Designing multi-objective multi-armed bandits algorithms: a study

Madalina M. Drugan and Ann Nowe

Abstract—**We propose an algorithmic framework for multi-objective multi-armed bandits with multiple rewards. Different partial order relationships from multi-objective optimization can be considered for a set of reward vectors, such as scalarization functions and Pareto search. A scalarization function transforms the multi-objective environment into a single objective environment and are a popular choice in multi-objective reinforcement learning. Scalarization techniques can be straightforwardly implemented into the current multi-armed bandit framework, but the efficiency of these algorithms depends very much on their type, linear or non-linear (e.g. Chebyshev), and their parameters. Using Pareto dominance order relationship allows to explore the multi-objective environment directly, however this can result in large sets of Pareto optimal solutions. In this paper we propose and evaluate the performance of multi-objective MABs using three regret metric criteria. The standard UCB1 is extended to scalarized multi-objective UCB1 and we propose a Pareto UCB1 algorithm. Both algorithms are proven to have a logarithmic upper bound for their expected regret. We also introduce a variant of the scalarized multi-objective UCB1 that removes online inefficient scalarizations in order to improve the algorithm's efficiency. These algorithms are experimentally compared on multi-objective Bernoulli distributions, Pareto UCB1 being the algorithm with the best empirical performance.**

I. Introduction

Many real-world problems are inherently multi-objective environments with conflicting objectives. Multi-armed bandits is a machine learning paradigm used to study and analyse resource allocation in stochastic and noisy environments. We consider the classical definition for the multi-armed bandits where only one arm is played at a time and each arm is associated with a fixed equal range stochastic reward vectors. When arm i is played at time steps t_1,t_2,..., the corresponding reward vectors $\mathbf{X}_{i,t_1}$, $\mathbf{X}_{i,t_2}, \ldots$ are independently and identically distributed according to an unknown law with unknown expectation vector. The independence holds between the arms.

In this paper, we design a novel multi-armed bandit framework that considers multi-objective (or multi-dimensional) rewards and that imports techniques from multi-objective optimization into the multi-armed bandits algorithms. We call this framework *multi-objective multi-armed bandits* (MO-MABs). Some of these techniques were also imported in other related learning paradigms: multi-objective Markov Decision Processes (MDPs) [5], [10], and multi-objective reinforcement learning [8], [9].

Multi-objective MABs lead to important differences compared to the standard MABs. There could be several arms considered to be the best according to their reward vectors. In Section II, we consider two order relationships. *Scalarization functions* [3], like linear and Chebyshev functions, transform the reward vectors into scalar rewards. *Pareto partial order* [11] allows to maximize the reward vectors directly in the multi-objective reward space. By means of an example, we compare these approaches on a non-convex distribution of the best arms. We highlight the difficulty of the linear scalarization functions for optimizing non-convex shapes. Linear scalarization is currently a popular choice in designing multi-objective reinforcement learning algorithms, like the multi-objective MDPs from [5] but these algorithms have the same limitations as scalarized MO-MABs in exploring non-convex shapes. We consider a variety of scalarisation functions, and compare their performance to our Pareto MAB algorithm.

In Section III, we propose three regrets metrics for multi-objective MAB algorithms. A straightforward regret for scalarized multi-objective MAB transforms the regret vector into a value using scalarization functions. This regret measure, however, does not give any information on the dynamics of the multi-objective MAB algorithm as a whole. Multi-objective MAB algorithms should pull *all* optimal arms frequently, therefore we also introduce an *unfairness* indicator to measure lack of variance in pulling the optimal arms. This measure is similar to the risk analysis metric from economy and it is especially useful in pointing out the weakness of scalarized multi-objective MAB in discovering and choosing a variety of optimal arms. An adequate regret definition for the Pareto MAB algorithm measures the distance between the *set* of optimal reward vectors and a suboptimal reward vector. Our measure is inspired by ϵ-dominance as proposed in [4].

Section IV introduces a Pareto UCB1 algorithm that uses the Pareto dominance relationship to store and identify all the optimal arms in each iteration. The regret bound for the multi-objective UCB1 algorithm using Pareto regrets is logarithmic in the number of plays for a suboptimal arm, the size of the reward vectors and the number of optimal arms. Section V proposes two scalarization multi-objective variants of the UCB1 classical multi-armed bandits [1], [2]: i) a straightforward generalization of the single-objective UCB1 by arbitrarily alternate different scalarization-based UCB1s, and ii) an improved UCB1 that removes scalarization functions considered not to be useful.

In Section VI, we compare runs of the proposed multi-objective UCB1 algorithms on multi-objective Bernoulli reward distributions, the standard stochastic environment used to test multi-armed bandits. Section VII concludes the paper.

Madalina M. Drugan and Ann Nowe are with the Computational Modeling lab, Artificial Intelligence group of Computer Science Department, Vrije Universiteit Brussels, Pleinlaan 2, B-1050 Brussels, Belgium (email: {mdrugan, anowe}@vub.ac.be).

TABLE I. Relations between reward vectors.

relationship	notation	relationships
μ dominates ν	$\nu \prec \mu$	$\exists j, \nu^j < \mu^j$ and $\forall o, j \neq o, \nu^o \leq \mu^o$
μ weakly domin ν	$\nu \preceq \mu$	$\forall j, \nu^j \leq \mu^j$
μ is incomp with ν	$\nu \| \mu$	$\nu \nsucc \mu$ and $\mu \nsucc \nu$
μ is non-domin by ν	$\nu \nsucc \mu$	$\nu \prec \mu$ or $\nu \| \mu$

II. Order relationships for reward vectors

Let's consider a K-armed bandit, $K \geq 2$. In the multi-objective setting, the expected reward of each bandit i is multi-dimensional, $\mu_i = (\mu_i^1, \ldots, \mu_i^D)$, where D is a fixed number of dimensions, or objectives. We consider the general case where a reward vector can be better than another reward vector in one dimension, and worse than another reward vector in another dimension. This means that the objectives might be conflicting.

A. The Pareto partial order

We consider that the reward vectors are ordered using the partial order on multi-objective spaces [11]. The following order relationship between two reward vectors, μ and ν, are considered. A reward vector μ is considered better than, or *dominating*, another reward vector ν, $\nu \prec \mu$, if and only if there exists at least one dimension j for which $\nu^j < \mu^j$, and for all other dimensions o we have $\nu^o \leq \mu^o$. We say that μ is weakly-dominating ν, $\nu \preceq \mu$, if and only if for all dimensions j, we have $\nu^j \leq \mu^j$. A reward vector μ is considered *incomparable* with another reward vector ν, $\nu \| \mu$, if and only if there exists at least one dimension j for which $\nu^j < \mu^j$, and there exists another dimension o, for which $\nu^o > \mu^o$. We say that μ is *non-dominated* by ν, $\nu \nsucc \mu$, if and only if there exists at least one dimension j for which $\nu^j < \mu^j$. These Pareto relationships are summarized in Table I.

Let the *Pareto optimal reward set* $\mathcal{O}^*$ be the set of reward vectors that are non-dominated by any of the reward vectors. Let the *Pareto optimal set of arms* $\mathcal{A}^*$ be the set of arms whose reward vectors belong to $\mathcal{O}^*$. Then:

$$\forall \mu_\ell^* \in \mathcal{O}^*, \quad \text{and} \quad \forall \mu_o, \quad \text{we have} \quad \mu_\ell^* \not\prec \mu_o$$

All the Pareto optimal rewards are incomparable:

$$\forall \mu_\ell^*, \, \mu_o^* \in \mathcal{O}^*, \quad \text{we have} \quad \mu_\ell^* \| \mu_o^*$$

We further assume that it is impossible, from the application point of view, to determine a-priori which arm in $\mathcal{A}^*$ is better than another arm from $\mathcal{A}^*$. Therefore, the reward vectors in the Pareto optimal reward set $\mathcal{O}^*$ are considered equally important.

B. Order relationships for scalarization functions

A conventional way to transform a multi-objective environment into a single-objective environment is to use *scalarization functions*. However, since single-objective environments, in general, results in a single optimum, and we need a set of scalarization functions to generate a variety of elements belonging to the Pareto optimal set. We consider two types of scalarization functions that weight the values of the reward vector, but with different properties because of their (non)-linearity. *We consider each set of weights to generate a scalarization function.*

The linear scalarization is the most popular scalarization function due to its simplicity. It weighs each value of the reward vector and the result is the sum of these weighted values. The linear scalarized reward is

$$f(\mu_i) = \omega^1 \cdot \mu_i^1 + \ldots \omega^D \cdot \mu_i^D, \quad \forall i$$

where $(w^1, \ldots, w^D)$ is a set of predefined weights and $\sum_{j=1}^{D} \omega^j = 1$. A known problem with linear scalarization is its incapacity to potentially find all the points in a non-convex Pareto set.

The Chebyshev scalarization has the advantage that in certain conditions it can find all the points in a non-convex Pareto set. The Chebyshev transformation was originally designed for minimization problems, but we adapt it for the maximization goal of multi-armed bandits. The *Chebyshev scalarization reward* is

$$f(\mu_i) = \min_{1 \leq j \leq D} \omega^j \cdot (\mu_i^j - z^j), \quad \forall i$$

where $\mathbf{z} = (z^1, \ldots, z^D)$ is a reference point that is dominated by all the optimal reward vectors μ_i^*. For each objective j, this reference point is the minimum of the current optimal rewards minus a small positive value, $\epsilon^j > 0$. Then:

$$z^j = \min_{1 \leq i \leq D} \mu_i^j - \epsilon^j, \quad \forall j \tag{1}$$

[6] shows that all the points in a Pareto set can be found by moving the reference point $\mathbf{z}$.

The optimum reward value μ^* is the reward for which the function f, linear or Chebyshev, attains its maximum value

$$f(\mu^*) =^{def} \max_{1 \leq i \leq K} f(\mu_i) \tag{2}$$

We denote the Pareto optimal set of arms identifiable by the linear scalarization with $\mathcal{A}_L^*$ and the Chebyshev scalarization with $\mathcal{A}_C^*$. The corresponding set of Pareto optimal reward set is $\mathcal{O}_L^*$ for linear scalarization and $\mathcal{O}_C^*$ for Chebyshev scalarization. *The set of optimal arms might be different for the different scalarizations.*

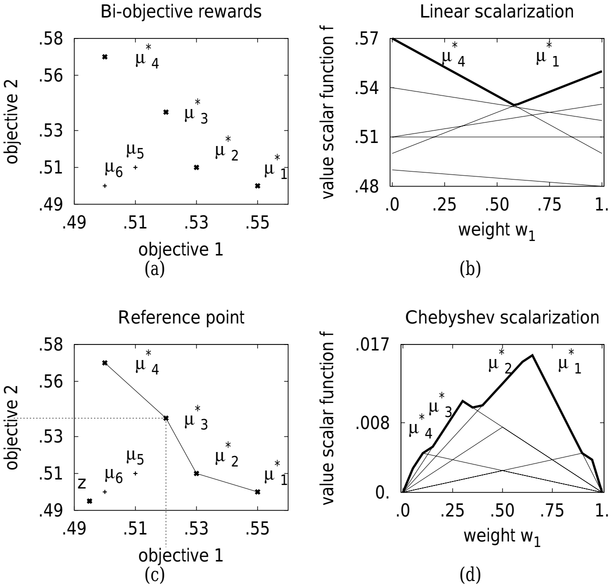
C. Comparing the partially ordered reward vector sets

To highlight differences in these scalarizations, we consider in Figure 1 a set of bi-objective rewards with a non-convex Pareto optimal reward set $\mathcal{O}^*$. We show by means of an example that $\mathcal{O}_L^* \subseteq \mathcal{O}^*$.

Example 1: Consider six bi-dimensional reward vectors from Figure 1 a). Let there be four optimal reward vectors, $\mu_1^* = (0.55, 0.5)$, $\mu_2^* = (0.53, 0.51)$, $\mu_3^* = (0.52, 0.54)$ and $\mu_4^* = (0.5, 0.57)$ in $\mathcal{O}^*$ and two suboptimal reward vectors, $\mu_5 = (0.51, 0.51)$ and $\mu_6 = (0.5, 0.5)$. Note that the suboptimal reward μ_5 is non-dominated by two optimal reward vectors from $\mathcal{O}^*$, μ_1^* and μ_4^*, but μ_5 is dominated by μ_2^* and μ_3^*. μ_6 is dominated by all the other reward vectors.

Figures 1 b) and d) show the values of the linear and Chebyshev scalarization functions for different weight sets

Fig. 1. a) Six reward vectors, where four are optimal: μ^*_1, μ^*_2, μ^*_3 and μ^*_4. Computing the reward values for b) the linear scalarization, and d) the Chebyshev scalarization. c) The reference point $\mathbf{z}$.



$(w_1, 1 - w_1)$. In Figure 1 b), the optimum reward vectors for any set of weights are μ^*_1 and μ^*_4 but never μ^*_2 or μ^*_3. $\mathcal{O}^*_L$ therefore does not include μ^*_2 and μ^*_3 from the non-convex Pareto optimal reward set $\mathcal{O}^*$. In Figure 1 c) and d), we consider Chebyshev scalarization with the reference point $\mathbf{z} = (0.495, 0.495)$ as it allows to discover all the optimal rewards from $\mathcal{O}^*$, and thus $\mathcal{O}^*_C = \mathcal{O}^*$.

With the Pareto partial order, all the arms in $\mathcal{A}^*$ are identifiable because of the dominance relationships described in Table I. For each of the non-selected arms, μ_5 and μ_6, there exists at least one arm in $\mathcal{A}^*$ that dominates that arm. □

Scalarized multi-objective Markov Decision Processes. We now argue that the above observations on the inefficiency of linear scalarization function are valid for more general types of reinforcement learning algorithms, like the Markov Decision Processes (MDPs). Multi-armed bandits is considered a special case of reinforcement learning with only a single state. An MDP is defined as a dynamical and uncertain environment where the states and the actions of an agent are associated with density functions and reward probabilities that are explored/exploited, for example, using a reinforcement learning algorithm. Lizotte et al [5] propose a linear scalarized multi-objective MDPs with a bi-dimensional reward vector that iteratively discovers the Pareto front using different sets of weights. For a convex Pareto front, they show that the entire Pareto shape can be learned with a set of linear scalarization functions whose weights can be analytically determined.

However, they do not discuss the behaviour of their algorithm for non-convex Pareto fronts. Let's reconsider the previous example. As shown in Figure 1 b), there is *no* set of weights for the linear scalarization that can express the middle optimal rewards μ^*_2 and μ^*_3. We conclude that the learning of a non-convex Pareto front is impossible with linear scalarization. The Chebyshev scalarization functions can potentially expresses the full Pareto front, however it should be recognized that this depends on the chosen reference point, which also needs to be optimized.

A multi-objective MDP that uses Pareto relationships is presented in Wiering and De Jong [10]. It can learn any discrete Pareto front but it has a rather complicated and computational expensive mechanism to update and propagate the probabilities of the states and falls out of the scope of the paper. Moreover, it should be noted that it is limited to deterministic settings, and therefore not applicable to the MO-MABs we consider here.

III. Performance of multi-objective MAB algorithms

The goal of multi-objective multi-armed bandits is to simultaneously minimize the regret in all objectives by fairly playing all the arms in the Pareto optimal arm set using a policy π. In this section, we propose three metrics to measure the performance of multi-objective MABs. The *Pareto regret metric* measures the distance between a reward vector and the Pareto optimal reward set, whereas the *scalarized regret metric* measures the distance between the maximum value of a scalarized function and the scalarized value of an arm. The *unfairness metric* is related to the variance in pulling all the optimal arms.

A. A Pareto regret metric

Intuitively, a regret metric measures how far a suboptimal reward vector μ_i is from being an optimal arm. Inspired by ϵ-dominance technique that measures the distance between μ_i and the Pareto optimal reward set $\mathcal{O}^*$, we propose a new regret metric. Because the considered Pareto optimal reward set has a discrete number of component rewards, to approximate the closest distance between $\mathcal{O}^*$ and μ_i, we need to construct a *virtual* reward vector that is incomparable with *all* the reward vectors from $\mathcal{O}^*$. By definition, we add to μ_i a positive value ϵ in all objectives, resulting in a so called virtual reward vector $\nu_{i,\epsilon}$. Thus:

$$\nu_{i,\epsilon} = \mu_i + \epsilon \;\text{ where }\; \forall j \;\; \nu^j_{i,\epsilon} = \mu^j_i + \epsilon, \;\text{ and }\; \epsilon > 0$$

The virtual optimal reward for the arm i, ν^*_i has the minimum value for ϵ for which $\nu_{i,\epsilon}$ is incomparable to all the rewards in $\mathcal{O}^*$. Thus,

$$\nu^*_i \leftarrow \min_{\epsilon \rightarrow \infty} \nu_{i,\epsilon}, \;\text{ for which }\; \forall \mu^*_\ell \in \mathcal{O}^*, \;\; \nu^*_i \| \mu^*_\ell$$

We denote the corresponding ϵ for ν^*_i with ϵ^*_i.

The regret of the arm μ_i is equal to the distance between the virtual optimal reward vector of the arm i, ν^*_i, and the reward vector of the same arm, μ_i. Thus,

$$\Delta_i = \nu^*_i - \mu_i = \epsilon^*_i \qquad (3)$$

Since by definition ϵ^*_i is always positive, this regret is always positive. Note that the regret of the arms belonging to $\mathcal{O}^*$ is 0 since the virtual reward coincides with the optimal reward vector itself.

B. A scalarized regret metric

Now we introduce the regret measure that will be used in combination with the scalarisation functions. The *scalarized* *regret* for a particular scalarized function f^j and for the arm i is

$$\Delta_i^j =^{def} \max_{k \in \mathcal{A}} f^j(\mu_k) - f^j(\mu_i) \qquad (4)$$

Thus, the scalarized regret is the difference between the maximum value for a scalarization function on the set of arms $\mathcal{A}$ and the scalarized value for an arm i.

The *linear scalarized regret* for an arm i is

$$\Delta_i^j = \sum_{t=1}^{D} \omega_j^t \cdot (\mu^{*t} - \mu_i^t), \text{ where } f^j(\mu^*) = \max_{k \in \mathcal{A}} f^j(\mu_k)$$

where f^j is a linear scalarization function with the set of weights $\omega_j = \{\omega_j^1, \dots, \omega_j^D\}$. The *Chebyshev scalarized regret* for an arm i and a Chebyshev scalarization function f^j is

$$\Delta_i^j = \max_{1 \leq t \leq D} \omega_j^t \cdot (\mu^{*t} - \mu_i^t), \text{ where } f^j(\mu^*) = \max_{k \in \mathcal{A}} f^j(\mu_k)$$

It is straightforward to show that the maximum value for any set of weights in the linear and Chebyshev functions is one of the Pareto optimal arms. Thus,

$$\forall j \in S, \ \exists! \, i \in \mathcal{A}^* \text{ such that } f^j(\mu_i^*) = \max_{k \in \mathcal{A}} f^j(\mu_k)$$

with S referring to the set of weights. While this definition of regret seems natural, it is improper for our goal because it gathers a collection of independent regrets instead of minimizing the regret of a multi-objective strategy in all objectives. In Example 1, we have shown that the scalarization functions do not always identify all the arms in $\mathcal{A}^*$ as such. Therefore, there is a measure needed to indicate the exploration of all optimal arms.

C. Play fairly the optimal arms

We consider $T_i^*(n)$ the number of times an optimal arm i is pulled, and $\mathbb{E}[T^*(n)]$ the expected number of times optimal arms are selected. The *unfairness* of a multi-objective multi-armed bandits algorithm is defined as the *variance* of the arms in $\mathcal{A}^*$,

$$\phi = \frac{1}{|\mathcal{A}^*|} \cdot \sum_{i \in \mathcal{A}^*} \left(T_i^*(n) - \mathbb{E}[T^*(n)]\right)^2 \qquad (5)$$

For a perfectly fair usage of optimal arms, we have that $\phi \to 0$. When a multi-objective strategy uses only samples a subset of $\mathcal{A}^*$, then the variance is large. Note the resemblance between this measure and the risk metric from economy. In other words, a well performing multi-objective MAB algorithm has low risk of unevenly choosing between optimal arms.

IV. The Pareto UBC1 bandits algorithm

The following multi-objective UCB1 instance uses the Pareto dominance relationships from Table I to order the reward vectors of arms. Like for the single-objective UCB1, the index for this policy has two terms: the mean vector, and the second term related to the size of a one-sided confidence interval of the average reward according with the Chernoff-Hoeffding bounds. The pseudo-code for the Pareto UCB1 is given in Algorithm 1.

Algorithm 1: Pareto UCB1

```
  Play each arm $i$ once
  $n \leftarrow K$; $n_i \leftarrow 1, \forall i$
  repeat
    Find the Pareto set $\mathcal{A}'$ such that $\forall i \in \mathcal{A}', \forall \ell$,
      $\bar{\mathbf{x}}_\ell + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_\ell}} \not\succ \bar{\mathbf{x}}_i + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_i}}$
    Pull $i$ uniform randomly chosen from $A$
    $n \leftarrow n + 1$; $n_i \leftarrow n_i + 1$
    Update $\bar{\mathbf{x}}_i$
  until stopping condition is met
```

As initialization step, each arm is played once. Each iteration, for each arm, we compute the sum of its mean reward vector and its associated confidence interval. A Pareto optimal reward set $\mathcal{A}'$ is calculated from these resulting vectors. Thus, for all the non-optimal arms $\ell \notin \mathcal{A}'$, there exists a Pareto optimal arm $i \in \mathcal{A}'$ that dominates the arm ℓ:

$$\bar{\mathbf{x}}_\ell + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_\ell}} \not\succ \bar{\mathbf{x}}_i + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_i}}$$

We select uniformly at random an optimal arm from $\mathcal{A}'$ and pull it. Thus, by design, this algorithm is fair in selecting Pareto optimal arms. After selection, the mean value $\bar{\mathbf{x}}_i$ and the common counters are updated. A possible stopping criteria is a maximum number of iterations.

In Figure 2, the dynamics of Algorithm 1 is illustrated. A suboptimal arm μ_5 that is closer to the Pareto front according to the Pareto regret metric from Equation 3 is more often selected than an worse arm μ_6, but less than the Pareto optimal arms. The following theorem provides an upper bound for the Pareto regret of the Pareto UCB1 strategy.

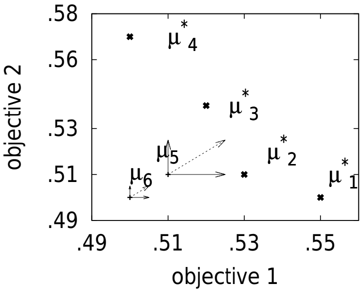
Theorem 1: Let policy Pareto UCB1 from Algorithm 1 be run on a K-armed D-objective bandit problem, $K > 1$, having arbitrary reward distributions $\mathbf{P}_1, \dots \mathbf{P}_K$ with support in $[0, 1]^D$. Consider the Pareto regret defined in Equation 3.

The expected Pareto regret of a policy π after any number of n plays is at most

$$\sum_{i \notin \mathcal{A}^*} \frac{8 \cdot \log(n\sqrt[4]{D|\mathcal{A}^*|})}{\Delta_i} + (1 + \frac{\pi^2}{3}) \cdot \sum_{i \notin \mathcal{A}^*} \Delta_i$$

where $\mathcal{A}^*$ is the set of Pareto optimal arms.

Fig. 2. The dynamics of Pareto UCB1.



Algorithm 2: Scalarized UCB1

```
Require: The scalarization function $f^j = (w^j_1, \dots, w^j_m)$
    Play each arm once
    $\forall j, i,\ n^j \leftarrow K;\ n^j_i \leftarrow 1;$
    Pull arm $i$ that maximizes $\mathbb{E}[f^j(\mathbf{x}_i)] + \sqrt{2\ln(n^j)/n^j_i}$
    Update $\bar{\mathbf{x}}^j_i;\ n^j \leftarrow n^j + 1\ ;\ n^j_i \leftarrow n^j_i + 1;$
```

Algorithm 3: Scalarized multi-objective UCB1

```
Require: $S = (f^1, \dots, f^S)$ scalarized functions
    Initialize the scalarized UCB1 for all $f^j$
    $n \leftarrow S \cdot K;\ n_i \leftarrow S$
    repeat
        Choose uniform randomly a function $f^j$
        Play one time scalarized UCB1 for $f^j$
        Update $\bar{\mathbf{x}}_i;\ n_i \leftarrow n_i + 1;\ n \leftarrow n + 1$
    until a stopping condition is met
```

The expected upper bound of the Pareto regret for Pareto UCB1 is logarithmic in the number of plays n, the number of dimensions D and the number of optimal arms $\mathcal{A}^*$. The worst-case performance of this algorithm is when the number of arms K equals the number of optimal arms $|\mathcal{A}^*|$. The algorithm reduces to the standard UCB1 for $D = 1$. Then, in most of the cases, $|\mathcal{A}^*| \approx 1$. In general, this Pareto UCB1 performs similarly with the standard UCB1 for small number of objectives and small Pareto optimal sets.

Empirical Pareto UCB1. The need to know apriori the size of the Pareto optimal set of arms $\mathcal{A}^*$ is inconvenient. If we consider this unknown, we may replace the second term of the Pareto UCB1's index $\sqrt{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})/n_i}$ with a problem independent index $\sqrt{2\ln(n\sqrt[4]{DK})/n_i}$ that upper bounds this index. This is a standard approach in PAC-learning. We call this algorithm the *empirical Pareto UCB1*. In terms of bounds, this means an increase in the bound magnitude and decrease in the confidence interval. However, for some experimental settings, as considered in the next section, the difference between the two indexes is small and thus does not really affect the performance.

V. Scalarizated multi-objective UCB1 bandits

The following algorithms are extensions of single-objective UCB algorithms where the scalarized order relationship from Section II is considered. The multi-objective UCB1 algorithm using a set of scalarization functions introduced in Section V-A is a straightforward generalization of the standard UCB1 where scalarization functions are alternated uniformly at random. In Section V-C, a scalarized multi-objective UCB1 algorithm is described which removes scalarization functions deemed redundant.

A. The scalarized multi-objective UBC1

This algorithm is a UCB1 algorithm that assumes a set of scalarizated reward vectors $S = (f^1, \dots, f^S)$, $S \geq 1$, with different weights. The pseudo-code for a scalarized UCB1 that uses a scalarization function f^j to reward arms is given in Algorithm 2. The pseudo-code for scalarizated multi-objective UCB1 that alternates scalarized UCB1 with different scalarization functions is given in Algorithm 3.

In Algorithm 2, let n^j be the number of times the function f^j is pulled, and let n^j_i be the number of times the arm i under function f^j is pulled. Let $\mathbb{E}[f^j(\mu_i)]$ be the expected reward of arm i under scalarization function f^j. Given a scalarization function f^j, pull the arm that maximizes the term $\mathbb{E}[f^j(\mu_i)] + \sqrt{2\ln(n^j)/n^j_i}$, and update the counters, where $\bar{\mathbf{x}}^j_i$ is the counter for arm i and scalarization function f^j.

To initialize Algorithm 3, each scalarization function from S for each arm is considered once. Until a stopping criteria is met, choose a scalarization function from S uniformly at random and run the corresponding scalarized UCB1. Update the counters and the expected value of $\bar{\mathbf{x}}_i = (\bar{x}^1_i, \dots, \bar{x}^D_i)$.

Note that each scalarized UCB1 has its own counter, and its individual expected value for each arm is updated separately. Therefore, the upper scalarized regret bound is the same as in [1]. The next proposition shows that the upper bound for the scalarized regret of the scalarized multi-objective UCB1 is the sum of all upper bounds of the scalarized UCB1s.

Proposition 1: Let policy scalarized multi-objective UCB1 from Algorithm 3 be run on a K-armed bandit problem having arbitrary reward distributions $\mathbf{P}_1, \dots, \mathbf{P}_K$ with support in $[0,1]^D$. Consider a set of scalarization functions $S = (f^1, \dots, f^S)$ and the corresponding scalarized regret Δ^j_i described in Equation 4.

We consider that the expected number of plays for all the uniformly chosen functions is $\mathbb{E}[n^j] = n/S$. The expected scalarized regret of strategy A after any number of $n = \sum_{j=1}^S n^j$ plays is equal to the sum of expected regret of each scalarization UCB1 f^j and is at most

$$\sum_{j=1}^{S} \sum_{i \notin \mathcal{A}^*} \frac{8 \cdot \ln(n/S)}{\Delta^j_i} + (1 + \frac{\pi^2}{3}) \cdot \sum_{j=1}^{S} \sum_{i \notin \mathcal{A}^*} \Delta^j_i$$

Proof: The proof follows immediately if we consider that the scalarized multi-objective UCB1 is a uniform random alternation of scalarized UCB1. ■

The leading constant of the above proposition is dominated by the term $\sum_{j=1}^S \sum_{i \notin \mathcal{A}^*} \frac{8 \cdot \ln(n/S)}{\Delta^j_i}$. Thus, the scalarized multi-objective UCB1 should be run S times longer than a single-objective UCB1, the bound in Proposition 1 increasing with the number of scalarization functions in S.

For a general Pareto optimal reward set, it is not known which as well as how many function instances should be used, therefore a uniform distribution of sets of weights is used.

B. Discussion on the scalarized multi-objective UBC1

In case we can assume the Pareto front is convex and bounded we can use Lizotte et al [5]'s method, and obtain the minimum set of weights needed to generate the entire Pareto front. Then, the scalarized multi-objective UCB1 is fair in selecting the Pareto optimal arms. However, [5]'s approach does

Algorithm 4: Improved scalarized multi-objective UCB1

```
**Require:** $S$ scalarized functions, $K$ arms, and $T$ horizon
  Set $\tilde{\Delta}_0 \leftarrow 1$, and $B_0 \leftarrow S$
  **for all** rounds $m = 0, 1, \dots, \lfloor \frac{1}{2} \log_2 \frac{T}{e} \rfloor$ **do**
    **if** $|B_m| > 1$ **then**
      **for all** $f^j \in B_m$ **do**
        Play the scalarized UCB1 for $n_m \leftarrow \lceil \frac{2 \cdot \log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} \rceil$ times
        For each arm $i$, update $\bar{\mathbf{x}}_i$
      **end for**
    **else**
      Chose the only function in $B_m$ until $T$ is reached
    **end if**
    Find the Pareto optimal reward set of round $m$, $\mathcal{A}_m^*$, using the mean reward vectors $\bar{\mathbf{x}}_i$
    **for all** $f^j \in B_m$ **do**
      **if** $\min_{\ell \in S_m^i} \phi_m^\ell + \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n_m}} < \phi_m^j - \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2n_m}}$ **then**
        Delete $f^j$
      **end if**
    **end for**
    Update $B_{m+1}$ to the remaining scalarizations
    $\tilde{\Delta}_{m+1} \leftarrow \tilde{\Delta}_m / 2$
  **end for**
```

not allow stochastic reward vectors, an important assumption in MAB. Furthermore, it has computational problems in highly dimensional reward spaces with irregular shapes that require a large number of weight sets.

Non-convex Pareto optimal sets. In a general setup, where the shape of the Pareto optimal sets is unknown, several sets of weights should be tried out in a scalarized multi-objective UCB1.

Consider linear scalarization functions. As we have showed in Example 1, not all the reward vectors from *any* Pareto optimal reward set are reachable with this scalarization. In this case, there will be always a positive regret between $\mathcal{O}^*$ and $\mathcal{O}_L^*$. The unfairness of this algorithm is increasing with the number of plays because an arm from $\mathcal{A}_L^*$ identified as optimal is increasingly pulled whereas other optimal arms that are not recognized as optimal from $\mathcal{A}^*$ are scarcely pulled.

Consider the Chebyshev scalarization function. It is possible to obtain all the solutions in $\mathcal{A}^*$ by varying the reference points, but there is no indication on how to search for these sets of reference points and we need to search for these points while minimizing the unfairness regret. If there are more Pareto optimal arms identified with the Chebyshev multi-objective UCB1 than with the linear multi-objective UCB1, then the former UCB1 has a lower unfairness than its linear counter part.

C. Improving scalarized multi-objective UCB1

A solution to the above described problem of scalarized multi-objective UCB1 is to design an algorithm that keeps only a minimal set with the best performing scalarized UCB1, and deletes the redundant scalarized UCB1s. We call the scalarized UCB1 which pulls *all* Pareto optimal arms often and evenly a *useful* UCB1. Thus, a scalarized UCB1 with low unfairness is considered useful. A scalarized UCB1 is removed if the Pareto optimal arms are pulled seldom or unevenly and when a certain confidence level is attained.

The pseudo-code for the improved scalarized multi-objective UCB1 algorithm is given in Algorithm 4. The horizon T is assumed to be known and the starting scalarization set is $B_0 \leftarrow S$. Each scalarization function f^j is associated with a scalarized UCB1 instance from Algorithm 2. Each scalarized UCB1 instance is run for a fix number of times n_m. The improved scalarized multi-objective UCB1 algorithm is run m rounds, in each round the number of times each scalarized UCB1 is run, n_m, increases. After running all the scalarized UCB1 instances, a Pareto optimal set of arms for the round m, $\mathcal{A}_m^*$, is computed from the mean reward vectors $\bar{\mathbf{x}}_i$ over all the scalarized UCB1 instances. For each scalarized UCB1, we compute its unfairness ϕ_m^j in the current round m. A scalarization function is deleted if its unfairness minus the confidence interval is larger than the smallest unfairness plus the confidence interval. The process is repeated after updating the set of remaining scalarizations, B_{m+1}, and the factor related with the confidence interval $\tilde{\Delta}_{m+1}$.

Note that the proposed algorithm is an adapted version of the improved UCB algorithm from [2] but here scalarized UCB1 instances are considered for elimination instead of arms. In Algorithm 4, the quality indicator is the unfairness and thus there is a minimization MAB problem. To give an intuition on the behaviour of this improved UCB algorithm, we consider the reward vectors from Example 1. This algorithm prefers the set of weights for which the Pareto optimal arms have about the same value. In Figure 1 b) and d), the set of weights with low unfairness corresponds with the intersection between the upper lines. Thus, for the linear scalarization, the set of weights situated at the intersection between the two upper lines is $\approx (0.6, 0.4)$. For the Chebyshev scalarization, there are three sets of weights with low unfairness corresponding to the three intersection points.

Proving the upper bound of this algorithm is beyond the scope of this paper, but we hint the reader to [7]. Algorithm 4 shows that there are techniques that can ameliorate the performance of the scalarized multi-objective MAB algorithms.

VI. Experiments

The scope of this section is to experimentally compare the behaviour of the three instances of multi-objective UCB1: 1) linear multi-objective UCB1 (lin), 2) Chebyshev multi-objective UCB1 (Cheb), and 3) empirical Pareto UCB1 (Pareto). We have performed experiments also with the canonical Pareto UCB1 but the results were similar with the results of the empirical Pareto UCB1 because the values of the two indices are similar. If $K = 20$ and $|\mathcal{A}^*| = 4$, then $\sqrt[4]{DK} = 2.51$ and $\sqrt[4]{D|\mathcal{A}^*|} = 1.68$. We measure: a) the percentage of time one of the optimal arms is used, b) the percentage of time each of the optimal arms is pulled, c) the regret, and d) the unfairness in using the optimal arms.

Each algorithm is run 100 times. We consider 11 weight sets for the linear and Chebyshev scalarization functions, $\{(1,0), (0.9, 0.1), \dots, (0.1, 0.9), (0, 1)\}$. For the Chebyshev

TABLE II. THE USAGE OF OPTIMAL ARMS IN TWO MULTI-OBJECTIVE BERNOULLI DISTRIBUTIONS

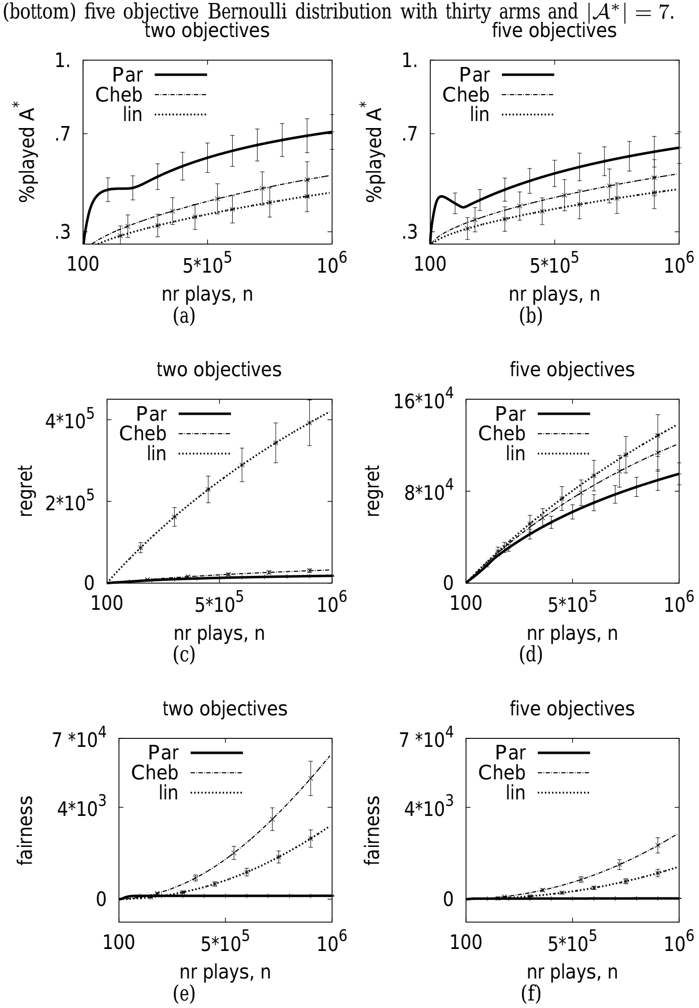
A. The bi-objective Bernoulli distribution with $K = 20$

MO-UCB1	Percent played arm(s)				
	$\mathcal{A}^*$	μ_1^*	μ_2^*	μ_3^*	μ_4^*
Pareto	71 ± 7	18 ± 2	17 ± 2	18 ± 2	18 ± 2
Cheb	53 ± 8	14 ± 2	7 ± 1	8 ± 1	23 ± 3
lin	46 ± 7	11 ± 2	8 ± 1	10 ± 1	17 ± 2

B. The five-objective Bernoulli distribution with $K = 30$

MO-UCB1	Percent played arm(s)							
	$\mathcal{A}^*$	μ_1^*	μ_2^*	μ_3^*	μ_4^*	μ_5^*	μ_6^*	μ_7^*
Pareto	64 ± 6	9 ± 1	9 ± 1	9 ± 1	9 ± 1	9 ± 1	9 ± 1	9 ± 1
Cheb	54 ± 8	13 ± 2	7 ± 1	4 ± 1	5 ± 1	8 ± 1	6 ± 1	11 ± 2
lin	47 ± 7	9 ± 1	6 ± 1	5 ± 1	6 ± 1	8 ± 1	6 ± 1	8 ± 1

Fig. 3. The performance of the three multi-objective UCB1 on (top) the bi-objective Bernoulli distribution with twenty arms and $|\mathcal{A}^*| = 4$ and on (bottom) five objective Bernoulli distribution with thirty arms and $|\mathcal{A}^*| = 7$.



scalarization, we uniformly at random generate the parameters $\epsilon^j \propto [0, 0.1]$ from Equation 1 for the reference point $\mathbf{z}$.

A. Adding arms to Example 1. For the first simulation we have added another 14 arms in Example 1, resulting in 20 armed bandits. The added arms are dominated by all the arms in $\mathcal{A}^*$. We take $\mu_7 = \ldots = \mu_{20} = (0.48, 0.48)$, leaving the Pareto optimal reward set unchanged.

Figure 3 a), b) and c) and Table II.A show good performance for Pareto UCB1, i.e. high and fair usage of the optimal arms and low regret, as compared with the scalarized multi-objective UCB1 algorithms. The worst multi-objective UCB1 algorithm, with the highest regret and lowest usage of the optimal arms, is the linear multi-objective UCB1. Chebyshev multi-objective UCB1 has a better performance than its linear counter part and worse than Pareto UCB1.

B. Adding dimensions to Example 1. In order to test the algorithms on a more complex multi-objective environment, we add three dimensions for each reward vector in the previous bi-objective environment and 10 suboptimal arms. The Pareto optimal set of arms $\mathcal{A}^*$ contains now 7 arms.

Figure 3 d), e) and f) and Table II.B show a similar performance of the three multi-objective UCB1 algorithms on the five objective Bernoulli distribution. Pareto UCB1 is again the best algorithm, the linear multi-objective UCB1 is the worst algorithm, and Chebyshev multi-objective UCB1 has an intermediate performance.

Discussion. Let's compare the performance of the multi-objective UCB1 algorithms on the two multi-objective Bernoulli distributions. The percentage of time an optimal arm is played with one of the scalarized multi-objective UCB1 is about the same, because the proportion of optimal arms in the two distributions is about the same. The percentage of playing an optimal arm with Pareto UCB1 decreases with the increased number of elements in the Pareto optimal reward set. For $K = 30$, the Pareto regret is larger and the unfairness is smaller than for $K = 20$ indicating a shortage in samples for the larger multi-objective environment. Furthermore, Pareto UCB1 is more fair than both scalarization multi-objective UCB1 algorithms. In conclusion, Pareto UCB1 performs the best and is the most robust from the three tested algorithms.

VII. CONCLUSION

We introduced multi-objective multi-armed bandits algorithms with multiple, possibly conflicting, reward values for an arm. We considered partial order relationships associated with reward vectors as well as linear and Chebyshev scalarization. By means of an example, we explain the difference between these approaches and we show that the discussion is valid in a more general multi-objective reinforcement learning setting. Three regret metrics that measure the performance of multi-objective MAB are introduced. The Pareto regret metric measures the distance between a reward vector and the Pareto optimal reward set, whereas a scalarized regret metric measures the distance to a single optimal arm. The unfairness, an extra performance measure complementary to the scalarized regret metric, measures the variance in the usage of *all* the optimal arms. Instances of multi-objective UCB1 algorithms extending the standard UCB1 are designed using the partially ordered reward vector sets. We showed that even though the straightforward scalarized multi-objective UCB1 is not efficient, there are variants that can improve its performance. We have proven logarithmic upper regret bounds for the Pareto UCB1 and compared the proposed multi-objective UCB1 algorithms on two multi-objective Bernoulli reward distributions. To conclude, our Pareto UCB1 algorithm is the most suited to explore/exploit the multi-arm bandits with reward vectors.

REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2/3):235–256, 2002.

[2] P. Auer and R. Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[3] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer, 2008.

[4] C. Horoba and F. Neumann. Benefits and drawbacks for the use of ϵ-dominance in evolutionary multi-objective optimization. In *Proc of Genetic and Evolutionary Computation Conference (GECCO'08)*, 2008.

[5] D.J. Lizotte, M. Bowling, and S.A. Murphy. Efficient reinforcement learning with multiple reward functions for randomized clinical trial analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[6] K.M. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research and Management Science. Springer, 1998.

[7] A. Sani, A. Lazaric, and R. Munos. Risk-aversion in multi-armed bandits. In *NIPS*, pages 3284–3292, 2012.

[8] K. van Moffaert, M.M. Drugan, and A. Nowe. Hypervolume-based multi-objective reinforcement learning. In *Proc of Evolutionary Multi-objective Optimization (EMO)*, pages 352–366. Springer, 2013.

[9] W. Wang and M. Sebag. Multi-objective Monte Carlo tree search. In *Asian conference on Machine Learning*, pages 1–16, 2012.

[10] M.A. Wiering and E.D. de Jong. Computing optimal stationary policies for multi-objective Markov decision processes. In *Proc of Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 158–165. IEEE, 2007.

[11] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE T. on Evol. Comput.*, 7:117–132, 2003.

APPENDIX

For this proof, we follow the corresponding proof from [1]. Let $\mathbf{X}_{i,1}$,...,$\mathbf{X}_{i,n}$ be random D-dimensional variables generated for arm i with common range $[0,1]^D$. The expected reward vector for the arm i after n pulls is

$$\bar{\mathbf{X}}_{i,n} = 1/n \cdot \sum_{t=1}^{n} \mathbf{X}_{i,t} \quad \Rightarrow \quad \forall j, \quad \bar{X}^j_{i,n} = 1/n \cdot \sum_{t=1}^{n} X^j_{i,t}$$

Chernoff-Hoeffding bound. We use a straightforward generalization of the standard Chernoff-Hoeffding bound for D dimensional spaces. Consider that $\forall j$, $1 \leq j \leq D$, $I\!\!E[\mathbf{X}^j_{i,t} \mid \mathbf{X}^j_{i,1}, \ldots, \mathbf{X}^j_{i,t-1}] = \mu^j_i$. Recall, the non-dominated relationship from Table I. There, $\bar{\mathbf{X}}_{i,n} \nprec \mu_i + a$ if there exists *at least* a dimension j for which $\bar{X}^j_{i,n} > \mu^j_i + a$. Translated in Chernoff-Hoeffding bound, using union bound, for all $a \geq 0$, we have

$$I\!\!P\{\big(\bar{\mathbf{X}}_{i,n} \nprec \mu_i + a\big)\} = \tag{6}$$

$$I\!\!P\{\big(\bar{X}^1_{i,n} > \mu^1_i + a\big) \vee \ldots \vee \big(\bar{X}^D_{i,n} > \mu^D_i + a\big)\} \leq De^{-2na^2}$$

Following the same line of reasoning

$$I\!\!P\{\big(\bar{X}^1_{i,n} < \mu^1_i - a\big) \vee \ldots \vee \big(\bar{X}^D_{i,n} < \mu^D_i - a\big)\} \leq De^{-2na^2} \tag{7}$$

Let $\ell > 0$ an arbitrary number. We denote with $c_{t,s} = \sqrt{2 \cdot \ln\left(t \sqrt[4]{D|\mathcal{A}^*|}\right)/s}$. Let $\mathcal{A}^*$ be the set of optimal arms. We now upper bound $T_i(n)$ on any sequence of plays by bounding for each $t \geq 1$ the indicator $(I_t = i)$. We have $(I_t = i) = 1$ if arm i is played at time t and $(I_t = i) = 0$ otherwise. Here, we consider that an arm i can be selected if it is non-dominated by *all* the optimal arms from $\mathcal{A}^*$.

$$T_i(n) = 1 + \sum_{t=K+1}^{n} \{I_t = i\} \leq \ell + \sum_{t=K+1}^{n} \{I_t = i, T_i(t-1) \geq \ell\}$$

$$\nsucc \bar{\mathbf{X}}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}, T_i(t-1) \leq \ell\}$$

$$\leq \ell + \sum_{t=K+1}^{n} \sum_{h=1}^{|\mathcal{A}^*|} \{\bar{\mathbf{X}}^*_{h,T^*_h(t-1)} + c_{t-1,T^*_h(t-1)}$$

$$\nsucc \bar{\mathbf{X}}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}\} \leq \begin{array}{l} s^*_h \leftarrow T^*_h(t-1) \\ s_i \leftarrow T_i(t-1) \end{array}$$

$$\ell + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=\ell}^{t-1} \sum_{h=1}^{|\mathcal{A}^*|} \{\bar{\mathbf{X}}^*_{h,s^*_h} + c_{t-1,s^*_h} \nsucc \bar{\mathbf{X}}_{i,s_i} + c_{t-1,s_i}\} \tag{8}$$

From Algorithm 1, we have that $\bar{\mathbf{X}}^*_{h,s^*_h} + c_{t,s^*_h} \nsucc \bar{\mathbf{X}}_{i,s_i} + c_{t,s_i}$ which implies that at least one of the conditions must hold

$$\bar{\mathbf{X}}^*_{h,s^*_h} \nsucc \mu^*_h - c_{t,s^*_h} \,;\, \bar{\mathbf{X}}_{i,s_i} \nprec \mu_i + c_{t,s_i} \,;\, \mu^*_h \nsucc \mu_i + 2 \cdot c_{t,s_i} \tag{9}$$

We bound the probability of events from Inequality 9 using the straightforward generalized Chernoff-Hoeffding bound to D dimensional reward vectors from Inequalities 6 and 7

$$I\!\!P\{\bar{\mathbf{X}}_{i,s_i} \nprec \mu_i + c_{t,s}\} \leq \frac{D}{D} \cdot \frac{t^{-4}}{|\mathcal{A}^*|} = \frac{t^{-4}}{|\mathcal{A}^*|}$$

$$I\!\!P\{\bar{\mathbf{X}}^*_{s^*_h} \nsucc \mu^*_h - c_{t,s^*_h}\} \leq \frac{t^{-4}}{|\mathcal{A}^*|}$$

For $s_i \geq \frac{8 \cdot \ln(n \sqrt[4]{D|\mathcal{A}^*|})}{\Delta_i^2}$, we have that

$$\nu^*_i - \mu_i - 2 \cdot c_{t,s_i} = \nu^*_i - \mu_i - 2 \cdot \sqrt{\frac{2 \cdot \ln(n \sqrt[4]{D|\mathcal{A}^*|})}{s_i}} \geq \nu^*_i - \mu_i - \Delta_i$$

Thus, we take $\ell = \lceil \frac{8 \cdot \ln(n \sqrt[4]{D|\mathcal{A}^*|})}{\Delta_i^2} \rceil$.

Then,

$$I\!\!E[T_i(n)] \leq \lceil \frac{8 \cdot \ln(n \sqrt[4]{D|\mathcal{A}^*|})}{\Delta_i^2} \rceil + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i = \lceil \frac{8 \cdot \ln\,(n \sqrt[4]{D|\mathcal{A}^*|})}{\Delta_i^2} \rceil} \sum_{h=1}^{|\mathcal{A}^*|}$$

$$\left(I\!\!P\{\bar{\mathbf{X}}^*_{h,s^*_h} \nsucc \mu^*_h - c_{t,s^*_h}\} + I\!\!P\{\bar{\mathbf{X}}_{i,s_i} \nprec \mu_i + c_{t,s_i}\}\right) \leq$$

$$\lceil \frac{8 \cdot \ln\left(n \sqrt[4]{D|\mathcal{A}^*|}\right)}{\Delta_i^2} \rceil + 2 \cdot \sum_{t=1}^{\infty} t^2 \cdot |\mathcal{A}^*| \frac{t^{-4}}{|\mathcal{A}^*|}$$

Approximating the last term with the Riemann zeta function $\zeta(2) = \sum_{t=1}^{\infty} t^{-2} \approx \frac{\pi^2}{6}$ we obtain the bound from the theorem. □