Indian J. Pure Appl. Math., **47**(2): 195-212, June 2016
 DOI: 10.1007/s13226-016-0184-5

MULTI-ARMED BANDITS BASED ON A VARIANT OF SIMULATED ANNEALING

Mohammed Shahid Abdulla* and Shalabh Bhatnagar**

**IT and Systems Area, Indian Institute of Management, Kozhikode, India*

***Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India*

e-mails: shahid@iimk.ac.in, shalabh@csa.iisc.ernet.in

(*Received* 28 *May* 2015; *accepted* 16 *September* 2015)

A variant of Simulated Annealing termed Simulated Annealing with Multiplicative Weights (SAMW) has been proposed in the literature. However, convergence was dependent on a parameter $\beta(T)$, which was calculated *a-priori* based on the total iterations T the algorithm would run for. We first show the convergence of SAMW even when a diminishing stepsize $\beta_k \rightarrow 1$ is used, where k is the index of iteration. Using this SAMW as a kernel, a stochastic multi-armed bandit (SMAB) algorithm called SOFTMIX can be improved to obtain the minimum-possible $\log$ regret, as compared to $\log^2$ regret of the original. Another modification of SOFTMIX is proposed which avoids the need for a parameter that is dependent on the reward distribution of the arms. Further, a variant of SOFTMIX that uses a comparison term drawn from another popular SMAB algorithm called UCB1 is then described. It is also shown why the proposed scheme is computationally more efficient over UCB1, and an alternative to this algorithm with simpler stepsizes is also proposed. Numerical simulations for all the proposed algorithms are then presented.

Key words : Stochastic processes; applied probability; statistics; discrete optimization.

1. INTRODUCTION

An algorithm called Simulated Annealing with Multiplicative Weights (SAMW) was introduced in [10]. Though the objective in [10] was to compute optimal policies for Finite-Horizon Markov Decision Processes, a simplified description of this algorithm can be given as follows. In the setting of SAMW, a random reward X_k^i is obtained at each step k, where the action a^i is such that it belongs to a finite set $|A|$. An empirical mean of the reward for each action a^i, defined as $\mu_k^i = \frac{\sum_{s=1}^k X_s^i}{k}$, is updated after the step k. The $|A|-$size probabilty vector ϕ_k is now updated to produce a ϕ_{k+1}, the objective being that $\phi_k^* \rightarrow 1$ as $k \rightarrow \infty$ for the action a^* with the highest average payoff (with

$\phi_k^j \to 0$ for actions $a^j \in A\backslash\{a^*\}$). We assume that a unique best action a^* exists. The improvement step of ϕ_k is performed as:

$$\phi_{k+1}^i := \frac{\phi_k^i \beta^{\mu_k^i}}{\sum_{a^j \in A} \phi_k^j \beta^{\mu_k^j}}, \tag{1}$$

where $\beta > 0$, a small constant which [10] requires to be $\beta(T)$ - a constant calculated apriori after the maximum iteration number T, i.e., $k \leq T$, is set. For a finite T, a constant β would only result in approximate performance as the proofs in [10] employed bounds that were 'asymptotically efficient', i.e., they held tightly only if $T \to \infty$. In §2 below, we eliminate this dependance of β on T. Note that in the setting that SAMW dealt with, each action is to be employed to obtain a random reward (that follows an unknown, but fixed, distribution) at each step k. It is this protocol that is changed in the following section, to introduce our second problem.

1.1 *Stochastic Multi-Armed Bandit*

The Stochastic Multi-Armed Bandit (SMAB) problem has the goal of detecting the action a^* that has the highest expected reward, as early as possible, or with as few applications of the sub-optimal actions $b \in A\backslash a^*$. We assume in the following that there is a unique best action a^*. To explain this problem in brief: a finite set of actions, called arms, A is available. Each arm a^j in A yields a reward drawn from a fixed distribution, whose mean is μ^j. The player who pulls the arms maintains an empirical mean for each arm j, but has to discover soonest the best arm $a^* \in A$ (the arm which corresponds to highest mean μ^*), so that this arm can continue to be pulled for maximum expected profit. We maintain a probability vector ϕ_k over A, that is updated iteratively to yield convergence (to a ϕ^*) so that as $k \to \infty$, $\phi_k(a^*) \to 1$ and $\phi_k(a^j) \to 0$ for all $a^j \in A\backslash\{a^*\}$. One may consider this as a randomized stationary policy for a one-state MDP.

The expected regret of the algorithm is defined as the expected total loss incurred due to not playing the best arm a^* in each iteration k. The arm played at each iteration k, inferred according to one's algorithm, is $\hat{a}_k$ and the reward obtained by pulling this arm is $\hat{X}_k$. Thus $k \cdot \mu^* - \sum_{p=1}^{k} E\left(\hat{X}_p | \mathcal{F}_{p-1}\right)$ is the expected regret of this algorithm, which samples arms $\hat{a}_k$ according to probability iterate ϕ_k. Here, $\mathcal{F}_{p-1}$ is a $\sigma-$algebra, and contains all information available to the algorithm at iteration p. In §3, we employ the SAMW kernel inside a SMAB algorithm called SOFTMIX [7], and propose a new algorithm SAMWMIX which achieves $O(\log k)$ expected regret in k steps, improved from $O(\log^2 k)$ of SOFTMIX. It is a classical result in machine learning that the best possible regret any SMAB algorithm can achieve is log regret (cf. [6, Theorem 2.2]). Also, [15] suggests that a log-regret analog

of SOFTMIX was yet to be found.

Both SOFTMIX and SAMWMIX have the disadvantage that a parameter d has to be provided *a-priori* as input to the algorithm satisfying the property that $d < \min_{a^i \in A \setminus \{a^*\}}\{\mu^* - \mu^i\}$. We eliminate this disadvantage in §4 by proposing a blind-SAMWMIX. While it behaves quite well in practice, blind-SAMWMIX has the disadvantage that it can only achieve $O(\log^{1+2\alpha}(k))$ regret, where $0 < \alpha < 0.5$ (with better properties for α closer to 0.5). A popular log-regret algorithm for SMAB, called UCB1 (Upper Confidence Bound 1), is of a different type. Here, at step k, a confidence bound b_k^i is calculated for each arm i and the arm corresponding to $\text{argmax}_{a^i \in A}\{\mu_k^i + b_k^i\}$ is played as the 'winner arm' $\hat{a}_{k+1}$. Consequently, the empirical mean μ_{k+1}^j for $a^j = \hat{a}_{k+1}$ is re-computed to include new sample $\hat{X}_{k+1}$, whereas all other empirical means are carried over, i.e. $\mu_{k+1}^j = \mu_k^j$ for $a^j \neq \hat{a}_{k+1}$. Thus it is not a randomized policy and performs an implicit explore-exploit trade-off using the bound b_k^i. Just as the SAMW kernel was used in SOFTMIX to produce SAMWMIX, we employ the UCB1 mechanism for our algorithm UCB1MIX in §5. The advantage lies in avoidance of the $O(|A|)$ complexity operation to find $\text{argmax}_{a^i \in A}\{\mu_k^i + b_k^i\}$.

1.2 *Survey of SAMW and SMAB*

The algorithm in (1) as described in [10] is drawn from the work in [12] where an optimal strategy for non-cooperative repeated bi-matrix zero-sum games is the objective. In [12] too, the quantity β in (1) is constant and dependent on the total number of iterations T. Our choice of β_k is a diminishing parameter with the further difference that $\beta_k > 1\ \forall k$. SAMW differs from Simulated Annealing in that it does not perform any local search in A but updates the probability distribution ϕ_k over A. It also has a simpler tuning process.

As we explain in §6, there are computation as well as precision-related advantages over UCB1 for algorithms like SOFTMIX that use a Boltzmann exploration structure. The work in [14] summarizes the advantages that a stochastic policy-based algorithm (that additionally uses 'Importance Sampling', the device in (9) below) has over other deterministic algorithms like UCB1. However, UCB1 remains popular in both extensions of SMABs as well as applications. Among extensions, recent work on SMAB has focused on using a norm in the Kullback-Liebler neighbourhood to produce UCB1's confidence bounds, bringing it into the *nearly* constant regime as in the KL-UCB of [13]. However, KL-UCB is optimal for Bernoulli arms (i.e. arms having reward 1 with a Bernoulli bias probability), and also requires an input parameter $\alpha > 1$. For general distributions, there is the α-UCB algorithm introduced in [6, §2.2] whose regret is always better than KL-UCB, but where there

is an input parameter too, this time $\alpha > 2$. We make specific comparisons with two other algorithms (EXP3 and Thomson Sampling) on the basis of their regret bounds in §3 below. A useful summary of stochastic multi-armed bandit algorithms' results can be found in [6, Chapters 1-2].

Of note among applications are the 'Mortal' bandits in [8], where there is the possibility that certain arms will not be available after an index k or that new arms may be made available. This work also proposes a variant of UCB1 known as UCB1-$\frac{K}{C}$ (K and C are parameters that are chosen *a-priori*). The work in [8] applies the algorithm to a web-advertisement model, where pulling an arm corresponds to choosing a particular ad to be delivered on a given webpage, the reward being when the viewer clicks this ad. A problem similar to [8], is the 'Irrevocable' Multi-Armed Bandit problem in [11], representing fashion retailers' procurement scenarios.

1.2.1 *Contributions and Outline* : Our contributions in this paper and the relevant sections that cover these are:

1. An algorithm with a decreasing stepsize β_k used in (1) above, and a proof of asymptotic convergence for the same (§2). This is different from the asymptotically efficient algorithm proposed in [10].

2. Modification of the proposed asymptotically convergent algorithm in §3 to obtain the log-regret algorithm SAMWMIX, on the lines of log-squared regret SOFTMIX in [7].

3. A 'blind' algorithm that does not require an input parameter d which both SOFTMIX and SAMWMIX need (§4).

4. An algorithm that adapts the existing UCB1 algorithm to a Boltzmann exploration scheme and is numerically observed to require lesser computation and has better 'precision', i.e., a higher probability of pulling the best arm a larger number of times (§5).

We also provide numerical results for all these algorithms (§6).

2. SAMW WITH DIMINISHING β

We propose the update step in SAMW to be:

$$\phi_{k+1}^i := \frac{\phi_k^i \beta_k^{\mu_k^i}}{\sum_{j=1}^{|A|} \phi_k^j \beta_k^{\mu_k^j}} \tag{2}$$

where μ_k^j is the $k-$sample mean corresponding to arm a^j, and is obtained from rewards $X_k^j \in [0,1]$ obtained by pulling arm a^j. In [10], the β_k are held constant to a small $\beta > 1$ (the result shown, [10, Lemma 3.2], is that for a given number of iterations T, $\beta := \psi(T)$ and $\psi(T) \to 1$ as $T \to \infty$).

Using the optimum policy ϕ^*, define the Kullback-Liebler entropy term as $D_l^* \stackrel{\Delta}{=} \sum_{j=1}^{|A|} \phi^*(j) \cdot \log \frac{\phi^*(j)}{\phi_l^j}$ which, since $\phi^*(a^*) = 1$ with $\phi^*(j) = 0$ for $j \neq a^*$, equals $-\log \phi_l^{a^*}$.

***Theorem* 1** — *For stepsize $\beta_k \stackrel{\Delta}{=} 1 + \beta_k' = 1 + \frac{1}{\log k}$, $k \geq 2$ and update step (2), the average reward from the current policy ϕ_k is s.t. $\sum_{j=1}^{|A|} \phi_k^j \mu_k^j \to \mu^*$ as $k \to \infty$.*

$$
\begin{aligned}
\text{PROOF} : D_{l+1}^* - D_l^* &= \sum_{j=1}^{|A|} \phi^*(j) \log \frac{\phi_l^j}{\phi_{l+1}^j} \\
&= \log \left(\frac{\sum_{j=1}^{|A|} \phi_l^j \beta_l^{X_l^j}}{\beta_l^{X_l^*}} \right) \\
&= X_l^* (-\log \beta_l) + \log \sum_{j=1}^{|A|} \phi_l^j \beta_l^{X_l^j} \\
&\leq X_l^* (-\log \beta_l) + \log \sum_{j=1}^{|A|} \phi_l^j (1 + (\beta_l - 1) X_l^j) \\
&\quad \text{using } \beta^a \leq 1 + (\beta - 1)a, \text{ for } \beta > 1,\ a < 1 \\
&\leq X_l^* (-\log \beta_k) + \log \left(1 + (\beta_l - 1) \sum_{j=1}^{|A|} X_l^j \cdot \phi_l^j \right) \\
&\quad \text{for a large } k \text{ s.t. } l < k, \text{ since } 1 < \beta_k < \beta_l \\
X_l^* &\leq \frac{D_l^* - D_{l+1}^*}{\log \beta_k} + \frac{\beta_l' \bar{X}_l}{\log \beta_k} \\
&\quad \text{with } \bar{X}_l \stackrel{\Delta}{=} \sum_{j=1}^{|A|} \phi_l^j X_l^j \text{ and using } \log(1+a) \leq a, \text{ for } a < 1 \\
\frac{1}{k} \sum_{l=1}^{k} X_l^* &\leq \frac{D_1^* - D_{k+1}^*}{k \log \beta_k} + \frac{1}{k \log \beta_k} \sum_{l=1}^{k} \beta_l' \bar{X}_l \qquad (3) \\
&\quad \text{obtained by adding over all } l \text{ till } k \text{ and dividing by } k.
\end{aligned}
$$

Note that $(D_1^* - D_{k+1}^*) \leq D_1^* = \log |A|$ due to the initialization $\phi_1^j = \frac{1}{|A|}$ and $k \log \beta_k \to \infty$ as $k \to \infty$. Hence, $\frac{D_1^* - D_{k+1}^*}{k \log \beta_k} \to 0$ as $k \to \infty$. Since $\beta_k \to 1$, it is the case that $\phi_k \to \bar{\phi}$ for some probability mass function $\bar{\phi}$ over all actions. The term $\hat{X}_k \stackrel{\Delta}{=} \frac{1}{k \log \beta_k} \sum_{l=1}^{k} \beta_l' \bar{X}_l$ may be viewed as the iterate in a stochastic approximation recursion:

$$\hat{X}_{k+1} \;:=\; \hat{X}_k + \frac{\beta'_{k+1}}{(k+1)\log\beta_{k+1}}\left(\bar{X}_{k+1} - \frac{((k+1)\log\beta_{k+1} - k\log\beta_k)}{\beta'_{k+1}}\hat{X}_k\right).$$

Let $E(\bar{X})$ denote $\sum_{i=1}^{|A|}\bar{\phi}^i\mu^i$ and the martingale difference M_{k+1} be such that $M_{k+1} = \bar{X}_{k+1} - E[\bar{X}_{k+1} \mid \mathcal{F}_k]$. The sequence M_{k+1} above is defined w.r.t. the sigma algebra $\mathcal{F}_k = \sigma(\phi_l^i, \bar{X}_l, 1 \leq l \leq k, 1 \leq i \leq |A|)$. One can then rewrite the above recursion as:

$$\begin{aligned}\hat{X}_{k+1} \;&:=\; \hat{X}_k + \frac{\beta'_{k+1}}{(k+1)\log\beta_{k+1}}\cdot \\ &\qquad \left(E[\bar{X}_{k+1} \mid \mathcal{F}_k] - \frac{((k+1)\log\beta_{k+1} - k\log\beta_k)}{\beta'_{k+1}}\hat{X}_k + M_{k+1}\right) \qquad (4)\\ &:=\; \hat{X}_k + \hat{\beta}_{k+1}\left(E[\bar{X}_{k+1} \mid \mathcal{F}_k] - \bar{\beta}_{k+1}\hat{X}_k + M_{k+1}\right), \qquad (5)\end{aligned}$$

where, $\hat{\beta}_{k+1} \stackrel{\Delta}{=} \frac{\beta'_{k+1}}{(k+1)\log\beta_{k+1}}$ and $\bar{\beta}_{k+1} \stackrel{\Delta}{=} \frac{((k+1)\log\beta_{k+1} - k\log\beta_k)}{\beta'_{k+1}}$, respectively. Now observe that (*i*) $\sum_{l=1}^{k}\hat{\beta}_l \to \infty$ as $k \to \infty$ and (*ii*) $\sum_{l=1}^{\infty}\hat{\beta}_l^2 < \infty$. By letting, $\bar{\beta}_0 = \bar{\beta}_1 = 0.1$, it is easy to see that $0.1 \leq \bar{\beta}_k < 1, \forall k$, and that $\bar{\beta}_k \leq \bar{\beta}_{k+1}, \forall k$. Further, $\bar{\beta}_k \to 1$ as $k \to \infty$.

Define a sequence $\{t(n)\} \subset [0, \infty)$ according to $t(0) = 0$ and $t(n) = \sum_{m=0}^{n-1}\hat{\beta}_m$, $n \geq 1$. Let $\bar{\beta}(t), t \geq 0$ be defined by $\bar{\beta}(t(k)) = \bar{\beta}_{k+1}, k \geq 0$ with linear interpolation (between end points) for $t \in [t(k), t(k+1)], k \geq 0$. Note from the construction that $\bar{\beta}(t) \in [0.1, 1] \forall t \geq 0$. Consider now the following ODE associated with (5):

$$\dot{\hat{X}}(t) = h_t(\hat{X}(t)) \stackrel{\triangle}{=} (E[\bar{X}] - \bar{\beta}(t)\hat{X}(t)). \qquad (6)$$

We shall now apply a key result from [4] (cf. Theorems 2.1-2.2) (alternatively, Theorem 7, Chapter 3.2 of [5]) that is however for the case of a time homogeneous objective function $h(\hat{X}(t))$ unlike ours (that depends explicitly on t). As can be seen, the result of [4] carries over easily in our case as well. Note that $\forall X, Y \in \mathcal{R}$,

$$|h_t(X) - h_t(Y)| \leq \bar{\beta}(t)|X - Y| \leq |X - Y|.$$

Thus, $h_t(X)$ is Lipschitz continuous in X, uniformly in t. Note also that $M_{k+1} = \bar{X}_{k+1} - E[\bar{X}_{k+1} \mid \mathcal{F}_k], k \geq 0$ is a mean-zero process w.r.t. $\mathcal{F}_k$ and that $M_{k+1}^2, k \geq 0$ is bounded above by 1. Further, the step-sizes $\hat{\beta}_l$, $l \geq 1$ satisfy (*i*) and (*ii*) above. Now, let

$$h_{t,r}(X) \stackrel{\Delta}{=} \frac{h_t(rX)}{r} = \frac{E[\bar{X}]}{r} - \bar{\beta}(t)X.$$

Thus, $h_{t,\infty}(X) \stackrel{\Delta}{=} \lim_{r\to\infty} h_{t,r}(X) = -\bar{\beta}(t)X$. Since $\bar{\beta}(t) \geq 0.1 > 0$ (uniformly over t), it follows that the origin is the unique globally asymptotically stable equilibrium for the ODE $\dot{\hat{X}}(t) = h_{t,\infty}(\hat{X}(t))$. Note also that for any $0 < T < \infty$, $\int_0^T \beta(\tau)d\tau \leq \beta(T)T < \infty$.

In the light of the observations made in the previous paragraph, it is easy to verify that the sequence of results in Chapter 3.2 of [5], specifically given in Lemmas 1-2, Corollary 3, Lemmas 4-6 and also Theorem 7 there continue to hold in our setting. Thus, $\sup_k \parallel \hat{X}_k \parallel < \infty$ almost surely (from Theorem 7, Chapter 3.2 of [5]), thereby ensuring almost sure boundedness of the iterates in (5).

Consider now the following ODE in place of (6):

$$\dot{\hat{X}}(t) = E[\bar{X}] - \hat{X}(t), \tag{7}$$

that has $\hat{X}^* = E[\bar{X}]$ as its unique globally asymptotically stable equilibrium. Now rewrite (5) as follows:

$$\hat{X}_{k+1} = \hat{X}_k + \hat{\beta}_{k+1}\left(E[\bar{X}_{k+1} \mid \mathcal{F}_k] - \hat{X}_k + M_{k+1} + \epsilon(k)\right),$$

where $\epsilon(k) = \hat{X}_k - \bar{\beta}_{k+1}\hat{X}_k$ is uniformly bounded by the foregoing and further, $\epsilon(k) \to 0$ as $k \to \infty$ almost surely (since $\bar{\beta}_{k+1} \to 1$ as $k \to \infty$). It now follows as a consequence of the third extension in Chapter 2.2 of [5] that $\hat{X}_k \to \hat{X}^*$ almost surely as $k \to \infty$. However, from (3), this implies that the empirical mean μ_k^* obtained from the best arm is s.t. $\mu_k^* \leq E(\bar{X})$ as $k \to \infty$. This cannot be true unless equality holds and therefore $\phi_k \to \phi^*$. □

Each arm a^j in A is sampled once per iteration k and this contributes to the high sampling budget of the algorithm in (2) (as well as the original in [10]). This high sampling budget comes about because all sub-optimal actions a^j are taken at each iteration k before a high level of confidence to infer the best action a^* is achieved at some iteration $k << K$. To mitigate this problem, we employ the structure of SAMW in the SOFTMIX algorithm of [7]. In SOFTMIX, $k-$sample means μ_k^i are used at the $k-$th step of the iteration (by a suitable construction), even though only a 'winner' action (call it $\hat{a}_k$) is actually taken at each iteration k. Such behaviour is typical of Stochastic Multi-Armed Bandit (SMAB) algorithms where only one arm is pulled in each iteration k of the algorithm.

3. LOG-REGRET SOFTMIX

We wish to obtain the effect of using $k-$sample means for each action j while performing update k in (2). This is achieved in a separate algorithm named SOFTMIX in [7] by considering reward $X_k^j = \frac{\hat{X}_k}{\phi_k^j}$

if arm a^j is the winner arm $\hat{a}_k$ (i.e. the arm pulled after ϕ_{k-1} is sampled), and 0 otherwise. Here $\hat{X}_k$ is the reward obtained by pulling the winner arm $\hat{a}_k$. A change in SOFTMIX algorithm, by giving it an SAMW-like kernel (see (2) above), helps us to obtain logarithmic regret. Thus any sub-optimal action $a^i \in A\backslash\{a^*\}$ will be taken only $O(\log k)$ times in k trials of the system. This has been proved as the best possible performance of any algorithm designed for the SMAB problem, with SOFTMIX capable of only $O(\log^2 k)$ regret. Also note the remark in [15] that a diminishing step-size algorithm (termed 'decreasing $\epsilon-$greedy') like SOFTMIX that achieves logarithmic regret was yet to be found.

We summarize the SOFTMIX algorithm in the following. Define an indicator function I^j_{k+1} which takes value 1 for a^j if j is selected by sampling iterate ϕ_k, $I^j_{k+1} = 0$ for all other actions in $A\backslash\{a^j\}$. For each arm $i \in \{1, 2, \ldots, |A|\}$, perform the following updates:

$$\phi^i_{k+1} = (1-\gamma_k)\frac{e^{\eta_k \hat{s}^i_k}}{\sum_{j=1}^{|A|} e^{\eta_k \hat{s}^j_k}} + \frac{\gamma_k}{|A|}, \tag{8}$$

$$\hat{s}^j_{k+1} = \hat{s}^j_k + \frac{\hat{X}_k}{\phi^j_k} I^j_k. \tag{9}$$

The step-sizes γ_k and η_k are calculated as: $\gamma_k = 1$ for $k = 1, 2$

$$\gamma_k = \min\left(1, \frac{5|A|\log(k-1)}{d^2 \cdot (k-1)}\right) \text{ for } k > 2,$$

$$\eta_k = \frac{1}{\frac{|A|}{\gamma_k} + 1} \log\left(1 + \frac{d(\frac{|A|}{\gamma_k} + 1)}{2\frac{|A|}{\gamma_k} - d^2}\right).$$

In the above, the quantity d is a heuristic value input by the programmer and satisfies the condition that $0 < d < \min\{\Delta^i, a^i \neq (a^*)\}$. Here, $\Delta^i \stackrel{\Delta}{=} \mu^* - \mu^i$ is the mean loss incurred upon taking action a^i.

The proposed algorithm SAMWMIX is:

$$\phi^i_{k+1} = (1-\gamma_k)\frac{e^{\sum_{s=1}^k \eta_s \hat{X}^i_s}}{\sum_{j=1}^{|A|} e^{\sum_{s=1}^k \eta_s \hat{X}^i_s}} + \frac{\gamma_k}{|A|}. \tag{10}$$

Here, $\hat{X}^i_k$ is obtained using the reward-modification scheme of SOFTMIX, i.e. $\hat{X}^i_k \stackrel{\Delta}{=} \frac{\hat{X}_k}{\phi^i_k} I^i_k$. The stepsizes γ_k and η_k are computed as follows:

$$\gamma_k = \min\left(1, \frac{5|A|}{d^2 \cdot k}\right) \text{ for } k > 0. \tag{11}$$

The stepsize η_k is the same as in SOFTMIX. Note the absence of $\log(k-1)$ from the numerator in the expression for γ_k above. Also observe the term $\frac{\gamma_k}{|A|}$ in (10) which is essentially an 'explore' component that was missing in the original SAMW algorithm in (2).

The proposed algorithm (10) performs a gradual weighting of samples $\{\hat{X}_s^i\}_{s=1}^k$ with all scale-factors $\{\eta_s\}_{s=1}^k$ whereas SOFTMIX performs a multiplication of the entire sum $\sum_{s=1}^k \hat{X}_s^i$ with η_k. Due to the similarities, a comparison of the time-varying algorithm EXP3 in [14] (based on the original EXP3 in [3]) with the proposed SAMWMIX is also in order:

- The high-probability result in [14] bounds the regret as O($\sqrt{k}$) in k plays of the bandit.
- The companion algorithm EXP3ELM (read as 'EXP3 with Action Elimination') in [14] does achieve a logarithmic regret bound. However, in all the typical experiments no action elimination takes place and the algorithm is treated to be on par with EXP3 itself.

A brief comparison with the stochastic policy -based method in [1], which uses Thomson Sampling and is the first to show logarithmic expected regret, is also due. Using d from (11) above, for the general N-armed bandit problem, the expected regret in [1] is proportional to $\frac{N}{d^4}$ whereas ours is proportional to $\frac{N}{d^2}$ (see (15) below).

We now give a proof of logarithmic regret in SAMWMIX, and obtain an upper bound of $O(\frac{1}{k})$ on $E\{\phi_k^i\}$ for any suboptimal arm a^i. This implies logarithmic regret, since $\sum_{p=1}^k \frac{\Delta_i}{p} = O(\log k)$. Define the $\sigma-$algebra $\mathcal{F}_k^i$ as $\sigma(\hat{X}_p^i, 1 \leq p \leq k)$, $k \geq 1$. In the following theorems, we let $\phi_k^i = P\{I_{k+1}^i = 1|\mathcal{F}_k\}$.

***Theorem* 2** — *The probability of selecting action $a^i \in A\backslash\{a^*\}$ at step k of (10) is s.t. $E(\phi_k^i) = O\left(\frac{1}{k}\right)$.*

PROOF : The early part of our treatment is similar to [7, Proof of Theorem 3.1, eq. (10)]:

$$\phi_k^i \quad \leq \quad (1-\gamma_k)\exp\Big(\sum_{p=1}^{k-1}\eta_s(\hat{X}_p^i - \hat{X}_p^*)\Big) + \frac{\gamma_k}{|A|}. \tag{12}$$

Now consider $Z_p^i \stackrel{\Delta}{=} \hat{X}_p^i - \hat{X}_p^* + \Delta_i$ and note that $E\{Z_p^i|\mathcal{F}_{p-1}\} = 0$. Also, due to the form of $\hat{X}_p^i = \frac{\hat{X}_p}{\phi_p^i} I_p^i$ we have for $c_p = \frac{|A|}{\gamma_p} + 1$, the inequality $Z_p^i \leq c_p$. Using the same form of $\hat{X}_p^i$, we obtain that $E\{(Z_p^i)^2|\mathcal{F}_{p-1}\} \leq \frac{2|A|}{\gamma_p} - \Delta_i^2 \leq \sigma_p^2 \stackrel{\Delta}{=} \frac{2|A|}{\gamma_p} - d^2$. Since $c_p > 0$, for the function $\xi_{c_p}(\eta) = (e^{c_p\eta} - 1 - c_p\eta)/{c_p}^2$ we may use the inequality $e^{\eta z} \leq 1 + \eta z + \xi_{c_p}(\eta)z^2$ for every $z < c_p$.

Thus, we also have for each p:

$$\begin{aligned} E\{e^{\eta_p Z_p^i}|\mathcal{F}_{p-1}\} &\leq E\{1+\eta_p Z_p^i + \left(Z_p^i\right)^2 \phi_{c_p}(\eta_p)|\mathcal{F}_{p-1}\} \\ &\leq 1+\sigma_p^2\phi_{c_p}(\eta_p) \leq e^{\sigma_p^2\phi_{c_p}(\eta_p).} \end{aligned}$$

Thus, we have that:

$$E(\phi_k^i) \leq \frac{\gamma_k}{|A|} + (1-\gamma_k)\exp\left(-\sum_{p=1}^{k-1}\left(\eta_p d - \xi_{c_p}(\eta_p)\sigma_p^2\right)\right). \tag{13}$$

Consider $K_p \triangleq d$ and $\sigma_p^2 \triangleq \frac{2|A|}{\gamma_p} - d^2$, making each term in the sum of the RHS above as $-K_p\eta_p + \xi_{c_p}(\eta_p)\sigma_p^2$. Now replace, as per the definition in [7, Fact 5.1], $\xi_{c_p}(\eta_p) = \frac{e^{c_p\eta_p}-1-c_p\eta_p}{c_p^2}$. From [7, Proof of Theorem 3.1], applying $\log(1+x) \geq \frac{2x}{2+x}$ to η_p (which can be represented as $\frac{1}{c_p}\log(1+\frac{K_p c_p}{\sigma_p^2})$), we get that $\eta_p \geq \frac{2K_p}{2\sigma_p^2+c_pK_p}$. Consequently, $\xi_{c_p}(\eta_p) \leq \frac{e^{c_p\eta_p}-1}{c_p^2} - \frac{2K_p}{c_p(2\sigma_p^2+c_pK_p)}$ where $e^{c_p\eta_p}-1 = \frac{K_pc_p}{\sigma_p^2}$ due to the representation of η_p as $\frac{1}{c_p}\log(1+\frac{K_pc_p}{\sigma_p^2})$. Subtraction among the two fractions above yields $\xi_{c_p}(\eta_p) \leq \frac{K_p^2}{\sigma_p^2(2\sigma_p^2+c_pK_p)}$.

For each term in the sum on the RHS of (13), we have: $-K_p\eta_p + \xi_{c_p}(\eta_p)\sigma_p^2 \leq \frac{-K_p^2}{2\sigma_p^2+c_pK_p}$. Rewrite (13) as $E(\phi_k^i) \leq \exp\left(-\sum_{p=1}^{k-1}\frac{K_p^2}{2\sigma_p^2+c_pK_p}\right) + \frac{\gamma_k}{|A|}$.

After substituting for c_p, K_p, σ_p^2 and choosing an upper bound of $5|A|$, with $p \geq P$ for a finite P, for the resulting denominator $4|A| + d|A| + \gamma_p(d-2d^2)$, we have:

$$\frac{K_p^2}{2\sigma_p^2+c_pK_p} \geq \frac{d^2\gamma_p}{5|A|}, \text{ and,} \tag{14}$$

$$E(\phi_k^i) \leq c\cdot\exp(-\log(k-1)) + \frac{5|A|}{d^2k}. \tag{15}$$

Constant c represents regret accumulated for indices $p < P$ and step-size γ_p is inferred using (11) above. Thus, $E(\phi_k^i) = O(\frac{1}{k})$. □

This result gives expected value of regret for any iteration k (and not only as $k \to \infty$), and is achieved without use of any concentration inequalities.

Modify the stepsize $\eta_k = \frac{1}{c_k}\log(1+\frac{dc_p}{\sigma_p^2})$ such that $c_k = 2|A|/\gamma_k + 1$ and $\sigma_k^2 = c_k - 1$. Log-regret holds as the inequalities $Z_k^i \leq c_k$ and $E\{(Z_k^i)^2|\mathcal{F}_{k-1}\} \leq \sigma_k^2$ in the proof above continue to hold. Assuming that the algorithm runs for a maximum of P iterations, a result indicating the low risk of taking sub-optimal action a^i at step k is now obtained.

***Theorem* 3** — *For $k \leq P$, using $\eta_k = \frac{1}{c_k}\log(1+\frac{dc_p}{\sigma_p^2})$ with $c_k = \frac{2|A|}{\gamma_k}+1$, $\sigma_k^2 = \frac{2|A|}{\gamma_k}$ and $\gamma_k = \min(1, \frac{5|A|}{d^2k})$, the quantity ϕ_k^i for $a^i \in A\backslash\{a^*\}$ in (10) is such that*

$$\frac{\gamma_k}{|A|} \leq \phi_k^i \leq (1-\gamma_k)\cdot\frac{1}{k} + \frac{\gamma_k}{|A|},$$

with probability $1-k(\alpha_P^k)$ for an α_P s.t. $0<\alpha_P<1$

PROOF : The LHS of the inequality is easy to observe: it is the exploration probability for action a^i. Observe from [7, eq. (9)] that $\phi_k^i \leq (1-\gamma_k)\exp(\sum_{s=1}^{k-1}\eta_s(\hat{X}_s^i - \hat{X}_s^*)) + \frac{\gamma_k}{|A|}$, where we define $\hat{Z}_s^i \stackrel{\Delta}{=} \hat{X}_s^i - \hat{X}_s^*$. Using the Markov inequality, we have $P\{\exp(\sum_{s=1}^{k-1}\eta_s\hat{Z}_s^i) > \frac{1}{k}\} \leq kE\{\exp(\sum_{s=1}^{k-1}\eta_s\hat{Z}_s^i)\}$. Consider another random variable $Z_{k-1}^i = \Pi_{s=1}^{k-1}\exp(\eta_s\hat{Z}_s^i)$ and note that $Z_{k-1}^i = \hat{Z}_{k-1}^i\cdot Z_{k-2}^i$. Also, $E(Z_{k-1}^i) \stackrel{\Delta}{=} E(Z_{k-1}^i|\mathcal{F}_{k-2})$ where $\mathcal{F}_{k-1}$ is $\sigma(\hat{X}_p^i, 1\leq p\leq k-2)$, $k\geq 3$. From the proof in Theorem 2, we have that $E\{\hat{Z}_{k-1}^i|\mathcal{F}_{k-2}\} = -\Delta^i$, $E\{(\hat{Z}_{k-1}^i)^2|\mathcal{F}_{k-2}\} \leq \frac{2|A|}{\gamma_{k-1}}$ and $\hat{Z}_{k-1}^i \leq \frac{|A|}{\gamma_{k-1}}$, hence $\frac{1}{c_{k-1}}\hat{Z}_{k-1}^i \leq \frac{1}{2}$. Further, since $\log(1+\frac{dc_{k-1}}{\sigma_{k-1}^2}) = \log(1+(1+\frac{\gamma_{k-1}}{2|A|})d) < 1$ for a small d, we have that $|\eta_{k-1}\hat{Z}_{k-1}^i| \leq 1$. Use the inequality $e^a \leq 1+a+a^2$ for $|a|\leq 1$ to obtain:

$$\begin{aligned} E(Z_k^i) &\leq E(1+\eta_{k-1}\hat{Z}_{k-1}^i + \eta_{k-1}^2(\hat{Z}_{k-1}^i)^2)\cdot Z_{k-1}^i \\ E(Z_k^i) &\leq (1-\eta_{k-1}\Delta^i + \eta_{k-1}^2\frac{2|A|}{\gamma_{k-1}})\cdot Z_{k-1}^i \end{aligned} \tag{16}$$

In the above, $\eta_{k-1}(\Delta^i - \eta_{k-1}\frac{2|A|}{\gamma_{k-1}}) \geq \eta_{k-1}(d - \frac{d}{\sqrt{d/\xi_{k-1}+1}}) \geq \bar{\alpha}_P$, where $\xi_{k-1} = \frac{\sigma_{k-1}^2}{c_{k-1}}$ and $\bar{\alpha}_P > 0$. To see this, notice that $\eta_{k-1}\frac{2|A|}{\gamma_{k-1}} = \xi_{k-1}\log(1+\frac{d}{\xi_{k-1}})$ where $\xi_{k-1}<1$, and then employ the inequality $\log(1+a) \leq \frac{a}{\sqrt{1+a}}$ for $a>0$. Thus, with $\alpha_P = 1-\bar{\alpha}_P$, the above (16) becomes: $E(Z_k^i) \leq \alpha_P\cdot Z_{k-1}^i$. Thus the adverse probablity $P\{\exp(\sum_{s=1}^{k-1}\eta_s\hat{Z}_s^i) > \frac{1}{k}\}$ is less than $k\alpha_P^k$. □

4. BLIND SAMWMIX

In SAMWMIX, an intelligent guess of d was still needed as input to the stepsize γ_k. Let us suppose we have a simple technique to avoid guessing d and hence we choose $\gamma_k = \frac{5|A|\log k}{k}$. Then, using (14), for all $k > e^{\frac{1}{d^2}}$, we would have that $\frac{d^2\log k}{k} \geq \frac{1}{k}$. However, this would result in *log–squared* regret due to a trailing term $\frac{5|A|\log k}{k}$ corresponding to $\frac{5|A|}{d^2k}$ in (15). We propose an alternative where d is not needed as input, and nearly log–regret is achieved. We retain the update step (10), define $c_p = 1+\frac{|A|}{\gamma_p}$ and $\sigma_p^2 = 2\frac{|A|}{\gamma_p}$. With $\alpha\in(0,0.5)$, we use modified step-sizes γ_p and η_p: $\gamma_p = \min\left(1, 5|A|\cdot\frac{\log^{2\alpha}p}{p}\right)$, $\eta_p = \frac{1}{c_p}\log\left(1+\frac{\frac{1}{\log^\alpha p}c_p}{\sigma_p^2}\right)$.

Theorem 4 *— Using stepsizes γ_p and η_p, the probability of selecting an action $a^i \in A\backslash\{a^*\}$ at step k of (10) is such that $E(\phi_k^i) = O\left(\frac{log^{2\alpha} k}{k}\right)$.*

PROOF : We define a diminishing stepsize $K_p \stackrel{\Delta}{=} \frac{1}{\log^{\alpha} p}$. Thus we note, as regards the term within the summation in RHS of (13) above, that for $p \geq P$:

$$-\eta_p d + \xi_{c_p}(\eta_p)\sigma_p^2 - \xi_{c_p}(\eta_p)d^2 \quad \leq \quad -\eta_p K_p + \xi_{c_p}(\eta_p)\sigma_p^2,$$

where we use the fact that $\xi_{c_p}(\eta_p) \geq 0$ and $P = e^{d^{-\frac{1}{\alpha}}}$. As in the proof of Theorem 2, we can bound the quantity in RHS by $\frac{-K_p^2}{2\sigma_p^2 + c_p K_p}$ and further by $\frac{\gamma_p \cdot \log^{-2\alpha} p}{5|A|}$. Now substitute for γ_p to obtain $\frac{1}{p}$ as the upper-bound for this term. However, the 'explore' term $\frac{\gamma_k}{|A|}$ in (10) results in $E(\phi_k^i) = O(\frac{\log^{2\alpha} k}{k})$ as in the statement of this theorem. □

Thus the regret accumulated is $O(\log(k)^{1+2\alpha})$, which can be made arbitrarily close to log-regret by choosing a small α. However, the smaller an α is, the larger will be the iterate index P (the bounds derived would hold only for $p > P$).

5. UCB1-LIKE SOFTMIX

The log-regret algorithm UCB1 in [2] has been applied in the past to finite–horizon MDPs (cf. [9]), in order to reduce the number of times simulated transitions are made to arrive at an optimal policy. We now propose a way to use the UCB1 of [2] within the SOFTMIX algorithm, thereby achieving log-regret as also avoiding a search for a maximum among $|A|$ values needed in each iteration k of UCB1. Finding a maximum requires $|A|$ comparisons, although there exist optimizations. Define the c_k^i-sample mean for arm i as $\mu_k^i = \frac{1}{c_k^i}\sum_{p=1}^{k} \hat{X}_p I_p^i$, and a term similar to the UCB1 'confidence' term $b_k^i \stackrel{\Delta}{=} \sqrt{\frac{2\log k}{c_k^i}}$, where $c_k^i = \sum_{p=1}^{k} I_p^i$ is the number of times arm i has been played. The method UCB1 adopts is to find the maximum among $\{\mu_k^i + b_k^i\}$, $\forall i \in A$, at each step k and to play this action as $\hat{a}_{k+1}$.

For the proposed algorithm UCB1MIX, we will use an 'explore' component similar to (10). Also note that the result obtained is a bound on the *instantaneous* regret - unlike UCB1 which bounds *total* regret. Consider step-sizes:

$$\delta_k = 2\log^{1+2\alpha} k, \qquad b_k^i = \sqrt{\frac{1.5\log^{1+\alpha} k}{c_k^i}}, \qquad S_k = \log^{-\alpha} k,$$
$$T_k = \tfrac{1}{k^2}, \qquad \eta_k = \tfrac{1}{T_k}\log(1 + S_k T_k), \quad \gamma_k = \tfrac{(1+\beta)\log^{\beta} k}{k}.$$

Using the condition $0 < \alpha < \beta < 0.5$, our algorithm is:

$$\phi^i_{k+1} \quad := \quad (1-\gamma_k)\frac{e^{\eta_k \delta_k(\mu^i_k + b^i_k)}}{\sum_{j=1}^{|A|} e^{\eta_k \delta_k(\mu^j_k + b^j_k)}} + \frac{\gamma_k}{|A|}. \tag{17}$$

***Theorem* 5** — *Assuming that $\mu^i_k \geq \delta > 0$, the probability of selecting an action $a^i \in A \backslash \{a^*\}$ at step k of (17) is s.t. $E(\phi^i_k) = O\left(\frac{log^\beta k}{k}\right)$.*

PROOF : Define Z^i_k as $(\mu^i_k + b^i_k) - (\mu^*_k + b^*_k)$ and note that $\eta_k \approx S_k$ due to $S_k T_k \to 0$. Note that $E(c^i_k) \geq \log^{1+\beta} k$ for all $a^i \in A$, due to the $\frac{\gamma_k}{|A|}$ 'explore' term in (17). Consequently, $Z^i_k \leq \mu^i_k + b^i_k \leq 1$ once $c^i_k \geq \frac{1.5 \log^{1+\alpha} k}{(1-\mu^i_k)^2}$. If we assume that $\mu^i_k \geq \delta > 0$, $\forall k, i$, then $Z^i_k \leq \mu^i_k + b^i_k \leq 1$ is achieved within finite k since $\beta > \alpha$.

Further, conditioning on $Z^i_k > 0$,

$$\begin{aligned} e^{\eta_k \delta_k Z^i_k} &\leq e^{2\log^{1+\alpha} k} \text{ since } \eta_k \approx S_k. \\ P(Z^i_k \geq 0) &\leq e^{-3\log^{1+\alpha} k} \text{ for } k > \frac{1.5\log^{1+\alpha} s}{(\Delta^i)^2}. \end{aligned}$$

In the above, we have drawn from the analysis preceding [2, (6)]. Hence, after a finite k,

$$E(Z^i_k | Z^i_k > 0) \cdot P(Z^i_k > 0) \leq e^{-\log^{1+\alpha} k} \leq \frac{1}{k}.$$

We condition next on $Z^i_k \leq 0$. For each action a^i, there exists a $\tilde{d^i} > 0$ s.t. $\tilde{d^i}\Delta_i \leq 1$ and $(\tilde{d^i}+1)\Delta_j \geq 1$. Now, note that $E(Z^i_k) \leq -d^i$ for some $d^i > 0$ once $c^i_k \geq \frac{1.5\log^{1+\alpha} k}{(1-\tilde{d^i}\Delta^i)^2}$. Thus, $e^{\eta_k \delta_k Z^i_k} \leq \Pi_{r=1}^{\lfloor \delta_k \rfloor} e^{\eta_k \cdot Z^i_k}$ and therefore, using [7, Fact 5.1], $e^{\eta_k Z^i_k} \leq 1 + \eta_k Z^i_k + \xi_{T_k}(\eta_k) \cdot (Z^i_k)^2$.

Applying expectations, we have that $E(e^{\eta_k Z^i_k}) \leq 1 - \eta_k d^i + \xi_{T_k}(\eta_k)$ since $E((Z^i_k)^2 | \mathcal{F}_{k-1}) \leq 1$ for $k > k_0$. Thus, $E(e^{\eta_k Z^i_k}) \leq e^{(-\eta_k S_k + \xi_{T_k}(\eta_k)\sigma^2_k)}$ and hence, $E(e^{\eta_k Z^i_k}) \leq \exp(\frac{-S^2_k}{2+S_k T_k})$. Therefore, $E(e^{\eta_k \delta_k \cdot Z^i_k}) \leq \exp(\frac{-\lfloor \delta_k \rfloor S^2_k}{2+S_k T_k})$. The condition $E((Z^i_k)^2 | \mathcal{F}_{k-1}) \leq 1$ is achieved for suitable $k > k_0$, i.e., $k > \max\left(\frac{1.5\log^{1+\alpha} k}{(1-\mu^*_k)^2}, \frac{1.5\log^{1+\alpha} k}{(1-\mu^i_k)^2}\right)$. We have used again, the previous assumption that $\mu^i_k > \delta > 0$, $\forall i, k$. Now, the fact that $S_k T_k \to 0$ and $-\delta_k S^2_k = -2\log k$ mean that $E(Z^i_k | Z^i_k \leq 0) \cdot P(Z^i_k \leq 0) \leq \frac{1}{k}$. However, the 'explore' term in (17), $\frac{\gamma_k}{|A|} = \frac{(1+\beta)\log^\beta k}{|A| \cdot k}$, results in the statement of the theorem. □

The β in the 'explore' term can be made arbitrarily small. However, just as for the α in §4, an arbitrarily small β will result in a larger threshold P (only above this threshold would the nearly log-regret bounds hold).

5.1 *Alternative UCB1-like scheme*

An alternative algorithm to obtain logarithmic regret in the manner of UCB1 but without the explicit maximization is given below. The advantage of this algorithm is the use of a single, simple stepsize γ_k as also achievement of exact log-regret (although there is a scale-factor exponential in the number of arms $|A|$).

Consider $\gamma_k = \frac{1}{k}$ and confidence terms $b_k^i = \sqrt{\frac{\log k}{c_k^i}}$ to define this update step:

$$\phi_{k+1}^i \quad := \quad (1-\gamma_k)\frac{e^{\mu_k^i + b_k^i}}{\sum_{j=1}^{|A|} e^{\mu_k^j + b_k^j}} + \frac{\gamma_k}{|A|}. \tag{18}$$

***Theorem* 6** — *Using algorithm (18), probability ϕ_k^i of playing an action a^i for $a^i \in A\backslash\{a^*\}$ is $O(\frac{1}{k})$.*

PROOF : Define difference terms Z_k^i as used above: $Z_k^i = \mu_k^i + b_k^i - \mu_k^* - b_k^*$. Also note that $E(\phi_{k+1}^i) \leq (1-\frac{1}{k})E(e^{Z_k^i}) + \frac{1}{k\cdot|A|}$. Now condition on $Z_k \leq 0$ to obtain a bound on ϕ_{k+1}^i when $a^i \in A\backslash\{a^*\}$. Using [2, (7)-(9), Theorem 1] for $k \geq \frac{8\log k}{\Delta^i}$ (where, as before, $\Delta^i = \mu^* - \mu^i$), we have that $P(\mu_k^i + b_k^i \geq \mu_k^* + b_k^*) \leq \frac{1}{k}$. To obtain a bound on the random variable $\exp\left(\mu_k^i + b_k^i - \mu_k^* - b_k^*\right)$, note that $\mu_k^i - \mu_k^* \leq 1$ and that $E(c_k^i) \geq \frac{\log k}{|A|}$ due to the 'explore' term, i.e. $\phi_p^i \geq \frac{1}{p\cdot|A|}$ for $p \in \mathcal{Z}_+$. Also, for the same reason, $E(\sqrt{\frac{\log k}{c_k^i}}) \leq \sqrt{|A|}$. Thus, we have that $e^{\mu_k^i + b_k^i - \mu_k^* - b_k^*} \leq e^{1+\sqrt{|A|}}$. Hence, $E(\phi_{k+1}^i) \leq (1-\frac{1}{k})(e^{1+\sqrt{|A|}}\cdot\frac{1}{k}) + \frac{1}{k\cdot|A|}$ and therefore, $E(\phi_{k+1}^i) \leq \frac{e^{|A|}}{k}$. This proves the statement.□

This confirms $(|A|-1)e^{|A|}\log k$ as the upper–bound for total expected regret till step k.

6. NUMERICAL RESULTS

We performed numerical experiments on all proposed algorithms using the computational software package SciLab. We assumed all rewards $\hat{X}_k \in (0,1)$ and that the rewards were being drawn uniformly from intervals (A_i, B_i) s.t. $0 < A_i < B_i < 1$. We used the rule that $|A| = 5$, and $|\mu^i - \mu^j| < 0.3$, $\forall i,j \in \{1,\ldots,|A|\}$, thus placing all means μ^i close apart and inducing a high degree of difficulty for all algorithms. To compare SAMW of constant stepsize with SAMW of diminishing stepsize, we conducted 1000 experiments for both algorithms with the additional constraint that $|\mu^i - \mu^j| < 0.1$, and with total pulls T being $T = l$ s.t. $1 \leq l \leq 500$. For constant stepsize SAMW, we use the assignment $\beta := \psi(T)$ from [10, Lemma 3.2], where $\psi(T) = 1 + \frac{1}{\log T}$ (this satisfies the condition that $\psi(T) \to 1$ as $T \to \infty$). For diminishing stepsize SAMW of §2 above, we

require that $\beta_k = 1 + \frac{1}{\log k}$. We plot the average value of ϕ_T^*, note in Figure 1 that using diminishing stepsize β_k results in upto 15% higher ϕ_T^*.

To maintain well-posedness of the SMAB algorithms, we used the condition $|\mu^i - \mu^*| > 0.1$, $\forall i \in \{1, \ldots, |A|\}$. For each SMAB algorithm, we considered 1000 cases each of $5-$armed bandits with A_i, B_i chosen randomly and with proximity conditions on μ^i as given above. In each case, we calculated the number of pulls of the best arm and the number of pulls for the second-best arm within a total of 2000 pulls. For better averaging effects in the SAMWMIX and blind-SAMWMIX algorithms, we applied the update rule $\phi_{k+1}^i := (1-\gamma_k)\frac{e^{\sum_{p=1}^k \eta_p \frac{X_p^i}{\phi_p^i}}}{\sum_{j=1}^{|A|} e^{\sum_{p=1}^k \eta_p \frac{X_p^j}{\phi_p^j}}} + \frac{\gamma_k}{|A|}$. This does not change the analyses presented in Theorems 2 and 3 above.

We used an optimization module available in SciLab to compute the intial index k_0 for SOFTMIX as well as blind-SAMWMIX. We observed better stability of the algorithm when $\alpha = 0.5$ for blind-SAMWMIX. Also, $\beta = 0.2$ and $\alpha = 0.1$ were used for UCB1MIX. We averaged the number of pulls of the best-arm as well as the second-best arm over the 1000 cases. Note the superior performance of SAMWMIX (Figure 2) and, even more so, of blind-SAMWMIX (Figure 3). Notice that only the first UCB1-like algorithm, UCB1MIX, performs better than SAMWMIX (Figure 4), yet slightly worse than blind-SAMWMIX.

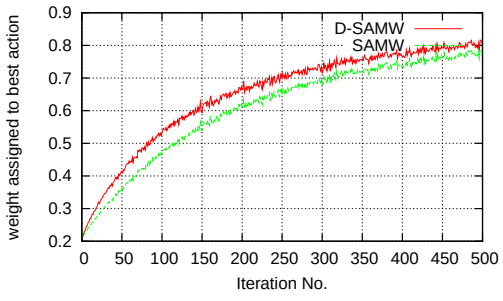

Figure 1 : Comparison of Constant-Stepsize SAMW with Diminishing-Stepsize SAMW

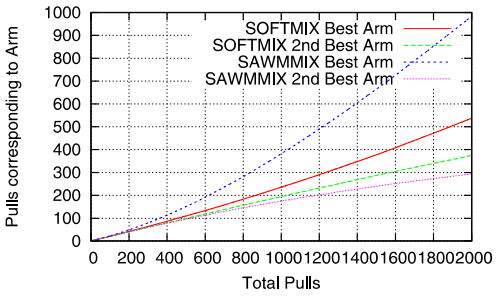

Figure 2 : SAMWMIX vs. SOFTMIX

As regards computational complexity, we ran an optimized version of UCB1MIX where the sum $\mu_k^i + b_k^i$, as in (17) above, is not re-calculated at every k, rather $\mu_{k_i}^i + b_{k_i}^i$ is used, where index k_i is the latest index prior to k at which arm a^i was pulled. The scalar multipliers of $\mu_k^i + b_k^i$ in (17) are also chosen to be $\eta_{k_i}^i$ and $\delta_{k_i}^i$ rather than η_k^i and δ_k^i. The proof in §5 doesn't change, but the calculation of

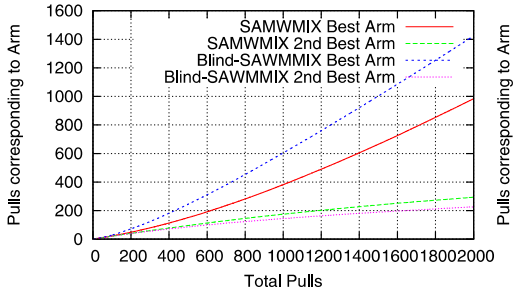

Figure 3 : Comparison of SAMWMIX with blind-SAMWMIX

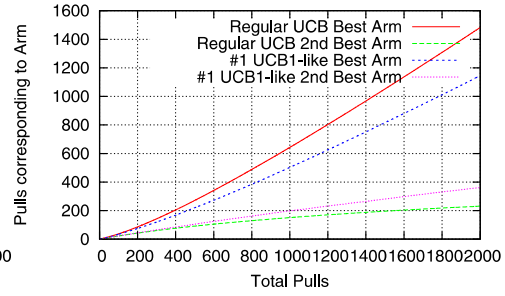

Figure 4 : Regular UCB 1 vs. UCB1MIX

the fraction $\frac{e^{\eta_{k_i}\delta_{k_i}\left(\mu^i_{k_i}+b^i_{k_i}\right)}}{\sum_{j=1}^{|A|} e^{\eta_{k_j}\delta_{k_j}\left(\mu^j_{k_j}+b^j_{k_j}\right)}}$ is easier as only one of the $|A|$ possible numerators has changed from iteration $k-1$. There are only $O(\log|A|)$ comparisons due to the binary search employed in generating an action $\hat{a}_k$ from iterate ϕ_{k-1}. In contrast, the UCB1 algorithm requires, in each iteration k, $O(|A|)$ comparisons to determine the maximum $\mu^i_{k-1}+b^i_{k-1}$ term. For $|A|>50$, the total number of arithmetic/relational operations in UCB1 overtook UCB1MIX. We also compared the number of 'bad runs' for UCB1 vis-a-vis UCB1MIX, analogous to the comparison in Figure 1. A bad run was declared if $c^*_{5000} < \frac{5000}{|A|}$ or if $c^*_{5000} < 1.1\cdot\left(c^{2,*}_{5000}\right)$ where $c^{2,*}_{5000}$ stands for the number of pulls of the second best arm in a total of 5000 pulls. In UCB1, 154 experiments out of a total of 1000 (around 15%) were bad runs, whereas only 1 out of 1000 experiments for UCB1MIX displayed this characteristic. However, whenever a good run is observed, UCB1 was much better in regret terms: in a good run (resp. bad run) it showed an average of 4641 (resp. 2) pulls of the best arm compared to just 254 (resp. 200) in UCB1MIX.

7. Conclusions and Future Directions

In this paper we have proposed a horizon-independent version of Simulated Annealing with Multiplicative Weights (SAMW) by modifying the learning rate. We have also modified the existing SOFTMIX algorithm, a stochastic policy -based stochastic multi-armed bandit (SMAB) algorithm, to obtain the lowest possible logarithmic expected regret in the proposed SAMWMIX (the original SOFTMIX is log-squared regret). An inconvenience with both SOFTMIX and SAMWMIX was the need to specify an input parameter d, which we eliminated with Blind-SAMWMIX - although it obtains slightly worse than logarithmic regret as a result. Finally, we proposed UCB1MIX, a stochastic policy -based algorithm adapting the existing UCB1 to a Boltzmann exploration scheme like SAMWMIX. We have

also given a description of the numerical experiments with each algorithm, comparing them with predecessor algorithms such as SAMW, SOFTMIX and UCB1. As part of future work, an algorithm that uses a tighter version of the inequality in (12) above is under development. Also, the SAMWMIX kernel appears to be of use for 'Contextual Bandits' (cf. [6, Chapter 4]) - a category of bandit problems different from SMABs - and an algorithm for the same is also under development.

REFERENCES

1. S. Agrawal and N. Goyal, Analysis of Thompson sampling for the multi-armed bandit problem, in: *Proc. Intl. Conf. on Learning Theory (COLT)*, (2012).
2. P. Auer, N. Cesa-Bianchi and P. Fischer, Finite-time analysis of the multiarmed bandit problem, *Machine Learning*, **47** (2002a), 235-256.
3. P. Auer, N. Cesa-Bianchi, Y. Freund and R. E. Schapire, The non-stochastic multiarmed bandit problem, *SIAM Journal of Computing*, **32** (2002b), 48-77.
4. V. Borkar and S. Meyn, The ODE method for convergence of stochastic approximation and reinforcement learning, *SIAM Journal on Control and Optimization*, **38** (2000), 447-469.
5. V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*, Cambridge University Press and Hindustan Book Agency (Jointly Published) (2008).
6. S. Bubeck and N. Cesa-Bianchi, Regret analysis of stochastic and non-stochastic multi-armed bandit problems, *Foundations and Trends in Machine Learning*, **5** (2012), 1-122.
7. N. Cesa-Bianchi and P. Fischer, Finite-time regret bounds for the multi-armed bandit problem, in: *Proc. 15th International Conf. on Machine Learning (ICML)* (1998).
8. D. Chakrabarti, R. Kumar, F. Radlinski and E. Upfal, Mortal multi-armed bandits, in: *Proc. 25th International Conference on Machine Learning (ICML)* (2008).
9. H. S. Chang, M. Fu, J. Hu and S. I. Marcus, An adaptive sampling algorithm for solving Markov decision processes, *Operations Research*, **53** (2005), 126-139.
10. H. S. Chang, M. C. Fu and S. I. Marcus, An asymptotically efficient algorithm for finite horizon stochastic dynamic programming problems, *IEEE Transactions on Automatic Control*, **52** (2007), 89-94.
11. V. Farias and R. Madan, The irrevocable multiarmed bandit problem, *Operations Research*, **59** (2011), 383-399.
12. Y. Freund and R. Schapire, Adaptive game playing using multiplicative weights, *Games and Economic Behavior*, **29** (1999), 79-103.

13. A. Gavirier and O. Cappe, The KL-UCB algorithm for bounded stochastic bandits and beyond, in: *Proc. Intl. Conf. on Learning Theory (COLT)* (2011).

14. Y. Seldin, C. Szepesvari, P. Auer and Y. Abbasi-Yadkori, Evaluation and analysis of the performance of the EXP3 algorithm in stochastic environments, *JMLR Workshop and Conference Proceedings*, **24** (2012), 103-116.

15. J. Vermorel and M. Mohri, Multi-armed bandit algorithms and empirical evaluation, in: *Proceedings of the 16th European Conference on Machine Learning (ECML)* (2005).