Annealing-Pareto Multi-Objective Multi-Armed Bandit Algorithm

Saba Q. Yahyaa, Madalina M. Drugan and Bernard Manderick
Vrije Universiteit Brussel, Department of Computer Science
Pleinlaan 2, 1050 Brussels, Belgium
Email: {syahyaa,mdrugan,bmanderi}@vub.ac.be

***Abstract*—In the stochastic multi-objective multi-armed bandit (or MOMAB), arms generate a vector of stochastic rewards, one per objective, instead of a single scalar reward. As a result, there is not only one optimal arm, but there is a set of optimal arms (Pareto front) of reward vectors using the Pareto dominance relation and there is a trade-off between finding the optimal arm set (exploration) and selecting fairly or evenly the optimal arms (exploitation). To trade-off between exploration and exploitation, either Pareto knowledge gradient (or Pareto-KG for short), or Pareto upper confidence bound (or Pareto-UCB1 for short) can be used. They combine the KG-policy and UCB1-policy, respectively with the Pareto dominance relation. In this paper, we propose Pareto Thompson sampling that uses Pareto dominance relation to find the Pareto front. We also propose annealing-Pareto algorithm that trades-off between the exploration and exploitation by using a decaying parameter ϵ_t in combination with Pareto dominance relation. The annealing-Pareto algorithm uses the decaying parameter to explore the Pareto optimal arms and uses Pareto dominance relation to exploit the Pareto front. We experimentally compare Pareto-KG, Pareto-UCB1, Pareto Thompson sampling and the annealing-Pareto algorithms on multi-objective Bernoulli distribution problems and we conclude that the annealing-Pareto is the best performing algorithm.**

I. Introduction

The Multi-Objective Multi-Armed Bandit (MOMAB) problem is a sequential stochastic learning problem. At each time step t, an agent pulls one arm i from an available set of arms A and receives a reward vector $\boldsymbol{r}_i$ of the arm i with D dimensions (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 true probability of success vector parameter of the arm i. The reward vector that the agent receives from the arm i is independent from all other arms and independent from the past reward vectors of the selected arm i. Moreover, the probability of success vector of the arm i has *independent* D distributions. We assume that the true probability of success vector parameter of each arm i is unknown parameter to the agent. Thus, by drawing each arm i, the agent maintains estimation of the true probability of success vector which is known as $\hat{\boldsymbol{p}}_i$.

The MOMAB problem has a set of Pareto optimal arms (Pareto front) A^*, that are incomparable, i.e. can not be classified using a designed partial order relations. The agent has not to only find the optimal arms (exploring), to minimize the total Pareto loss of not pulling the optimal arms, but also has to play them fairly (exploiting), to minimize the total unfairness loss. This problem is known as the *trade-off between exploration and exploitation in the multi-objective optimization*, or the trade-off problem [1]. At each time step t, the Pareto loss (or Pareto regret) is the distance between the set mean (or the true probability of success set) of Pareto optimal arms and the mean of the selected arm (or the true probability of success vector). While, the unfairness loss (or unfairness regret) is the variance in selecting the optimal arms [2]. Thus, the total Pareto regret and the total unfairness regrets are the cumulative summation of the Pareto and unfairness regret over t time steps, respectively. Since, the total unfairness regret grows exponentially on the number of time steps and does not take into its account the total number of selecting optimal arms, we propose to use the *entropy measure* [3] to compute the unfairness regret. The entropy regret is the measurement of disarray or disorder on selecting the optimal arms in the Pareto front A^*.

The Pareto front A^* can be found for example, by using Pareto partial order relation (or Pareto dominance relation) which finds the Pareto front A^* by optimizing directly the Multi-Objective (MO) space [4]. To solve the trade-off problem directly in the MO space, [2] used Upper Confidence Bound (UCB1) [5] policy and [6] used Knowledge Gradient (KG) [7] policy in the MOMAB problem. Both UCB1 and KG policies trade-off between exploration and exploitation by adding an exploration bound to the estimated mean vector (estimated probability of success vector $\hat{\boldsymbol{p}}_i$) to each arm i in each objective $d, d \in D$ and select the optimal arms by using Pareto dominance relation. However, the exploration bound of UCB1 for arm i requires only knowledge about that arm, while in case of KG it also requires knowledge about the other arms.

In this paper, we are interested in the trade-off between exploration and exploitation by using *randomness* instead of adding an exploration bound to the estimated mean vectors. In the one-objective, multi-armed bandit problem, Thompson Sampling [8] trades-off between exploration and exploitation by assigning to each arm i a random probability of selection P_i which is generated from Beta distribution. The random probability of selection P_i of an arm i depends on the performance of the arm i. It will be high value if the arm i has high estimated probability of success $\hat{p}_i$ value. For that reason, firstly, we extend Thompson sampling [8] to the MOMAB to find the optimal arms in the MO space. Pareto Thompson sampling trades-off between exploration and exploitation by assigning to each arm i in each objective d a random probability of selection P_i^d which is generated from Beta distribution. Pareto Thompson sampling uses Pareto dominance relation on the random probability of selection vectors $\boldsymbol{P}_i$, $\boldsymbol{P}_i = [P_i^1, \cdots, P_i^D]^T$ of arms i to find the Pareto front A^*. Secondly, we propose

annealing-Pareto algorithm. The annealing-Pareto trades-off between exploration and exploitation by using a decaying parameter ϵ_t, $\epsilon_t \in (0, 1)$ in combination with the Pareto dominance relation. The ϵ_t parameter has a high value at the beginning of time step t to explore all the available arms and increase the confidence in the estimated means, but as the time step t increases, the ϵ_t parameter decreases to exploit the arms that have maximum estimated mean. To keep track on all the optimal arms in the Pareto front A^*, at each time step t, the annealing-Pareto uses Pareto dominance relation.

The rest of the paper is organized as follows: In Section II, we give background information on the algorithms and the used notation. In Section III, we present Thompson sampling and Pareto Thompson sampling. In Section IV, we introduce the annealing Pareto algorithm. In Section V, we present the performance measures in MOMAB including the proposed entropy measure. In Section VI, we describe the experiments set up followed by experimental results. Finally, we conclude and discuss future work.

II. Background

In this section, we introduce MOMAB framework, Pareto dominance relation, and MOMAB algorithms.

A. 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 selects one arm i and receives a reward vector $\boldsymbol{r}_i$. The reward r_i^d, $r_i^d \in \{0, 1\}$ in each objective d is drawn from a corresponding Bernoulli probability distribution with unknown probability of success p_i^d, i.e. the probability of getting reward equals 1. Thus, by drawing each arm i, the agent estimates the probability of success $\hat{p}_i^d(t)$ for the arm i in the objective d, $d \in D$. Using Bayesian view, the probability of success $\hat{p}_i^d$ can be estimated by using Beta distribution [9]. The probability density of Beta(α_i^d, β_i^d), $\alpha_i^d > 0, \beta_i^d > 0$ is $f(x; \alpha_i^d, \beta_i^d) = \frac{\Gamma(\alpha_i^d + \beta_i^d)}{\Gamma(\alpha_i^d)\Gamma(\beta_i^d)} x^{\alpha_i^d - 1}(1-x)^{\beta_i^d - 1}$, where α_i^d, and β_i^d are the number of successes and failures of the arm i in the objective d, respectively and $\Gamma(y) = y!$. After receiving the reward r_i^d, the updated estimated probability of success $\hat{p}_i^d(t+1)$ at time step $t+1$ is calculated as follows:

$$\hat{p}_i^d(t+1) = \frac{\alpha_i^d(t+1)}{\alpha_i^d(t+1) + \beta_i^d(t+1)} \quad (1)$$

$$\text{where} \quad \alpha_i^d(t+1) = \alpha_i^d(t) + 1, \quad \text{if } r_i^d = 1$$
$$\beta_i^d(t+1) = \beta_i^d(t) + 1, \quad \text{if } r_i^d = 0$$

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

The probability of success vector of arm i, $i \in A$ is represented as $\boldsymbol{p}_i = [p_i^1, \cdots, p_i^D]^T$, where T is the transpose. When the objectives are conflicting with one another then the probability of success p_i^d of arm i corresponding with objective d, $d \in D$, can be better than the component p_j^d of another arm j but worse if we compare the components for another objective d': $p_i^d > p_j^d$ but $p_i^{d'} < p_j^{d'}$ for objectives d and d', respectively. The agent has a set of optimal arms (Pareto front) A^* which can be found by the *Pareto dominance relation*.

B. Pareto Dominance Relation

Pareto dominance relation finds the Pareto front A^* directly in the multi-objective space [4]. It uses the following relations between the probability of success vectors of two arms. We use i and j to refer to the probability of success (estimated or true) vector of arms i and j, respectively.

1) Arm i dominates or is better than j, $i \succ j$, if there exists at least one dimension d for which $i^d \succ j^d$ and for all other dimensions d' we have $i^{d'} \succeq j^{d'}$.
2) Arm i weakly-dominates j, $i \succeq j$, if and only if for all dimensions d, i.e. $d = 1, \cdots, D$ we have $i^d \succeq j^d$.
3) Arm i is incomparable with j, $i \parallel j$, if and only if there exists at least one dimension d for which $i^d \succ j^d$ and there exists another dimension d' for which $i^{d'} \prec j^{d'}$.
4) Arm i is not dominated by j, $j \not\succ i$, if and only if there exists at least one dimension d for which $j^d \prec i^d$. This means that either $i \succ j$ or $i \parallel j$.

Using the above relations, Pareto front $A^*, A^* \subset A$ be the set of arms that are not dominated by all other arms. Moreover, the optimal arms in A^* are incomparable with each other.

C. Multi Objective Multi Armed Bandit Algorithms

Pareto-UCB1 [2] and Pareto-KG [6] trade-off between exploration and exploitation by combination one-objective, Multi-Armed Bandits (MAB) algorithms (or policies) with Pareto dominance relation.

Pareto-UCB1 is the extension of the UCB1 policy [5] to the multi-objective multi-armed bandits. Pareto-UCB1 plays initially each arm i once. At each time step t, it estimates the probability of success vector $\hat{\boldsymbol{p}}$ of each of the MO arms i, i.e. $\hat{\boldsymbol{p}}_i = [\hat{p}_i^1, \cdots, \hat{p}_i^D]^T$ and adds to each objective d an upper confidence bound which represents the *exploration bound* ExpB_i^d in the objective d to trade-off between exploration and exploitation. The exploration bound ExpB_i^d in the objective d of the arm i is calculated as follows:

$$\mathrm{ExpB}_i^d = \sqrt{\frac{2\ln(t\sqrt[4]{D|A^*|})}{N_i}}$$

where D is the number of objectives, $|A^*|$ is the number of optimal arms, t is the current time step, and N_i is the number of times arm i has been selected. Pareto-UCB1 uses the Pareto dominance relation, Section II-B to find the Pareto UCB1 optimal arm set A^*_{UCB1}. Thus, for all the non-optimal arms $k \notin A^*_{UCB1}$ there exists a Pareto optimal arm $j^* \in A^*_{UCB1}$ that is not dominated by the arms k:

$$\hat{\boldsymbol{p}}_k + \mathbf{ExpB}_k \not\succ \hat{\boldsymbol{p}}_{j^*} + \mathbf{ExpB}_{j^*}$$

where $\mathbf{ExpB}_{j^*}$, $\mathbf{ExpB}_{j^*} = [\mathrm{ExpB}_{j^*}^1, \cdots, \mathrm{ExpB}_{j^*}^D]$ is the exploration bound vector of the arm j^*. Pareto-UCB1 selects uniformly at random one of the arms j^* in the set A^*_{UCB1} and receives the corresponding reward vector $\boldsymbol{r}_{j^*}$. Pareto-UCB1, updates the estimated probability of success $\hat{\boldsymbol{p}}_{j^*}$ vector, the number of times arm j^* is chosen N_{j^*} and computes the Pareto and the unfairness regrets.

Pareto-KG is the extension of the KG policy [7] to the multi-objective multi-armed bandits. Pareto-KG, plays each

arm initial $Steps$. At each time step t, it calculates an exploration bound $\mathbf{ExpB}_i$, $\mathbf{ExpB}_i = [\mathrm{ExpB}_i^1, \cdots, \mathrm{ExpB}_i^D]^T$ for each arm i. The exploration bound of arm i depends on the estimated probability of success of all arms. The exploration bound of arm i for objective d (ExpB_i^d) is calculated as follows:

$$\mathrm{ExpB}_i^d = (L-t) * |A| D * v_i^d, \quad \text{where}$$

$$v_i^d \hat{=} \begin{cases} \frac{\alpha_i^d}{\alpha_i^d+\beta_i^d}\left(\frac{\alpha_i^d+1}{\alpha_i^d+\beta_i^d+1} - C_i^d\right), & \text{if} \frac{\alpha_i^d}{\alpha_i^d+\beta_i^d} \leq C_i^d < \frac{\alpha_i^d+1}{\alpha_i^d+\beta_i^d+1} \\ \frac{\beta_i^d}{\alpha_i^d+\beta_i^d}\left(C_i^d - \frac{\alpha_i^d}{\alpha_i^d+\beta_i^d+1}\right), & \text{if} \frac{\alpha_i^d}{\alpha_i^d+\beta_i^d+1} \leq C_i^d < \frac{\alpha_i^d}{\alpha_i^d+\beta_i^d} \\ 0 & \text{otherwise} \end{cases}$$

$C_i^d = \max_{i \neq j} \alpha_j^d/(\alpha_j^d + \beta_j^d)$. The parameters α_i^d, β_i^d, and v_i^d are the number of successes, number of failures, and the index of an arm i for dimension d, respectively [9]. The total number of arms is $|A|$, and L is the horizon of an experiment which is the total number of time steps.

After computing the exploration bound $\mathbf{ExpB}_i$ for each arm i, Pareto-KG sums the $\mathbf{ExpB}_i$ of the arm i with the corresponding estimated probability of success $\hat{\boldsymbol{p}}_i$. It uses Pareto dominance relation, Section II-B to find the Pareto-KG optimal arm set A_{KG}^*. Thus, the optimal arm j^*, $j^* \in A_{KG}^*$ is not dominated by all other arms $k, k \in |A|$,

$$\hat{\boldsymbol{p}}_k + \mathbf{ExpB}_k \not\succ \hat{\boldsymbol{p}}_{j^*} + \mathbf{ExpB}_{j^*}$$

Pareto-KG chooses uniformly at random one of the optimal arms in A_{KG}^*. After pulling the optimal arm j^*, it observes the reward vector $\boldsymbol{r}_{j^*}$, updates the estimated probability of success $\hat{\boldsymbol{p}}_{j^*}$ vector, and computes the Pareto and the unfairness regrets. Note that the authors in [6] used Pareto-KG in the MOMAB problems with normal distributions. In this paper, we use Pareto-KG in the MOMAB problems with Bernoulli distributions.

Pareto-UCB1 adds to the estimated probability of success p_i^d of an arm i in the objective d an exploration bound ExpB_i^d and each objective d has the same exploration bound $\mathrm{ExpB}_i^{d=1} = \cdots = \mathrm{ExpB}_i^{d=D} = \mathrm{ExpB}_i$. The exploration bound ExpB_i of the arm i decreases if we are certain in its estimated probability of success p_i^d. While, Pareto-KG adds to the estimated probability of success p_i^d of the arm i in the objective d an exploration bound ExpB_i^d, but this exploration depends on the estimated probability of success of all available arms in the objective d.

III. Pareto Thompson Sampling

In the Bernoulli one-objective, multi-armed bandit MAB, the reward is a stochastic scalar value, therefore, there is only one optimal arm. The reward r_i for an arm i is either 0, or 1 with unknown probability of success p_i. Thompson sampling [8] does not trade-off between exploration and exploitation by adding an exploration bound ExpB_i to the estimated probability of success $\hat{p}_i$ of the arm i, instead it uses randomness of the Beta distribution.

With Bayesian priors on the Bernoulli probability of success p_i of each arm i, Thompson sampling assumes initially the number of successes, α_i and the number of failures, β_i for each arm i is 1. At each time t, Thompson sampling samples the probability of selection P_i for each arm i, $i \in A$ (the probability that an arm i is optimal) from Beta distribution, i.e. $P_i = \mathrm{Beta}(\alpha_i, \beta_i)$. Thompson sampling selects the optimal arm i^* that has the maximum probability of selection P_{i^*}, i.e. $i^* = \mathrm{argmax}_{i \in A} P_i$ and observes the reward r_{i^*}. If $r_{i^*} = 1$, then Thompson sampling updates the number of successes $\alpha_{i^*} = \alpha_{i^*} + 1$ for the arm i^*. If $r_{i^*} = 0$, then Thompson sampling updates the number of failures $\beta_{i^*} = \beta_{i^*} + 1$ for the arm i^*.

Since, Thompson sampling is very easy to implement, empirically performs better than UCB1 policy [10] and it has been shown to be close to optimal [11], we will extend it to MOMABs.

Pareto Thompson sampling explores all the arms by using randomness, it calculates a probability of selection $\boldsymbol{P}_i, \boldsymbol{P}_i = [P_i^1, \cdots, P_i^D]$ for each arm i. Also, it uses Pareto dominance relation to exploit the optimal arms. The pseudocode of the Pareto Thompson sampling algorithm is given in Fig. (1).

```
1. Input: Horizon of an experiment L;time step t;
 arm set A;number of dimensions |D|;number of
 arms |A|;reward distribution r ~ B(p).

2. Initialize: α_i^d = 1; β_i^d = 1; p̂_i^d = 0.5 ∀_{i∈ A, d∈ D}.

3. For time step t = 1, ···, L
4.    For arm i = 1, ···, A
5.       For objective d = 1, ···, D
6.        Sample P_i^d from Beta(α_i^d, β_i^d)
7.       End For
8.    End For
9. Find the Pareto optimal arms set A*_{P_TS}
    such that ∀_i ∈ A*_{P_TS} and ∀_j ∉ A*_{P_TS}
                P_j ⊁ P_i
10. Select i uniformly, randomly from A*_{P_TS}
11. Observe:reward vector r_i, r_i = [r_i^1, ···, r_i^D]^T
12. Update:α_i; β_i; N_i ← N_i + 1
13. Compute:unfairness and Pareto regret
14. End For

15. Output:Unfairness regret;Pareto regret
```

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

As initialization step (step: 2), Pareto Thompson sampling assumes each arm is pulled two times and the number of successes α_i^d, $\alpha_i^d = 1$ equals to number of failures β_i^d, $\beta_i^d = 1$ in each objective d. At each time step t, it samples the probability of selection vector $\boldsymbol{P}_i$ for each arm i, $i \in A$ (the probability that an arm i is optimal). The probability of selection P_i is sampled by using Beta distribution, $\boldsymbol{P}_i = \mathrm{Beta}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ (steps: 4-8). Note that, Pareto Thompson does not use Beta distribution to estimated the probability of success $\boldsymbol{p}_i$ of an arm i, instead it uses Beta distribution to sample the probability of selection $P_i^d, P_i^d \in (0,1)$ of each arm i in each objective d. Pareto Thompson sampling selects its optimal arms $i^*, i^* \in A_{P_{TS}}^*$ that are not dominated by all other arms using Pareto dominance relation, Section II-B, where $A_{P_{TS}}^*$ is the Pareto Thompson sampling optimal arm set (step: 9). Pareto Thompson sampling pulls uniformly at random one of the arms i^* and observes the corresponding reward vector $\boldsymbol{r}_{i^*}$ (step: 11). It updates the number of successes vector $\boldsymbol{\alpha}_{i^*}, \boldsymbol{\alpha}_{i^*} = [\alpha_{i^*}^1, \cdots, \alpha_{i^*}^D]^T$, where $\alpha_{i^*}^d = \alpha_{i^*}^d + 1$ if $r_{i^*}^d = 1$, the

number of failures vector $\boldsymbol{\beta}_{i^*}$, $\boldsymbol{\beta}_{i^*} = [\beta^1_{i^*}, \cdots, \beta^D_{i^*}]^T$, where $\beta^d_{i^*} = \beta^d_{i^*} + 1$ if $r^d_{i^*} = 0$, and the number of times N_{i^*} arm i^* is selected (step: 12). Then, it calculates the Pareto and the unfairness regrets (step: 15). This procedure is repeated until the end of playing L steps.

Pareto Thompson sampling does not trade-off between exploration and exploitation by adding an exploration bound, instead it modifies the estimated probability of success p^d_i of the arm i in each objective d to explore widely the arm i in the objective d.

IV. The Annealing Pareto Algorithm

Annealing-Pareto algorithm has a specific mechanism to control the trade-off between exploration and exploitation. It uses an exponential decay ϵ_t, $\epsilon_t = \epsilon^t_{decay}/(|A|D)$, where ϵ_{decay} is the decay factor parameter and Pareto dominance relation. At the beginning of time step t, ϵ_t has a high value to explore all the available arms. As the time step t is increased, ϵ_t has a low value to exploit only the optimal arms. To keep track on all the optimal arms in the Pareto front A^*, the annealing-Pareto algorithm uses Pareto dominance relation, Section II-B. The decay factor parameter ϵ_{decay}, $\epsilon_{decay} \in (0, 1)$, when the decay factor parameter $\epsilon_{decay} = 0$ means the annealing-Pareto is a fully Pareto dominance relation and when the decay factor parameter $\epsilon_{decay} = 1$ means the annealing-Pareto uses a fixed exponential decay. The pseudocode of the algorithm is given in Fig. (2).

As initialization step, each arm i is pulled two times and the number of successes α^d_i, $\alpha^d_i = 1$ in each objective d, $d \in D$ equals to number of failures β^d_i, $\beta^d_i = 1$. [1] The ϵ-Pareto optimal arm set A^*_ϵ contains all the arms in the arm set A. At each time step t, the annealing-Pareto trades-off between exploration and exploitation by using the decay factor parameter ϵ_{decay}, $\epsilon_{decay} \in (0, 1)$ in the exponential decay ϵ_t to (step: 4). High decay factor parameter ϵ_{decay} value means high exploration and small decay factor parameter ϵ_{decay} value means high exploitation. In each objective d, $d \in D$, the annealing-Pareto detects the optimal arm in that objective $i^{*,d}$, $i^{*,d} = \text{argmax}_{i \in A} \hat{p}^d_i$, where $\hat{p}^d_i$ is the estimated probability of success for arm i in the objective d (step: 7). The annealing-Pareto selects all the arms in the objective d that have estimated probability of success between $[p^{*,d} - \epsilon_t, p^{*,d}]$ and includes them in the corresponding selected arm set S^d (steps: 8-12), where $p^{*,d}$, $p^{*,d} = \max_{i \in A} p^d_i$ is the probability of success of the optimal arm $i^{*,d}$ in the objective d. The annealing-Pareto constructs the total selected arm set $S(t)$ at time step t by reunion of the selected arm set S^d in each objective d (step: 14). To keep track on the Pareto front, the annealing-Pareto uses Pareto dominance relation (step: 17) on the arms j that are elements in the previous ϵ-Pareto optimal arm set $A^*_\epsilon(t-1)$ and are not element in the total selected arm set $S(t)$. If the arm j is not dominated by all other arms, then this arm will be added to the total selected arm set $S(t)$ (step: 18). The annealing-Pareto updates its ϵ-Pareto optimal arm set $A^*_\epsilon(t)$ to be the total selected arm set $S(t)$ (step: 21). It pulls uniformly at random one of the arms i^* that is an element in the ϵ-Pareto optimal arm set $A^*_\epsilon(t)$ (step: 22) and observes the corresponding reward vector $\boldsymbol{r}_{i^*}$ (step: 23). It updates the number of successes vector $\boldsymbol{\alpha}_{i^*}$, $\boldsymbol{\alpha}_{i^*} = [\alpha^1_{i^*}, \cdots, \alpha^D_{i^*}]^T$, where $\alpha^d_{i^*} = \alpha^d_{i^*} + 1$ if $r^d_{i^*} = 1$, the number of failures vector $\boldsymbol{\beta}_{i^*}$, $\boldsymbol{\beta}_{i^*} = [\beta^1_{i^*}, \cdots, \beta^D_{i^*}]^T$, where $\beta^d_{i^*} = \beta^d_{i^*} + 1$ if $r^d_{i^*} = 0$, and estimated the probability of success vector $\hat{\boldsymbol{p}}_{i^*}$, $\hat{\boldsymbol{p}}_{i^*} = [\hat{p}^1_{i^*}, \cdots, \hat{p}^D_{i^*}]^T$, Equation II-A of the pulled arm i^* and the number of times N_{i^*} arm i^* is selected (step: 23). Then, it calculates the Pareto and unfairness regrets (step: 25). This procedure is repeated until the end of playing L time steps which is the horizon of an experiment.

```
1. Input: Horizon of an experiment L;time step t;
 arm set A;number of objectives |D|;number of
 arms |A|;reward distribution r ~ B(p);selected
 arm set S^d (t) = { }∀_d;decay factor ε_decay ∈ (0,1).

2. Initialize: α_i^d = 1; β_i^d = 1; p̂_i^d = 0.5 ∀_{i∈ A, d∈ D};
 initial ε-Pareto front set A*_ε(0) = A.

3. For time step t = 1, ··· , L
4.   The decay factor parameter ε_t = ε^t_decay/(|A||D|)
5.   For objective d = 1, ··· , D
6.     S^d (t) = {φ}
7.     p̂^{*,d} = max_{1≤ i≤ A} p̂_i^d
8.     For arm i = 1, ··· , A
9.       If p̂_i^d ∈ [p̂^{*,d} − ε_t, p̂^{*,d}]
10.        S^d(t) ← {S^d(t), i}
11.      End If
12.    End For
13.  End For
14.  S(t) ← S^1(t) ∪ S^2(t) ∪ ··· ∪ S^D(t)
15.  S_difference ← A*_ε (t − 1) − S(t)
16.  For arm j ∈ S_difference do
17.      If p̂_k ⊁ p̂_j, ∀_k ∈ A
18.      S(t) ← S(t) ∪ j
19.      End If
20.  End For
21.  A*_ε(t) ← S(t)
22.  Select i* uniformly, randomly from A*_ε(t)
23.  Observe:reward vector r_{i*}, r_{i*} = [r^1_{i*}, ··· , r^D_{i*}]^T
24.  Update:α_{i*}; β_{i*}; p̂_{i*}; N_{i*} ← N_{i*} + 1
25.  Compute:unfairness and Pareto regret
26.  End For

27.  Output:Unfairness regret;Pareto regret
```

Fig. 2. Algorithm: (annealing Pareto algorithm).

In Fig. (3), the dynamic of the algorithm is illustrated on bi-objective 5-armed bandit. The optimal arms a^*_1, a^*_2, and a^*_3 have the probability of success p^*_1, p^*_2 and p^*_3, respectively. The non-optimal arms a_4, and a_5 have the probability of success p_4 and p_5, respectively. At the beginning of time step, i.e. $t = 1$ the total selected arm set $S(t)$ almost contains all the arms (optimal arms and non-optimal arms), and the ϵ-Pareto optimal arm set A^*_ϵ contains all the arms as shown in subfigure (a). As the time step increases, $S(t)$ contains some of the optimal arms, i.e. a^*_2 as shown in subfigure (b and c), therefore, to maintain all the Pareto front, the annealing Pareto constructs its updated ϵ-Pareto optimal arm set $A^*_\epsilon(t)$ to be the set that contains the non dominated arms (a^*_1 and a^*_3) in the previous $A^*_\epsilon(t-1)$ and the arms in the set $S(t)$.

[1] We use Bayesian view to update the estimated probability of success $\hat{p}^d_i$ of an arm i in the objective d, therefore, prior knowledge is required.

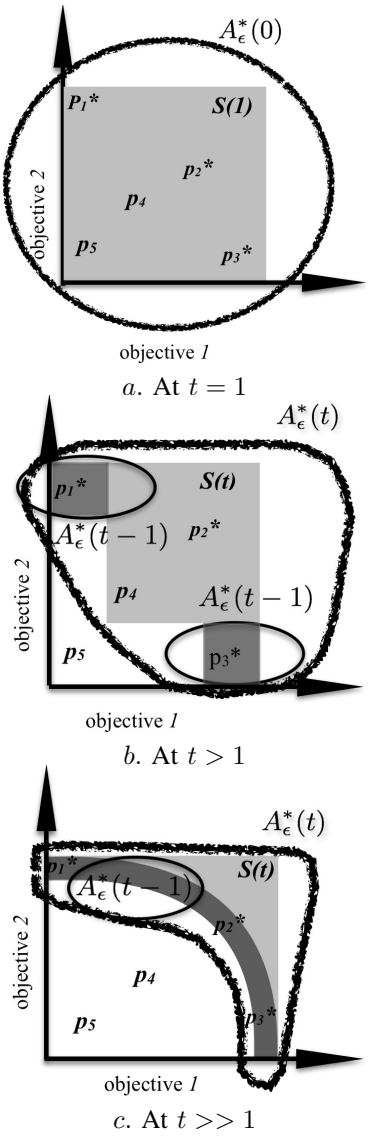


Fig. 3. The dynamic of the annealing-Pareto algorithm.

V. Performance Measures

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.

Pareto regret measure (R_{Pareto}) [2] measures the distance between a probability of success vector of an arm i that is pulled at time step t and the Pareto front A^*. R_{Pareto} is calculated by finding firstly the virtual distance dis^*. The virtual distance dis^* is defined as the minimum distance that is added to the probability of success vector of the pulled arm $\boldsymbol{p}_t$ at time step t in each objective to create a virtual probability of success vector $\boldsymbol{p}^*_t$ that is incomparable with all the arms in Pareto set A^*, where $\boldsymbol{p}^*_t || \boldsymbol{p}_{i^*} \; \forall_{i^* \in A^*}$ as follows:

$$\boldsymbol{p}^*_t = \boldsymbol{p}_t + \boldsymbol{\varepsilon}^*$$

where $\boldsymbol{\varepsilon}^*$, $\boldsymbol{\varepsilon}^* = [dis^{*,1}, \cdots, dis^{*,D}]^T$ is a vector. The Pareto regret is:

$$R_{Pareto} = dis(\boldsymbol{p}_t, \boldsymbol{p}^*_t) = dis(\boldsymbol{\varepsilon}^*, \boldsymbol{0}) \tag{2}$$

where dis, $dis(\boldsymbol{p}_t, \boldsymbol{p}^*_t) = \sqrt{\sum_{d=1}^{D} (p^{*,d}_t - p^d_t)^2}$ is the Euclidean distance between the probability of success vector of the virtual arm $\boldsymbol{p}^*_t$ and the probability of success vector of the pulled arm $\boldsymbol{p}_t$ at time step t. Thus, the regret of the Pareto front is 0, i.e. the mean of the optimal arm coincides itself ($dis^* = 0$ for the arms in the Pareto front).

Pareto regret metric is two variants of the entropy measure, Shannon's entropy and relative entropy [3]. *Shannon's entropy measure* is a measure of disorder (or disarray) on the Pareto front A^*. The higher the entropy, the higher the disorder. The Shannon entropy unfairness regret $R_{SE}(t)$ at time step t is as follows:

$$R_{SE}(t) = -\frac{1}{N_{|A^*|}(t)} \sum_{i^* \in A^*} p_{i^*}(t) \ln(p_{i^*}(t)) \tag{3}$$

where $p_{i^*}(t)$, $p_{i^*}(t) = {}^{N_{i^*}(t)}/_{N(t)}$ is the probability 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 and $N(t)$ is the number of times all arms $i = 1, \cdots, A$ have been selected at time step t, and $N_{|A^*|}(t)$ is the number of times the optimal arms, $i^* = 1, \cdots, |A^*|$ have been selected at time step t.

The *relative entropy measure* is the Kullback-Leibler divergence which is a measure of the difference between two probability distributions Q^* and Q as follows:

$$R_{RE}(t) = \sum_{i=1}^{A} Q^*_i(t) \ln(\frac{Q^*_i(t)}{Q_i(t)}) \tag{4}$$

where $Q^*(t)$ is the optimal distribution of selecting all the available arms $i = 1, \cdots, A$ at time step t, while $Q(t)$ is the distribution of selecting the available arms by an algorithm. The optimal probability of observing of an arm i in the optimal distribution $Q^*(t)$ at time step t is $Q^*_i(t), Q^*_i(t) = {}^{N^*_i(t)}/_t$, where $N^*_i(t)$ is the optimal number of times arm i has been selected at time step t. While, the probability of observing of an arm i in the distribution of an algorithm $Q(t)$ at time step t is $Q_i(t), Q_i(t) = {}^{N_i(t)}/_t$, where $N_i(t)$ is the number of times arm i has been selected at time step t.

The relative entropy takes in its account all the available arms, while Shannon entropy takes in its account only the arms in the Pareto front A^*.

For instance, for 2-objective, 6-armed MOMABs with Pareto front $A^*, A^* = \{a^*_1, a^*_2, a^*_3, a^*_4\}$, where a^*_i is an optimal arm. The number of selecting each arm vector $\boldsymbol{N}$ by an algorithm is $\boldsymbol{N} = [32, 22, 22, 17, 12, 7]^T$ and the optimal number $\boldsymbol{N}^*$ of selecting each arm is $\boldsymbol{N}^* = [27, 27, 27, 27, 2, 2]^T$ at time step $t = 100$ with playing initially each arm 2 times. The Shannon entropy is 0.0143, while the relative entropy is 0.1151. Since we are interesting in playing fairly only the optimal arms, we use Shannon entropy measure to measure the unfairness regret.

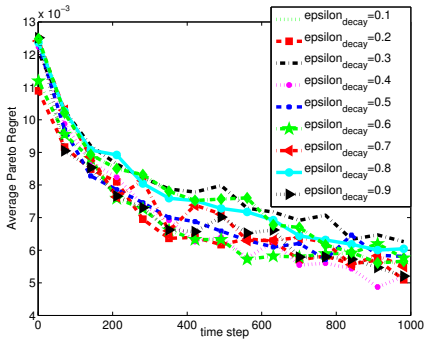


Fig. 4. Average Pareto regret performance of the annealing-Pareto algorithm on 2-objective, 6-armed with non-convex probability of success vector set using different values of the decay factor ϵ_{decay}.

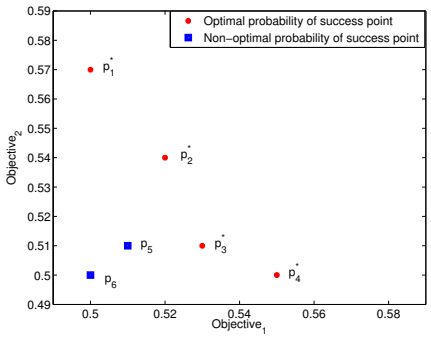
VI. Experiments

In this section, we experimentally compare Pareto-UCB1, Pareto-KG, Section II-C, Pareto Thompson sampling, Section III, and annealing-Pareto, Section IV. The performance measures are: 1) The average Pareto and the cumulative average Pareto regret at each time step which are averaged of M experiments. 2) The average unfairness and the cumulative average unfairness regret at each time step which are averaged of M experiments.

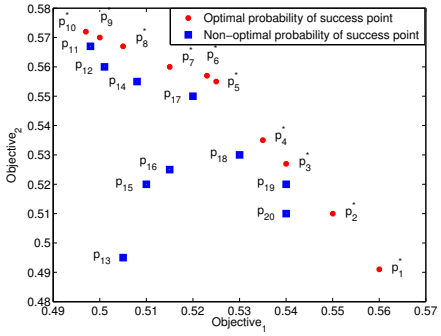
The number of experiments M and the horizon of each experiment L are 1000. The rewards of each arm i in each objective d, $d = 1, \cdots, D$ are drawn from Bernoulli distribution $\mathrm{B}(\boldsymbol{p}_i)$ where $\boldsymbol{p}_i = [p_i^1, \cdots, p_i^D]^T$ is the unknown true probability of success of the arm i. As in the one-objective MABs [11], each arm i is played initially two times and the number of successes α_i^d, $\alpha_i^d = 1$ equals to the number of failures β_i^d, $\beta_i^d = 1$ in each objective d. To get best performance for the annealing-Pareto, the decay factor ϵ_{decay} parameter in the exponential decay ϵ_t, $\epsilon_t = \epsilon_{decay}^t/(|A|D)$ has to be tuned. For example, for 6-armed 2-objective with non-convex probability of success set, Experiment 1, Fig. 4 gives the average Pareto regret performance by using different values of the decay factor ϵ_{decay}, i.e. 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. The x-axis is the time steps. The y-axis is the average Pareto regret. Fig. 4 shows, the performance of $\epsilon_{decay} = 0.4$ outperforms the performance of others ϵ_{decay}, where the average Pareto regret is minimized. This means that the annealing-Pareto does not need high exploration. To get rid of tuning the parameter ϵ_{decay} in each experiment, we generate uniformly at random the decay factor parameter $\epsilon_{decay} \in (0, 1)$. However, *the annealing-Pareto performance will increase if we tune the decay parameter* ϵ_{decay}.

A. Non-Convex Mean Vector Set

Experiment 1. We use the same example in [2], since it is simple to understand and the Pareto probability of success set contains values close to each others. The number of arms $|A|$ equals 6, the number of objectives D equals 2. The true probability of success set vector 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, the Pareto front is $A^* = (a_1^*, a_2^*, a_3^*, a_4^*)$ where a_i^* refers to the optimal arm i^*.



a. Non-convex mean set



b. Convex mean set

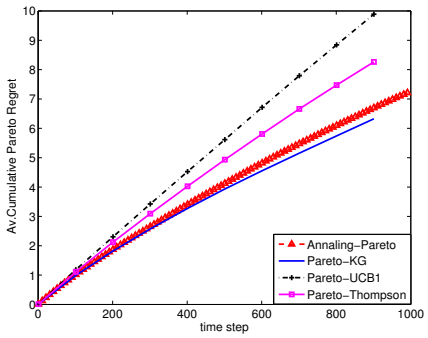
Fig. 5. Non-convex and convex probability of success vector set. Sub-figure a shows a non-convex set with Bi-objective, 6-armed. Sub-figure b shows a convex set with Bi-objective, 20-armed.

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 arms. Fig. 5 shows a set of bi-objective true probability of success with a non-convex set.

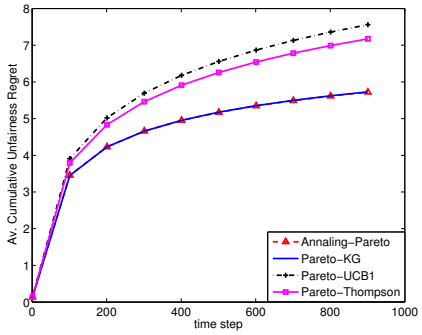
Fig. 6 gives the average cumulative Pareto and unfairness regrets performances. The y-axis is either the average of the cumulative Pareto or unfairness regret performance. The x-axis is the horizon of each experiment. According to the average cumulative Pareto regret performance, Fig. 6 shows Pareto-KG is the best algorithm and Pareto-UCB1 is the worst one. Annealying-Pareto performs better than Pareto-Thompson sampling and worse than Pareto-KG. According to the average cumulative unfairness regret performance, Fig. 6 shows the annealing-Pareto algorithm performs as same as Pareto-KG algorithm. Annealing-Pareto and Pareto-KG are the best algorithms. Pareto Thompson sampling performs better than Pareto-UCB1.

Experiment 2. We add extra 3 objectives and 14 arms to Experiment 1, resulting in 5-objective, 20-armed, we add three optimal arms and 11 dominated arms by all the arms in Pareto front A^*. Pareto front contains 7 optimal arms.

Fig. 7 gives the average cumulative Pareto and unfairness regrets performances. Fig. 7 shows the annealing-Pareto is the best algorithm according to the average cumulative Pareto and unfairness regret performances. Pareto-KG performs better than Pareto Thompson sampling and worse than annealing-Pareto. And, Pareto Thompson sampling performs better than Pareto-UCB1.

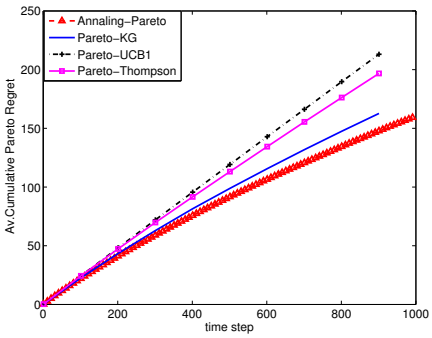


a. Pareto cumulative regret

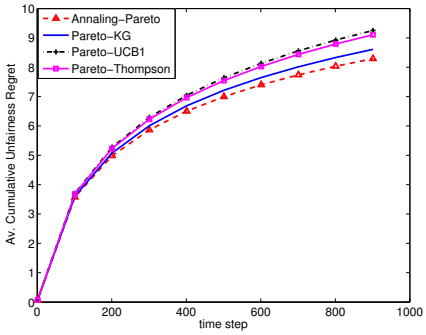


b. Unfairness cumulative regret

Fig. 6. Performance comparison on 2-objective, 6-armed with non-convex probability of success vector set. Sub-figure a shows the average Pareto cumulative regret. Sub-figure b shows the average cumulative unfairness regret.



a. Pareto cumulative regret



b. Unfairness cumulative regret

Fig. 7. Performance comparison on 5-objective, 20-armed with non-convex probability of success vector set. The average cumulative Pareto and unfairness regret performances are shown in sub-figures a and b, respectively.

From Experiment 1 and 2, we see that as the number of optimal equals 4, Fig. 6 shows the Pareto-KG algorithm outperforms all the other algorithms according to the average cumulative Pareto regret performance, while according to the average unfairness regret Pareto-KG performs as same as the annealing-Pareto and they outperform all the other algorithms. As the number of optimal arms and objectives are increased, Fig. 7 shows the annealing-Pareto performance outperforms the performance of all other algorithms.

B. Convex Mean Vector Set

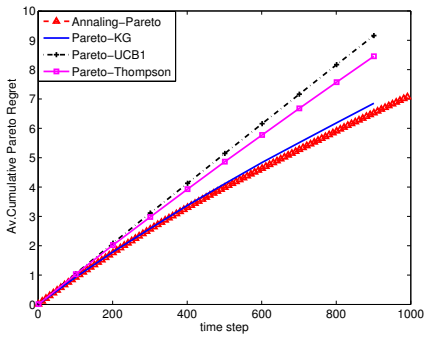
Experiment 3. With number of objectives D equals 2, number of arms $|A|$ equals 20 and convex Pareto probability of success set, ($\boldsymbol{p}_1 = [.56, .491]^T, \boldsymbol{p}_2 = [.55, .51]^T, \boldsymbol{p}_3 = [.54, .527]^T, \boldsymbol{p}_4 = [.535, .535]^T, \boldsymbol{p}_5 = [.525, .555]^T, \boldsymbol{p}_6 = [.523, .557]^T, \boldsymbol{p}_7 = [.515, .56]^T, \boldsymbol{p}_8 = [.505, .567]^T, \boldsymbol{p}_9 = [.5, .57]^T, \boldsymbol{p}_{10} = [.497, .572]^T, \boldsymbol{p}_{11} = [.498, .567]^T, \boldsymbol{p}_{12} = [.501, .56]^T, \boldsymbol{p}_{13} = [.505, .495]^T, \boldsymbol{p}_{14} = [.508, .555]^T, \boldsymbol{p}_{15} = [.51, .52]^T, \boldsymbol{p}_{16} = [.515, .525]^T, \boldsymbol{p}_{17} = [.52, .55]^T, \boldsymbol{p}_{18} = [.53, .53]^T, \boldsymbol{p}_{19} = [.54, .52]^T, \boldsymbol{p}_{20} = [.54, .51]^T$). The Pareto front A^* contains 10 optimal arms, $A^* = (a_1^*, a_2^*, a_3^*, a_4^*, a_5^*, a_6^*, a_7^*, a_8^*, a_9^*, a_{10}^*)$. Fig. 5 shows a convex set of bi-objective true probability of success.

Experiment 4. We add extra 3 objectives and 10 arms to Experiment 3, resulting in 5-objective, 20-armed, we add dominated arms by all the arms in A^*. Pareto front still contains 10 optimal arms.

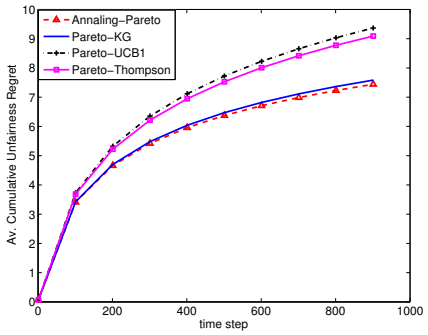
Fig. 8 and Fig. 9 give the average cumulative Pareto and unfairness regrets performances. Fig. 8 and Fig. 9 show the annealing-Pareto is the best algorithm, and the Pareto-UCB1 is the worst algorithm. The Pareto-KG performs worse than the annealing-Pareto and better than Pareto Thompson sampling. Pareto Thompson sampling performs better than Pareto-UCB1 and worse than Pareto-KG.

From Experiment 3 and 4, we see that as the number of objectives equals 2, the performance of the annealing-Pareto is slightly better than Pareto-KG and dramatically better than Pareto-UCB1 and Pareto Thompson sampling. However, as the number of objectives is increased, the performance of the annealing-Pareto is slightly better than Pareto-KG, Pareto-UCB1 and Pareto Thompson sampling.

From the above experiments, we see that Pareto-KG performs better than Pareto-UCB1. The intuition is the added exploration bound. Pareto-UCB1 adds the same exploration bound $\text{ExpB}_i^1 = \cdots = \text{ExpB}_i^D$ to the estimated probability of success $\hat{p}_i^d$ of an arm i in each objective d, $d \in D$, each arm has the same exploration bound over all the objectives D. The added exploration bound by Pareto-UCB1 decreases faster to 0 after some confidence in the estimated probability of success. While, Pareto-KG adds different exploration bound (each objective d of an arm i has its own exploration bound) which depends on all arms and the added exploration bound does not decrease faster to 0, since Pareto-KG explores better than Pareto-UCB1. Pareto Thompson sampling performs better than Pareto-UCB1 and worse than Pareto-KG because it uses randomness of Beta distribution, as a result it explores widely all the arms in the arm set A. The annealing-Pareto performs

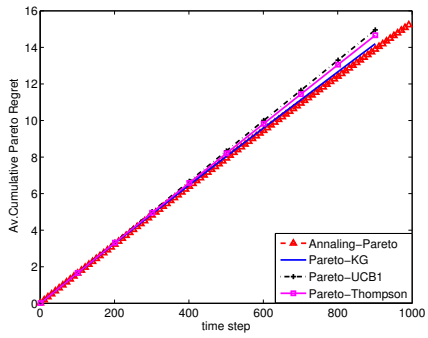


a. Pareto cumulative regret

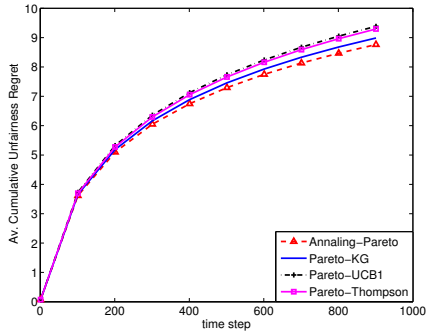


b. Unfairness cumulative regret

Fig. 8. Performance comparison on 2-objective, 20-armed with convex probability of success vector set. Sub-figure a shows the Pareto cumulative regret performance. Sub-figure b shows the unfairness regret performance.



a. Pareto cumulative regret



b. Unfairness cumulative regret

Fig. 9. Performance comparison on 5-objective, 20-armed with convex probability of success vector set. Sub-figure a shows the Pareto cumulative regret performance. Sub-figure b shows the unfairness regret performance.

the best because it explores widely only the optimal arms and the near to the optimal arms not all the arms, i.e. the optimal and the non-optimal arms.

VII. Conclusion

We introduced Bernoulli multi-objective, multi-armed bandit problem MOMAB and Pareto dominance relations. We also introduced Pareto-KG and Pareto-UCB1. We extended Pareto Thompson sampling to the MOMAB problem. We proposed annealing-Pareto algorithm. We introduced the performance measure in the MOMAB. We proposed using the entropy measure as a performance measure in the MOMAB. We studied empirically the trade-off between exploration and exploitation in the MOMAB. Pareto-KG and Pareto-UCB1 trade-off between exploration and exploitation by using knowledge gradient policy KG and upper confidence bound policy UCB1, respectively. Pareto Thompson sampling trades-off between exploration and exploitation by using randomness of Beta distribution. While, the annealing-Pareto trades-off between exploration and exploitation by using a decay factor parameter. Finally, we compared Pareto-KG, Pareto-UCB1, Pareto Thompson sampling and the annealing-Pareto and concluded that: the annealing-Pareto is the outperformed one according to both the Pareto regret performance measure and the unfairness regret performance measure.

References

[1] S. Q. Yahyaa, M. M. Drugan, B. Manderick, "The Scalarized Multi-Objective Multi-Armed Bandit Problem: An Empirical Study of its Exploration vs. Exploration Tradeoff", in *Proc. International Joint Conference on Neural Networks (IJCNN'14)*, Beijing, China, July 2014.

[2] M. M. Drugan and A. Nowe, "Designing Multi-Objective Multi-Armed Bandits Algorithms: A study", in *Proc. International Joint Conference on Neural Networks (IJCNN'13)*, Texas, USA, Aug. 2013.

[3] J. Sethna, *Statistical Mechanics: Entropy, Order Parameters and Complexity*. Oxford University Press, 2006.

[4] E. Zitzler and et al., "Performance Assessment of Multiobjective Optimizers: An Analysis and Review", *IEEE Trans. on Evolutionary Computation*, vol. 7, pp. 117-132. 2002.

[5] P. Auer, N. Cesa-Bianchi and P. Fischer, "Finite-Time Analysis of the Multiarmed Bandit Problem", *Machine Learning*, vol. 47, no. 2-3, pp. 235–256. 2002.

[6] S. Q. Yahyaa, M. M. Drugan and B. Manderick, "Knowledge Gradient for Multi-Objective Multi-Armed Bandit Algorithms", in *Proc. International Conference on Agents and Artificial Intelligence (ICAART'14)*, Angers, France, 2014.

[7] I. O. Ryzhov, W. B. Powell and P. I. Frazier, "The Knowledge Gradient Policy for a General Class of Online Learning Problems", *Operation Research* 2011.

[8] W. R. Thompson, "On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Samples", *Biometrika*, vol. 25, no. 3-4, pp. 285–294. 1933.

[9] W. B. Powell and I. O. Ryzhov, *Optimal Learning*. John Willey and Sons, Canada, 2012.

[10] O. Chapelle, L. Li, "An Empirical Evaluation of Thompson Sampling", in *Proc. Advances in Neural Information Processing Systems (NIPS'11)*, 2011.

[11] S. Agrawal, N. Goyal, "Analysis of Thompson Sampling for the Multi-armed Bandit Problem", in *Proc. and work Annual Conference on Learning Theory (COLT'12)*, vol. 23, pp. 22492257. 2012.