Correlated Gaussian Multi-Objective Multi-Armed Bandit across Arms Algorithm

Saba Q. Yahyaa and Madalina M. Drugan
Vrije Universiteit Brussel, Department of Computer Science
Pleinlaan 2, 1050 Brussels, Belgium
Email: sabayahyaa@gmail.com,madalina.drugan@vub.ac.be

***Abstract*—Stochastic multi-objective multi-armed bandit problem, ($MOMAB$), is a stochastic multi-armed problem where each arm generates a vector of rewards instead of a single scalar reward. The goal of ($MOMAB$) is to minimize the regret of playing suboptimal arms while playing fairly the Pareto optimal arms. In this paper, we consider Gaussian correlation across arms in ($MOMAB$), meaning that the generated reward vector of an arm gives us information not only about that arm itself but also on all the available arms. We call this framework the correlated-$MOMAB$ problem. We extended Gittins index policy to correlated ($MOMAB$) because Gittins index has been used before to model the correlation between arms. We empirically compared Gittins index policy with multi-objective upper confidence bound policy on a test suite of correlated-$MOMAB$ problems. We conclude that the performance of these policies depend on the number of arms and objectives.**

I. Introduction

The Multi-Objective Optimization (MOO) problem with conflicting objectives is omnipresent in the real-world. For instance, in shipping firm, the conflicting objectives might consist of the shipping time and the cost. At the same time, a short shipping time is needed in order to improve customer satisfaction and a few number of used ships is required in order to reduce the operating cost. It is obvious that adding more ships will reduce the needed shipping time but will increase the operating cost. The goal of the MOO with conflicting objectives is to trade off the conflicting objectives [1].

The Multi-Objective Multi-Armed Bandit ($MOMAB$) problem [2], [3] is a straightforward synergy between multi-objective optimization and stochastic Multi-Armed Bandits (MAB) in the sense that MAB is adapted to reward vectors and all Pareto optimal arms are considered equally important. Similarly with MAB, $MOMAB$ is a sequential stochastic learning problem. At each time step n, an agent pulls one arm a from an available set of arms A and receives a reward vector $\boldsymbol{r}_a$ from the arm a with D dimensions (or objectives) as a feedback signal. The reward vector $\boldsymbol{r}_a$ is drawn from a *corresponding* stationary probability distribution vector, for example from a normal probability distribution $N(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2)$, where $\boldsymbol{\mu}_a$ is the *unknown* true mean vector and $\boldsymbol{\sigma}_a^2$ is the *known* variance vector parameters of the arm a. The reward vector $\boldsymbol{r}_a$ that the agent receives from the arm a is *independent* from all other arms and independent from the past reward vectors of the selected arm a. Moreover, the mean vector of the arm a has *independent* D distributions. For each objective $d \in D$, we assume that the agent has a prior multivariate normal distribution belief across all the available arms.

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 [4]. The agent has to trade off between minimizing the Pareto regret, i.e. the total loss of not pulling the optimal arms and thus exploring the sub-optimal arms, and selecting fairly the optimal arms in the Pareto front that minimizes the unfairness loss, i.e. exploiting the Pareto optimal arms [5]. At each time step n, the Pareto regret is defined as the distance between the reward vectors of the Pareto optimal arms and the selected arm [2]. The unfairness regret is the Shannon entropy on the frequency of selecting the optimal arms in the Pareto front [6].

Linear scalarized function [7] is a simple and intuitive method to identify the Pareto front, i.e. the set of Pareto optimal reward vectors, in MOO. Each linear scalarized function has a predefined corresponding weight vector. Given a predefined weight vector $\boldsymbol{w}$, the linear scalarized function weighs each value of the mean vector $\boldsymbol{\mu}_a$ of each arm a, converts the Multi-Objective (MO) space to a single-objective one by summing the weighted mean values and selects the optimal arm a^* that has the maximum scalarized function. Since solving a MOO problem means finding the Pareto front A^*, we need a set of linear scalarized functions to generate a representative set on the Pareto front. For a discrete Pareto front, there is no guarantee that linear scalarized functions can find all the optimal arms in the Pareto front A^* [7]. To improve the performance of the linear scalarized functions in finding and playing fairly the optimal arms, the authors in [2] have used upper confidence bound (UCB_1) policy [8] in the $MOMAB$ problems.

In this paper, we introduce the Correlated Multi-Objective Multi-Armed Bandit ($CMOMAB$) problem where selecting an arm a gives us information about all the available arms A. The $CMOMAB$ can be applied in a lot of MOO problems, e.g. in wireless ad hoc networks [9] when there are shared paths among ways in sending packets from a source node to a destination node or in the shipping firm when there is an overlap among ship's ways. We extend Gittins Index (GI) [10] policy to the $CMOMAB$ in order to find and select fairly the optimal arms (i.e. trade off between exploration and exploitation) since GI policy has been used before to model the correlation in the single-

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.93

objective Multi-Armed Bandit with normal correlated beliefs across arms ($CMAB$) [11]. In $CMOMAB$, GI policy computes for each arm and objective a GI index that will be added to the corresponding mean of the multivariate normal distribution belief in order to trade off between exploring suboptimal arms and fairly selecting the optimal arms. GI policy performs linear scalarized function on the GI index plus the mean of the multivariate normal distribution belief to transform the multi-objective problem into a single-objective one. Finally, we compare GI and UCB_1 policies on a test suite of $CMOMAB$ problems and we conclude that the performance of GI and UCB_1 policies depend on the amount of correlation across arms, the number of arms and objectives, and the used parameters.

The rest of the paper is organized as follows: Section II discusses single-objective multi-armed bandit with normal correlated beliefs across arms, and Gittins index in $CMAB$ and UCB_1 policies in the $CMAB$. Section III introduces the correlated multi-objective multi-armed bandit across arms framework. Section IV extends the GI policy to the $CMOMAB$ problems. Section V extends the UCB_1 policy to the $CMOMAB$ problems. Section VI describes the experiments set up followed by experimental results. Section VII concludes the paper and discusses future work.

II. Background in multi-armed bandit with Gaussian correlated beliefs ($CMAB$)

In this section, we discuss: 1) The multi-armed bandit with Gaussian correlated beliefs ($CMAB$) [11] problem to understand the framework of the $CMOMAB$. $CMAB$ arises in a lot of applications such as drug treatments for human patients when the treatments consist of overlapping sets of drugs, [11], [12]. See [12] for more applications on the $CMAB$. 2) Gittins index policy for the $CMAB$ [11], and 3) UCB_1 policy for the $CMAB$ [11].

The standard single-objective multi-armed bandit with Gaussian correlated beliefs ($CMAB$) is a stochastic MAB problem. At each time step n, an agent selects one arm $a \in A$ from the arm set A and observes a *scalar* reward r_a from the arm a. The observed reward r_a is *independent* from all other arms and independent from the past rewards of the arm a. The reward r_a is drawn from a corresponding normal probability distribution $N(\mu_a, \sigma_a^2)$ with *unknown mean* μ_a and *known variance* σ_a^2. Since the true mean μ_a is unknown, it is a normal random variable according to Bayesian view [13]. The agent has a prior multivariate normal distribution belief $N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ across the arms A, where $\boldsymbol{\mu}_n = [\mu_1, \cdots, \mu_a, \cdots, \mu_A]^T$ is the prior mean vector belief of size $|A|$ and $\boldsymbol{\Sigma}_n$ is the prior covariance matrix belief of size $|A| \times |A|$. The number of arms is $|A|$ and T is the transpose. After observing the reward r_a, the agent updates its prior belief distribution to get the posterior belief distribution $N(\boldsymbol{\mu}_{n+1}, \boldsymbol{\Sigma}_{n+1})$ which is a multivariate normal distribution according to Bayesian view [13]. The mean belief vector $\boldsymbol{\mu}_{n+1}$ and the covariance matrix belief $\boldsymbol{\Sigma}_{n+1}$ of the posterior belief distribution can be updated recursively as [13]:

$$\boldsymbol{\mu}_{n+1} = \boldsymbol{\mu}_n + \frac{r_a - \mu_{a,\,n}}{\sigma_a^2 + \sigma_{a,\,n}^2}\, \boldsymbol{\Sigma}_n\, \boldsymbol{e}_a$$
$$\boldsymbol{\Sigma}_{n+1} = \boldsymbol{\Sigma}_n - \frac{\boldsymbol{\Sigma}_n\, \boldsymbol{e}_a\, \boldsymbol{e}_a^T\, \boldsymbol{\Sigma}_n}{\sigma_a^2 + \sigma_{a,\,n}^2} \tag{1}$$

where $\boldsymbol{e}_a$ is a unit vector corresponding to arm a, $r_a \sim N(\mu_a, \sigma_a^2)$ is the observed reward, and $\mu_{a,\,n}$ and $\sigma_{a,\,n}^2$ are the mean and variance of the arm a of the belief distribution $N(\mu_{a,\,n}, \sigma_{a,\,n}^2)$ at time step n.

The goal of the agent is to maximize the total expected cumulative reward $R = \mathbb{E}[\sum_{n=1}^{N} r_n]$ by finding the optimal arm $a^* = \mathrm{argmax}_{a \in A}\, \mu_a$ and , where N is the total number of time steps that the agent played and r_n is the observed reward at time step n. Maximize the total expected reward is equivalent to minimizing the total regret (or total loss) $L = N\, \mu_{a^*} - \sum_{a=1}^{A} \mathbb{E}[n_a]\mu_a$, where n_a is the total number of pulling an arm a, μ_a is the true mean of the arm a and μ_{a^*} is the true mean of the optimal arm a^*.

A exploration/exploitation policy decides which arms to pull next to maximize the total expected cumulative reward. Here, we consider the Gittins index policy and the UCB_1-$Tuned$ policy to select the next arm to pull with the $CMAB$ problem.

A. Gittins Index Policy for CMAB

The approximated Gittins index (GI) policy [10] is used for $CMAB$ because it is based on the current beliefs about all the available arms. At each time step n, the GI policy calculates for each arm a, a corresponding index V_a^{GI}. The index of an arm a depends only on the mean belief $\mu_{a,\,n}$ and the variance belief $\sigma_{a,\,n}^2$ ($\sigma_{a,\,n}$ is the standard error belief) of the arm a. Given the normal belief distribution $N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, the GI policy selects the arm that has the maximum mean belief plus its GI index as [10]:

$$\begin{aligned} a_{GI}^* &= \underset{a \in A}{\mathrm{argmax}}\, (\mu_{a,\,n} + V_a^{GI}) \qquad (2)\\ &= \underset{a \in A}{\mathrm{argmax}} \left(\mu_{a,\,n} + \sigma_a \sqrt{-\log \gamma}\, b(-\frac{\sigma_{a,\,n}^2}{\sigma_a^2 \log \gamma}) \right), \end{aligned}$$

where σ_a^2 is the *known* variance of the reward distribution for selecting arm a, γ is the discount rate, and the function $b(s)$ is approximated as [10]:

$$b(s) = \begin{cases} \dfrac{s}{\sqrt{2}} & \text{for } s \le \frac{1}{7} \\ e^{-0.02645(\log s)^2 + 0.89106 \log s - 0.4873} & \text{for } \frac{1}{7} < s \le 10^2 \\ \sqrt{s}(2\log s - \log\log s - \log 16\pi)^{\frac{1}{2}} & \text{for } s > 100 \end{cases} \tag{3}$$

B. UCB1 Policy for CMAB

UCB_1 [8] is a very popular index policy. UCB_1 is a family of policies of which we only consider UCB_1-$Tuned$ [8]. Since UCB_1-$Tuned$ takes into account the variance beside the mean of the normal belief distribution, it performs well in practice. Like all UCB_1, UCB_1-$Tuned$ (or UCB_T for short) plays initially each arm a once. UCB_T

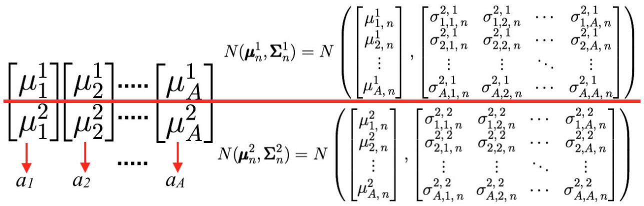
μ_1^1 μ_2^1 ..... μ_A^1
μ_1^2 μ_2^2 ..... μ_A^2
a_1 a_2 ..... a_A
$N(\boldsymbol{\mu}_n^1, \boldsymbol{\Sigma}_n^1) = N\left(\begin{bmatrix}\mu_{1,n}^1\\ \mu_{2,n}^1\\ \vdots\\ \mu_{A,n}^1\end{bmatrix}, \begin{bmatrix}\sigma_{1,1,n}^{2,1} & \sigma_{1,2,n}^{2,1} & \cdots & \sigma_{1,A,n}^{2,1}\\ \sigma_{2,1,n}^{2,1} & \sigma_{2,2,n}^{2,1} & \cdots & \sigma_{2,A,n}^{2,1}\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{A,1,n}^{2,1} & \sigma_{A,2,n}^{2,1} & \cdots & \sigma_{A,A,n}^{2,1}\end{bmatrix}\right)$
$N(\boldsymbol{\mu}_n^2, \boldsymbol{\Sigma}_n^2) = N\left(\begin{bmatrix}\mu_{1,n}^2\\ \mu_{2,n}^2\\ \vdots\\ \mu_{A,n}^2\end{bmatrix}, \begin{bmatrix}\sigma_{1,1,n}^{2,2} & \sigma_{1,2,n}^{2,2} & \cdots & \sigma_{1,A,n}^{2,2}\\ \sigma_{2,1,n}^{2,2} & \sigma_{2,2,n}^{2,2} & \cdots & \sigma_{2,A,n}^{2,2}\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{A,1,n}^{2,2} & \sigma_{A,2,n}^{2,2} & \cdots & \sigma_{A,A,n}^{2,2}\end{bmatrix}\right)$

Fig. 1: A correlated A-armed 2-objective bandit problem across arm. In each objective, the agent has a multivariate normal distribution belief.

computes for each arm a the corresponding index $V_a^{UCB_T}$, and adds it to the mean belief of the arm a. The index $V_a^{UCB_T}$ of an arm a is computed as:

$$V_a^{UCB_T} = \sqrt{\frac{\ln n}{n_a} \min\left(\frac{1}{4},\, (\sigma_{a,\,n} + \sqrt{\frac{2\ln n}{n_a}})\right)} \tag{4}$$

where n_a is the number of times arm a has been pulled, n is the current time step, and $\sigma_{a,\,n}$ is the standard error belief of the belief distribution $N(\mu_{a,\,n}, \sigma_{a,\,n}^2)$ for the arm a at time step n. UCB_T selects the arm that has the maximum mean belief $\mu_{a,\,n}$ plus its UCB_T index ($V_a^{UCB_T}$) as [8]:

$$a_{UCB_T}^* = \underset{a \in A}{\operatorname{argmax}}\, (\mu_{a,\,n} + V_a^{UCB_T}) \tag{5}$$

III. The correlated multi-objective multi-armed bandit problem ($CMOMAB$)

In this section, we combine the framework of correlated single-objective multi-armed bandit with the multi-objective optimization problem to introduce the correlated multi-objective multi-armed bandit ($CMOMAB$) problem in which the correlation is considered across *arms* only.

Let us consider the $CMOMAB$ across arms with $|A| \geq 2$ *dependent* arms and with D *independent* objectives per arm. At each time step n, an agent pulls one arm a and observes a reward vector $\boldsymbol{r}_a$. The reward $r_a^d \sim N(\mu_a^d, \sigma_a^2)$ in each objective $d \in D$ is drawn from a corresponding normal probability distribution, where μ_a^d is the *unknown* mean and σ_a^2 is the *known* variance of the reward distribution for the arm a in the objective d. We assume that the agent has a prior multivariate normal distribution belief across the arms. Since the objectives are independent and the agent has a prior multivariate normal distribution belief across the arms, each objective is a $CMAB$. For each objective d, the multivariate normal prior distribution belief is $N(\boldsymbol{\mu}_n^d, \boldsymbol{\Sigma}_n^d)$, where $\boldsymbol{\mu}_n^d$ is the mean vector of size of $|A|$ and $\boldsymbol{\Sigma}_n^d$ is the covariance matrix of size $|A| \times |A|$ of the multivariate normal prior distribution for the objective d at time step n. Figure 1 shows a correlated 2-objective A-armed bandit problem across arms. After observing the reward vector $\boldsymbol{r}_a$, the agent uses the observed reward r_a^d *in each objective* d to update its prior multivariate normal distribution belief using Equation 1.

When the objectives are conflicting, the mean μ_a^d of an arm a corresponding with objective d, can be better than the component $\mu_{a'}^d$ of another arm a' but worse if we compare the components for another objective d': $\mu_a^d > \mu_{a'}^d$ but $\mu_a^{d'} < \mu_{a'}^{d'}$ for objectives d and d', respectively. The agent has a set of optimal arms (Pareto front) A^* which are (partially) ordered either by the *Pareto dominance relation* (PDR) [4], or *linear scalarization dominance* (LSF) [7].

Pareto dominance relation $\boldsymbol{PDR}$ identifies the Pareto front A^* directly in the multi-objective space [4]. It uses the following relations between the mean vectors of two arms: 1) Arm a dominates a', $a \succ a'$, if there exists at least one objective d for which $\mu_a^d \succ \mu_{a'}^d$ and for all other objectives d' we have $\mu_a^{d'} \succeq \mu_{a'}^{d'}$. 2) Arm a is incomparable with a', $a \parallel a'$, if and only if there exists at least one objective d for which $\mu_a^d \succ \mu_{a'}^d$ and there exists another objective d' for which $\mu_a^{d'} \prec \mu_{a'}^{d'}$. 3) Arm a is not dominated by a', $a' \not\succ a$, means that either $a \succ a'$ or $a \parallel a'$. Using these relations, Pareto front $A^* \subset A$ is the set of arms that contains not dominated arms.

Linear scalarization dominance $\boldsymbol{LSF}$ converts the MOO problem into a single-objective one [7]. Given a predefined weight vector $\boldsymbol{w} = [w^1, \cdots, w^D]^T$ such that $\sum_{d=1}^D w^d = 1$, LSF assigns to each value of the mean vector $\boldsymbol{\mu}_a$ of an arm a a weight w^d and sums these weighted mean values as:

$$f(\boldsymbol{\mu}_a) = w^1\mu_a^1 + \cdots + w^D\mu_a^D \tag{6}$$

where $f(\boldsymbol{\mu}_a)$ is a LSF on the mean vector $\boldsymbol{\mu}_a$ of the arm a. After transforming the multi-objective problem to a single one, the LSF selects its optimal arm $a^* = \operatorname{argmax}_{1 \leq a \leq A} f(\boldsymbol{\mu}_a)$ that has the maximum LSF value.

To find all the optimal arms in the Pareto front, we need a set of scalarized functions $\boldsymbol{F} = \{f^1, \cdots, f^S\}$ to generate a variety of elements belonging to the Pareto front A^*. Each scalarized function $f^s \in \boldsymbol{F}$ has a corresponding predefined weight vector $\boldsymbol{w}^s \in \boldsymbol{W}$, where $\boldsymbol{W} = [\boldsymbol{w}^1, \cdots, \boldsymbol{w}^S]$ is a predefined total weight matrix. It is common practice in MOO to generate the matrix $\boldsymbol{W}$ uniformly random spread in the weighted space [14]. LSF is very popular due to its simplicity but it cannot identify all the rewards in a concave shape Pareto front A^* [14].

A. Measuring the Performance of $CMOMAB$

As in the $MOMAB$, the agent has to find both the Pareto front A^* (exploring the optimal arms) and play the optimal arms fairly (exploiting the optimal arms). There are two regret measures: Pareto regret measure (R_P) and unfairness regret measure (R_{SE}).

Pareto regret [2] measures the distance between a mean vector of an arm a that is pulled at time step n and the Pareto front A^*. The R_P is calculated by finding firstly the virtual distance dis^*. The virtual distance dis^* is defined as the minimum distance that will be added to the mean vector $\boldsymbol{\mu}_a$ of the pulled arm a at time step n in each objective to create a virtual mean vector $\boldsymbol{\mu}_v^* = \boldsymbol{\mu}_a + \boldsymbol{\varepsilon}^*$ that is incomparable with all the arms in Pareto set A^*, i.e. $\boldsymbol{\mu}_v^* || \boldsymbol{\mu}_{a^*} \forall_{a^* \in A^*}$. Where $\boldsymbol{\varepsilon}^*$ is a vector, $\boldsymbol{\varepsilon}^* = [dis^{*,1}, \cdots, dis^{*,D}]^T$. Then, the Pareto regret $R_P = dis(\boldsymbol{\mu}_a, \boldsymbol{\mu}_v^*) = dis(\boldsymbol{\varepsilon}^*, \boldsymbol{0})$ is the Euclidean distance between the mean vectors of the virtual arm $\boldsymbol{\mu}_v^*$ and the pulled arm $\boldsymbol{\mu}_a$ at time step n. Note that the regret of the Pareto front is 0 for optimal arms.

The unfairness regret measure [6] is the Shannon entropy R_{SE}. It is a measure of disorder on the frequency of pulling the optimal arms in the Pareto front A^*. The higher the entropy, the higher the disorder. The $R_{SE}(n)$ at time step n is:

$$R_{SE}(n) = -\frac{1}{\sum_{a^* \in A*} n_{a^*}} \sum_{a^* \in A^*} p_{a^*} \ln(p_{a^*}),$$

where $p_{a^*} = {}^{n_{a^*}}/_{\sum_{a \in A} n_a}$ is the frequency of pulling an optimal arm a^*, n_{a^*} is the number of times the optimal arm a^* has been pulled, $\sum_{a \in A} n_a$ is the number of times all arms $a = 1, \cdots, A$ have been pulled, and $\sum_{a^* \in A*} n_{a^*}$ is the number of times the optimal arms, $a^* = 1, \cdots, |A^*|$ have been pulled at time step n.

IV. Gittins Index Policy for CMOMAB

In a $CMOMAB$ across arms problem, each objective is a $CMAB$ problem with a given prior multivariate normal distribution belief. Gittins index policy computes which arm to pull next using $CMAB$. For each objective, for each arm, Gittins index GI computes the corresponding index and adds it to the prior mean belief of that arm. It performs linear scalarized function to convert the MOO into a single-objective one and selects the optimal arm that has the maximum scalarized value. The pseudo-code of the GI policy for the $CMOMAB$ is given in Algorithm 1.

```
1. Input: Action set $A$; Scalarized function set
 $\boldsymbol{F} = \{f^1, \cdots, f^S\}$; Number of objective $D$; Horizon of
 a run $N$; Discount rate $\gamma$; Reward distributions.

2. Initialize:  For objective $d = 1, \cdots, D$
                Prior mean $\boldsymbol{\mu}_0^d$; Prior covariance $\boldsymbol{\Sigma}_0^d$
               End

3. For time step  $n = 1, \cdots, N$
4.   For objective $d = 1, \cdots, D$
5.     For arm $a = 1, \cdots, |A|$
6.       Compute: $V_{a,\,n}^{GI,\,d}$
7.       $V\text{-}GI_{a,\,n}^d \leftarrow \mu_{a,\,n}^d + V_{a,\,n}^{GI,\,d}$
8.     End
9.   End
10.Select $f^s$ uniformly, randomly from $\boldsymbol{F}$
11.$f^s(\boldsymbol{V\text{-}GI}_{a,n}) = w^1 V\text{-}GI_{a,\,n}^1 + \cdots + w^D V\text{-}GI_{a,\,n}^D$
12.Select: the optimal arm $a^*$ that maximizes
             the scalarized function $f^s$
13.Observe: reward vector $\boldsymbol{r}_{a^*}$, $\boldsymbol{r}_{a^*} = [r_{a^*}^1, \cdots, r_{a^*}^D]^T$;
    Update: $n_{a^*} \leftarrow\ n_{a^*} + 1$
14.    For objective $d = 1, \cdots, D$
15.      $r_a = r_{a^*}^d$
16.      Update: $\boldsymbol{\mu}_n^d$ and $\boldsymbol{\Sigma}_n^d$
17.    End
18.    Compute: Pareto & unfairness regrets
19. End

20.  Output: Pareto & unfairness regrets
```

Algorithm: 1 (GI algorithm for $CMOMAB$).

In Algorithm 1, let A be the arm set, D be the number of objectives, N be the horizon of a run, γ be the discount rate, $\boldsymbol{F} = \{f^1, \cdots, f^S\}$ be the given scalarized function set, each scalarized function $f^s \in \boldsymbol{F}$ has a corresponding predefined weight vector $\boldsymbol{w} \in \boldsymbol{W}$ of size D, where $\boldsymbol{W}$ is the total predefined weight matrix of size $D \times S$, and the observed reward vector $\boldsymbol{r}_a$ of each arm $a \in A$ be drawn from a corresponding normal distribution $\boldsymbol{r}_a \sim N(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2)$, where $\boldsymbol{\mu}_a = [\mu_a^1, \cdots, \mu_a^D]^T$ is the unknown mean and $\boldsymbol{\sigma}_a^2 = [\sigma_a^{2,1}, \cdots, \sigma_a^{2,D}]^T$ is the known variance vectors of size D (Step 1).

As initialization step, for each objective $d \in D$, we have a prior multivariate normal distribution belief $N(\boldsymbol{\mu}_0^d, \boldsymbol{\Sigma}_0^d)$ since we have correlation across arms only, the objectives are independent with each other, see Section III. The $\boldsymbol{\mu}_0^d$ is the prior mean belief vector and $\boldsymbol{\Sigma}_0^d$ is the prior covariance belief matrix of the belief distribution of the objective d (Step 2).

At each time step n, for each objective d, the algorithm computes for each arm a the corresponding GI index $V_{a,\,n}^{GI,\,d} = \sigma_a^d \sqrt{-\log \gamma}\ b(-\frac{\sigma_{a,\,n}^{2,d}}{\sigma_a^{2,d} \log \gamma})$, where $\sigma_{a,\,n}^{2,d}$ is the variance of the arm a in the objective d of the normal belief distribution. The function $b(.)$ can be calculated using Equation 3 (Step 6). It adds the index $V_{a,\,n}^{GI,\,d}$ of an arm a in the objective d to the mean belief $\mu_{a,\,n}^d$ of that arm to compute the final value $V\text{-}GI_{a,\,n}^d$ (Step 7). The scalarized function $f^s \in \boldsymbol{F}$ that has a predefined weight vector $\boldsymbol{w}^s$ is selected uniformly at random (Step 10). Algorithm 1 performs linear scalarized function on the final value vector $\boldsymbol{V\text{-}GI}_{a,\,n} = [V\text{-}GI_{a,\,n}^1, \cdots, V\text{-}GI_{a,\,n}^D]^T$ of each arm a at time step n (Step 11). It selects the optimal arm $a^* = \text{argmax}_{a \in A} \boldsymbol{V\text{-}GI}_{a,\,n}$ that has the maximum scalarized value (Step 12), observes the corresponding reward vector $\boldsymbol{r}_{a^*}$ and updates the number of times n_{a^*} arm a^* is selected (Step 13). For each objective d, the parameters of the prior belief can be updated using Equations 1 (Steps 14-17). Algorithm 1 computes the Pareto and unfairness regrets, see Section III. This procedure is repeated until the end of playing N time steps.

V. UCB$_T$ Policy for CMOMAB

As GI policy, UCB_T finds the Pareto front A^* and plays fairly the optimal arms in the set A^* by considering each objective is a $CMAB$ problem with a given prior multivariate normal distribution belief. At each time step n, for each objective d, UCB_T computes for each arm the corresponding index $V_{a,\,n}^{UCB_T,d}$, see Equation 4, and adds it to the prior mean belief of that arm a in the objective d. For each arm a, it performs linear scalarized function on the prior mean belief vector $\boldsymbol{\mu}_{a,\,n} = [\mu_{a,\,n}^1, \cdots, \mu_{a,\,n}^D]^T$ plus the corresponding UCB_T index $\boldsymbol{V}_{a,\,n}^{UCB_T} = [V_{a,\,n}^{UCB_T,\,1}, \cdots, V_{a,\,n}^{UCB_T,\,D}]^T$ to convert the MOO problem into a single-objective one and selects the optimal arm a^* that has the maximum scalarized value. UCB_T policy observes the corresponding reward vector $\boldsymbol{r}_{a^*}$ and increases the number of pulling the arm a^*. For each objective d, the parameters of the prior belief can be updated using Equations 1.

Note that, Algorithm 1 can be used as a pseudo-code of the UCB_T policy for the $CMOMAB$ by computing the UCB_T index $\boldsymbol{V}_{a,\,n}^{UCB_T}$ instead of the GI index $\boldsymbol{V}_{a,\,n}^{GI}$ (Step 6 and 7) and applying linear scalarized function on the UCB_T

index plus the mean belief instead of the GI index plus the mean belief (Step 11) for each arm a at each time step n.

VI. Empirical Comparison

In this section, we compare Gittins index policy (Section IV) with UCB_T policy (Section V) on a test set of correlated multi-objective multi-armed bandit problem where we have correlation across arms only with number of arms $|A|$ and number of objectives D.

A. Performance Measures and Parameters Setting

The used performance measures are:

1) The average cumulative Pareto regret at each time step t.
2) The cumulative average unfairness regret at each time step t.

The above performance measures are the average of M runs.

Parameters setup:

The number of runs $M = 100$ and the horizon of each run $N = 10000$ as [2]. For *each run*, the reward vector $\boldsymbol{r}_a$ of each arm a is drawn from a corresponding normal distribution vector $N(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2)$, where $\boldsymbol{\mu}_a$ is the unknown mean vector and $\boldsymbol{\sigma}_a^2$ is the known variance vector of the reward of arm a. For simplicity, we assume that each arm a has an equal variance vector, i.e. $\boldsymbol{\sigma}_a^2 = \boldsymbol{\sigma}_\epsilon^2$ and each objective has the same variance, i.e. $\boldsymbol{\sigma}_\epsilon^2 = [\sigma_\epsilon^{2,1}, \cdots, \sigma_\epsilon^{2,D}]^T$. For each objective $d \in D$, $\sigma_\epsilon^{2,d} = \sigma_\epsilon^2 = 100$ as [11]. We assume that we have correlation across arms only. Since the objectives of an arm are independent, we consider each objective is a correlated single-objective MAB problem. We assume that we have a multivariate normal distributed belief $N(\boldsymbol{\mu}_0^d, \boldsymbol{\Sigma}_0^d)$ for each objective d, where $\boldsymbol{\mu}_0^d$ is the prior mean belief vector and $\boldsymbol{\Sigma}_0^d$ is the prior covariance matrix belief for the objective d.

We *follow* [11] in setting the prior belief parameters and the true mean of the reward distribution. The prior covariance matrix belief $\boldsymbol{\Sigma}_0^d$ for each objective d is set by the power-exponential rule:

$$\sigma^d_{a,a',\,0} = \sigma^2_\epsilon\; e^{-\lambda(a-a')^2} \qquad (7)$$

where $\sigma^{d,\,2}_{a,a',\,0}$ is the prior covariance value for the row a and the column a' in the matrix $\boldsymbol{\Sigma}_0^d$, σ^2_ϵ is the variance of the reward distribution and λ is a constant. The correlation parameter λ is set to 0.01. The prior mean belief $\mu^d_{a,\,0}$ for each arm a and objective d is generated from a normal distribution $N(0, \sigma^2_\epsilon)$. The true mean of the reward distribution μ^d_a for each arm a and objective d is generated from the prior belief parameters. The μ^d_a is taken from the prior normal distribution $N(\mu^d_{a,\,0}, \sigma^{2,\,d}_{a,a,\,0})$. Since the true mean is generated for each run, each run has a specific Pareto front A^* which is unknown. We find the Pareto front A^* for each run to compute the cumulative Pareto regret. Each arm is drawn initially once time to compute the unfairness regret.

We consider 11 scalarized function f^s that are uniformly randomly spread [14], see Section III. For instance, for number of objectives equals 2, the total weight matrix $\boldsymbol{W}$ can be set to $\boldsymbol{W} = [[1,0]^T, [0.9, 0.1]^T, \cdots, [0.1, 0.9]^T, [0,1]^T]$.

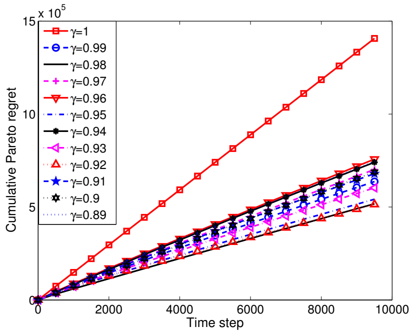

Fig. 2: The average cumulative Pareto regret performance measure of the GI using different values of the discount rate on a correlated 10-armed 2-objective bandit problem across arms.

B. Experimental Results

1) Parameters effect on GI and UCB_T policy: With number of objectives D equals 2, number of arms $|A|$ equals 10, we examine the effect of the parameters setting on the performance of GI and UCB_T policy.

The effect of the discount rate γ: Firstly, we examine the consequence of changing the discount rate γ on the performance measures of the GI policy, i.e. the discount rate γ is a tunable parameter. The best value of the discount rate γ^* is determined using cross-validation, i.e. γ^* is selected empirically from the set $\{0.89, 0.9, 0.91, \cdots, 0.98, 0.99, 0.999\}$. The best γ^* is the one that performs better than all the others discount rate according to the average cumulative Pareto regret performance measure, i.e. the average cumulative Pareto regret is decreased using the best discount rate γ^*.

Figure 2 gives the cumulative Pareto regret performance measure using different discount rate γ. The y-axis is the average cumulative performance measure. The x-axis is the time step. Figure 2 shows the cumulative Pareto regret of GI policy is decreased when the discount rate γ equals to 0.92, i.e. the best discount rate $\gamma^* = 0.92$.

Secondly, we compare GI policy using the best discount rate $\gamma^* = 0.92$ with the UCB_T policy. Figure 3 gives the average cumulative Pareto regret and unfairness regret performance measures. The y-axis is the average cumulative performance measure. The x-axis is the time step. Figure 3 shows that GI policy performs better than the UCB_T policy according to average cumulative Pareto and unfairness regret performance measures.

The effect of the variance σ^2_ϵ: To examine the result of changing the variance value σ^2_ϵ of the reward distributions, we compare GI policy with the UCB_T policy using different values of the variance $\sigma^2_\epsilon, \sigma^2_\epsilon = 0.001, 0.01, 0.1, 10,$ and 100. For each σ^2_ϵ, we simulate GI and UCB_T policies and compare GI with UCB_T using the performance measure at $10,000$ time step. We used the best discount rate $\gamma^* = 0.92$ for GI. The performance measure is the average cumulative Pareto regret. Figure 4 gives the average cumulative Pareto regret performance measure of GI and UCB_T using different value of σ^2_ϵ. The y-axis is the performance measure.

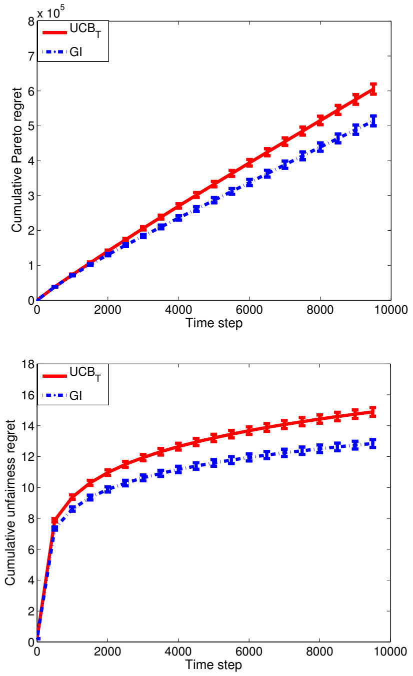

Fig. 3: Performance comparison of GI policy using $\gamma^* = 0.92$ and UCB_T policy on 2-objective, 10-armed bandit problem. Upper-figure shows the cumulative Pareto regret performance measure. Lower-figure shows the cumulative unfairness regret performance measure.

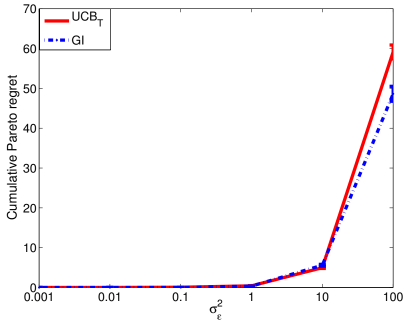

Fig. 4: The average cumulative Pareto regret performance measure for different values of the variance σ_ϵ^2 of the reward distributions on a correlated 10-armed 2-objective bandit problem across arms.

The x-axis is the variance σ_ϵ^2 of the reward distributions. Figure 4 shows: 1) for $\sigma_\epsilon^2 \leq 1$ (small value), GI policy performs as same as UCB_T policy, 2) for $1 < \sigma_\epsilon^2 \leq 10$, UCB_T policy performs slightly better than the GI policy, and 3) for $\sigma_\epsilon^2 > 10$, UCB_T policy performs slightly better than the GI policy.

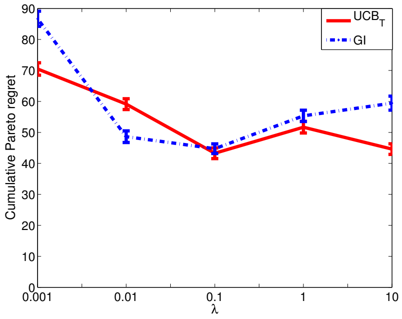

Fig. 5: The average cumulative Pareto regret performance measure for different values of the correlation parameter λ on a correlated 10-armed 2-objective bandit problem across arms.

The effect of the correlation parameter λ: To examine the effect of changing the correlation parameter value λ, we compare GI and UCB_T policy using different values of the λ, $\lambda = 0.001, 0.01, 0.1, 1,$ and 10. For each λ, we simulate GI and UCB_T policies and compare GI with UCB_T using the performance measure at $10,000$ time step. The variance σ_ϵ^2 of the reward distribution is set to 100. The best discount rate $\gamma^* = 0.92$ for GI. The performance measure is the average cumulative Pareto regret. Figure 5 gives the average cumulative Pareto regret performance measure of GI and UCB_T using different value of λ. The y-axis is the performance measure. The x-axis is the correlation parameter λ. Figure 5 shows the performance of the GI and UCB_T policies depend on the correlation parameter λ. The UCB_T policy performs better than GI when there is a weak correlation across arms (high values of the correlation parameter λ means weak correlation across arms since the negative sign in the power exponential rule, see Equation 7).

Discussion: The above experiments on 10-armed 2-objective bandit problems show that: 1) the performance measures (the cumulative average Pareto regret and the cumulative average unfairness regret) of the GI and UCB_T policies depend on the used parameters. As the variance of the reward distribution σ_ϵ^2 is increased, GI policy performs better than UCB_T policy. The intuition is that, the approximated index V_a^{GI} of the GI policy takes into account the value of the variance belief $\sigma_{a,\,n}^2$ of arm a, each value of the variance belief has a specific GI index V_a^{GI} calculation, see Equation 2. As the correlation parameter λ is increased (, i.e. when the arms are lightly correlated), UCB_T policy performs better than GI policy. The intuition is that, the approximated index $V_a^{UCB_T}$ does not consider the effect of the correlation parameter. We also see that, the performance of GI policy decreases as the discount rate γ is increased and this is because GI policy is a myopic policy which considers the current rewards only.

2) Adding Arms: We add extra arms to the 2-objective 10-armed bandit problem to examine the ***effect of increasing the number of arms*** $|A|$. We compare GI policy with the UCB_T policy using $|A| = 50$. We set the discount rate γ to 0.92 for GI policy, the correlation parameter λ to 0.01, and the variance of the reward distributions σ_ϵ^2 to 100. Figure 6 gives the cumulative Pareto and unfairness regret performance measures. The y-axis is the average cumulative

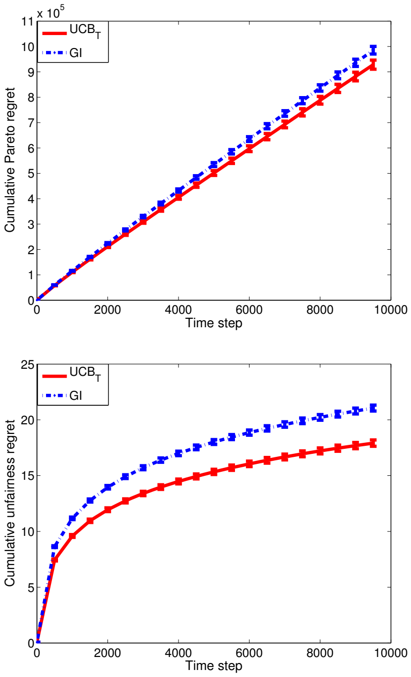

Fig. 6: Performance comparison on 2-objective, 50-armed bandit problem. Upper-figure shows the cumulative Pareto regret performance measure. Lower-figure shows the cumulative unfairness regret performance measure.

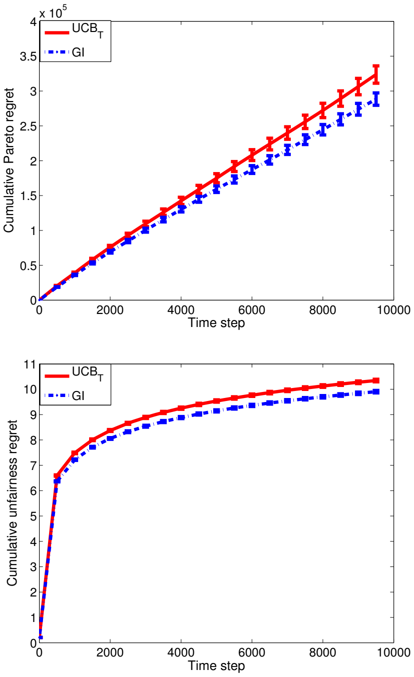

Fig. 7: Performance comparison on 5-objective, 10-armed bandit problem. Upper-figure shows the cumulative Pareto regret performance measure. Lower-figure shows the cumulative unfairness regret performance measure.

performance measure. The x-axis is the time step. Figure 6 shows UCB_T policy performs better than the GI policy according to the average cumulative Pareto and unfairness regret.

Discussion: When the number of arms is increased (i.e. $|A| = 20$) UCB_T policy outperforms GI policy, although we used the best parameters values for the GI policy (i.e. the discount rate $\gamma = 0.92$, see Figure 3, the variance of the reward distributions $\sigma_\epsilon^2 = 100$, see Figure 4, and the correlation parameter $\lambda = 0.01$, see Figure 5). The intuition is that the index of the UCB_T policy considers the number of times each arm a is selected.

3) Adding Objectives: We add extra objectives to the 2-objective 10-armed bandit problem to examine the *effect of increasing the number of objectives* D. We compare GI policy with the UCB_T policy using $D = 5$ and number of arms $|A| = 10$. We used the best parameters values for the GI policy (i.e. the discount rate $\gamma = 0.92$, see Figure 3, the variance of the reward distributions $\sigma_\epsilon^2 = 100$, see Figure 4, and the correlation parameter $\lambda = 0.01$, see Figure 5). Figure 7 gives the cumulative Pareto and unfairness regret performance measures. Figure 7 shows GI policy performs better than the UCB_T policy according to the average cumulative Pareto and unfairness regrets.

Discussion: The performance measures of GI policy is increased when the number of objectives is increased, $D = 5$, while the performance measures of UCB_T policy is decreased. The intuition is that the index of the UCB_T policy does not consider the number of objectives.

4) Adding Arms and Objectives: We add extra arms and objectives to the 2-objective 10-armed bandit problem to examine the ***effect of increasing the number of arms*** $|A|$ ***and objectives*** D. We compare GI policy with the UCB_T policy using $D = 5$ and number of arms $|A| = 50$. We used the best parameters values for the GI policy (i.e. the discount rate $\gamma = 0.92$, see Figure 3, the variance of the reward distributions $\sigma_\epsilon^2 = 100$, see Figure 4, and the correlation parameter $\lambda = 0.01$, see Figure 5). Figure 8 gives the cumulative Pareto and unfairness regret performance measures. Figure 8 shows GI policy performs better than the UCB_T policy according to the average cumulative Pareto and unfairness regrets. While according to the average cumulative unfairness regret, GI and UCB_T policies have the same performance.

Discussion: As the number of objectives and arms is increased GI policy outperforms UCB_T policy. This means that using the best parameters values for GI policy, increasing the number of objectives and arms will not change the performance of GI policy.

5) Conclusion of the above Experiments: The performance (the cumulative Pareto regret and the cumulative unfairness regret performance measure) of the UCB_T and GI policies depend on:

- The used parameters (i.e., the discount rate γ, the correlation parameter λ and the variance of the reward distributions σ_ϵ^2). GI policy does not need high discount rate γ, the performance of GI policy is

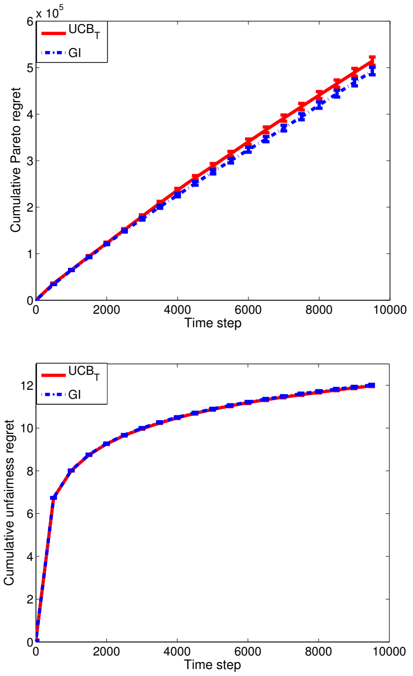

Fig. 8: Performance comparison on 5-objective, 50-armed bandit problem. Upper-figure shows the cumulative Pareto regret performance measure. Lower-figure shows the cumulative unfairness regret performance measure.

decreased for high values of the discount rate. As the correlation across arms or the variance of the reward distributions is increased, GI policy outperforms UCB_T policy.

- The number of arms $|A|$ and objectives D. For $|A| = 10$ and $D = 2$, GI policy outperforms UCB_T policy using the best parameters values for the GI policy. As the number of arms is only increased, UCB_T policy performs better than GI policy although we used the best parameters values for the GI policy. As the number of objectives is only increased GI policy outperforms UCB_T policy. As the number of arms and objectives is increased GI policy outperforms UCB_T policy.

VII. Conclusion

We introduced the correlated Gaussian multi-objective multi-armed bandit problem, where the objectives are independent and the arms are correlated. We extended UCB_1-$Tuned$ (or UCB_T) and Gittins index (or GI) policies to the Gaussian $CMOMAB$ across arms. We empirically compared UCB_T and GI policies on a test set of $CMOMAB$ problems. We concluded that: the performance measures of GI and UCB_T policies depend on the parameters values, and the number of arms and objectives.

References

[1] S. Q. Yahyaa, M. M. Drugan and B. Manderick, ”Thompson Sampling in the Adaptive Linear Scalarized Multi Objective Multi Armed Bandit”, in *Proc. International Conference on Agents and Artificial Intelligence (ICAART’15)*, Lisbon, Portugal, 2015.

[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] 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.

[4] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. Da Fonseca, ”Performance Assessment of Multiobjective Optimizers: An Analysis and Review”, *IEEE Trans. on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132. 2002.

[5] 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.

[6] S. Q. Yahyaa, M. M. Drugan and B. Manderick, ”Annealing-Pareto Multi-Objective Multi-Armed Bandits Algorithm”, in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL15)*, Orlando, Florida, USA, December 2014.

[7] G. Eichfelder, ”An adaptive scalarization method in multiobjective optimization”, *SIAM Optimization*, vol. 19, no. 4, pp. 1694-1718, Jan. 2009.

[8] 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.

[9] K. Jaffrès-Runser, M. R. Schurgot, C. Comaniciu and J.-M. Gorce, ”A Multiobjective Performance Evaluation Framework for Routing in Wireless Ad Hoc Networks”, in *Proc. International Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks (WiOpt’10)*, Avignon, France, 2010.

[10] S. E. Chick and N. Gans, ”Economic Analysis of Simulation Selection Problems”, *Management Science*, vol. 55, no. 3, pp. 421-437, 2009.

[11] I. O. Ryzhov, W. B. Powell and P. I. Frazier, ”The Knowledge-gradient policy for a general class of online learning problems”, *Operations Research*, vol. 60, no. 1, pp. 180-195, 2012.

[12] P. I. Frazier, W. B. Powell and S. Dayanik, ”The Knowledge Gradient Policy for Correlated Normal Rewards” *INFORMS*, vol. 21, no. 4, pp. 599-613, 2009.

[13] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.

[14] I. Das and J. E. Dennis, ”A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems”, *Structural Optimization*, vol. 14, no. 1, pp. 63-69, 1997.