in all stages.

2) **Clinical guideline benchmark (CGB)**: In the first stage of a round, CGB always recommends MG. Then, in addition to MG, it recommends ultrasound to patients with high breast density and MR to patients who are categorized as high-risk patients based on their contexts. After the BI-RADS scores of all stages are observed, CGB predicts malignant if at least one BI-RADS score is greater than or equal to 4.

3) **Contextual bandit benchmark (CBB)**: The context is taken to be the initial state of the patient, which can take one of the four possible values. Each action for CBB corresponds to a permutation of non-empty subsets of \( A \). CBB runs a different instance of the UCB1 algorithm \([22]\) for each context. After the BI-RADS scores are observed by the chosen action, CBB uses the same decision tree as FAL to predict whether the patient has cancer or not.

### D. Results

Table I and Fig. 2 summarize the average performance of the considered algorithms. FAL achieves 0.27 gain in true positive rate (TPR) and 0.061 gain in positive predictive value (PPV) compared to CBB, and 0.038 gain in TPR and 0.294 gain in PPV compared to CGB. In addition, we also observe that FAL has the highest average utility, which is defined as the average terminal reward minus the average cost. From Fig. 2 it is observed that FAL is strictly better than its competitors in both TPR and PPV.

### VI. Related Work

#### A. Combinatorial, Semi and Matroid Bandits

S-MAB is related to various existing classes of MAB with large action sets. These include combinatorial bandits \([23]\)–\([25]\), combinatorial semi-bandits \([26]\), matroid bandits \([27]\).
and bandits in metric spaces [23]. In these works, at each
time, the learner (simultaneously) chooses an action tuple
and obtains a reward that is a function of the chosen action
tuple. Unlike these works, in S-MAB actions in a round are
chosen sequentially, and the previously chosen actions in a
round guide the action selection process within that round.
The differences between S-MAB and these various classes of
MAB are given in Table III.

**TABLE III: Comparison of S-MAB with combinatorial, semi
and matroid bandits.**

| Arm selection in each round | [23]–[27] | S-MAB |
|----------------------------|----------|-------|
| Reward in each round       | sum of rewards of selected arms | general function of selected arms and observed feedbacks |
| Action sequence length     | fixed and limited by action set size | variable, not limited by action set size |

### B. Bandits with Knapsacks

Another related strand of literature studies MAB with
knapsacks [29], [30]. In these problems, there is a budget,
which limits the number of times a particular action can
be selected. The goal is to maximize the total reward given
the budget constraints. However, similar to standard MAB
problems, in these problems it is also assumed that the reward
is immediately available after each selected action, and the
current reward only depends on the current action unlike S-
MAB in which the current reward depends on a sequence of
actions and feedbacks through a state. Although the S-MAB
also has a budget constraint which restricts the length of the
action sequence that can be taken in each round, this constraint
is completely different from the budget constraint in MAB
with knapsacks. In the S-MAB problem, the budget is renewed
after each round; and hence, does not limit the number of
rounds in which a certain action can be selected as in MAB
with knapsacks.

### C. Adaptive Submodular Bandits

One of the most closely related prior works is the work
on adaptive submodularity [17]. In this work, existence of an
underlying joint state that is realized from a prior distribution
before the start of the action selection process in each round is
assumed. The learner is constrained to pick sequentially \( K \)
of the actions in the action set in each round without replacement
in order to maximize its reward. The states of the selected
actions are instantly revealed to the learner; hence, at decision
stage \( t \) of a round, the learner knows the states of all arms
selected before stage \( t \). The reward the learner gets at the end
of a round is a submodular function of the selected arms and
their states.

If we translate the above setting to our S-MAB formulation,
the joint state can be viewed as a hidden state vector
for actions, whose components are revealed only after the
corresponding actions are taken. Hence, this is a special case
of the S-MAB, where the joint state does not depend on the
chosen actions and observed feedbacks.

It is shown in [17] that for adaptive submodular reward
functions, a simple adaptive greedy policy (which resembles
our benchmark) is \( 1 - 1/e \) approximately optimal. Hence, any
learning algorithm that has sublinear regret with respect to the
greedy policy is guaranteed to be approximately optimal. This
work is extended to an online setting in [13], where prior
distribution over the state is unknown and only the reward
of the chosen sequence of actions is observed. However, an
independence assumption is imposed over action states to
estimate the prior in a fast manner.

Our work differs from these works: (i) In [13], [17]
the *adaptive stochastic maximization* problem is considered,
where the goal is to select the optimal sequence of items or
actions (without replacement) given a fixed budget (on the
number of stages). In our formulation, the same action can be
taken in different stages and the number of stages is not fixed
but is adapted based on the feedback. (ii) In [13], [17] the
item states (feedbacks) are realized before the round begins.
In our formulation, feedback in the current stage depends on
actions and feedbacks in prior stages of the current round.

We show in Section II-D that our benchmark is approxi-
mately optimal when the reward function is adaptive monotone
submodular, an action can only be selected in a single stage
and the feedback related with each action is realized at the
beginning of each round before action selection takes place.
Hence, work on adaptive submodular learning can be viewed
as a special case of the S-MAB.

Other variants such as [31], [32] use monotone submodu-
lar property in online resource allocation problems, where
the reward of a round is evaluated based on the monotone
submodular utility function of that round, which changes from
round to round. The action sequence is fixed at the beginning
of each round before observing the utility function. In these
works, learning algorithms with sublinear regret with respect
to the best (fixed) action sequence in hindsight are developed.
Unlike these works, we compare the regret of S-MAB against
an adaptive (not fixed) benchmark.

### D. Optimization and Reinforcement Learning in Markov De-
cision Processes (MDPs)

Our problem is also related to reinforcement learning in
MDPs. Regret bounds only exists for highly structured MDPs.
For instance, in [14], [15] algorithms with logarithmic regret
with respect to the optimal policy are derived for finite,
positive recurrent MDPs. Episodic MDPs are studied in [33],
and sublinear regret bounds are derived assuming that the \( \log \)
sequence is generated by an adversary. S-MAB differs from
these works as follows: (i) the number of visited states (stages)
in each round is not fixed; (ii) During a round, only feedbacks
are observed and no reward observations are available for the
intermediate states. Rewards of the intermediate states are only
revealed at the end of the round.

[17] also considers other optimization settings, but they are not related
to our work.
In [34] PAC bounds are derived for continuous state MDPs with unknown but deterministic state transitions and geometrically discounted rewards, using a metric called policy-mistake count. In contrast, we develop regret bounds for S-MAB which hold uniformly for time over unknown, random state transitions, and undiscounted rewards.

While it is possible to translate the S-MAB into an MDP, the optimal solution cannot be found by policy iteration (PI) or value iteration (VI) because the transition probabilities are unknown. In addition to the above problems, reinforcement learning in MDPs require either computationally intractable algorithms (exponential in the length of each round) with sublinear regret or computationally tractable algorithms with linear regret (and no approximation guarantee). Moreover, model-free methods like Q-learning [35] and TD(λ) [36] will be highly inefficient due to the size of the sequence of actions that can be taken, and the sequence of feedbacks that can be observed in each round. In contrast, regret of the S-MAB depends only polynomially on the round length and logarithmically on the number of rounds. The differences between S-MAB, and optimization and reinforcement learning algorithms for MDPs are given in Table IV.

We would also like to note that S-MAB is very different from PAC learning [37]. In our context, the goal of a PAC learning algorithm will be to minimize the number of exploration rounds that is required to identify an ϵ optimal policy with probability at least 1 − δ. In contrast, the goal of the S-MAB is to maximize the total reward summed over all rounds.

**TABLE IV: Comparison of S-MAB with optimization and reinforcement learning algorithms.**

|                              | PI, VI     | Q-learning, TD(λ) | S-MAB     |
|------------------------------|------------|-------------------|-----------|
| Transition probabilities     | known      | unknown           | unknown   |
| Convergence to optimal       | always     | may converge     | asymptotically converges asymptotically |
| Regret in time               | zero       | may be           | logarithmic |
| Efficient for:               | small action sequences | small action sequences | large action sequences |

_E. Online and Stochastic Convex Optimization_

We will finish our discussion of the related work by differentiating S-MAB from Online Convex Optimization (OCO) and Stochastic Convex Optimization (SCO) problems.

In OCO, there is a learner which sequentially chooses actions (from a convex set) over time, and incurs a loss (from a convex function) after each chosen action. The loss function is generated by an adversary and is unknown to the learner beforehand. The goal of the learner is to minimize its regret, which is the difference between the total loss it accumulates and the loss of the best fixed action it could have followed (best fixed strategy in hindsight). Many versions of OCO exist including full feedback [38]–[40], in which the learner observes the entire loss function after each decision step, and bandit feedback [41], [42], in which the learner partially observes the loss evaluated at the chosen action.

OCO and S-MAB have two fundamental differences: (i) In S-MAB, the learner selects multiple actions during a round, and the selected actions effect the actions that will be selected in future; whereas in OCO full or partial loss function is observed after every taken action, and the reward only depends on the current taken action. (ii) The regret of an S-MAB algorithm is measured with respect to the adaptive benchmark, which myopically adapts the next action to select in a round based on the previously selected actions and observed feedbacks; whereas in OCO the regret is measured with respect to the best fixed action in hindsight. While the action sequence selected by the adaptive benchmark can change from round to round based on the sequence of observed feedbacks, the action sequence selected by the benchmark of the OCO is fixed among rounds. Hence S-MAB and OCO are different both in terms of the way rewards are generated and performance is evaluated.

In SCO, the goal is to minimize a convex loss function using finite number queries obtained from a gradient oracle [43]. Numerous methods have been proposed to efficiently solve this problem, such as a batch reduction from an OCO problem and alternating direction method of multipliers based methods [44]. The differences of SCO from S-MAB are similar in flavor to that of OCO from S-MAB. In addition to this, the objective function of SCO is also different from that of OCO and S-MAB.

**VII. CONCLUSION**

In this paper, we proposed a new class of online learning methods called Staged Multi-armed Bandits. Although the number of possible sequences of actions increases exponentially with the length of the round in an S-MAB, we proved that an efficient online learning algorithm which has expected regret that grows polynomially in the number of stages and states, and logarithmically in the number of rounds exists. This algorithm enjoys high probability confidence bounds on the expected gain of selected actions, and its regret is shown to be bounded with high probability. Possible future research directions are listed below:

- Some actions may generate feedbacks that are highly informative about the reward at the end of the round, while some actions may generate feedbacks whose effect on the reward is negligible. Differentiating the quality of the actions in terms of the informativeness of the feedbacks they generate has the potential to reduce the regret from linear in the number of actions to linear in the number of informative actions.
- In some applications of S-MAB such as clinical decision support for complex diseases, multiple actions (e.g., drugs) are required to be taken concurrently, and the feedback may only be available for the group of actions taken together, but not for the individual actions. This turns the problem into a combinatorial S-MAB, which we plan to analyze in the future.
APPENDIX A
A CONFIDENCE BOUND FOR STAGE-STATE-ACTION TRIPLET

Firstly, we consider the confidence bound for the stop action. Fix $t \in [l_{\text{max}}]$ and $x \in X$. Let $\epsilon_\rho = I(t \leq T_\rho, x_t^\rho = x)$. By Assumption 1, $\{\kappa^\rho_t\}_{t=1}^\infty$ is a sequence of independent $\sigma$-sub-Gaussian random variables. Using the result of Theorem 1 in [18], it can be shown that given any $\delta > 0$ with probability at least $1 - \delta$ we have for all $\rho \geq 2$

$$
\left( \frac{\sum_{t=1}^{\rho-1} \epsilon_t \kappa^\rho_t}{\sqrt{N_{t,x}^\rho}} \right)^2 \leq 2\sigma^2 \log \left( \frac{\sqrt{1 + N_{t,x}^\rho}}{\delta} \right)
$$

\Rightarrow \left| \sum_{t=1}^{\rho-1} \epsilon_t \kappa^\rho_t \right| \leq \sqrt{(1 + N_{t,x}^\rho)2\sigma^2 \log \left( \frac{\sqrt{1 + N_{t,x}^\rho}}{\delta} \right)}.

(12)

Observe that

$$
\hat{r}_{t,x}^\rho = \sum_{t=1}^{\rho-1} \frac{r_{t,x} \epsilon_t + \kappa^\rho_t \epsilon_t}{N_{t,x}^\rho} = r_{t,x} + \sum_{t=1}^{\rho-1} \frac{\epsilon_t \kappa^\rho_t}{N_{t,x}^\rho}.
$$

Hence

$$
|\hat{r}_{t,x}^\rho - r_{t,x}| = \frac{1}{N_{t,x}^\rho} \left| \sum_{t=1}^{\rho-1} \epsilon_t \kappa^\rho_t \right|.
$$

Combining this with (12) we obtain with probability at least $1 - \delta$

$$
|\hat{r}_{t,x}^\rho - r_{t,x}| \leq \sqrt{(1 + N_{t,x}^\rho)2\sigma^2 \log \left( \frac{\sqrt{1 + N_{t,x}^\rho}}{\delta} \right)}.
$$

(13)

Since by definition $g_{t,x,\text{stop}} = r_{t,x}$ and $\hat{g}_{t,x,\text{stop}} = \hat{r}_{t,x}$ we get with probability at least $1 - \delta$

$$
\forall \rho \geq 2 \quad |\hat{g}_{t,x,\text{stop}}^\rho - g_{t,x,\text{stop}}| \leq \sqrt{(1 + N_{t,x,a}^\rho)2\sigma^2 \log \left( \frac{\sqrt{1 + N_{t,x,a}^\rho}}{\delta} \right)}.
$$

(14)

Observe that

$$
\hat{g}_{t,x,a}^\rho = \frac{\sum_{t=1}^{\rho-1} \left( g_{t,x,a} \epsilon_t + \beta_t \kappa^\rho_t \epsilon_t \right)}{N_{t,x,a}^\rho} = g_{t,x,a} + \frac{\sum_{t=1}^{\rho-1} \beta_t \epsilon_t}{N_{t,x,a}^\rho}.
$$

Hence

$$
|\hat{g}_{t,x,a}^\rho - g_{t,x,a}| = \frac{1}{N_{t,x,a}^\rho} \left| \sum_{t=1}^{\rho-1} \epsilon_t \beta_t \kappa^\rho_t \right|.
$$

Combining this with (14) we obtain with probability at least $1 - \delta$

$$
\forall \rho \geq 2 \quad |\hat{g}_{t,x,a}^\rho - g_{t,x,a}| \leq \sqrt{(1 + N_{t,x,a}^\rho)2\sigma^2 \log \left( \frac{\sqrt{1 + N_{t,x,a}^\rho}}{\delta} \right)}.
$$

(15)

APPENDIX B

LEMMA 8 OF [19]

Let $a > 0$. For any

$$
\tau > 2 \left( \log \left( \frac{1}{a} \right) - b \right)^+ + a \tau + b \geq \log \tau,
$$

we have $a \tau + b \geq \log \tau$, where $a^+ = \max(a,0)$.

REFERENCES

[1] T. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” Advances in Applied Mathematics, vol. 6, pp. 4–22, 1985.
[2] 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, 2010.
[3] A. Anandkumar, N. Michael, and A. Tang, “Opportunistic spectrum access with multiple players: Learning under competition,” in Proc. IEEE INFOCOM, pp. 1–9.
[4] Y. Gai and B. Krishnamachari, “Distributed stochastic online learning policies for opportunistic spectrum access,” IEEE Trans. Signal Process., vol. 63, no. 23, pp. 6184–6193, 2014.
[5] S. Bagheri and A. Scaglione, “The restless multi-armed bandit formulation of the cognitive compressive sensing problem,” IEEE Trans. Signal Process., vol. 63, no. 5, pp. 1183–1198, 2015.
[6] 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, 2015.
[7] A. Slivkins, “Contextual bandits with similarity information,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 2533–2568, 2014.
[8] L. Li, W. Chu, J. Langford, and R. Schapire, “A contextual-bandit approach to personalized news article recommendation,” in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 661–670.
[9] J. Nassour, V. Hugel, F. B. Ouezdou, and G. Cheng, “Qualitative adaptive reward learning with success failure maps: applied to humanoid robot walking,” IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 1, pp. 81–93, 2013.
