ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Thompson Sampling for Multi-Objective Multi-Armed Bandits Problem

Saba Yahyaa and Bernard Manderick

Computational Modeling group, Artificial Intelligence Lab, Computer Science Department
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
syahyaa, bmanderi@vub.ac.be

Abstract. The multi-objective multi-armed bandit ($MOMAB$) problem is a sequential decision process with stochastic rewards. Each arm generates a vector of rewards instead of a single scalar reward. Moreover, these multiple rewards might be conflicting. The $MOMAB$-problem has a set of Pareto optimal arms and an agent's goal is not only to find that set but also to play evenly or fairly the arms in that set. To find the Pareto optimal arms, linear scalarized function or Pareto dominance relations can be used. The linear scalarized function converts the multi-objective optimization problem into a single objective one and is a very popular approach because of its simplicity. The Pareto dominance relations optimizes directly the multi-objective problem. In this paper, we extend the Thompson Sampling policy to be used in the $MOMAB$ problem. We propose Pareto Thompson Sampling and linear scalarized Thompson Sampling approaches. We compare empirically between Pareto Thompson Sampling and linear scalarized Thompson Sampling on a test suite of $MOMAB$ problems with Bernoulli distributions. Pareto Thompson Sampling is the approach with the best empirical performance.

1 Introduction

The $MOMAB$ problem is a sequential stochastic learning problem [1, 2]. At each time step t, an agent pulls one arm i from an available arm set A and receives a reward vector $\boldsymbol{r}_i$ of the arm i with D dimensions (or objectives) as feedback signal. The reward vector is drawn from a corresponding stationary probability distribution vector, e.g. Bernoulli distribution $\mathrm{B}(\boldsymbol{p}_i)$, where $\boldsymbol{p}_i$ is the *success rate* vector of the arm i. The reward vector that the agent receives from the arm i is independent from all other arms and from the past reward vectors of the pulled arm i. Moreover, the success rate vector $\boldsymbol{p}_i$ of the arm i has *independent* D distributions. We assume that the success rate vector of each arm i is unknown parameter to the agent. Thus, by drawing each arm i, the agent maintains estimation of the true success rate vector which is known as $\hat{\boldsymbol{p}}_i$.

The $MOMAB$ problem has a set of optimal arms (Pareto front) A^*, that are incomparable, cannot be classified using a partial order relation [3]. The agent has to discover the optimal arms (exploring), to reduce the total Pareto loss of not pulling the optimal arms, and has to play them fairly (exploiting), to reduce the total unfairness loss. [4]. At each time step t, the Pareto loss (Pareto regret) is the distance between the success rate set of the Pareto front and the success rate vector of the selected arm [1]. The unfairness loss (unfairness regret) is the Shannon entropy which is the measure of disarray on the frequency of selecting the optimal arms in the Pareto front A^*. The higher entropy is the higher disorder [5]. Thus, the total Pareto and unfairness regrets are the cumulative summation of the Pareto and unfairness regrets over t time steps, respectively.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

The Pareto front A^* can be found either by scalarized function, e.g. Linear Scalarized Function (LSF) [6], or Pareto Dominance Relation (PDR) [3]. The LSF converts the Multi-Objective (MO) space into a single one. The LSF is simple and intuitive, however, can not find all the optimal arms in a non-convex success rate set. While, PDR finds the Pareto front by optimizing directly the MO space, however, for large number of optimal arms and objectives it can not find out all the Pareto optimal arm set.

In this paper, we extend the Thompson Sampling (TS) [7] to be used in the MOO in order to improve the performance of the scalarized function and Pareto dominance relation. Linear scalarized Thompson Sampling function (LSF-TS) and Pareto Thompson Sampling (PTS) trade off between exploration and exploitation by assigning to each arm i in each objective d a random probability of selection P_i^d that is generated from Beta distribution. The LSF-TS transforms the multi-objective problem into a single one using linear scalarized function on the random probability of selection vectors $\boldsymbol{P}_i = [P_i^1, \cdots, P_i^D]^T$ of arms i and selects the arm that has the maximum scalarized function, where T is the transpose. The PTS uses Pareto dominance relation on the random probability of selection vectors $\boldsymbol{P}_i$ of arms i to find the Pareto front A^*.

The rest of the paper is organized as follows: In Section 2 we introduce the $MOMAB$ problem, PDR, LSF, and the regret measures in the $MOMAB$. In Section 3 we introduce the Pareto and linear scalarized Thompson sampling. In Section 4, we describe the experiments set up followed by experimental results. Finally, we conclude the paper.

2 Multi Objective Multi Armed Bandit Framework

Let us consider the $MOMAB$ problems with $|A| \geq 2$ arms and with *independent* D objectives per arm. At each time step t, the agent pulls one arm i and receives a reward vector $\boldsymbol{r}_i$. The reward $r_i^d \in \{0, 1\}$ in each objective $d \in D$ is drawn from a corresponding Bernoulli distribution with unknown success rate p_i^d, the probability of getting reward equals 1. Thus, by drawing each arm i, the agent estimates the success rate $\hat{p}_i^d(t)$ of the arm i in the objective d. Using Bayesian view, the success rate $\hat{p}_i^d$ can be estimated by using Beta distribution [5] after receiving the reward r_i^d as follows:

$$\hat{p}_i^d(t+1) \leftarrow \frac{\alpha_i^d(t+1)}{\alpha_i^d(t+1) + \beta_i^d(t+1)}, \quad \text{where} \tag{1}$$

$$\alpha_i^d(t+1) \leftarrow \alpha_i^d(t) + 1, \text{ if } r_i^d = 1, \;\; \beta_i^d(t+1) \leftarrow \beta_i^d(t) + 1, \text{ if } r_i^d = 0 \tag{2}$$

where $\alpha_i^d(t)$, and $\beta_i^d(t)$ are the number of successes and failures, respectively at time step t and $\alpha_i^d(t+1)$, and $\beta_i^d(t+1)$ are the updated number of successes and failures, respectively at time step $t+1$ of the arm i in the objective d.

The success rate vector of arm $i \in A$ is represented as $\boldsymbol{p}_i = [p_i^1, \cdots, p_i^D]^T$. The agent has a set of optimal arms (Pareto front) A^* which can be found by the *Pareto dominance relation* or *linear scalarized function*.

The Pareto dominance relation (PDR) finds the Pareto front A^* directly in the MO space [3]. It uses the following relations between the success rate vectors of two arms. 1) Arm i dominates j, $i \succ j$, if there exists at least one objective d for which $p_i^d \succ p_j^d$ and for all other objectives d' we have $p_i^{d'} \succeq p_j^{d'}$. Arm i is incomparable with j, $i \parallel j$,

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

if and only if there exists at least one objective d for which $p_i^d \succ p_j^d$ and there exists another objective d' for which $p_i^{d'} \prec p_j^{d'}$. 2) Arm i is not dominated by j, $j \not\succ i$, means that either $i \succ j$ or $i \parallel j$. Using these relations, Pareto front $A^* \subset A$ be the arm set that are not dominated by all other arms.

Linear scalarization function (LSF) converts the MOO problem into a single one [6]. However, solving a MOO problem means finding the Pareto front A^*. Thus, we need a set of scalarized functions $\boldsymbol{F} = \{f^1, \cdots, f^S\}$ to generate a variety of elements belonging to the Pareto front A^*. Each scalarized function $f^s \in \boldsymbol{F}$ has a corresponding predefined weight set $\boldsymbol{w}^s \in \boldsymbol{W}$, where $\boldsymbol{W} = (\boldsymbol{w}^1, \cdots, \boldsymbol{w}^S)$. The predefined total weight set $\boldsymbol{W}$ is uniformly random spread sampling in the weighted space [8].

The LSF assigns to each value of the success rate vector $\boldsymbol{p}_i$ of an arm i a weight w^d and the result is the sum of these weighted mean values. Given a predefined set of weights $\boldsymbol{w}^s = (w^1, \cdots, w^D)$ such that $\sum_{d=1}^{D} w^d = 1$, the LSF across $\boldsymbol{p}_i$ is:

$$f^s(\boldsymbol{p}_i) = w^1 p_i^1 + \cdots + w^D p_i^D \tag{3}$$

where $f^s(\boldsymbol{p}_i)$ is a linear scalarized function $s \in S$ on the success rate vector $\boldsymbol{p}_i$ of the arm i. After transforming the MO problem to a single one, the LSF f^s selects the arm $i^*_{f^s} = \mathrm{argmax}_{1 \le i \le A} f^s(\boldsymbol{p}_i)$ that has the maximum LSF value. The LSF is very popular because of its simplicity. However, it can not find all the optimal arms in the Pareto front A^*, if the A^* has a non-convex success rate vectors [8].

In the $MOMAB$, the agent has to find both the Pareto front A^* (or exploring the optimal arms) and play the optimal arms fairly (or exploiting the optimal arms). As a result, there are two regret measures. *The Pareto regret measure* (R_P) [1] measures the distance between a success rate vector of an arm i that is pulled at time step t and the Pareto front A^*. Pareto regret R_P is calculated by finding firstly the virtual distance dis^*. The virtual distance dis^* is defined as the minimum distance that is added to the success rate vector of the pulled arm $\boldsymbol{p}_t$ at time step t in each objective to create a virtual success rate vector $\boldsymbol{p}_t^* = \boldsymbol{p}_t + \boldsymbol{\varepsilon}^*$ that is incomparable with all the arms in Pareto set A^*, i.e. $\boldsymbol{p}_t^* || \boldsymbol{p}_i \ \forall_{i \in A^*}$. Where $\boldsymbol{\varepsilon}^*$ is a vector, $\boldsymbol{\varepsilon}^* = [dis^{*,1}, \cdots, dis^{*,D}]^T$. Then, the Pareto regret $R_P = dis(\boldsymbol{p}_t, \boldsymbol{p}_t^*) = dis(\boldsymbol{\varepsilon}^*, \boldsymbol{0})$ is the Euclidean distance between the success rate vectors of the virtual arm $\boldsymbol{p}_t^*$ and the pulled arm $\boldsymbol{p}_t$ at time step t, Thus, the regret of the Pareto front is 0 for optimal arms. *The unfairness regret measure* [5] is the Shannon entropy R_{SE} which is the measure of disorder on the frequency of selecting the optimal arms in the Pareto front A^*. The higher the entropy, the higher the disorder. The Shannon entropy at time step t, $R_{SE}(t) = -({}^1/_{N_{|A^*|}(t)}) \sum_{i^* \in A^*} p_{i^*}(t) \ln(p_{i^*}(t))$, where $p_{i^*}(t) = {}^{N_{i^*}(t)}/_{N(t)}$ is the frequency of selecting an optimal arm i^* at time step t, where $N_{i^*}(t)$ is the number of times the optimal arm i^* has been selected, $N(t)$ is the number of times all arms $i = 1, \cdots, A$ have been selected, and $N_{|A^*|}(t)$ is the number of times the optimal arms, $i^* = 1, \cdots, |A^*|$ have been selected at time step t.

3 Multi-Objective Thompson sampling

In the Bernoulli one-objective multi-armed bandit, the reward is a stochastic scalar value, there is only one optimal arm. The reward r_i of an arm i is either 0, or 1 with unknown success rate p_i. Thompson sampling (TS) [7] tradesoff between exploration

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

1. **Input:** number of objectives $|D|$; number of arms $|A|$; reward $r \sim \mathrm{B}(\boldsymbol{p})$
2. **Initialize:** $\alpha_i^d = 1;\ \beta_i^d = 1;\ \hat{p}_i^d = 0.5\ \forall_i \in A, \forall_d \in D$
3. **For time step** $t = 1, \cdots, T$
4. For arm $i = 1, \cdots, A$
5. For all objectives $d \in D$, Sample P_i^d from Beta(α_i^d, β_i^d)
6. End For
7. Find: Pareto optimal arms $\mathcal{I}^*$ such that $\forall_i \in \mathcal{I}^*$ and $\forall_j \notin \mathcal{I}^*$, $\boldsymbol{P}_j \not\succ \boldsymbol{P}_i$
8. Select i^* uniformly, at random from $\mathcal{I}^*$
9. Observe: $\boldsymbol{r}_{i^*}$; Update: $\boldsymbol{\alpha}_{i^*}, \boldsymbol{\beta}_{i^*}$; Compute: unfairness & Pareto regrets
10. **End For**
11. **Output:** Unfairness regret; Pareto regret

Fig. 1: Algorithm: (Pareto Thompson sampling PTS).

and exploitation by using randomness of the Beta distribution. With Bayesian priors on the success rate p_i of each arm i, TS assumes initially the number of successes, α_i and the number of failures, β_i of each arm i is 1. At each time t, TS samples the *probability of selection* P_i of each arm $i \in A$ (*the probability that an arm i is optimal*) from Beta distribution, i.e. $P_i = \text{Beta}(\alpha_i, \beta_i)$. TS selects the optimal arm i^* that has the maximum probability of selection P_{i^*}, $i^* = \text{argmax}_{i \in A}\, P_i$ and observes the reward r_{i^*}. If $r_{i^*} = 1$, then TS updates the number of successes $\alpha_{i^*} \leftarrow \alpha_{i^*} + 1$ of the arm i^*. If $r_{i^*} = 0$, then TS updates the number of failures $\beta_{i^*} \leftarrow \beta_{i^*} + 1$ of the arm i^*. Since, TS is very easy to implement [9], we will extend it to $MOMAB$.

Pareto Thompson Sampling (PTS) [5] explores all the arms by using randomness, it calculates a probability of selection $\boldsymbol{P}_i = [P_i^1, \cdots, P_i^D]^T$ of each arm i. It uses Pareto dominance relation to exploit the optimal arms. The pseudocode of the PTS algorithm is given in Figure (1). Initially (Step 2), PTS assumes each arm i is pulled two times and the number of successes $\alpha_i^d = 1$ and failures $\beta_i^d = 1$ in each objective d are equal. At each time step t, it samples the probability of selection vector $\boldsymbol{P}_i$ of each arm $i \in A$ from Beta distribution, $\boldsymbol{P}_i = \text{Beta}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ (Steps 4-6). Note that, PTS does not use Beta distribution to estimated the success rate $\boldsymbol{p}_i$ of an arm i, instead it uses Beta distribution to sample the probability of selection $P_i^d \in (0, 1)$ of each arm i in each objective d. PTS selects its optimal arms $i^* \in \mathcal{I}^*$ that are not dominated by all other arms using Pareto dominance relation, where $\mathcal{I}^*$ is the PTS optimal arm set (Step 7). PTS pulls uniformly at random one of the arms i^*, observes the corresponding reward vector $\boldsymbol{r}_{i^*}$, updates the number of successes $\boldsymbol{\alpha}_{i^*}$, and failures $\boldsymbol{\beta}_{i^*}$ vectors and computes the Pareto and unfairness regrets (Step 9). This procedure is repeated T steps.

Linear scalarized Thompson Sampling function (LSF-TS) converts the MO problem into a single objective one by performing LSF on the probability of selection vector $\boldsymbol{P}_i$ of each arm $i \in A$. The pseudocode of the LSF-TS is given in Figure 2.

Given the scalarized function set $S = (f^1, \cdots, f^S)$ where each scalarized function s (we use s to refer the scalarized function f^s) has different predefined weight set, $\boldsymbol{w}^s = (w^{1,s}, \cdots, w^{D,s})$. For each $s \in S$, LSF-TS assumes each arm i is pulled two times and the number of successes $\alpha_i^{d,s} = 1$ equals to number of failures $\beta_i^{d,s} = 1$ in each objective d (Step 2). After initial playing, LSF-TS pulls uniformly at random

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

1. **Input:** number of arms $|A|$ and objectives $|D|$; reward $r \sim \mathrm{B}(\boldsymbol{p})$; set of scalarized function $S = (f^1, \cdots, f^s, \cdots, f^S)$
2. **Initialization:** $\forall_s \in S$, Set: $\alpha_i^{d,s} = 1; \beta_i^{d,s} = 1; \hat{p}_i^{d,s} = \frac{1}{2} \forall_i \in A, \forall_d \in D$
3. **Repeat**
4. Select: a function $s \in S$ uniformly, at random
5. For arm $i = 1, \cdots, A$
6. For all objectives $d \in D$, Sample $P_i^{d,s}$ from Beta$(\alpha_i^{d,s}, \beta_i^{d,s})$
7. End For
8. Select: the optimal arm $i^{*,s}$ that maximizes f^s
9. Observe: $\boldsymbol{r}_{i^{*,s}}$; Update: $\boldsymbol{\alpha}_{i^*}^s, \boldsymbol{\beta}_{i^*}^s$; Compute: unfairness & Pareto regrets
10. **Until** T
11. **Output:** Unfairness regret; Pareto regret.

Fig. 2: Algorithm: (Linear scalarized Thompson sampling function $LSF\text{-}TS$).

one of the scalarized function (Step 4), samples the probability of selection vector $\boldsymbol{P}_i^s$ of each arm $i \in A$ under s by using Beta distribution, $\boldsymbol{P}_i^s = \text{Beta}(\boldsymbol{\alpha}_i^s, \boldsymbol{\beta}_i^s)$ (Steps 5-7), converts the MO problem into one objective by performing linear scalarized function on the $\boldsymbol{P}_i^s$, Equation 3, and selects the optimal arm $i^{*,s}$ that maximizes the scalarized function s (Step 8). $LSF\text{-}TS$ observes the reward vector of $i^{*,s}$, updates the number of successes $\boldsymbol{\alpha}_{i^*}^s$, and failures $\boldsymbol{\beta}_{i^*}^s$ vectors, Equation 2 and calculates the Pareto, and unfairness regrets (Step 9). This procedure is repeated until the end of playing T steps.

4 Experiments

In this section, we experimentally compare Pareto and linear scalarized function Thompson Sampling, Section 3. The performance measures are: the average cumulative Pareto and unfairness regrets at each time step which are averaged on M experiments. The number of experiments M and the horizon of each experiment T are 1000. The rewards $\boldsymbol{r}_i$ of arms i are drawn from Bernoulli distribution $\mathrm{B}(\boldsymbol{p}_i)$ with unknown true success rate $\boldsymbol{p}_i$. As [5], each arm i is played initially two times and the number of successes $\alpha_i^d = 1$ equals to the number of failures $\beta_i^d = 1$ in each objective d.

Experiment: We use the same example in [1] with extra arms and objectives. The example in [1] contains non convex success rate set with $|A| = 6$ and $D = 2$. The success rate set is $(\boldsymbol{p}_1 = [0.55, 0.5]^T, \boldsymbol{p}_2 = [0.53, 0.51]^T, \boldsymbol{p}_3 = [0.52, 0.54]^T, \boldsymbol{p}_4 = [0.5, 0.57]^T, \boldsymbol{p}_5 = [0.51, 0.51]^T, \boldsymbol{p}_6 = [0.5, 0.5]^T)$. Note that, Pareto front is $A^* = (a_1^*, a_2^*, a_3^*, a_4^*)$ where a_i^* refers to the optimal arm i^*. The suboptimal a_5 is not dominated by the two optimal arms a_1^* and a_4^*, but a_2^* and a_3^* dominates a_5 while a_6 is dominated by all the other mean vectors. In order to compare the variants TS, i.e. PTS and $LSF\text{-}TS$ performances on a more complex MOMAB problems, we add another 14 arms and another 3 objectives resulting in 5-D, 20-A. The Pareto front A^* contains now 7 arms. We consider 11 weight sets, $\boldsymbol{W} = \{(1, 0)^T, (0.9, 0.1)^T, \cdots, (0.1, 0.9)^T, (0, 1)^T\}$ for $LSF\text{-}TS$. Figure 3 gives the average cumulative Pareto and unfairness regret performances. The x-axis is the horizon of each experiment. The y-axis is either the cumulative Pareto or unfairness regret performance which is the average of 1000 ex-

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

periments. Figure 3 shows PTS outperforms $LSF\text{-}TS$ according to both regrets performance, where PTS performs slightly better than $LSF\text{-}TS$ according to the Pareto regret and PTS performs dramatically better than $LSF\text{-}TS$ according to the unfairness regret performance.

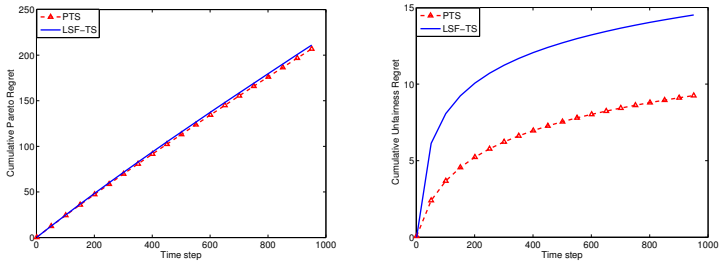

Fig. 3: Performance on 5-objective, 20-armed with $|A^*| = 7$. Left figure shows the cumulative Pareto regret. Right figure shows the cumulative unfairness regret.

5 Conclusions

We presented MOMAB framework. We extended Thompson Sampling TS to the $MOMAB$. We proposed two variants of TS in the $MOMAB$, $LSF\text{-}TS$ and PTS. Finally, we compared PTS, and $LSF\text{-}TS$ and concluded that: PTS outperforms $LSF\text{-}TS$ according to the Pareto and unfairness regrets.

References

[1] M.M. Drugan and A. Nowe, Designing Multi-Objective Multi-Armed Bandits Algorithms: A study, *proceedings of the International Joint Conference on Neural Networks* (IJCNN 2013), 2013.

[2] S.Q. Yahyaa, M.M. Drugan and B. Manderick, Knowledge Gradient for Multi-Objective Multi-Armed Bandit Algorithms, *proceedings of the* 6^{th} *International Conference on Agents and Artificial Intelligence* (ICAART), France, 2014.

[3] E. Zitzler and et al.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7, pages 117–132, 2002.

[4] S.Q. Yahyaa, M.M. Drugan and B. Manderick, The Scalarized Multi-Objective Multi-Armed Bandit Problem: An Empirical Study of its Exploration vs. Exploration Tradeoff, *proceedings of the International Joint Conference on Neural Networks* (IJCNN), 2014.

[5] S.Q. Yahyaa, M.M. Drugan and B. Manderick, Annealing-Pareto Multi-Objective Multi-Armed Bandits Algorithm, *proceedings of IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (ADPRL), Florida, USA, 2014.

[6] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer-Verlag Berlin Heidelberg, 2008.

[7] W.R. Thompson, On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Samples, *Biometrika*, 25(3-4), pages 285–294, 1933.

[8] I. Das and J. E. Dennis, A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems, *Structural Optimization*, 1997.

[9] O. Chapelle and L. Li, An Empirical Evaluation of Thompson Sampling, *proceedings of Advances in Neural Information Processing Systems* (NIPS), 2011.