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 \( r_a \) from the arm \( a \) with \( D \) dimensions (or objectives) as a feedback signal. The reward vector \( r_a \) is drawn from a corresponding stationary probability distribution vector, for example from a normal probability distribution \( N(\mu_a, \sigma_a^2) \), where \( \mu_a \) is the unknown true mean vector and \( \sigma_a^2 \) is the known variance vector parameters of the arm \( a \). The reward vector \( r_a \) 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 \( w \), the linear scalarized function weighs each value of the mean vector \( \mu_a \) 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 \( a^\star \) 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 (UCB1) 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-
objective Multi-Armed Bandit with normal correlated beliefs across arms (CMOMAB) [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 UCB1 policies on a test suite of CMOMAB problems and we conclude that the performance of GI and UCB1 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 UCB1 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 UCB1 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 CMAB. 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) UCB1 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, 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 \( N(\mu_a, \sigma^2_a) \) with unknown mean \( \mu_a \) and known variance \( \sigma^2_a \). Since the true mean \( \mu_a \) is unknown, it is a normal random variable according to Bayesian view [13]. The agent has a prior multivariate normal distribution belief \( N(\mu_n, \Sigma_n) \) across the arms \( A \), where \( \mu_n = [\mu_1, \ldots, \mu_a, \ldots, \mu_A]^T \) is the prior mean vector belief of size \( |A| \) and \( \Sigma_n \) 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 \( N(\mu_{n+1}, \Sigma_{n+1}) \) which is a multivariate normal distribution according to Bayesian view [13]. The mean belief vector \( \mu_{n+1} \) and the covariance matrix belief \( \Sigma_{n+1} \) of the posterior belief distribution can be updated recursively as [13]:

\[
\begin{align*}
\mu_{n+1} &= \mu_n + \frac{r_a - \mu_{a,n}}{\sigma^2_a + \sigma^2_{a,n}} \Sigma_n \epsilon_a \\
\Sigma_{n+1} &= \Sigma_n - \frac{\Sigma_n \epsilon_a \epsilon_a^T \Sigma_n}{\sigma^2_a + \sigma^2_{a,n}}
\end{align*}
\]

(1)

where \( \epsilon_a \) is a unit vector corresponding to arm \( a \), \( r_a \sim N(\mu_a, \sigma^2_a) \) is the observed reward, and \( \mu_{a,n} \) and \( \sigma^2_{a,n} \) are the mean and variance of the arm \( a \) of the belief distribution \( N(\mu_{a,n}, \sigma^2_{a,n}) \) 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 \( a^* = \arg \max_{a \in A} \mu_a \) and \( \sigma^2_a \), 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 \( a, \mu_a \) is the true mean of the arm \( a \) and \( \mu_{a^*} \) is the true mean of the optimal arm \( a^* \).

An 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 UCB1-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 \( V_{a,n} \). The index of an arm \( a \) depends only on the mean belief \( \mu_a \) and the variance belief \( \sigma^2_a \) (\( \sigma^2_a \) is the standard error belief) of the arm \( a \). Given the normal belief distribution \( N(\mu_a, \Sigma_n) \), the GI policy selects the arm that has the maximum mean belief plus its GI index as [10]:

\[
a^*_{GI} = \arg \max_{a \in A} \left( \mu_a, n + V_{a,n}^{GI} \right)
\]

(2)

where \( V_{a,n}^{GI} = \arg \max_{a \in A} \left( \mu_a + \sigma_a \sqrt{- \log \gamma b(-\frac{\sigma^2_a}{\gamma})} \right) \),

where \( \sigma^2_a \) is the known variance of the reward distribution for selecting arm \( a \), \( \gamma \) is the discount rate, and the function \( b(s) \) is approximated as [10]:

\[
b(s) = \begin{cases} 
\frac{8}{\sqrt{s}} & \text{for } s \leq \frac{1}{7} \\
0.02645 \log(s)^2 + 0.89106 \log(s) - 0.4873 & \text{for } \frac{1}{7} < s \leq 10^2 \\
\sqrt{2(\log s - \log \log s - \log 16\pi)^2} & \text{for } s > 100
\end{cases}
\]

(3)

B. UCB1 Policy for CMAB

UCB1 [8] is a very popular index policy. UCB1 is a family of policies of which we only consider UCB1-Tuned [8]. Since UCB1-Tuned takes into account the variance beside the mean of the normal belief distribution, it performs well in practice. Like all UCB1, UCB1-Tuned (or UCBT for short) plays initially each arm \( a \) once. UCBT
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

$$V_{a, UCB}^T = \ln n_a \min_n \left( \frac{1}{4}, \frac{n_a}{n_a} \right)$$

(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 $N(\mu_{a,n}, \sigma^2_{a,n})$ 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]:

$$a_{UCB}^* = \underset{a \in A}{\arg\max} (\mu_{a,n} + V_{a, UCB}^T)$$

(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 $r_a$. The reward $r_a^d \sim N(\mu_a^d, \sigma_a^d)$ in each objective $d \in D$ is drawn from a corresponding normal probability distribution, where $\mu_a^d$ is the unknown mean and $\sigma_a^d$ 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 $N(\mu_a^d, \Sigma_a^d)$, where $\mu_a^d$ is the mean vector of size $|A| \times |A|$ and $\Sigma_a^d$ 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 $r_a$, 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 $\mu_a^d$ 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 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 $\mu_a^d > \mu_{a'}^d$ and for all other objectives $d'$ we have $\mu_a^{d'} \geq \mu_{a'}^{d'}$. 2) Arm $a$ is incomparable with $a'$, $a \parallel a'$, if and only if there exists at least one objective $d$ for which $\mu_a^d > \mu_{a'}^d$ and there exists another objective $d'$ for which $\mu_a^{d'} < \mu_{a'}^{d'}$. 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 LSF converts the MOO problem into a single-objective one [7]. Given a predefined weight vector $w = [w_1, \ldots, w_D]^T$ such that $\sum_{d=1}^D w_d = 1$, LSF assigns to each value of the mean vector $\mu_a$ of an arm $a$ a weight $w_a$ and sums these weighted mean values as:

$$f(\mu_a) = w_1^{\mu_1} + \cdots + w_D^{\mu_D}$$

(6)

where $f(\mu_a)$ is a LSF on the mean vector $\mu_a$ of the arm $a$. After transforming the multi-objective problem to a single one, the LSF selects its optimal arm $a^* = \underset{a \in A}{\arg\max} f(\mu_a)$ that has the maximum LSF value.

To find all the optimal arms in the Pareto front, we need a set of scalarized functions $F = \{f^1, \ldots, f^S\}$ to generate a variety of elements belonging to the Pareto front $A^*$. Each scalarized function $f^s$ has a corresponding predefined weight vector $w^s \in W$, where $W = [w^1, \ldots, w^D]$ is a predefined total weight matrix. It is common practice in MOO to generate the matrix $W$ 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 $\text{dis}^*$. The virtual distance $\text{dis}^*$ is defined as the minimum distance that will be added to the mean vector $\mu_a$ of the pulled arm $a$ at time step $n$ in each objective to create a virtual mean vector $\mu_a^* = \mu_a + \epsilon^*$ that is incomparable with all the arms in Pareto set $A^*$, i.e. $\mu_a^* \parallel \mu_{a'}$ for $a' \in A^*$. Where $\epsilon^*$ is a vector, $\epsilon^* = [\text{dis}^{*,1}, \ldots, \text{dis}^{*,D}]^T$. Then, the Pareto regret $R_P = \text{dis}(\mu_a, \mu_a^*) = \text{dis}(\epsilon^*, 0)$ is the Euclidean distance between the mean vectors of the virtual arm $\mu_a^*$ and the pulled
arm \( \mu_a \) at time step \( n \). Note that the regret of the Pareto front is 0 for optimal arms.

The unfairness regret measure \cite{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 \( a = 1, \ldots, A \) have been pulled, and \( \sum_{a' \in A} n_{a'} \) is the number of times the optimal arms, \( a^* = 1, \ldots, |A^*| \), have been pulled at 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 \( F = \{f_1, \ldots, f_S\} \); Number of objective \( D \); Horizon of a run \( N \); Discount rate \( \gamma \); Reward distributions.

2. **Initialize**: For objective \( d = 1, \ldots, D \)
   - Prior mean \( \mu_0^d \)
   - Prior covariance \( \Sigma_0^d \)
3. \[ \text{For time step } n = 1, \ldots, N \]
4. \[ \text{For objective } d = 1, \ldots, D \]
5. \[ \text{For arm } a = 1, \ldots, |A| \]
6. \[ \text{Compute: } V_{GI}^d, a \]
7. \[ V_{-GI}^d, n \leftarrow \mu_{a, n} + V_{GI}^d, a \]
8. \[ \text{End} \]
9. \[ \text{End} \]
10. Select \( f^* \) uniformly, randomly from \( F \)
11. \[ f^*(V_{-GI}^d, n) = w^1 V_{-GI}^{d_1, n} + \cdots + w^D V_{-GI}^{d_D, n} \]
12. Select: the optimal arm \( a^* \) that maximizes the scalarized function \( f^* \)
13. Observe: reward vector \( r_{a^*}, r_{a^*} = [r_{a^*, 1}, \ldots, r_{a^*, D}]^T \)
14. \[ \text{For objective } d = 1, \ldots, D \]
15. \[ r_{a} = r_{a^*} \]
16. Update: \( \mu_n^d \) and \( \Sigma_n^d \)
17. \[ \text{End} \]
18. Compute: Pareto & unfairness regrets
19. \[ \text{End} \]
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, \( \gamma \) be the discount rate, \( F = \{f_1, \ldots, f_S\} \) be the given scalarized function set, each scalarized function \( f^* \in F \) has a corresponding predefined weight vector \( w \in W \) of size \( D \), where \( W \) is the total predefined weight matrix of size \( D \times S \), and the observed reward vector \( r_a \) of each arm \( a \in A \) be drawn from a corresponding normal distribution \( r_a \sim N(\mu_a, \Sigma_a) \), where \( \mu_a = [\mu_{a, 1}^d, \ldots, \mu_{a, D}^d]^T \) is the unknown mean and \( \sigma_a^2 = [\sigma_{a, 1}^2, \ldots, \sigma_{a, D}^2]^T \) 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 \( N(\mu_0^d, \Sigma_0^d) \) since we have correlation across arms only, the objectives are independent with each other, see Section III. The \( \mu_0^d \) is the prior mean belief vector and \( \Sigma_0^d \) 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 \( V_{GI}^d, a \) as defined in Equation 3 (Step 6). It adds the index \( V_{GI}^d, a \) to the objective \( d \) to the mean belief \( \mu_{a, n}^d \) of that arm to compute the final value \( V_{-GI}^d, n \) (Step 7). The scalarized function \( f^* \in F \) that has a predefined weight vector \( w^* \) is selected uniformly at random (Step 10). Algorithm 1 performs linear scalarized function on the final value vector \( V_{-GI}^d, n = [V_{-GI}^{d_1, n}, \ldots, V_{-GI}^{d_D, n}]^T \) of each arm \( a \) at time step \( n \) (Step 11). It selects the optimal arm \( a^* = \arg\max_{a \in A} V_{-GI}^d, n = \max_{a \in A} V_{-GI}^d, n \) that has the maximum scalarized value (Step 12), observes the corresponding reward vector \( r_{a^*} \) 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\(_B\) POLICY FOR CMOMAB

As GI policy, UCB\(_B\) 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\(_B\) computes for each arm the corresponding index \( V_{UCB}^d, a \), 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 \( \mu_{a, n} = [\mu_{a, 1}^d, \ldots, \mu_{a, D}^d]^T \) plus the corresponding UCB\(_B\) index \( V_{UCB}^d, a = [V_{UCB}^{d_1, a}, \ldots, V_{UCB}^{d_D, a}]^T \) to convert the MOO problem into a single-objective one and selects the optimal arm \( a^* \) that has the maximum scalarized value. UCB\(_B\) policy observes the corresponding reward vector \( r_{a^*} \) 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\(_B\) policy for the CMOMAB by computing the UCB\(_B\) index \( V_{UCB}^d, a \) (Step 6 and 7) and applying linear scalarized function on the UCB\(_B\)
In this section, we compare the 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 $r_a$ of each arm $a$ is drawn from a corresponding normal distribution vector $N(\mu_a, \sigma_a)$, where $\mu_a$ is the unknown mean vector and $\sigma_a^2$ 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. $\sigma_a^2 = \sigma^2_a$ and each objective has the same variance, i.e. $\sigma_d^2 = [\sigma_1^2, \ldots, \sigma_D^2]^T$. For each objective $d \in D$, $\sigma_d^2 = \sigma_d = 100$ 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 $N(\mu_0^d, \Sigma_0^d)$ for each objective $d$, where $\mu_0^d$ is the prior mean belief vector and $\Sigma_0^d$ 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 $\Sigma_0^d$ for each objective $d$ is set by the power-exponential rule:

$$\sigma_{0,a',a}^2 = \sigma_a^2 e^{-\lambda(a-a')^2}$$  \hspace{1cm} (7)

where $\sigma_{0,a',a}^2$ is the prior covariance value for the row $a$ and the column $a'$ in the matrix $\Sigma_0^d$, $\sigma_a^2$ is the variance of the reward distribution and $\lambda$ is a constant. The correlation parameter $\lambda$ is set to 0.01. The prior mean belief $\mu_{0,a}^d$ for each arm $a$ and objective $d$ is generated from a normal distribution $N(0, \sigma_a^2)$. The true mean of the reward distribution $\mu_a^d$ is taken from the corresponding normal distribution $N(\mu_a^d, \sigma_a^2)$. 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 $f^*$ that are uniformly randomly spread [14], see Section III. For instance, for number of objectives equals 2, the total weight matrix $W$ can be set to $W = [[1, 0]^T, [0, 9, 0.1]^T, \ldots, [0, 1, 0.9]^T, [0, 1]^T]$.

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 $r_a$ of each arm $a$ is drawn from a corresponding normal distribution vector $N(\mu_a, \sigma_a^2)$, where $\mu_a$ is the unknown mean vector and $\sigma_a^2$ 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. $\sigma_a^2 = \sigma^2_a$ and each objective has the same variance, i.e. $\sigma_d^2 = [\sigma_1^2, \ldots, \sigma_D^2]^T$. For each objective $d \in D$, $\sigma_d^2 = \sigma_d = 100$ 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 $N(\mu_0^d, \Sigma_0^d)$ for each objective $d$, where $\mu_0^d$ is the prior mean belief vector and $\Sigma_0^d$ 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 $\Sigma_0^d$ for each objective $d$ is set by the power-exponential rule:

$$\sigma_{0,a',a}^2 = \sigma_a^2 e^{-\lambda(a-a')^2}$$  \hspace{1cm} (7)

where $\sigma_{0,a',a}^2$ is the prior covariance value for the row $a$ and the column $a'$ in the matrix $\Sigma_0^d$, $\sigma_a^2$ is the variance of the reward distribution and $\lambda$ is a constant. The correlation parameter $\lambda$ is set to 0.01. The prior mean belief $\mu_{0,a}^d$ for each arm $a$ and objective $d$ is generated from a normal distribution $N(0, \sigma_a^2)$. The true mean of the reward distribution $\mu_a^d$ is taken from the corresponding normal distribution $N(\mu_a^d, \sigma_a^2)$. 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 $f^*$ that are uniformly randomly spread [14], see Section III. For instance, for number of objectives equals 2, the total weight matrix $W$ can be set to $W = [[1, 0]^T, [0, 9, 0.1]^T, \ldots, [0, 1, 0.9]^T, [0, 1]^T]$.

Fig. 2: The average cumulative Pareto regret performance of the G1 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 G1 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 G1 and UCB$_T$ policy.

The effect of the discount rate $\gamma$: Firstly, we examine the consequence of changing the discount rate $\gamma$ on the performance measures of the G1 policy, i.e. the discount rate $\gamma$ is a tunable parameter. The best value of the discount rate $\gamma^*$ is determined using cross-validation, i.e. $\gamma^*$ is selected empirically from the set $\{0.89, 0.9, 0.91, \ldots, 0.98, 0.99, 0.999\}$. The best $\gamma^*$ 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 $\gamma^*$.

Figure 2 gives the cumulative Pareto regret performance measure using different discount rate $\gamma$. The $x$-axis is the average cumulative Pareto regret measure. The $x$-axis is the time step. Figure 2 shows the cumulative Pareto regret of G1 policy is decreased when the discount rate $\gamma$ equals to 0.92, i.e. the best discount rate $\gamma^* = 0.92$.

Secondly, we compare G1 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 G1 policy performs better than the UCB$_T$ policy according to average cumulative Pareto and unfairness regret performance measures.

The effect of the variance $\sigma_2^d$: To examine the result of changing the variance value $\sigma_2^d$ of the reward distributions, we compare GI policy with the UCB$_T$ policy using different values of the variance $\sigma_2^d = 0.001, 0.01, 0.1, 1.0, 100$. For each $\sigma_2^d$, we simulate G1 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 $\sigma_2^d$. The $y$-axis is the performance measure.
The $x$-axis is the variance $\sigma_i^2$ of the reward distributions. Figure 4 shows: 1) for $\sigma_i^2 \leq 1$ (small value), GI policy performs as same as $UCB_T$ policy, 2) for $1 < \sigma_i^2 \leq 10$, $UCB_T$ policy performs slightly better than the GI policy, and 3) for $\sigma_i^2 > 10$, $UCB_T$ policy performs slightly better than the GI policy.

The effect of the correlation parameter $\lambda$: To examine the effect of changing the correlation parameter value $\lambda$, we compare GI and $UCB_T$ policy using different values of the $\lambda$, $\lambda = 0.001, 0.01, 0.1, 1, \text{ and } 10$. For each $\lambda$, we simulate GI and $UCB_T$ policies and compare GI with $UCB_T$ using the performance measure at 10,000 time step. The variance $\sigma_i^2$ 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 $\lambda$. The $y$-axis is the performance measure. The $x$-axis is the correlation parameter $\lambda$. Figure 5 shows the performance of the GI and $UCB_T$ policies depend on the correlation parameter $\lambda$. The $UCB_T$ policy performs better than GI when there is a weak correlation across arms (high values of the correlation parameter $\lambda$ 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 $\sigma_i^2$ is increased, GI policy performs better than $UCB_T$ policy. The intuition is that, the approximated index $V_{a, n}^{GI}$ 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 $\lambda$ 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 $V_{a, n}^{UCB_T}$ does not consider the effect of the correlation parameter. We also see that, the performance of GI policy decreases as the discount rate $\gamma$ 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 $\gamma$ to 0.92 for GI policy, the correlation parameter $\lambda$ to 0.01, and the variance of the reward distributions $\sigma_i^2$ to 100. Figure 6 gives the cumulative Pareto unfairness regret performance measures. The $y$-axis is the average cumulative
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 $\sigma_q^2 = 100$, 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 $\sigma_q^2 = 100$, 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 $\sigma_q^2 = 100$, 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 $\gamma$, the correlation parameter $\lambda$ and the variance of the reward distributions $\sigma_q^2$). $GI$ policy does not need high discount rate $\gamma$, the performance of $GI$ policy is
Policies on a test set of 10,000 UCB policy outperforms another policy. As the number of arms increased, UCB policy outperforms GI policy although we used the best parameters values for the GI policy. As the number of objectives is only increased, UCB policy outperforms GI policy. The number of arms and objectives is increased, UCB policy outperforms GI policy. 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 policy.

- The number of arms $|A|$ and objectives $D$. For $|A| = 10$ and $D = 2$, GI policy outperforms UCB policy using the best parameters values for the GI policy. As the number of arms is only increased, UCB 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 policy. As the number of arms and objectives is increased GI policy outperforms UCB 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 $UCB_1$, $T_{\text{ucb}}$, (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


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.