Knowledge Gradient for Multi-objective Multi-armed Bandit Algorithms

Saba Q. Yahyaa, Madalina M. Drugan and Bernard Manderick
Department of Computer Science, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Keywords: Multi-armed Bandit Problems, Multi-objective Optimization, Knowledge Gradient Policy.

Abstract: We extend knowledge gradient (KG) policy for the multi-objective, multi-armed bandits problem to efficiently explore the Pareto optimal arms. We consider two partial order relationships to order the mean vectors, i.e. Pareto and scalarized functions. Pareto KG finds the optimal arms using Pareto search, while the scalarizations-KG transform the multi-objective arms into one-objective arm to find the optimal arms. To measure the performance of the proposed algorithms, we propose three regret measures. We compare the performance of knowledge gradient policy with UCB1 on a multi-objective multi-armed bandits problem, where KG outperforms UCB1.

1 INTRODUCTION

The single-objective multi-armed bandits (MABs) problem is a sequential Markov Decision Process (MDP) of an agent that tries to optimize its decisions while improving its knowledge on the arms. At each time step t, the agent pulls one arm and receives reward as a feedback signal. The reward that the agent receives is independent from the past implementations and independent from all other arms. The rewards are drawn from a static distribution, e.g. normal distributions $N(\mu, \sigma^2)$, where μ is the true mean and σ^2 is the variance. We assume that the true mean and variance parameters are unknown to the agent. Thus, by drawing each arm, the agent maintains estimations of the true mean and the variance which are known as $\hat{\mu}$ and $\hat{\sigma}^2$, respectively.

The goal of the agent is to minimize the *loss* of not pulling the best arm i^* that has the maximum mean all the time. The loss, or *total expected regret*, is defined for any fixed time steps L as:

$$R_L = L\mu^* - \sum_{t=1}^{L} \mu_t \quad (1)$$

where $\mu^* = \max_{i=1,\cdots,|A|} \mu_i$ is the true mean of the greedy (best) arm i^* and μ_t is the true mean of the selected arm i at time step t.

In the multi-armed bandits problem, at each time step t, the agent either selects the arm that has the maximum estimated mean (exploiting the greedy arm), or selects one of the non-greedy arms in order to be more confident about its estimations (exploring one of the available arms). This problem is known as *the trade-off between exploitation and exploration* (Sutton and Barto, 1998). To overcome this problem, (Yahyaa and Manderick, 2012) have compared several action selection policies on the multi-armed bandits problem (MABs) and have shown that Knowledge Gradient (KG) policy (I.O. Ryzhov and Frazier, 2011) outperforms other MABs techniques.

In this paper, we extend knowledge gradient KG policy (I.O. Ryzhov and Frazier, 2011) to vector means, obtaining the *Multi-Objective Knowledge Gradient* (MOKG). In the multi-objective setting, there is a set of Pareto optimal arms that are incomparable, i.e. can not be classified using a designed partial order relationship. Thus, the agent trades-off the conflicting objectives (or dimensions) of the mean vectors, the exploration (finding the Pareto front set) and the exploitation (selecting fairly the optimal arms).

The Pareto optimal arm set is found either by using: i) the Pareto partial order relationship (Zitzler and et al., 2002), or ii) the scalarized functions (Eichfelder, 2008). Pareto partial order finds the Pareto front set by optimizing directly the multi-objective space. The scalarized functions convert the multi-objective space to a single-objective space, i.e. the mean vectors are transformed in scalar values. There are two types of scalarization functions, linear and non-linear (or Chebyshev) functions. Linear scalar-

Q. Yahyaa S., M. Drugan M. and Manderick B..
Knowledge Gradient for Multi-objective Multi-armed Bandit Algorithms.
DOI: 10.5220/0004796600740083
In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence* (ICAART-2014), pages 74-83
ISBN: 978-989-758-015-4

ization function is simple and intuitive but can not find all the optimal arms in a non-convex Pareto front set. In opposition, Chebyshev scalarization function has an extra parameter to be tuned, however can find all the optimal arms in a non-convex Pareto front set. Recently, (Drugan and Nowe, 2013) have used a multi-objective version of the Upper Confidence Bound (UCB1) policy to find the Pareto optimal arm set (exploring) and select fairly the optimal arms (exploiting), i.e. solve the trade-off problem in the Multi-Objective, Multi-Armed Bandits (MOMABs) problem. We compare KG policy and UCB1 on the MOMABs problem.

The rest of the paper is organized as follows. In Section 2 we present background information on the algorithms and the used notation. In Section 3 we introduce multi-objective, multi-armed bandits framework and upper confidence bound policy UCB1 in multi-objective normal distributions bandits. In Section 4 we introduce knowldge gradient (KG) policy and we propose Pareto knowldge gradient algorithm, linear scalarized knowledge gradient across arms algorithm, linear scalarized knowledge gradient across dimensions algorithm, and Chebyshev scalarized knowledge gradient algorithm. In Section 5 we present scalarized multi-objective bandits. In Section 6, we describe the experiments set up followed by experimental results. Finally, we conclude and discuss future work.

2 BACKGROUND

In this section, we introduce the Pareto partial order relationship, order relationships for scalarization functions and regret performance measures of the multi-objective, multi-armed bandits problem.

Let us consider the multi-objective, multi-armed bandits (MOMABs) problem with $|A|, |A| \geq 2$ arms and with D objectives (or dimensions). Each objective has a specific value and the objectives are conflicting with each other. This means that the value of arm i can be better than the value of arm j in one dimension and worse than the value of arm j in other dimension.

2.1 The Pareto Partial Order Relationship

Pareto partial order finds the Pareto optimal arm set directly in the multi-objective space (Zitzler and et al., 2002). Pareto partial order uses the following relationships between the mean vectors of two arms. We use i and j to refer to the mean vector (estimated mean vector or true mean 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 o we have $i^o \succeq j^o$.
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 o for which $i^o \prec j^o$.
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 relationships, the Pareto optimal arm A^* set, $A^* \subset A$ be the set of arms that are not dominated by all other arms. Then:

$$\forall_{a^*} \in A^*, \text{ and } \forall_o \notin A^*(\forall_o \in A), \quad \text{we have } o \not\succ a^*$$

Moreover, the Pareto optimal arms A^* are incomparable with each other. Then:

$$\forall_{a^*,b^*} \in A^*, \quad \text{we have } a^* \parallel b^*$$

2.2 The Scalarized Functions Partial Order Relationships

In general, scalarization functions convert the multi-objective into single-objective optimization (Eichfelder, 2008). However, solving a multi-objective optimization problem means finding the Pareto front set. Thus, we need a set of scalarized functions S to generate a variety of elements belonging to the Pareto front set. There are two types of scalarization functions that weigh the mean vector, linear and non-linear (Chebyshev) scalarization functions.

The linear scalarization assigns to each value of the mean vector of an arm i a weight w^d and the result is the sum of these weighted mean values. The linear scalarized across mean vector is:

$$f^j(\mu_i) = w^1\mu_i^1 + \cdots + w^D\mu_i^D \qquad (2)$$

where $(w^1, \cdots, w^D)$ is a set of predefined weights for the linear scalarized function j, $j \in S$, such that $\sum_{d=1}^{D} w^d = 1$ and μ_i is the mean vector of arm i. The linear scalarization is very popular because of its simplicity. However, it can not find all the arms in the Pareto optimal set A^* if the corresponding mean set is a non-convex set.

The Chebyshev scalarization beside weights, Chebyshev scalarization has a D-dimensional reference point, i.e. $z = [z^1, \cdots, z^D]^T$. The Chebyshev

scalarized can find all the arms in a non-convex Pareto mean front set by moving the reference point (Miettinen, 1999). For maximization multi-objective multi-armed bandits problem, the Chebyshev scalarization is (Drugan and Nowe, 2013):

$$f^j(\mu_i) = \min_{1\leq d\leq D} w^d(\mu_i^d - z^d),\ \forall_i \quad (3)$$

$$z^d = \min_{1\leq i\leq A} \mu_i^d - \varepsilon^d,\ \forall_d$$

where ε is a small value, $\varepsilon > 0$. The reference point z is dominated by all the optimal mean vectors. Thus, it is the minimum of the current mean vector minus ε value.

After transforming the multi-objective problem to single-objective problem, the scalarized functions select the arm that has the maximum function value:

$$i^* = \max_{1\leq i\leq A} f^j(\mu_i)$$

2.3 The Regret Metrics

To measure the performance of the Pareto, scalarized functions partial order relationships, (Drugan and Nowe, 2013) have proposed three regret metric criteria.

1. *Pareto regret metric* R_{Pareto} measures the distance between a mean vector of an arm i that is pulled at time step t and the Pareto optimal mean set. 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 mean vector of the pulled arm μ_t at time step t in each dimension to create a virtual mean vector μ_t^* that is incomparable with all the arms in Pareto set A^*, where $\mu_t^* || \mu_i \forall_{i\in A^*}$ as follows:

$$\mu_t^* = \mu_t + \varepsilon^*$$

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

$$R_{Pareto} = dis(\mu_t, \mu_t^*) = dis(\varepsilon^*, 0) \quad (4)$$

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

2. *The scalarized regret metric* measures the distance between the maximum value of a scalarized function and the scalarized value of an arm that is pulled at time step t. Scalarized regret is the difference between the maximum value for a scalarized function f^j which is either Chebyshev or linear on the set of arms A and the scalarized value for an arm k that is pulled by the scalarized f^j at time step t,

$$R_{scalarized^j}(t) = \max_{1\leq i\leq A} f^j(\mu_i) - f^j(\mu_k)(t) \quad (5)$$

3. *The unfairness regret metric* is related to the variance in drawing all the optimal arms. The unfairness regret of multi-objective, multi-armed bandits problem is the variance of the times the arms in A^* are pulled:

$$R_{unfairness}(t) = \frac{1}{|A^*|}\sum_{i^*\in A^*}(N_{i^*}(t) - N_{|A^*|}(t))^2 \quad (6)$$

where $R_{unfairness}(t)$ is the unfairness regret at time step t, $|A^*|$ is the number of optimal arms, $N_{i^*}(t)$ is the number of times an optimal arm i^* has 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.

3 MOMABs FRAMEWORK

At each time step t, the agent selects one arm i and receives a reward vector. The reward vector is drawn from a normal distribution $N(\mu_i, \sigma_i^2)$, where $\mu_i = [\mu_i^1, \cdots, \mu_i^D]^T$ is the true mean vector and $\sigma_i = [\sigma_i^1, \cdots, \mu_i^D]^T$ is the standard deviation vector of arm i, and T is the transpose.

The true mean and standard deviation vectors of arms i are unknown to the agent. Thus, by drawing each arm i, the agent estimates the mean vector $\hat{\mu}_i$ and the standard deviation vector $\hat{\sigma}_i^2$. The agent updates the estimated mean $\hat{\mu}_i$ and the estimated variance $\hat{\sigma}^2$ in each dimension d as follows (Powell, 2007):

$$N_{i+1} = N_i + 1 \quad (7)$$

$$\hat{\mu}_{i+1}^d = (1 - \frac{1}{N_{i+1}})\,\hat{\mu}_i^d + \frac{1}{N_{i+1}} r_{t+1}^d \quad (8)$$

$$\hat{\sigma}_{i+1}^{2,d} = \frac{N_{i+1}-2}{N_{i+1}-1}\,\hat{\sigma}_i^{2,d} + \frac{1}{N_{i+1}}(r_{t+1}^d - \hat{\mu}_i^d)^2 \quad (9)$$

where N_i is the number of times arm i has been selected, $\hat{\mu}_{i+1}^d$ is the updated estimated mean of arm i for dimension d, $\hat{\sigma}_{i+1}^{2,d}$ is the updated estimated variance of arm i for dimension d and r_{t+1}^d is the collected reward from arm i in the dimension d.

3.1 UCB1 in Normal MOMABs

In the single-obtimization bandits problem, upper confidence bound UCB1 policy (P. Auer and Fischer,

2002) plays firstly each arm, then adds to the estimated mean $\hat{\mu}$ of each arm i an exploration bound. The exploration bound is an upper confidence bound which depends on the number of times arm i has been selected. UCB1 selects the optimal arm i^* that maximizes the function $\hat{\mu}_i + \sqrt{\frac{2\ln(t)}{N_i}}$ as follows:

$$i^* = \max_{1 \leq i \leq A} \left(\hat{\mu}_i + \sqrt{\frac{2\ln(t)}{N_i}} \right)$$

where N_i is the number of times arm i has been pulled.

In the multi-objective multi-armed bandits problem MOMABs with Bernoulli distributions, (Drugan and Nowe, 2013) have extended UCB1 policy to find the Pareto optimal arm set either by using UCB1 in Pareto order relationship or in scalarized functions. In this paper, we use UCB1 in the multi-objective multi-armed bandits problem with normal distributions.

3.1.1 Pareto-UCB1 in Normal MOMABs

Pareto-UCB1 plays initially each arm i once. At each time step t, it estimates the mean vector of each of the multi-objective arms i, i.e. $\hat{\mu}_i = [\hat{\mu}_i^1, \cdots, \hat{\mu}_i^D]^T$ and adds to each dimension an upper confidence bound. Pareto-UCB1 uses a Pareto partial order relationships, Section 2.1 to find the Pareto optimal arm set $A^*_{P_{UCB1}}$. Thus, for all the non-optimal arms $k \notin A^*_{P_{UCB1}}$ there exists a Pareto optimal arm $j \in A^*_{P_{UCB1}}$ that is not dominated by the arms k:

$$\hat{\mu_k} + \sqrt{\frac{2\ln(t\sqrt[4]{D|A^*|})}{N_k}} \not\succ \hat{\mu_j} + \sqrt{\frac{2\ln(t\sqrt[4]{D|A^*|})}{N_j}}$$

Pareto-UCB1 selects uniformly, randomly one of the arms in the set $A^*_{P_{UCB1}}$. The idea is to select most of the times one of the optimal arm in the Pareto front set, $i \in A^*$. An arm $j \notin A^*$ that is closer to the Pareto front set according to metric measure is more selected than the arm $k \notin A^*$ that is far from A^*.

3.1.2 Scalarized-UCB1 in Normal MOMABs

scalarized UCB1 adds an upper confidence bound to the pulled arm under the scalarized function j. Each scalarized function j has associated a predefined set of weights, $(w^1, \cdots, w^D)^j$, $\sum_{d=1}^{D} w^d = 1$. The upper bound depends on the number of times the scalarized function j has been selected, N^j and on the number of times the arm i has been pulled N_i^j under the scalarized function j. Firstly, the scalarized UCB1 plays each arm once and estimates the mean vector of each arm, $\hat{\mu}_i, i = 1, \cdots, |A|$. At each time step t, it pulls the optimal arm i^* as follows:

$$i^* = \max_{1 \leq i \leq A} \left(f^j(\hat{\mu}_i) + \sqrt{\frac{2\ln(N^j)}{N_i^j}} \right)$$

where f^j is either linear scalarized function, Equation 2, or Chebyshev scalarized function, Equation 3 with a predefined set of weights and $\hat{\mu}_i$ is the estimated mean vector of arm i.

4 MULTI OBJECTIVE KNOWLEDGE GRADIENT

Knowledge gradient (KG) policy (I.O. Ryzhov and Frazier, 2011) is an index policy that determines for arm i the index V_i^{KG} as follows:

$$V_i^{KG} = \tilde{\sigma}_i * x \left(-|\frac{\hat{\mu}_i - \max_{j \neq i, j \in |A|} \hat{\mu}_j}{\tilde{\sigma}_i}| \right)$$

where $\tilde{\sigma}_i = \hat{\sigma}_i / N_i$ is the Root Mean Square Error (RMSE) of the estimated mean of an arm i. The function $x(\zeta) = \zeta\Phi(\zeta) + \phi(\zeta)$ where $\phi(\zeta) = 1/\sqrt{2\pi}\exp(-\zeta/2)$ is the standard normal density and its cumulative distribution is $\Phi(\zeta) = \int_{-\infty}^{\zeta} \phi(\zeta')d\zeta'$. KG chooses the arm i with the largest V_i^{KG} and it prefers those arms about which comparatively little is known. These arms are the ones whose distributions around the estimate mean, $\hat{\mu}_i$ have larger estimated standard deviations, $\hat{\sigma}_i$. Thus, KG prefers an arm i over its alternatives if its confidence in the estimate mean $\hat{\mu}_i$ is low. This policy trades-off between exploration and exploitation by selecting its arm i^*_{KG} as follows:

$$i^*_{KG} = \underset{i \in |A|}{\text{argmax}} \left(\hat{\mu}_i + (L-t)V_i^{KG} \right) \quad (10)$$

where t is a time step and L is the horizon of experiment which is the total number of plays that the agent has. In (Yahyaa and Manderick, 2012), KG policy is the competitive policy for the single-objective multi-armed bandits problem according to the collected cumulated average reward and average frequency of optimal selection performances. Moreover, KG policy does not have any parameter to be tuned. Therefore, we used KG policy in the MOMABs problem.

4.1 Pareto-KG Algorithm

Pareto order knowledge gradient (Pareto-KG) uses the pareto partial order relationship (Zitzler and et al., 2002) to order arms. The pseudocode of Pareto-KG is given in Figure 1. At each time step t, Pareto-KG calculates an exploration bound ExpB for each arm a, ($\text{ExpB}_a = [\text{ExpB}_a^1, \cdots, \text{ExpB}_a^D]^T$). The exploration

bound of arm a depends on the estimated mean of all arms and on the estimated standard deviation of the arm a. The exploration bound of arm a for dimension d (ExpB_a^d) is calculated as follows:

$$\text{ExpB}_a^d = (L-t) * |A| D * v_a^d$$

$$v_a^d = \tilde{\hat{\sigma}}_a^d \, x \left(-|\frac{\hat{\mu}_a^d - \max\limits_{k \neq a, k \in A} \hat{\mu}_k^d}{\tilde{\hat{\sigma}}_a^d}| \right), \quad \forall_{d \in D}$$

where v_a^d is the index of an arm a for dimension d, L is the horizon of experiment which is the total number of time steps, $|A|$ is the total number of arms, D is the number of dimensions and $\tilde{\hat{\sigma}}_a^d$ is the root mean square error of an arm for dimension d which equals $\hat{\sigma}_a^d / \sqrt{N_a}$. N_a is the number of times arm a has been pulled. After computing the exploration bound for each arm, Pareto-KG sums the exploration bound of arm a with the corresponding estimated mean. Thus, Pareto-KG selects the optimal arms i that are not dominated by all other arms $k, k \in |A|$ (step: 4). Pareto-KG chooses uniformly, randomly one of the optimal arms in $A_{P_{KG}}^*$ (step: 5). Where $A_{P_{KG}}^*$ is a set that contains Pareto optimal arms using KG policy. After pulling the chosen arm i, Pareto-KG algorithm, updates the estimated mean $\hat{\mu}_i$ vector, the estimated standard deviation $\hat{\sigma}_i^2$ vector, the number of times arm i is chosen N_i and computes the Pareto and the unfairness regrets.

1. Input: length of trajectory L;time step t;
 number of arms $|A|$;number of dimensions D;
 reward distribution $r \sim N(\mu, \sigma_r^2)$.

2. Initialize: plays each arm *Initial* steps to
 estimate mean vectors $\hat{\mu}_i = [\hat{\mu}_i^1, \cdots, \hat{\mu}_i^D]^T$;
 standard deviation vectors $\hat{\sigma}_i = [\hat{\sigma}_i^1, \cdots, \hat{\sigma}_i^D]^T$.

3. For $t = 1$ to L
4. Find the Pareto optimal arms set $A_{P_{KG}}^*$
 such that $\forall_i \in A_{P_{KG}}^*$ and $\forall_j \notin A_{P_{KG}}^*$

 $\hat{\mu}_j + \text{ExpB}_j \not\succ \hat{\mu}_i + \text{ExpB}_i$

5. Select i uniformly, randomly from $A_{P_{KG}}^*$
6. Observe: reward vector r_i, $r_i = [r_i^1, \cdots, r_i^D]^T$
7. Update: $\hat{\mu}_i$; $\hat{\sigma}_i$; $N_i \leftarrow N_i + 1$
8. Compute: the unfairness regret;Pareto regret
9. End for

10. Output: Unfairness regret, Pareto regret, N.

Figure 1: Algorithm: (Pareto-KG).

4.2 Scalarized-KG Algorithm

Scalarized knowledge gradient (scalarized-KG) functions convert the multi-dimensions MABs to one-dimension MABs and make use of the estimated mean and estimated variance.

4.2.1 Linear Scalarized-KG Across Arms

Linear scalarized-KG across arms (LS1-KG) converts immediately the multi-objective estimated mean $\hat{\mu}_i$ and estimated standard deviation $\hat{\sigma}_i$ of each arm to one-dimension, then computes the corresponding exploration bound ExpB_i. At each time step t, LS1-KG weighs both the estimated mean vector, i.e. $([\hat{\mu}_i^1, \cdots, \hat{\mu}_i^D]^T)$ and estimated variance vector, i.e. $([\hat{\sigma}_i^{2,1}, \cdots, \hat{\sigma}_i^{2,D}]^T)$ of each arm i, converts the multi-dimension vectors to one-dimension by summing the elements of each vector. Thus, we have one-dimension multi armed bandits problem. KG calculates for each arm, an exploration bounds which depends on all other arms and selects the arm that has the maximum estimated mean plus exploration bounds. LS1-KG is as follows:

$$\widetilde{\hat{\mu}}_i = f^j(\hat{\mu}_i) = w^1 \hat{\mu}_i^1 + \cdots + w^D \hat{\mu}_i^D \qquad \forall i \qquad (11)$$

$$\widetilde{\hat{\sigma}}_i^2 = f^j(\hat{\sigma}_i^2) = w^1 \hat{\sigma}_i^{2,1} + \cdots + w^D \hat{\sigma}_i^{2,D} \qquad \forall_i \qquad (12)$$

$$\widetilde{\tilde{\hat{\sigma}}}_i^2 = \widetilde{\hat{\sigma}}_i^2 / N_i \qquad \forall_i \qquad (13)$$

$$v_i = \widetilde{\tilde{\hat{\sigma}}}_i \, x \left(-|\frac{\widetilde{\hat{\mu}}_i - \max\limits_{j \neq i, \, j \in A} \widetilde{\hat{\mu}}_j}{\widetilde{\tilde{\hat{\sigma}}}_i}| \right) \qquad \forall_i \qquad (14)$$

where f^j is a linear scalarization function that has a predefined set of weight $(w^1, \cdots, w^D)$, $\widetilde{\hat{\mu}}_i$, $\widetilde{\hat{\sigma}}_i^2$ are the modified estimated mean and variance of an arm i, respectively which are one-dimension values and $\widetilde{\tilde{\hat{\sigma}}}_i^2$ is the modified RMSE of an arm i which is a one-dimension value. v_i is the KG index of an arm i. $x(\zeta) = \zeta \Phi(\zeta) + \phi(\zeta)$ where Φ and ϕ are the cumulative distribution and the density of the standard normal density, respectively. Linear scalarized-KG across arms selects the optimal arm i^* according to:

$$i_{LS_1KG}^* = \underset{i=1,\cdots,|A|}{\text{argmax}} \; (\widetilde{\hat{\mu}}_i + \text{ExpB}_i) \qquad (15)$$

$$= \underset{i=1,\cdots,|A|}{\text{argmax}} \; (\widetilde{\hat{\mu}}_i + (L-t) * |A| D * v_i) \qquad (16)$$

where ExpB_i is the exploration bound of arm i, $|A|$ is the number of arms, D is the number of dimension, L is the horizon of an experiments, i.e. length of trajectories and t is the time step.

4.2.2 Linear Scalarized-KG across Dimensions

Linear scalarized-KG across dimensions (LS2-KG) computes the exploration bound ExpB_i for each arm, i.e. $\text{ExpB}_i = [\text{ExpB}_i^1, \cdots, \text{ExpB}_i^D]$, adds the ExpB_i to

the corresponding estimated mean vector $\hat{\mu}_i$, then converts the multi-objective problem to one dimension. At each time step t, LS2-KG computes exploration bounds for all dimensions of each arm, sums the estimated mean in each dimension with its corresponding exploration bound, weighs each dimension, then converts the multi-dimension to one-dimension value by taking the summation over each vector of each arm. Linear scalarized-KG across dimensions is as follows:

$$f^j(\hat{\mu}_i) = w^1(\hat{\mu}_i^1 + \text{ExpB}_i^1) + \cdots + w^D(\hat{\mu}_i^D + \text{ExpB}_i^D) \forall_i \quad (17)$$

where

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

$$v_i^d = \tilde{\hat{\sigma}}_i^d \; x \left(-|\frac{\hat{\mu}_i^d - \max\limits_{j \neq i,\, j \in A} \hat{\mu}_j^d}{\tilde{\hat{\sigma}}_i^d}| \right), \quad \forall_{d \in D}$$

$|A|$ is the number of arms, L is the horizon of each experiment, v_i^d is the index of arm i for dimension d, $\hat{\mu}_i^d$ is the estimated mean for dimension d of arm i, $\tilde{\hat{\sigma}}_i^d$ is the RMSE of arm i for dimension d, $ExpB_i^d$ is the exploration bound of arm i for dimension d and $x(\zeta) = \zeta\Phi(\zeta) + \phi(\zeta)$ where Φ and ϕ are the cumulative distribution and the density of the standard normal density, respectively. LS2-KG selects the optimal arm i^* that has maximum $f^j(\hat{\mu}_i)$ as follows:

$$i^*_{LS_2KG} = \underset{i=1,\cdots,|A|}{\text{argmax}} \; f^j(\hat{\mu}_i)$$

4.2.3 Chebyshev Scalarized-KG

Chebyshev scalarized-KG (Cheb-KG) computes the exploration bound of each arm in each dimension, i.e. $\text{ExpB}_i = [\text{ExpB}_i^1, \cdots, \text{ExpB}_i^D]$, then converts the multi-objective problem to one-dimension problem. Cheb-KG is as follows:

$$f^j(\hat{\mu}_i) = \min_{1 \leq d \leq D} w^d(\hat{\mu}_i^d + \text{ExpB}_i^d - z^d) \quad \forall_i \qquad (18)$$

where f^j is a Chebyshev scalarization function that has a predefined set of weights $(w^1, \cdots, w^D)$, ExpB_i^d is the exploration bound of arm i for dimension d which is calculated as follows:

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

$$v_i^d = \tilde{\hat{\sigma}}_i^d \; x \left(-|\frac{\hat{\mu}_i^d - \max\limits_{j \neq i,\, j \in A} \hat{\mu}_j^d}{\tilde{\hat{\sigma}}_i^d}| \right), \quad \forall_{d \in D}$$

And, $z = [z^1, \cdots, z^D]^T$ is a reference point. For each dimension d, the corresponding reference is the minimum of the current estimated means of all arms minus a small positive value, $\varepsilon^d > 0$. The reference z^d for dimension d is calculated as follows:

$$z^d = \min_{1 \leq i \leq |A|} \hat{\mu}_i^d - \varepsilon^d, \qquad \forall_d$$

Cheb-KG selects the optimal arm i^* that has maximum $f^j(\hat{\mu}_i)$ as follows:

$$i^*_{Cheb-KG} = \underset{i=1,\cdots,|A|}{\text{argmax}} \; f^j(\hat{\mu}_i)$$

5 THE SCALARIZED MULTI-OBJECTIEVE BANDITS

The pseudocode of the scalarized MOMABs problem (Drugan and Nowe, 2013) is given in Figure 2. Given the type of the scalarized function f, (f is either linear-scalarized-UCB1, Chebyshev-scalarized-UCB1, linear scalarized-KG across arms, linear scalarized-KG across dimensions or Chebyshev scalarized-KG) and the scalarized function set $(f^1, \cdots, f^S)$ where each scalarized function f^s has different weight set, $w^s = (w^{1,s}, \cdots, w^{D,s})$.

```
1. Input: length of trajectory L;reward vector
 r ~ N(μ,σ_r^2);type of scalarized function f;set
of scalarized function S = (f^1, ..., f^S).

2. Initialize: For s = 1 to S
 plays each arm Initial steps;
 observe (r_i)^s;
 update: N^s ← N^s + 1;N_i^s ← N_i^s + 1;(μ̂_i)^s;(σ̂_i)^s
          End
3. Repeat
4. Select a function s uniformly, randomly
5. Select the optimal arm i* that maximizes the
  scalarized function f^s
6. Observe: reward vector r_{i*}, r_{i*} = [r_{i*}^1, ..., r_{i*}^D]^T
7. Update: μ̂_{i*};σ̂_{i*};N_{i*}^s ← N_{i*}^s + 1;N^s ← N^s + 1
8. Compute: unfairness regret;scalarized regret
9. Until L

10. Output: Unfairness regret;Scalarized regret.
```

Figure 2: Algorithm: (Scalarized multi-objective function).

The algorithm in Figure 2 plays each arm of each scalarized function f^s, *Initial* plays (step: 2). N^s is the number of times the scalarized function f^s is pulled and N_i^s is the number of times the arm i under the scalarized function f^s is pulled. $(r_i)^s$ is the reward of the pulled arm i which is drawn from a normal distribution $N(\mu, \sigma_r^2)$ where μ is the true mean and σ_r^2 is the true variance of the reward. $(\hat{\mu}_i)^s$ and $(\hat{\sigma}_i)^s$ are the estimated mean and standard deviation vectors of the arm i under the scalarized function s, respectively. After initial playing, the algorithm chooses randomly at uniform one of the scalarized function (step: 4), selects the optimal arm i^* that maximizes

the type of this scalarized function (step: 5) and simulates the selected arm i^*. The estimated mean vector $(\hat{\mu}_{i^*})^s$, estimated standard deviation vector $(\hat{\sigma}_{i^*})^s$, and the number $N^s_{i^*}$ of the selected arm and the number of the pulled scalarized function are updated (step: 7). This procedure is repeated until the end of playing L steps which is the horizon of an experiment.

6 EXPERIMENTS

In this section, we experimentally compare Pareto-UCB1, and Pareto-KG and we compare linear-scalarized-UCB1, Chebyshev-scalarized-UCB1, linear scalarized-KG across arms, linear scalarized-KG across dimensions, and Chebyshev scalarized-KG. The performance measures are:

1. The percentage of time optimal arms are pulled, i.e. the average of M experiments that optimal arms are pulled.
2. The percentage of time each of the optimal arms is drawn, i.e. the average of M experiments that each one of the optimal arms is pulled.
3. The average regret at each time step which is the average of M experiments.
4. The average unfairness regret at each time step which is the average of M experiments.

We used the algorithm in Figure 2 for the scalarized functions, and the algorithm in Figure 1 for the Pareto-KG. To compute the Pareto regret, we need to calculate the virtual distance. The virtual distance dis^* that is added to the mean vector μ_t of the pulled arm at time step t (the pulled arm is not element in the Pareto front (Pareto optimal arm) set A^*) can be calculated by firstly ranking all the Euclidean distance dis between the mean vectors of the Pareto optimal arm set and 0 as follows:

$$dis(\mu^*_1,0) < dis(\mu^*_2,0) < \cdots < dis(\mu^*_{|A^*|},0)$$

$$dis_1 < dis_2 < \cdots < dis_{|A^*|}$$

where 0 is a vector, $0 = [0^1,\cdots,0^D]^T$. Secondly, finding the minimum added distance dis^* which is calculated as follows:

$$dis^* = dis_1 - dis(\mu_t,0) \tag{19}$$

where dis_1 is the Euclidean distance between 0 vector and the Pareto optimal mean vector μ^*_1, and $dis(\mu_t,0)$ is the Euclidean distance between the mean vector of the pulled arm that is not element in the Pareto front set and vector 0. Then, add dis^* to the mean vector of the pulled arm μ_t to create a mean vector that is element in the Pareto optimal mean set, i.e. $\mu^*_t = \mu_t + dis^*$ and check if $\mu^*_t$is a virtual vector that is incomparable with the Pareto front set. If μ^*_t is incomparable with the mean vectors of Pareto front set, then dis^* is the virtual distance, calculate the regret. Otherwise, reduce the added distance to find dis^* as follows:

$$dis^* = (dis_1 - \frac{dis_2 - dis_1}{1/D}) - dis(\mu_t,0)$$

where D is the number of dimensions. And, check if dis^* creates μ^*_t that is incomparable with the Pareto front set. If not reduce again the dis^* by using dis_3 instead of dis_2 and so on.

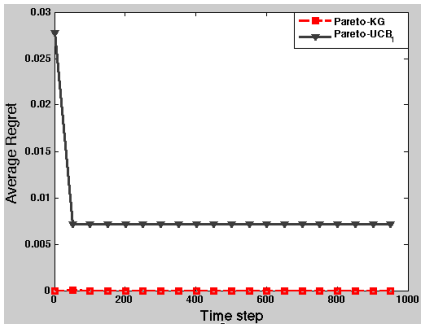
The number of experiments M is 1000. The horizon of each experiment L is 1000. The rewards of each arm i in each dimension d, $d = 1,\cdots,D$ are drawn from normal distribution $N(\mu_i,\sigma^2_{i,r})$ where $\mu_i = [\mu^1_i,\cdots,\mu^D_i]^T$ is the true mean and $\sigma_{i,r} = [\sigma^1_{i,r},\cdots,\sigma^D_{i,r}]^T$ is the true standard deviation of the reward. The true means and the true standard deviations of arms are unknown parameters to the agent.

First of all, we used the same example in (Drugan and Nowe, 2013) because it contains non-convex mean vector set. The number of arms $|A|$ equals 6, the number of dimensions D equals 2. The standard deviation for arms in each dimension is either equal and set to 1, 0.1, or 0.01 or different and generated from a uniform distribution over the closed interval $[0,1]$, i.e. taken from a normal distribution $N(0.5,1/12)$. The true mean set vector is ($\mu_1 = [0.55,0.5]^T$, $\mu_2 = [0.53,0.51]^T$, $\mu_3 = [0.52,0.54]^T$, $\mu_4 = [0.5,0.57]^T$, $\mu_5 = [0.51,0.51]^T$, $\mu_6 = [0.5,0.5]^T$). Note that the Pareto optimal arm set (Pareto front set) 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. For upper confidence bounce UCB1, each arm is played initially one time, i.e. $Initial = 1$ as (Drugan and Nowe, 2013) (for Pareto-, linear-, Chebyshev-UCB1), then the estimated mean of arms are calculated and the scalarized or Pareto selection is computed. Knowledge gradient KG needs the estimated standard deviation for each arm, $\hat{\sigma}_i$, therefore, each arm is either played initially 2 times, $Initial = 2$ which is the minimum number to estimate the standard deviation or each arm is considered unknown until it is visited $Initial$ times. If the arm is unknown, then the estimated mean of that arm has a maximum value, i.e. $\hat{\mu}^d_i = \max_{d\in D}\mu^d_j, \forall_{j,\ j\in|A|}$ and the estimated standard deviation, i.e. $\hat{\sigma}^d_i = \max_{d\in D}\sigma^d_j, \forall_{j,j\in|A|}$ to increase the exploration of arms. We compare the different setting for KG and found out that play-

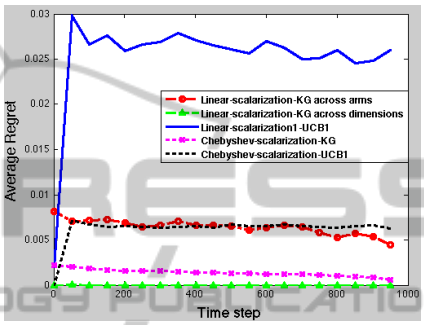
ing each arm initially 2 times, KG performance is increased, therefore, we used this to compare with UCB1. The number of Pareto optimal arms $|A^*|$ is unknown to the agent, therefore, $|A^*| = 6$. We consider 11 weight sets for the linear-, and Chebyshev-UCB1 and linear scalarized-KG across arms (LS1-KG), linear scalarized-KG across dimensions (LS2-KG), and Chebyshev-KG (Cheb-KG) functions, i.e. $w = \{(1,0)^T, (0.9,0.1)^T, \cdots, (0.1,0.9)^T, (0,1)^T\}$. For Chebyshev-UCB1 and Chebyshev-KG, ε was generated uniformly, randomly, $\varepsilon \in [0, 0.1]$.

Table 1 gives the average number $\pm$ the upper and lower bounds of the confidence interval that the optimal arms are selected in column A^*, the average number $\pm$ the upper and lower bounds of the confidence interval that one of the optimal arm a^* is pulled in columns a_1^*, a_2^*, a_3^*, and a_4^* using the scalarized functions in column Functions.

Table 1 shows the number of selecting the optimal arms is increased by using knowledge gradient. Pareto-KG plays fairly the optimal arms. Although ε set to a fixed value for all the scalarized functions set ($j = 1, \cdots, 11$), Chebyshev-KG performs better than the linear scalarization-KG across arms (LS1-KG) and linear scalarization-KG across dimensions (LS2-KG) in playing fairly the optimal arms. While, the performance of linear scalarized-KG across arms (LS1-KG) in playing fairly the optimal arms is as same as linear scalarized-KG across dimensions (LS2-KG). Moreover, LS1-KG prefers the optimal arms a_4^* and a_3^* then a_1^* and a_2^* and LS2-KG prefers the optimal arms a_1^* and a_2^* then a_4^* and a_3^*. Pareto-UCB1 performs better than linear- and Chebyshev-scalarization-UCB1, (LS-UCB1 and Cheb-UCB1, respectively) according to the number of selecting optimal arms. This is the same result in (Drugan and Nowe, 2013) when the rewards are drawn form Bernoulli distributions. Cheb-UCB1 performs better than LS-UCB1 in selecting the optimal arms. We also see that LS-UCB1 performs better than LS1-KG and LS2-KG in playing fairly the optimal arms. And, Cheb-UCB1 performs better than Cheb-KG in playing fairly the optimal arms. Figure 3 shows the average regret performances. The x-axis is the horizon of each experiments and the y-axis is the average of 1000 experiments. From Figure 3, we see that how the regret performance is improved by using KG policy. Minimum Pareto regret is achieved by using Pareto-KG in subfigure (*a*). Minimum scalarized regret is achieved by using LS2-KG in subfigure (*b*) and maximum regret is achieved by using linear-scalarized-UCB1. From subfigure (*b*), we also see Chebyshev-UCB1 performs better than linear-scalarized-UCB1 and linear-scalarized-KG across dimensions performs better than linear scalarized-KG across arms and Chebyshev scalarized-KG.



(*a*) Average Pareto regret performance



(*b*) Average scalarized regret performance

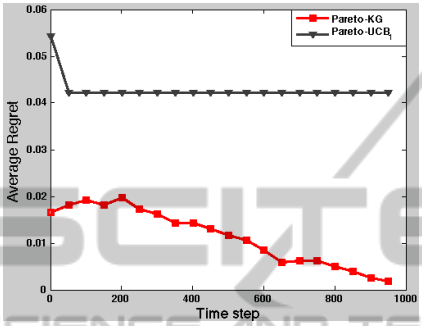
Figure 3: Average regret performance on bi-objective, 6-armed bandit problems.

Secondly, we added another 14 arms to the previous example as (Drugan and Nowe, 2013). The added arms are dominated by all other in A^* and have equal mean vectors, i.e. $\mu_7 = \cdots \mu_{20} = [0.48, 0.48]^T$. Figure 4 gives the average regret and the average unfairness regret performances of the Pareto-KG and Pareto-UCB1. The x-axis is the horizon of each experiments and the y-axis is the average of 1000 experiments. Figure 4 shows the average regret performance is improved by using Pareto-KG in subfigure (*a*), while, the average unfairness performance in subfigure (b) is improved using Pareto-UCB1.

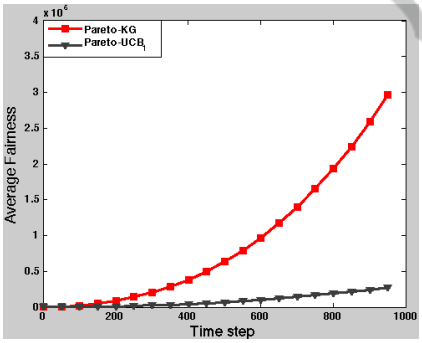
Thirdly, we added extra dimension to the previous example. The Pareto front set A^* contains 7 arms. Figure 5 gives the average regret performance using $\sigma_r = 0.01$. The y-axis is the average regret performance and the x-axis is the horizon of experiments. Figure 5 shows how the performance is improved using KG policy in the MOMABs. Subfigure *a* shows Pareto-KG performs Pareto UCB1. Subfigure *b* shows best performance (the average regret is decreased) for Chebyshev-KG and worst performance for linear-UCB1. Chebyshev-UCB1 performs better than linear-scalarized-KG across dimensions and worse than linear-scalarized-KG across arms. And, the Chebyshev scalarized- (KG and UCB1) is better

Table 1: Percentage of times optimal arms A^* are pulled and percentage of times each one of the optimal arm is pulled performances on bi-objective MABs with number of arms $|A| = 6$ and the standard deviation of rewards are equal for each arm $i, i \in A\ \sigma_{i,r} = 0.01$.

Functions	A^*	a_1^*	a_2^*	a_3^*	a_4^*
LS2-KG	$999 \pm .33$	368 ± 17.6	303 ± 18.2	96 ± 9.3	232 ± 8.5
Pareto-KG	$998 \pm .02$	$250 \pm .85$	$249 \pm .87$	$250 \pm .83$	$249 \pm .82$
LS1-KG	$998 \pm .04$	222 ± 9.7	122 ± 7.4	301 ± 14.4	353 ± 12.2
Cheb-KG	$998 \pm .25$	279 ± 6	228 ± 7	264 ± 6	227 ± 4.3
Pareto-UCB$_1$	$714 \pm .41$	$180 \pm .3$	$163 \pm .21$	$173 \pm .23$	$198 \pm .54$
Cheb-UCB$_1$	$677 \pm .07$	$168 \pm .08$	$166 \pm .06$	$170 \pm .06$	$173 \pm .07$
LS-UCB$_1$	$669 \pm .08$	$167 \pm .06$	$168 \pm .06$	$168 \pm .06$	$166 \pm .06$

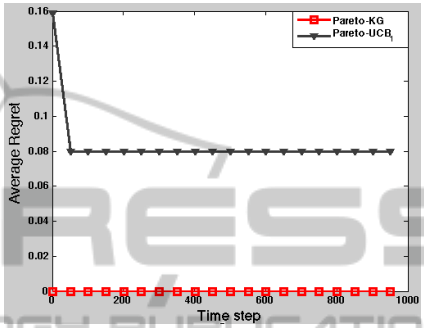


(*a*) Average regret performance.

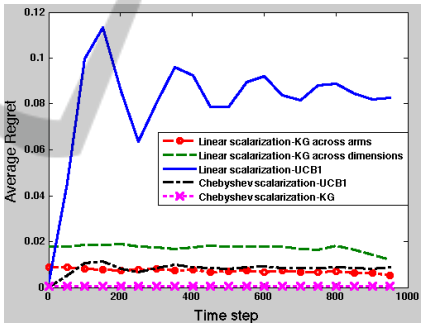


(*b*) Average unfairness regret performance.

Figure 4: Performance comparison of Pareto-KG and Pareto-UCB1on bi-objective MABs with 20 arms using standard deviation of reward $\sigma_r = 0.1$ for all arms. Subfigure (a) is the average regret performance and subfigure (b) is the average unfairness regret performance.



(*a*) Average Pareto regret performance



(*b*) Average scalarized regret performance

Figure 5: Average regret performance on triple-objective, 20-armed bandit problems.

than the linear scalarized- (KG and UCB1) according to the regret performance.

Finally, we added extra 2 objectives in the previous triple-objective in order to compare the KG and UCB1 performances on a more complex MOMABs problem. Table 2 gives the average number $\pm$ the upper and lower bounds of the confidence interval that the optimal arms are selected in column A^*, the average number $\pm$ the upper and lower bounds of the confidence interval that one of the optimal arm a^* is pulled in columns $a_1^*, a_2^*, a_3^*, a_4^*, a_5^*, a_6^*$, and a_7^* using the scalarized functions in column Functions.

Table 2 shows the number of selecting the optimal arms is increased by using KG policy. Pareto-KG outperforms Pareto-UCB1 in selecting and playing fairly the optimal arms. Scalarized functions-KG outperform scalarized functions-UCB1 in selecting the optimal arms, while scalarized functions-UCB1 outperform scalarized functions-KG in playing fairly the optimal arms. LS1-KG (linear scalarized-KG across arms) performs better than LS2-KG and Cheb-KG in selecting the optimal arms. Cheb-KG performs better than LS2-KG and worse than LS1-KG in selecting the optimal arms. LS2-KG performs better than LS1-KG and Cheb-KG in playing fairly the optimal arms and prefers playing $a_2^*, a_1^*, a_7^*, a_5^*, a_6^*, a_3^*$ then a_4^*. LS1-KG performs better than Cheb-KG and worse than LS2-KG in playing fairly the optimal arms and prefers $a_4^*, a_6^*, a_3^*, a_1^*, a_5^*, a_7^*$ then a_2^*. Cheb-KG prefers the optimal arms $a_1^*, a_6^*, a_3^*, a_5^*, a_2^*, a_4^*$, then a_7^*. LS-

Table 2: Percentage of times optimal arms A^* are pulled and percentage of times each one of the optimal arm is pulled performances on 5-objective MABs with number of arms $|A| = 20$ and the standard deviation of rewards are equal for each arm $i, i \in A$ $\sigma_{i,r} = 0.01$.

Functions	A^*	a_1^*	a_2^*	a_3^*	a_4^*	a_5^*	a_6^*	a_7^*
LS1-KG	1000 ± 0	143.1 ± 6.273	76.6 ± 4.566	154.1 ± 7.459	195 ± 7.633	135.8 ± 7.25	164.8 ± 8.353	130.6 ± 6.336
Cheb-KG	$999.7 \pm .023$	507.3 ± 4.111	63.6 ± 4.263	111.7 ± 4.043	29.2 ± 3.076	73.9 ± 4.752	193.5 ± 4.883	20.5 ± 2.574
LS2-KG	601.1 ± 8.993	109.2 ± 6.439	121.8 ± 6.827	57.9 ± 4.454	57.1 ± 4.093	79.3 ± 5.668	79.1 ± 5.536	96.7 ± 6.271
Pareto-KG	571.3 ± 3.54	$81.4 \pm .723$	$81.7 \pm .738$	$81.6 \pm .72$	$81.7 \pm .72$	$81.9 \pm .688$	$81.6 \pm .72$	$81.4 \pm .72$
Pareto-UCB$_1$	$455.1 \pm .21$	$64.2 \pm .095$	$60.3 \pm .066$	$62.7 \pm .073$	$69.1 \pm .116$	$65.1 \pm .076$	$65.1 \pm .077$	$68.6 \pm .114$
LS-UCB$_1$	$379.7 \pm .278$	$53.9 \pm .061$	$53.7 \pm .063$	$54.5 \pm .064$	$54.7 \pm .064$	$54.4 \pm .066$	$54.6 \pm .068$	$53.9 \pm .066$
Cheb-UCB$_1$	$367.9 \pm .219$	$53.4 \pm .073$	$54.1 \pm .075$	$52.9 \pm .073$	$52.7 \pm .075$	$51.9 \pm .074$	$51.6 \pm .077$	$51.3 \pm .077$

UCB1 and Cheb-UCB1 play fairly the optimal arms, while LS-UCB1 performs better than Cheb-UCB1 in selecting the optimal arms.

From the above figures and tables, we conclude that the average regret is decreased using KG policy in the MOMABs problem. Pareto-KG outperforms Pareto-UCB1 and scalarized functions-KG outperform scalarized functions-UCB1 according to the average regret performance. While Pareto-UCB1 outperforms Pareto-KG according to the unfairness regret, where the unfairness regret is increased using knowledge gradient policy. However, when the number of objective is increased Pareto-KG performs better than Pareto-UCB1 in playing fairly the optimal arms. According to the average regret performance, Chebyshev scalarized-KG performs better than linear scalarized-KG across arms and dimensions when the number of arms is increased, while LS1-KG outperforms all other scalarization functions when the number of objectives is increased to 5.

7 CONCLUSIONS AND FUTURE WORK

We presented multi-objective, multi-armed bandits problem MOMABs, the regret measures in the MOMABs and Pareto-UCB1, linear-UCB1, and Chebyshev-UCB1. We also presented knowledge gradient policy KG. We proposed Pareto-KG. We also proposed two types of linear scalarized-KG (linear scalarized-KG across arms (LS1-KG) and linear scalarized-KG across dimensions (LS2-KG) and Chebyshev-scalarized-KG. Finally we compared KG and UCB1 and concluded that the average regret is improved using KG policy in the MOMABs. Future work must provide theoretical analysis for the KG in MOMABs and must compare the family of upper confidence bound UCB1, and UCB1-Tuned policies (P. Auer and Fischer, 2002), and knowledge gradient KG policy on the correlated MOMABs. and must compare KG, UCB1, and UCB1-Tuned policies in sequential ranking and selection (P.I. Frazier and Dayanik, 2008) MOMABs.

REFERENCES

Drugan, M. and Nowe, A. (2013). Designing multi-objective multi-armed bandits algorithms: A study. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

Eichfelder, G. (2008). *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer-Verlag Berlin Heidelberg, 1st edition.

I.O. Ryzhov, W. P. and Frazier, P. (2011). The knowledge-gradient policy for a general class of online learning problems. *Operation Research*.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Springer, illustrated edition.

P. Auer, N. C.-B. and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

P.I. Frazier, W. P. and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM J. Control and Optimization*, 47(5):2410–2439.

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley and Sons, New York, USA, 1st edition.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA, 1st edition.

Yahyaa, S. and Manderick, B. (2012). The exploration vs exploitation trade-off in the multi-armed bandit problem: An empirical study. In *Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. ESANN.

Zitzler, E. and et al. (2002). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7:117–132.