INDIAN INSTITUTE OF MANAGEMENT KOZHIKODE

Working Paper

IIMK/WPS/218/ITS/2017/02

January 2017

effSAMWMIX: An efficient Stochastic Multi-Armed Bandit Algorithm based on a Simulated Annealing with Multiplicative Weights

Boby Chaitanya Villari[1]
Mohammed Shahid Abdulla[2]

[1] Doctoral Student, IT & Systems Area, Indian Institute of Management Kozhikode, IIMK Campus P.O, Kerala – 673570, India, E-mail: Bobycv06fpm@iimk.ac.in

[2] Associate Professor, IT & Systems Area, Indian Institute of Management Kozhikode, IIMK Campus P.O, Kerala – 673570, India, E-mail: shahid@iimk.ac.in, Phone: +91 - 495 - 2809254

IIMK WORKING PAPER

effSAMWMIX : An efficient Stochastic Multi-Armed Bandit Algorithm based on a Simulated Annealing with Multiplicative Weights

Boby Chaitanya Villari
Doctoral Student of IT & Systems Area, IIM Kozhikode

Mohammed Shahid Abdulla
Associate Professor of IT & Systems Area, IIM Kozhikode

Abstract—**SAMWMIX, a Stochastic Multi-Armed Bandit(SMAB) which obtains a $O(log$ T) where T being the number of steps in the time horizon, is proposed in the literature . A blind-SAMWMIX which incorporates an input parameter ,which has better empirical performance but obtains a regret of the order $O(log^{1+2\alpha} T)$.Current work proposes an efficient version of SAMWMIX which not only obtains a regret of $O(log$ K) but also exults a better performance. A proof for the same is given in this work. The proposed effSAMWMIX algorithm is compared with KL-UCB and Thompson Sampling(TS) algorithms over rewards which follow distributions like Exponential, Poisson, Bernoulli, Triangular, Truncated Normal distribution and a synthetic distribution designed to stress test SMAB algorithms with closely spaced reward means. It is shown that effSAMWMIX performs better than both KL-UCB & TS in both regret performance and execution time.**

Keywords—stochastic multi-armed bandit;stochastic processes; reward distributions; optimization;

I. INTRODUCTION

Decision making under uncertainty is a challenge in a turbulent environment. Partial feedback from the environment leads to incomplete information giving rise to uncertainty. The agent has to learn sequentially (i.e. in many iterations) by relying only on reinforcements with partial feedbacks obtained due to the previous decision. Thus, the decision maker should *explore* the entire set of available decision choices in an attempt to improve the knowledge about the problem's current solution and *exploit* the currently available knowledge on the problem's solution choices to choose the best choice. Multi-Armed Bandits (MAB), a family of Machine Learning algorithms, are tailor-made to handle such explore-exploit problem situations[1]. The MAB problem is a sequential decision-making task where the decision maker (*agent*) decides to choose (*pull*), at each time step, an action (*arm*) from a pool of M actions - based on some informed choosing strategy (*policy*). With the aim of maximizing the average payoff from this exercise in the long-run, the *agent* examines these payoffs to continuously improve the *policy* and decide on the future selection of arms. Alternatively, the same can be seen as a regret minimization problem where the regret is the difference between *rewards* of an oracle policy that chooses the best arm in every time step and the rewards of the learned MAB policy. It is known that in T pulls, $O(log\ T)$ regret is the lowest possible regret an MAB algorithm can achieve and thus the desirable target.

The Upper Confidence Bound (UCB) class of MAB algorithms, both in the stochastic and contextual settings, is considered a benchmark. However, in both settings, UCB requires a 'pick the best arm' behavior, which is performed by explicitly picking the maximum among the N arms. The metrics associated with each arm will, however, change from pull to pull. This makes the per-step complexity of the UCB class of algorithms $O(T)$. The proposed SMAB algorithm – SAMWMIX - differs in avoiding maximization and instead picking a 'Soft-Maximum' via a Boltzmann Exploration structure. Since the 'Soft Maximum' is encoded inside a probability vector, the generation of an arm's index using this probability vector contributes to a $O(log\ T)$ complexity – which makes the algorithm very competitive if T happens to be large. This advantage is over-and-above the improved precision of SAMWMIX and its variants due to a theoretical result[2] that bounds the 'finite sample regret' (i.e. the probability of choosing the wrong arm in each pull).

In a previous work, the authors proposed Gamma Optimized SAMWMIX(GO-SAMWMIX) [3] with a simple heuristic improvement over the original .The bounds on this algorithm largely remain the same as that of original but GO-SAMWMIX is reported to have better effectiveness than its predecessor in terms of the number of times the best of the choices are chosen by the algorithm over a given time horizon t.This work proposes an efficient version of SAMWMIX which not only obtains a regret of $O(log$ T) but also has a better performance in terms of efficiency i.e. the number of times the best action is pulled and thus lowering the regret or maximizing the cumulative reward over the horizon t.This work compares the performance of effSAMWMIX vis-à-vis that of the state-of-the-art KL-UCB algorithm [4] and Thompson Sampling(TS) algorithm[5, 6] which is seen to gain prominence in this decade. The tests are done over rewards which follow distributions like Exponential, Poisson, Bernoulli, Triangular and Truncated

Normal distribution's and a synthetic distribution designed to stress test the algorithm with closely spaced reward means μ. (see section IV). effSAMWMIX has achieved a superior performance as detailed further in this paper.

II. MULTI-ARMED BANDITS- A REVIEW

A. The Multi-Armed Bandit(MAB) problem in brief

A MAB problem is a sequential decision-making problem with discrete time steps or horizon T.The agent chooses an action in every time step from a set of N possible actions. Then the environment will provide the reward pertaining to action i at time step $t < T$. Thus process continues in the following way.

1. $Agent\ chooses\ action\ I_t\ \&\ thus\ interacts\ with\ environment$
2. $Environment\ provides\ the\ rewards\ x_{i,t} \in [0,1] for\ t \in \{1,T\}\ and\ reward\ x_{i,t}$
3. $Observing\ this\ reward, x_{i,t}, the\ agent\ tries\ to\ maximize\ the\ cumulative\ reward\ i.e. \sum^{T} x_{I_t,t}\ \ by\ the\ time\ step\ T$

Fig.1. The MAB problem

For a general MAB, it is to be noted that the rewards could follow any distribution but are to be bounded by $[0,1]$.An agent is tested to achieve a highest cumulative reward or lowest cumulative regret to be graded better. Imposing additional assumption on the generalized MAB problem will lead to a variant, of our interest, called the Stochastic Multi-Armed Bandit (SMAB).

B. The Stochastic Multi-Armed Bandit (SMAB)

In SMAB settings it is assumed that the reward x_t follows a fixed distribution $v_i on\ [0,1]$ unknown to the agent. The reward of each action i is independent of the rewards it obtained from any other time horizon (or pull) and independent of rewards of other actions [7].This means that the rewards $\{X_t^i\}_{i \epsilon N}$ are assumed to be i.i.d from v_i and all the rewards of K choices(arms) are also independent of each other.

In addition to Stochastic settings, an adversarial setting[8] that exists for MABs which is not the setting for the algorithms studied in this work.

C. Performance Measures for an SMAB

In an SMAB (also for any MAB) setting, an algorithm is evaluated based on the cumulative reward or the cumulative regret obtained due to the agent's decision-making processes over the time horizon. , regret $\mathcal{R}$ is a measure of how far the performance of contended algorithm is compared to an oracle policy that knows how to pick the best possible action in every each time step i in horizon $\mathcal{H}$.For all CMABs, regret $\mathcal{R}$ is a central conceptual variable to be considered and concentrated towards obtaining a logarithmically changing regret over time horizon. During experimentation the regret $\bar{\mathcal{R}}$ is plotted against horizon $\mathcal{H}$ so as to observe and compare with algorithms reported or contented against.

An expected payoff regret or simply expected regret $\bar{\mathcal{R}}$ is defined as follows

$$\bar{\mathcal{R}} = \sum_{t=1}^{T} \left(\max_{i=1,2,\dots N} E[X_t^i] \right) - \sum_{t=1}^{H} X_t^{I_t}$$

Thus $\bar{\mathcal{R}}$ is the difference between the payoff of an optimal arm as selected by an oracle policy (in expectation over the arm's distribution) and the actual payoff obtained by the MAB policy in contention. Since in real time environment, the oracle policy might not be known in advance, $\bar{\mathcal{R}}$ happens to have significance only in simulated environments where the testing is done to evaluate the algorithm's performance.

III. THE $effSAMWMIX$ ALGORITHM

The effSAMWMIX algorithm is based on SAMWMIX[2] which indeed is based on a variant of simulated annealing called Simulated Annealing with Multiplicative Weights(SAMW).The regret is calculated as given below. As mentioned earlier, the SAMWMIX and its variants use a 'Soft-Maximum' via a Boltzmann Exploration structure. Since the 'Soft Maximum' is encoded inside a probability vector, the generation of an arm's index using this probability vector contributes to a $O(log\ K)$ complexity – which makes the algorithm very competitive if K, the number of available arms, happens to be large. This binds the finite sample regret which is the probability of choosing a wrong action in each pull.

The effSAMWMIX algorithm chooses the arms by calculating and maintaining a vector ϕ_t^j where $j \epsilon [1,K]$ are the number of arms available and $t \epsilon [1,T]$ is the time horizon. The ϕ_t^j is given below.

$$\phi_{k+1}^j = (1-\gamma_k)\ \frac{\phi_k^j\ e^{\Sigma \eta_t \widehat{X_k^j}}}{\sum_{j=1}^{N} \phi_k^j\ e^{\Sigma \eta_t \widehat{X_k^j}}} + \frac{\gamma_k}{N} \qquad (1)$$

(1) is similar to (10) of [2] since it represents a Boltzmann exploration schema. The learning component which is the step-size γ_t and the inverse temperature parameter is given below.

$$\gamma_k = \frac{N\left(4 + (d + d_k)\right)}{k(d+d_k)^2 - (d + d_k - 2d^2)} \qquad (2)$$

$$\eta_k = \frac{1}{\frac{N}{\gamma_k} + 1} log \left(\frac{1 + d\left(\frac{n}{\gamma_k} + 1\right)}{\frac{2N}{\gamma_k} - d^2} \right) \qquad (3)$$

The construction of proof is based on the definition of an expected regret which is defined as the expected cumulative loss incurred due to not playing the best possible arm a_t^* during the iteration t where$t \epsilon [1,T]$.The calculation of the maximum expected regret is put below.

$$\begin{aligned} \text{Maximum Regret of MAB} &= \underset{1 \le j \le N}{Max}\ E[\textstyle\sum_{t=1}^{T}(X_t^i - X_t^{I_t})] \\ &= E[\sum_{t=1}^{T}(X_t^1 - X_t^{I_t})] \\ &= \sum_{t=1}^{T}\left[E\left(X_t^1 - X_t^{I_t}\right)\right] \\ &= \sum_{t=1}^{T} E\left[E\left(X_t^1 - X_t^{I_t}\right) / \mathcal{F}_{t-1}\right] \end{aligned}$$

$$= \sum_{t=1}^{T} E\left[\sum_{j=2}^{N} \Delta_j \phi_t^j\right]$$
$$= \sum_{t=1}^{T}\left[\sum_{j=2}^{N} \Delta_j E[\phi_t^j]\right]$$

Figure 2. Calculation of Maximum Expected Regret

The proof for logarithmic regret is similar to that of SAMWMIX and thus can be documented similarly. Since $\sum_{p=1}^{T}(\Delta_i/p) = \log(T)$, obtaining an upper bound of $O\left(\frac{1}{T}\right)$ on expectation $E\{\phi_t^i\}$ for any suboptimal arm implies a logarithmic regret in effSAMWMIX. From the following set of equations, we derive the value of γ_k which essentially differentiates the effSAMWMIX from SAMWMIX.

By altering the Boltzmann exploration schema of SAMWMIX from

$$\phi_{t+1}^j = (1-\gamma_t)\ \frac{\phi_t^j\, e^{\Sigma\eta_t \widehat{x_t^j}}}{\sum_{j=1}^{N} \phi_t^j\, e^{\Sigma\eta_t \widehat{x_t^j}}} + \frac{\gamma_t}{N}$$

which is (10) of [2] to that of effSAMWMIX which is

$$\phi_{t+1}^j = (1-\gamma_t)\ \frac{e^{\Sigma\eta_t \widehat{x_t^j}}}{\sum_{j=1}^{N} e^{\Sigma\eta_t \widehat{x_t^j}}} + \frac{\gamma_t}{N}$$

where $\eta_k = \frac{1}{\frac{N}{\gamma_k}+1} log\left(\frac{1+d\left(\frac{n}{\gamma_k}+1\right)}{\frac{2N}{\gamma_k}-d^2}\right)$

if $K_p = d, C_p = \frac{N}{\gamma_p} + 1\,, \sigma_p^2 = 2 * \frac{N}{\gamma_p} - d^2, then$

$$\eta_p = \frac{1}{C_p}\log\left(\frac{1+C_pK_p}{\sigma_p^2}\right)$$

The regret is defined as $\bar{\mathcal{R}} = \sum_{t=1}^{T}\left[\sum_{j=2}^{N} \Delta_j E[\phi_t^j]\right]$

But $\phi_{t+1}^j = (1-\gamma_t)\ \frac{e^{\Sigma\eta_t \widehat{x_t^j}}}{\sum_{j=1}^{N} e^{\Sigma\eta_t \widehat{x_t^j}}} + \frac{\gamma_t}{N}$

Thus $\bar{\mathcal{R}} = \sum_{t=1}^{T}\left[\sum_{j=2}^{N} \Delta_j E\left[(1-\gamma_t)\ \frac{e^{\Sigma\eta_t \widehat{x_t^j}}}{\sum_{j=1}^{N} e^{\Sigma\eta_t \widehat{x_t^j}}} + \frac{\gamma_t}{N}\right]\right]$

$$= \sum_{t=1}^{T}\left[\sum_{j=2}^{N} \Delta_j\, E\left[(1-\gamma_t)\ \frac{e^{\Sigma\eta_t \widehat{x_t^j}}}{\sum_{j=1}^{N} e^{\Sigma\eta_t \widehat{x_t^j}}}\ \right]\right] + \frac{\gamma_t}{N}$$

Since $e^{\Sigma\eta_1 \widehat{x_t^1}} \leq \sum_{j=1}^{N} e^{\Sigma\eta_t \widehat{x_t^j}}$,the following equation (4) holds true.

$$(1-\gamma_t)\ \frac{e^{\Sigma\eta_t \widehat{x_t^j}}}{\sum_{j=1}^{N} e^{\Sigma\eta_t \widehat{x_t^j}}} + \frac{\gamma_t}{N} \leq (1-\gamma_t)\ \frac{e^{\Sigma\eta_t \widehat{x_t^j}}}{e^{\Sigma\eta_1 \widehat{x_t^1}}} + \frac{\gamma_t}{N} \quad (4)$$

Considering the following inequality

$$e^{\Sigma_{k=1}^{K-1}\eta_k \widehat{x_k^j}} * e^{\eta_k(d_k)*\left(\widehat{x_t^j}+d_k\right)} \leq \sum_{j=1}^{N} e^{\Sigma_{k=1}^{K}\eta_k(d_k)\widehat{x_k^j}} \quad (5)$$

be true for every i including $i = 1\ to\ N$ and if a_k is the winner arm in k_{th}iteration, (5) can be written as follows

$$e^{\Sigma_{k=1}^{K-1}\eta_k \widehat{x_k^{a_k}}} * e^{\eta_k(d_k)*\left(\widehat{x_t^{a_k}}+d_k\right)} \leq \sum_{j=1}^{N} e^{\Sigma_{k=1}^{K}\eta_k(d_k)\widehat{x_k^j}} \quad (6)$$

Fetching η_p from SAMWMIX [2]

$$\eta_p^{samwmix} = \frac{1}{C_p}\log\left(\frac{1+C_pK_p}{\sigma_p^2}\right) \text{ where } k_p = d$$

$$\eta_p^{samwmix} = \frac{1}{C_p}\log\left(\frac{1+C_pd}{\sigma_p^2}\right)$$

Analogous to SAMWMIX we define $\eta_p^{effsamwmix}$ as

$$\eta_p^{effsamwmix} = \frac{1}{C_p}\log\left(\frac{1+C_p(d+d_p)}{\sigma_p^2}\right) \quad (7)$$

Getting the notations back incorporating $d_k, i.e.$

$$\eta_p^{effsamwmix} = \eta_p(d_p)\,, parameterized\ by\ d_k$$

$$\eta_p(d_p) = \frac{1}{C_p}\log\left(\frac{1+C_p(d+d_p)}{\sigma_p^2}\right)$$

$$\phi_{p+1}^j = (1-\gamma_p)\, e^{\sum_{p=1}^{K}\frac{-K_p^2}{2\sigma^2+K_pC_p}} + \frac{\gamma_t}{N}$$

To obtain logarithmic regret, the summation term in exponent should be a logarithmically increasing entity. Hence we equate it to $\log K$

$$\sum_{p=1}^{K} -\frac{K_p^2}{2\sigma^2 + K_pC_p} = -\log K$$

which implies that $\frac{K_p^2}{2\sigma^2+K_pC_p} = \frac{1}{k}$ (8)

Since $K_p = (d + d_k)$,equation (8) is rewritten as

$$\frac{(d+d_k)^2}{2(\frac{2N}{\gamma_k} - d^2) + (d+d_k)(\frac{N}{\gamma_k}+1)} = \frac{1}{k}$$

Thus $\gamma_k = \frac{N(4+(d+d_k))}{k(d+d_k)^2-(d+d_k-2d^2)}$

Thus an exploration parameter $\gamma_k = \frac{N(4+(d+d_k))}{k(d+d_k)^2-(d+d_k-2d^2)}$ will aid effSAMWMIX in achieving a logarithmic regret $\log T$ over the horizon T. The pseudocode for effSAMWMIX algorithm is put below

Algorithm 1 ***effSAMWMIX SMAB algorithm***

Input : Rewards Vector G_t,set of Arms N,number of rounds T

1. Using G_t Calculate d=min $\Delta(\mu_1, \mu_2 \ldots . \mu_N)$ where μ_i is the reward mean of Arm i.
2. Calculate
   a. $C_0 = N + 1;\ \sigma^2 = 2 * N;$
   b. $\eta_0 = \frac{1}{C_0}\log\left(\frac{1+C_p*d}{\sigma^2}\right)$
   c. $startIter = ((4 + d) * N + d)/d^2$
3. $for\ i\ =\ 1,\ldots,N\ do$
   a. Obtain reward $X^i_{t=i}$
   b. Initialize $\phi^i_t\ = \eta_0 * \left(\frac{startIter}{N}\right) * (\frac{X^i_{t=i}}{\frac{1}{N}})$
   c. Initialize pull count for $arm\ a^i\ as\ p_i$=1
4. $for\ t\ = (startIter + 1 + N), \ldots, (startIter + T) do$
   a. Obtain random probability r
   b. Choose an arm i as winner if $\sum \phi^i_t\ > r$ and store reward $G^i_{t-startIter} = a^*_t$ and normalize the reward using its probability $\hat{X}$=a^*_t/ϕ^i_t
   c. Update $p_i = p_i + 1$
   d. $for\ d_t\ =\ 1, \ldots, \frac{t-startIter}{T-N} in\ steps\ of\ d_{tstep}\ do$
      i. Calculate $k_t = d + d_t;$
      ii. $\gamma_{ttemp} = ((4 + k_t) * N + k_t)/(t * k_t^2)$
      iii. $C_t = \left(\frac{N}{\gamma_{ttemp}}\right) + 1$ and $\sigma^2_{temp} = 2 * N/\gamma_{ttemp}$
      iv. $\eta_t = \frac{1}{C_t}\log\left(\frac{1+C_t*k_t}{\sigma^2_{temp}}\right)$
      v. $\sum \phi^i_{ttemp} = \sum \phi^i_t + e^{\sum \eta_t \widehat{X^i_t}}$
      vi. If $e^{\sum \eta_t*(d_t* \widehat{X^i_t}) + \eta_t*(d_t* \widehat{X^{a^i}_t})} > \sum \phi^i_{ttemp}$ then assign $d_{ttrue} = d_t - d_{tstep}$
   e. Assign
      i. $K_t = d_t + d_{ttrue}$
      ii. $\gamma_t = ((4 + k_t) * N + k_t)/(t * k_t^2)$
      iii. $C_t = \left(\frac{N}{\gamma_{ttemp}}\right) + 1$ and $\sigma^2_t = 2 * N/\gamma_{ttemp}$
      iv. $\eta_t = \frac{1}{C_t}\log\left(\frac{1+C_t K_t}{\sigma^2_t}\right)$
   f. Now update ϕ^j_{t+1} using (1) which is

$$\phi^j_{t+1} = \ (1 - \gamma_t)\ \frac{e^{\sum \eta_t \widehat{X^j_t}}}{\sum_{j=1}^{N} e^{\sum \eta_t \widehat{X^j_t}}}\ + \frac{\gamma_t}{N}$$

Output : ϕ vector

IV. EMPIRICAL EVALUATION OF $effSAMWMIX$

effSAMWMIX is currently being compared to the performance of KL-UCB and TS. We chose KL-UCB for that it is prominently cited and compared to in MAB literature. Similarly, TS is gaining prominence[9] in the scientific fraternity in this decade. Also, effSAMWMIX has performed better than its predecessors GO-SAMWMIXM and SAMWMIX which in turn has as superior performance over UCB1 algorithm. This work reports the performance of effSAMWMIX vis-à-vis to that of KL-UCB and TS over the following reward distributions. The main purpose of these numerical experiments using various reward distributions is to compare the performance in terms of cumulated regret and the number of times the arm with the best mean reward (best arm henceforth) is pulled.

A. Customized Synthetic Distribution (CuSyn)

It is known that the mean reward of any arm influences the decision making of an SMAB. The closer the mean rewards of the available arms, the difficult it should be for an SMAB algorithm to latch on to the best arm for that iteration. A Customized Synthetic Distribution (CuSyn) is designed with an input parameter that allows distributing the rewards of the arms so that the minimum reward mean's difference between any two arms is controlled. For example, using 0.1 as a parameter, the arms' reward distributions will be placed so that the closest of the arms will have their reward means differing only by 10%. This ensures control over the closeness of the arms and allows for stress testing the algorithm.

B. A few other Reward distributions

Taking clues from[10], reward distributions of the arms are set so that their means are in an exponential distribution or a Bernoulli distribution. The algorithms are tested with arms that follow a Triangular Distribution [11] and a Normal distribution truncated to be bounded between [0,1] (Truncated Normal Distribution henceforth). The results of the numerical experiments are put below.

C. Design of the Experiment

Numerical experiments are performed on effSAMWMIX, KL-UCB and Thompson Sampling(TS) algorithms using the computational software package MATLAB. All the distributions are to have rewards $\hat{x}\epsilon(0,1)$ and the rewards are i.i.d for all the five arms ($K = 5$) in consideration. The time horizon is set to be 2000 pulls ($T = 2000$) and each run of the code is called an experiment. 100 such experiments are conducted and the results are averaged to remove any randomness in the results. The results are put below when the rewards followed each of the distributions named below.

TABLE I. MEASURED CPU TIME FOR ALGORITHMS.

CPU Time in milliseconds for each rewards Distribution	**Algorithm (time in milliseconds)**		
	effSAMWMIX	***KL-UCB***	***Thompson Sampling***
CuSyn	66.6875	199.6875	363.4531
Triangular	57.9844	195.9063	343.4531
Truncated Normal	65.2031	201.8594	357.6094
Bernoulli	66.5469	196.2344	353.9844
Poisson	65.2656	201.1719	360.0156
Exponential	67.5313	200.5156	360.5313

1) Customized Synthetic(CuSyn) Distribution

The cumulative regret accumulated by effSAMWMIX is the lowest as shown in Fig.1. effSAMWMIX has chosen the best

possible arm the most number of times (see Fig.2.) and thus aggregated to a lower regret compared to KL-UCB and TS.

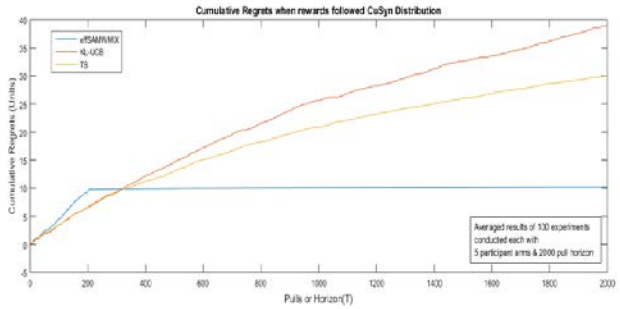

Fig. 1. Comparison of cumulative regrets when the rewards followed a Customized Synthetic Distribution.

Also, the average time taken, to run an experiment with effSAMWMIX is about 67millseconds which is the lowest (See Table.1). Thus effSAMWMIX is both effective (lower regret) and efficient (lower computation time) of the three algorithms.

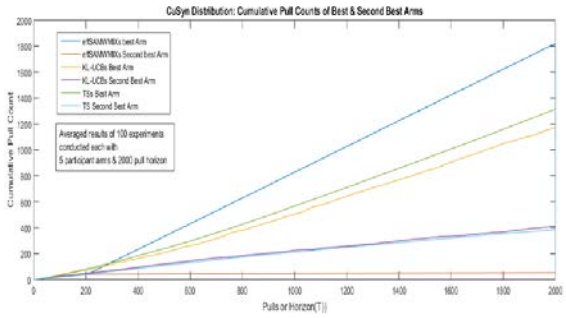

Fig. 2. Comparison of best arm's Pull Count when the rewards followed a Customized Synthetic Distribution.

2) Triangular Distribution

effSAMWMIX performed better than the other two algorithms both in terms of efficiency and effectiveness. The cumulative regret accumulated by effSAMWMIX is the lowest as shown in Fig.3.

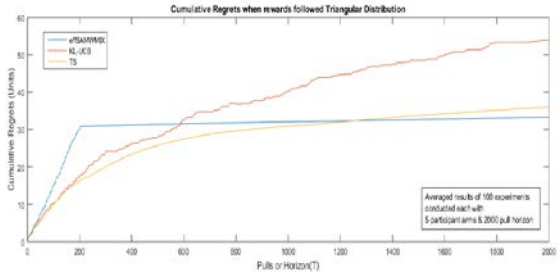

Fig. 3. Comparison of cumulative regrets when the rewards followed a Triangular Distribution

effSAMWMIX has chosen the best possible arm the most number of times (see Fig.4.) and thus aggregated to a lower regret compared to KL-UCB and TS. Here also the average time taken to run an experiment with effSAMWMIX is about 58 milliseconds which is the lowest (See Table.1) of that of the three algorithms.

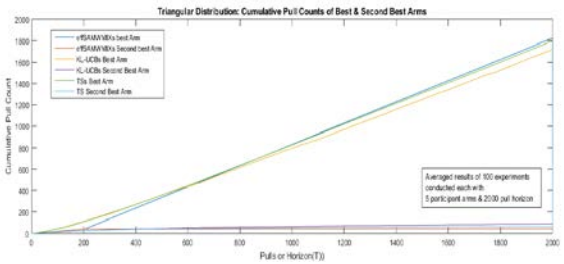

Fig. 4. Comparison of best arm's Pull Count when the rewards followed a Triangular Distribution.

3) Truncated Normal Distribution

effSAMWMIX's performance is superior to other two algorithms when the mean of arms' rewards followed a Truncated Normal Distribution.

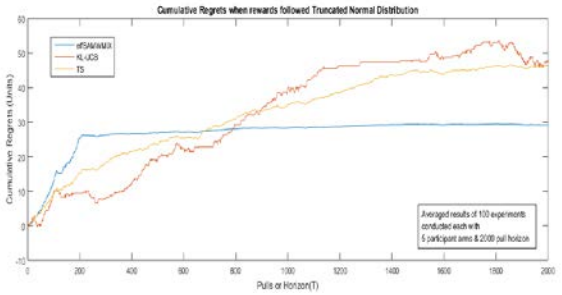

Fig. 5. Comparison of cumulative regrets when the rewards followed a Truncated Normal Distribution

effSAMWMIX is superior both in effectiveness and efficiency (see Fig.5 & Fig 6)

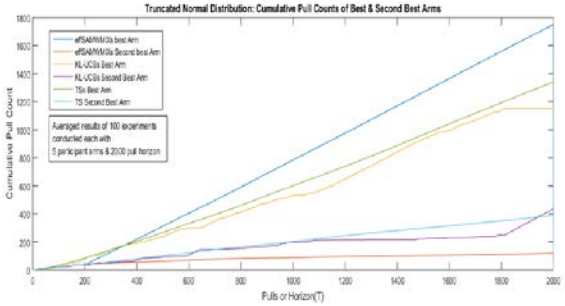

Fig. 6. Comparison of best arm's Pull Count when the rewards followed a Truncated Normal Distribution.

4) Bernoulli Distribution

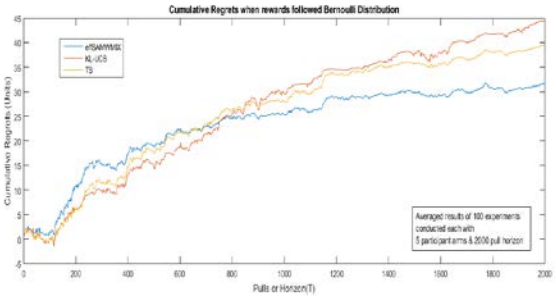

Fig. 7. Comparison of cumulative regrets when the rewards followed a Bernoulli Distribution

When the mean of arms' rewards followed a Bernoulli distribution the performance results are favorable to effSAMWMIX both of effectiveness and efficiency (see Fig.7 & Fig 8)

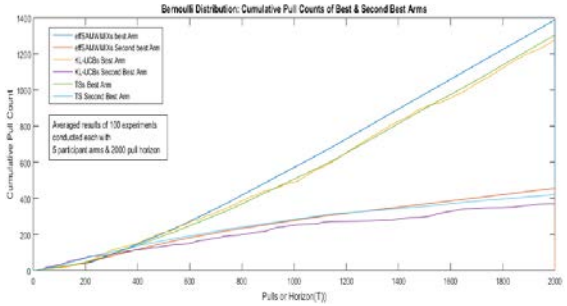

Fig. 8. Comparison of best arm's Pull Count when the rewards followed a Bernoulli Distribution.

5) Poisson Distribution

effSAMWMIX performed better than the other two algorithms in terms of time efficiency.

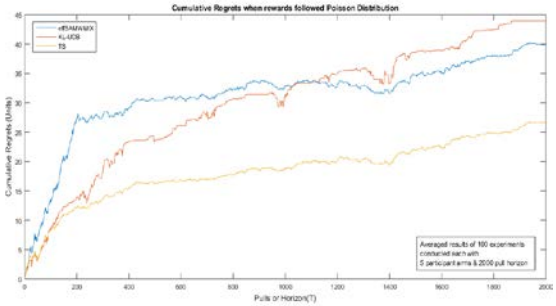

Fig. 9. Comparison of cumulative regrets when the rewards followed a Poisson Distribution

TS has performed better than both KL-UCB and effSAMWMIX (see Fig.1 & Fig.2.) even though TS's time efficiency is the lowest of the three (see Table .1.).

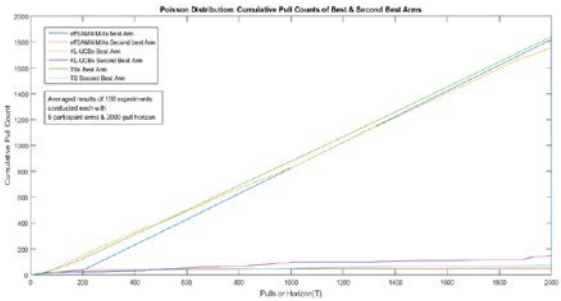

Fig. 10. Comparison of best arm's Pull Count when the rewards followed a P Distribution.

6) Exponential Distribution

The time efficiency of effSAMWMIX is still the best among the algorithms in comparison but the regret of KL-UCB is the lowest.

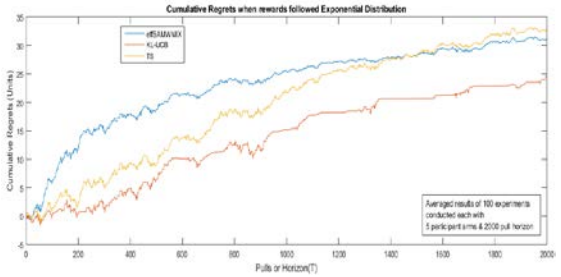

Fig. 11. Comparison of cumulative regrets when the rewards followed an Exponential Distribution

While effSAMWMIX performed with a regret lower than that of TS, KL-UCB has the best regret when the rewards followed an Exponential distribution.

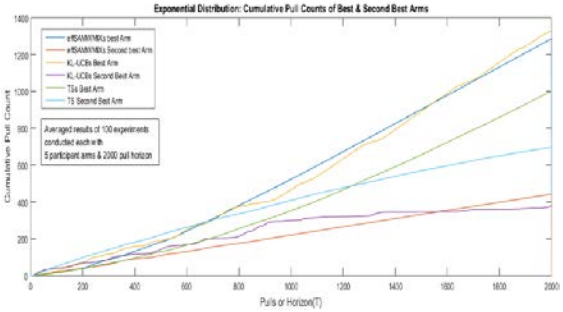

Fig. 12. Comparison of best arm's Pull Count when the rewards followed an Exponential Distribution.

V. Conclusions and Future Directions

The proposed SMAB algorithm effSAMWMIX is compared with the KL-UCB and Thompson Sampling MAB algorithms. The tests are run when the available arms follow the various distributions mentioned in the preceding sections. In all the cases, effSAMWMIX achieved a better time efficiency than those in comparison. Except for when the arms' rewards followed either an Exponential distribution or a Poisson distribution, effSAMWMIX outperformed KL-UCB and TS in terms of effectiveness i.e. with a minimum of the cumulative regret achieved over the horizon. In the case when the arms'

rewards followed an exponential distribution, KL-UCB achieved the lowest regret while effSAMWMIX ranked second. And in the Poisson distribution case, effSAMWMIX performed next to Thompson Sampling SMAB.

The effSAMWMIX obtains a log regret which is a desirable property for an SMAB algorithm. Also effSAMWMIX has a fast execution time and has the best efficiency among the three algorithms over all the cases in discussion. The authors intend to check the applicability of this algorithm by extending it to incorporate contextual information so as to address the explore-exploit problems that are common in any real-world business environment like that of a News article recommendation or cold-start problems in ecommerce domain.

VI. References

1 Vargas, A.M.: 'Linear Bayes policy for learning in contextual-bandits', Expert Systems with Applications, 2013, 40, (18), pp. 7400-7406

2 Abdulla, M.S., and Bhatnagar, S.: 'Multi-armed bandits based on a variant of Simulated Annealing', Indian Journal of Pure and Applied Mathematics, 2016, 47, (2), pp. 195-212

3 Boby Chaitanya, V., & Mohammed Shahid A: 'Bandit Algorithms for Contextual Advertising: An Evaluation of SOFTMIX algorithm over the benchmark Yahoo! FrontPage Today Dataset.'. Proc. 2nd Pan I.I.M World Management Conference, Kozhikode, India2014, November pp. Pages

4 Garivier, A., and Cappé, O.: 'The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond', in Editor (Ed.)^(Eds.): 'Book The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond' (2011, edn.), pp. 359-376

5 Kaufmann, E., Korda, N., and Munos, R.: 'Thompson sampling: An asymptotically optimal finite-time analysis', in Editor (Ed.)^(Eds.): 'Book Thompson sampling: An asymptotically optimal finite-time analysis' (Springer, 2012, edn.), pp. 199-213

6 Thompson, W.R.: 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples', Biometrika, 1933, 25, (3/4), pp. 285-294

7 Robbins, H.: 'Some aspects of the sequential design of experiments': 'Herbert Robbins Selected Papers' (Springer, 1985), pp. 169-177

8 Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R.E.: 'Gambling in a rigged casino: The adversarial multi-armed bandit problem', in Editor (Ed.)^(Eds.): 'Book Gambling in a rigged casino: The adversarial multi-armed bandit problem' (IEEE, 1995, edn.), pp. 322-331

9 Agrawal, S., and Goyal, N.: 'Analysis of Thompson Sampling for the Multi-armed Bandit Problem', in Editor (Ed.)^(Eds.): 'Book Analysis of Thompson Sampling for the Multi-armed Bandit Problem' (2012, edn.), pp. 39.31-39.26

10 Kaufmann, E., Cappé, O., and Garivier, A.: 'On Bayesian Upper Confidence Bounds for Bandit Problems', in Editor (Ed.)^(Eds.): 'Book On Bayesian Upper Confidence Bounds for Bandit Problems' (2012, edn.), pp. 592-600

11 Kotz, S., and Van Dorp, J.R.: 'Other Continuous Families of Distributions with Bounded Support and Applications' (World Scientific, 2004. 2004)

Research Office
Indian Institute of Management Kozhikode
IIMK Campus P. O.,
Kozhikode, Kerala, India,
PIN - 673 570
Phone: +91-495-2809238
Email: research@iimk.ac.in
Web: https://iimk.ac.in/faculty/publicationmenu.php