Approximative Pareto Front Identification

Madalina M. Drugan, *Member, IEEE*
Technical University of Eindhoven, Department of Computer Science
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
Email: m.m.drugan@tue.nl

***Abstract*—Techniques from multi-objective optimization are incorporated into the stochastic multi-armed bandit (MAB) problem to improve performance when the rewards obtained from pulling an arm are random vectors instead of random variables. We call this problem the stochastic multi-objective MAB (or MOMAB) problem. In this paper, we study the analytical and empirical proprieties of MOMABs with the goal of identifying multiple arms in the Pareto front that use the partial Pareto dominance relation to compare mean reward vectors. We introduce three algorithms: 1) *Pareto Front Identification* identifies the Pareto optimal arms using a fixed budget. 2) ϵ-*approximate Pareto Front Identification* uses the Pareto ϵ-dominance to identify a uniformly spread subset of the Pareto front. 3) *Pareto Subfront Identification* combines the last two algorithms to improve the accuracy of the ϵ-approximation Pareto front. We experimentally compare the proposed algorithms on several Pareto MAB-problems.**

I. Introduction

There are many interesting applications in the field of automatic control where one wants to simultaneously meet different objectives. Objectives can be aligned as well as conflicting and the Pareto front can have any shape, i.e. not necessarily convex, and any distribution of arms. Furthermore, it is hard to assign prior weights to these objectives and, as a result, approaches that combine these weighted objectives into a single objective function, e.g. scalarization, have problems in identifying the Pareto front.

Multi-armed bandits (MABs) [1] is a machine learning problem used to analyze resource allocation in stochastic environments. The *stochastic multi-objective multi-armed bandit* problem (or stochastic MOMAB) was introduced in [2] and can be considered a generalization of MABs scalar rewards to reward vectors. Some techniques from multi-objective optimization were already used in other related learning problems: multi-objective Markov decision processes [3], [4] and multi-objective reinforcement learning [5]–[7]. A reward vector of one arm can be better than the reward vector of another arm according to one objective but worse according to another objective. *Pareto dominance relation* [8], which is a partial order relationship, compares multi-objective rewards. According to this relation, the quality of several arms denoted as Pareto front can be considered to be equal.

The paper is organized as follows: Section II introduces a "pure-exploration" approach of Pareto front identification, or "EXPLORE"-m algorithms, where the goal is to identify with a given tolerance the arms in the Pareto front. Unlike the homologues "pure-exploration" algorithm [9] for single objective environments, the number of desired best arms is not known beforehand. The goal is an algorithm with a small probability to erroneously select a suboptimal arm after a number of steps. We propose a straightforward extension of PAC subset selection problem to Pareto MABs.

There are two classes of "EXPLORE"-m algorithms [10]: i) fixed budget, and ii) fixed tolerance algorithms. A fixed budged best arm identification algorithm, i.e. successive rejects algorithm [11], uses a limited number of arms pulls to select the best arm. A fixed tolerance algorithm assumes that two arms can be ordered only if they are further apart than a small tolerance value. [12] identifies the Pareto front using scalarization functions and the best arm identification algorithm from [11]. [13] uses the Pareto order relation to identify the Pareto front using the revised UCB algorithm from [14]. Both algorithms are fixed budget algorithms.

In Section III, we consider a fixed available budget of arm pulls to identify the Pareto front. Audibert et al. [11] propose a best arm identification-algorithm to identify a single optimal arm. Its generalization, the m-best arm identification-algorithm [15] identifies the m best arms of a MAB-problem. We extend these algorithms to Pareto MAB-problems. *Pareto front identification* is a fixed budget successive reject algorithm that sequentially removes suboptimal arms and stops after a fixed number of arm pulls. This is the fixed budget available to the algorithm.

Pareto MOMAB methods, and thus also the proposed approach, have computational problems when the Pareto front is large, because, each iteration, Pareto front is stored and compared. Section IV introduces the fixed confidence ϵ-*approximation Pareto Front Identification* problem where the ϵ-dominance relation is used to compare mean reward vectors. The multi-objective environment is considered to a hypergrid consisting of D-dimensional rectangles, or D-rectangles, and the arms are considered to be part of one of these D-rectangles, where D is the number of objectives. The algorithm assigns arms to D-rectangles in order to deterministically delete dominated D-rectangles instead of dominated arms. In each non-dominated D-rectangle, one arm, that is at most at an ϵ-distance from the Pareto front, is selected at random.

Section V introduces a hybrid Pareto MAB-algorithm called *Pareto subfront identification*. A single non-dominated arm is selected in each D-rectangle using a fixed budget successive reject algorithm. Unlike *Pareto front identification*, this hybrid algorithm does not assume that the cardinality of the Pareto front is known beforehand. This allows for a tighter upper confidence bound since the performance depends on the number of non-dominated D-rectangles. And

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.128

this is a parameter that can be tuned by the user.

In Section VI, we study the exploration vs. exploitation trade-off of the Pareto MAB algorithms in complex multi-objective stochastic environments. We compare the performance of the proposed Pareto MAB-algorithms on a bi-objective environment with a convex Pareto front where the components of the reward vectors are drawn according to independent multi-variate normal distributions. Section VII concludes the paper.

II. THE PARETO-"EXPLORE" PROBLEM

The goal of a Pareto-"EXPLORE" problem is to select (a subset) of the Pareto front with a small probability error in a finite number of samples. Thus, a *Pareto EXPLORE* algorithm is (ϵ, δ)-optimal if it selects Pareto optimal arms with accuracy ϵ and a small error probability of $1-\delta$. For an efficient exploration mechanism, we assume that the size of Pareto front is controlled by the user. Problem complexity is defined as the number of pulls an algorithm performs before termination.

Consider a set of K arms, $K \geq 2$, where $\mathcal{I}$ is the set of these K arms. Pulling an arm returns a random vector of rewards, one component per objective. The random vectors have a stationary distribution with support in the D-dimensional hypercube $[0,1]^D$ but the vector of true expected rewards $\boldsymbol{\mu}_i = (\mu_i^1, \ldots, \mu_i^D)$ is unknown. All rewards obtained from any arm i at time step t are independently and identically distributed. Rewards obtained from different arms are also assumed to be independent. The rewards are almost surely bounded random vectors so that we can apply the Hoeffding inequality [16].

Pareto dominance for uncertain environments. *Pareto dominance relation* is the natural order for the multi-objective environments allowing ordering the reward vectors directly in the multi-objective reward space. The usage of the Pareto dominance relationship in MOMABs is described in [2], [17] and in multi-objective reinforcement learning in [4]. Since the standard dominance relations are too crisp for their practical usage in stochastic environments, we import from multi-objective evolutionary optimisation [18] the *Pareto dominance relation for uncertain environments* (PDU) that is a generalisation of the standard Pareto dominance relationship. We adapt the original PDU definition that uses median and percentile to the usage in the MAB framework: 1) the unbiased estimator is a mean reward vector and 2) a confidence vector represent the uncertainty. The arms are identified within a confidence value that decreases over time and number of samples, a common practice in "EXPLORE"-m strategies.

For simplicity, we consider equal confidence values in all objectives $\boldsymbol{\epsilon} = (\epsilon, \ldots, \epsilon)$. We say that $\boldsymbol{\mu}_i$ *dominates* $\boldsymbol{\mu}_k$, $\boldsymbol{\mu}_k \prec \boldsymbol{\mu}_i$, iff there exists at least one objective j for which $\mu_k^j + \epsilon_i < \mu_i^j - \epsilon_i$ and for all other objectives k, we have $\mu_k^j + \epsilon_k \leq \mu_i^j - \epsilon_i$. We say that $\boldsymbol{\mu}_i$ *non-dominates* $\boldsymbol{\mu}_k$, $\boldsymbol{\mu}_k \not\prec \boldsymbol{\mu}_i$, iff there exists at least one objective j for which $\mu_i^j + \epsilon_i < \mu_k^j - \epsilon_k$. We say that $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_k$ are *incomparable*, iff either $\boldsymbol{\mu}_i$ dominates $\boldsymbol{\mu}_k$ nor $\boldsymbol{\mu}_k$ dominates $\boldsymbol{\mu}_i$.

```
Require: $\delta > 0$ error probability, $\epsilon$ accuracy
  for all arms $i \in \mathcal{I}$ do
    Sample $i$ for $\ell = \frac{4}{\epsilon^2} \cdot \ln\left(\frac{2KD}{\delta}\right)$ times
    Let $\tilde{\boldsymbol{\mu}}_i$ be the empirical reward vector of arm $i$
  end for
  return $\tilde{\mathcal{I}}^*$ is the empirical Pareto front
```

Algorithm 1: Naive (ϵ, δ)-Pareto PAC algorithm

In a practical setting, for each arm i, we need an unbiased estimator of the expected reward vector $\boldsymbol{\mu}_i$, which for the general MAB problem it is the empirical mean vector $\tilde{\boldsymbol{\mu}}_i$. The estimated sample mean of an arm i is the vector $\tilde{\boldsymbol{\mu}}_i(n) = \left(\sum_{s=1}^{n_i} \frac{X_i^1(s)}{n_i}, \ldots, \sum_{s=1}^{n_i} \frac{X_i^D(s)}{n_i}\right)$, where $\mathbf{X}_i(s) = (X_i^1(s), \ldots, X_i^D(s))$ is the sample s for arm i and n_i is the number of times a suboptimal arm i has been played by the policy π during the first n plays. The uncertainty vector $\boldsymbol{\epsilon}_i$ for arm i gives the uncertainty, or the probability, that $\tilde{\boldsymbol{\mu}}_i$ correctly approximates the true reward vector within a confidence value.

Let the *Pareto optimal set of arms* $\mathcal{I}^*$ be the set of arms whose reward vectors are non-dominated by all other reward vectors[1]. When compared with the regular Pareto dominance (PD), the Pareto front generated with PDU is larger and contains the regular Pareto front. In the limit, when the confidence value is approaching 0, the two Pareto fronts coincide.

The usage of PDU is demonstrated on the bi-objective example with 20 arms. When the standard Pareto dominance relation is considered, there are: i) ten Pareto optimal reward vectors, and ii) ten suboptimal arms. Naming, the Pareto front is $\boldsymbol{\mu}_1^* = (0.562, 0.493)$, $\boldsymbol{\mu}_2^* = (0.552, 0.515)$, $\boldsymbol{\mu}_3^* = (0.543, 0.527)$, $\boldsymbol{\mu}_4^* = (0.535, 0.535)$, $\boldsymbol{\mu}_5^* = (0.525, 0.555)$, $\boldsymbol{\mu}_6^* = (0.523, 0.557)$, $\boldsymbol{\mu}_7^* = (0.515, 0.563)$, $\boldsymbol{\mu}_8^* = (0.506, 0.568)$, $\boldsymbol{\mu}_9^* = (0.503, 0.571)$, $\boldsymbol{\mu}_{10}^* = (0.497, 0.573)$. Suboptimal arms are $\boldsymbol{\mu}_{11} = (0.498, 0.567)$, $\boldsymbol{\mu}_{12} = (0.502, 0.563)$, $\boldsymbol{\mu}_{13} = (0.505, 0.495)$, $\boldsymbol{\mu}_{14} = (0.508, 0.555)$, $\boldsymbol{\mu}_{15} = (0.512, 0.533)$, $\boldsymbol{\mu}_{16} = (0.514, 0.525)$, $\boldsymbol{\mu}_{17} = (0.522, 0.554)$, $\boldsymbol{\mu}_{18} = (0.531, 0.531)$, $\boldsymbol{\mu}_{19} = (0.542, 0.523)$, $\boldsymbol{\mu}_{20} = (0.547, 0.513)$.

For a very small confidence value $\epsilon = 10^{-4}$, the size of the Pareto front does not increase, whereas if $\epsilon = 5 \cdot 10^{-4}$ there are 2 extra arms in the Pareto front, $\boldsymbol{\mu}_{17}$ and $\boldsymbol{\mu}_{19}$. When the confidence value increases to $\epsilon = 10^{-3}$, there is one more Pareto optimal arm added to the Pareto front, i.e. $\boldsymbol{\mu}_{20}$. When $\epsilon = 5 \cdot 10^{-3}$, the number of Pareto optimal arms is 17 because four extra arms, i.e. $\boldsymbol{\mu}_{11}$, $\boldsymbol{\mu}_{12}$, $\boldsymbol{\mu}_{14}$, and $\boldsymbol{\mu}_{18}$, are added. If $\epsilon = 0.01$, then the Pareto front has size 19. Note that there is one arm that is suboptimal for all these confidence values, and that is $\boldsymbol{\mu}_{13}$.

Naive PAC for Pareto front identification is the simplest Pareto "EXPLORE" algorithm . Each arm is pulled for $\frac{4}{\epsilon^2} \cdot \ln\left(\frac{2KD}{\delta}\right)$ such that there is a probability of $1 - \frac{\delta}{K}$ the mean reward vector is an ϵ approximation of the true mean reward vector.

[1] We denote with the symbol $*$ all the quantities that are related to the Pareto front, e.g. the reward vector of an arm i in the Pareto front is $\boldsymbol{\mu}_i^*$.

```
Require: The number of Pareto optimal arms $|\mathcal{I}^*|$
  Let $\mathcal{I}^{(1)} \leftarrow \mathcal{I}$, and $n^{(1)} \leftarrow 0$, and
  $n^{(t)} \leftarrow \left\lceil \frac{1}{\overline{\log}(K/|\mathcal{I}^*|)+1} \cdot \frac{N-K}{K+1-t} \right\rceil$
  for all rounds $t = 1, 2, \ldots, K - |\mathcal{I}^*|$ do
    (1) $\forall i \in \mathcal{I}^{(t)}$, select arm $i$ for $n^{(t)} - n^{(t-1)}$ rounds
    (2) Let $argmin_{i \in \mathcal{I}^{(t)}} \tilde{\boldsymbol{\mu}}_i$ the arm to dismiss
    (3) $\mathcal{I}^{(t+1)} \leftarrow \mathcal{I}^{(t)} \setminus argmin_{i \in \mathcal{I}^{(t)}} \tilde{\boldsymbol{\mu}}_i$
  end for
  Let the remaining set of arms be $\tilde{\mathcal{I}}^* \leftarrow \mathcal{I}^{(K-|\mathcal{I}^*|)}$
```

Algorithm 2: Pareto successive rejects (PSR) algorithm

Theorem 1: The Naive (ϵ, δ)-Pareto PAC algorithm, cf Algorithm 1, has the sample complexity of $\mathcal{O}\left(\frac{K}{\epsilon^2} \cdot \ln\left(\frac{2KD}{\delta}\right)\right)$.

Proof: Let k be a suboptimal arm and i be a Pareto optimal arm. Then, for all objectives j and for all Pareto optimal arms, we have $\mu_k^j + \epsilon/2 < \tilde{\mu}_i^j - \epsilon/2$. We want to bound the event that exists an objective j for which $\tilde{\mu}_k^j > \tilde{\mu}_i^j$, thus, that arm k is non-dominated by the Pareto optimal arm i, $\tilde{\boldsymbol{\mu}}_k \not\prec \tilde{\boldsymbol{\mu}}_i$. Then, the probability of misclassifying the two arms is

$$\mathcal{P}(\tilde{\boldsymbol{\mu}}_k \not\prec \tilde{\boldsymbol{\mu}}_i) = \mathcal{P}(\exists j,\ \text{for which } \tilde{\mu}_k^j > \tilde{\mu}_i^j) =$$

$$\mathcal{P}(\exists j,\ \tilde{\mu}_i^j < \mu_i^j - \epsilon/2 \text{ or } \tilde{\mu}_k^j > \mu_k^j + \epsilon/2) \leq 2 \cdot D \cdot e^{-(\epsilon/2)^2 \ell}$$

where this inequality uses the Hoeffding inequality and union bound. Since $\ell = \frac{4}{\epsilon^2} \cdot \ln\left(\frac{2KD}{\delta}\right)$, we have that $\mathcal{P}(\tilde{\boldsymbol{\mu}}_k \not\prec \tilde{\boldsymbol{\mu}}_i) \leq \frac{\delta}{K}$. □ ■

Alternative algorithms that pull preferentially arms that are close to the Pareto front, or incrementally rule out suboptimal arms based on their distance to the Pareto front, have theoretically and empirically a better computational complexity.

III. Pareto Front Identification

In this section, we extend the fixed budget best arm identification algorithm to vector rewards to identify the entire set of Pareto optimal arms, or Pareto front. We call these class of algorithms *Pareto front identification* (PFI). We introduce the *Pareto successive rejects* (PSR) algorithm which is an extension of the fixed budget successive rejects algorithm [11] for the single-objective MAB-problem. The main idea is to successively delete dominated arms until all the remaining arms are non-dominated. As in [15], the number of best arms one want to identify is assumed to be known beforehand.

The pseudo-code for the *Pareto Successive Rejects* algorithm is given in Algorithm 2. The algorithm has $K - |\mathcal{I}^*|$ phases of increasing length carefully chosen to obtain logarithmic convergence. In the t-th phase, all the active arms, i.e. the ones that are not deleted yet, are equally pulled for $n^{(t)} - n^{(t-1)}$ times, where

$$n^{(t)} = \left\lceil \frac{1}{\overline{\log}(K/|\mathcal{I}^*|) + 1} \cdot \frac{N-K}{K+1-t} \right\rceil \tag{1}$$

To simplify the notation, we denote $C = \frac{1}{\overline{\log}(K/|\mathcal{I}^*|)+1}$, where by definition

$$\overline{\log}(K/|\mathcal{I}^*|) = \overline{\log}(K) - \overline{\log}(|\mathcal{I}^*|) = \sum_{t=1}^{K} \frac{1}{t} - \sum_{j=1}^{|\mathcal{I}^*|} \frac{1}{j}$$

$$= \sum_{t=|\mathcal{I}^*|+1}^{K} \frac{1}{t} = \frac{1}{|\mathcal{I}^*|+1} + \sum_{t=1}^{K-|\mathcal{I}^*|+1} \frac{1}{K+1-t}$$

C is a positive constant since $K \geq |\mathcal{I}^*|$.

At the end of each phase, the algorithm deletes the arm with the estimated mean reward vector dominated by the other estimated mean reward vectors. In case of a tie, i.e. when there are several reward vectors that are dominated by other reward vectors, we chose randomly an arm to delete. The remaining arms are non-dominated and recommended as the Pareto front $\mathcal{I}^*$.

The worst arm is pulled $n^{(1)} = \left\lceil C \cdot \frac{N-K}{K} \right\rceil$ times, the second worst arms is pulled $n^{(2)} = \left\lceil C \cdot \frac{N-K}{K-1} \right\rceil$ times, and the number of times an arm is pulled increases with its quality. The $|\mathcal{I}^*|$ best arms are pulled $n^{(K-|\mathcal{I}^*|)} = \left\lceil C \cdot \frac{N-K}{|\mathcal{I}^*|+1} \right\rceil$ times. Note that the fixed budget n which is not exceeded because

$$\sum_{k=1}^{K-|\mathcal{I}^*|} n^{(t)} + |\mathcal{I}^*| \cdot n^{(K-|\mathcal{I}^*|+1)} \leq$$

$$K + \sum_{t=1}^{K-|\mathcal{I}^*|} C \cdot \frac{N-K}{K-t} + |\mathcal{I}^*| \cdot C \cdot \frac{N-K}{|\mathcal{I}^*|+1} \leq$$

$$K + \frac{N-K}{\overline{\log}(K/|\mathcal{I}^*|)+1} \left(\frac{|\mathcal{I}^*|+1}{|\mathcal{I}^*|+1} + \sum_{t=1}^{K-|\mathcal{I}^*|+1} \frac{1}{K+1-t} \right)$$

$$= N$$

The proof of the following theorem follows closely the proof in [11]. Let us consider the complexity measure

$$H_2 = \max_{i \in \mathcal{I}} i \Delta_i^{-2} \tag{2}$$

where Δ_i is defined as the Euclidean distance between the mean reward vector $\boldsymbol{\mu}_i$ of an arm i and its projection $\boldsymbol{\nu}_i$ onto the Pareto front and H_2 quantifies the hardness of the problem [11]. This projection is obtained as follows: A vector $\boldsymbol{\epsilon}_i$ with equal components ϵ_i, i.e. $\boldsymbol{\epsilon}_i = (\epsilon_i, \epsilon_i, \cdots, \epsilon_i)$, is added to $\boldsymbol{\mu}_i$ such that ϵ_i is the smallest value for which $\boldsymbol{\nu}_i = \boldsymbol{\mu}_i + \boldsymbol{\epsilon}_i$ becomes Pareto optimal, $\Delta_i = \|\boldsymbol{\nu}_i - \boldsymbol{\mu}_i\|_2 = \|\boldsymbol{\epsilon}_i\|_2 = \sqrt{D}\epsilon_i$, where the last equality holds because we have D objectives and all components of $\boldsymbol{\epsilon}_i$ are equal. The projection regret is defined as the cumulative difference between always selecting the best arm and selecting suboptimal arms with an algorithm of reference.

Theorem 2: The probability of deleting a Pareto optimal arm after n plays is at most

$$e^{(n)} \leq D|\mathcal{I}^*| \binom{K - |\mathcal{I}^*|}{2} \cdot e^{-\frac{(n-K)}{H_2(\overline{\log}(K/|\mathcal{I}^*|)+1)}} \tag{3}$$

By a union bound and Hoeffding's inequality, the probability of erroneously deleting a Pareto optimal arm results directly.

The number of times an arm is played depends on the size of $|\mathcal{I}^*|$. This is a very strong limitation since in many cases its cardinality is not known. Note that when the Pareto optimal set is large, i.e. $|\mathcal{I}^*| \approx K$, the Pareto successive rejects algorithm uniformly pulls a large number of optimal arms. As a result it behaves poorly, like Pareto UCB1 in that case.

There are important differences between PSR, cf. Algorithm 2, and the m-best arm identification algorithm [15] where the m-best arms can be ordered. The number of times an arm is pulled is larger for PSR compared with the best arm identification algorithm for the single-objective MAB. If $\mathcal{I}^* \approx 1$ then that number is close to $\left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{N-K}{K+1-t} \right\rceil$. If $|\mathcal{I}^*| \approx K$ then that number reaches its maximum $\left\lceil \frac{N-K}{K+1-t} \right\rceil$. This shows the importance of correctly approximating $|\mathcal{I}^*|$ for the exploration vs. exploitation trade-off. If the size of $\mathcal{I}^*$ is overestimated, the size of the rounds is too large resulting in poor exploitation. If the size of $\mathcal{I}^*$ is underestimated, then some of the Pareto optimal arms will be deleted, resulting in poor exploration.

If the cardinality of Pareto front $|\mathcal{I}^*|$ is very large, and $|\mathcal{I}^*| \approx K$, then a simple uniform random sampler can perform similarly with an MO-MAB algorithm. Alternative best arm identification algorithms [19] are faster because they eliminate half of the suboptimal arms instead of a single one. How efficient are these alternatives for Pareto MAB-problems is subject to future work.

IV. $\boldsymbol{\epsilon}$-APPROXIMATE PARETO FRONT IDENTIFICATION

The intuition for this algorithm comes from multi-objective optimization. The goal is to deterministically delete dominated D-rectangles rather than dominated arms. Unlike *Pareto front identification*, this hybrid algorithm does not assume that the cardinality of the Pareto front is known beforehand. This allows for a tighter upper confidence bound since the performance depends on the number of non-dominated D-rectangles, and this is a parameter that can be tuned by the user.

D-objective hypergrid. The D-objective reward space is organized into a hypergrid of D-rectangles as follows. Let m^j and M^j be the upper and the lower limit, respectively, for the j-th objective. Then the interval $[m^j, M^j]$ is divided in $\lceil \frac{M^j - m^j}{\epsilon} \rceil$ disjoint subintervals $[m^j + o^j \epsilon^j, m^j + (o^j + 1)\epsilon^j[$ of length ϵ^j, where o^j ranges from 0 till $\frac{M^j - m^j}{\epsilon} - 1$ and o^j is used as index of the corresponding interval. A D-rectangle is the Cartesian product $\prod_{j=1}^{D} [m^j + o^j \epsilon, m^j + (o^j + 1)\epsilon[$ of D such subintervals, one per objective j and it is indexed by $\boldsymbol{o} = (o^1, o^2, \cdots, o^D)$. For simplicity, we assume that $\mathbf{m} = (0, \dots, 0)$ and $\mathbf{M} = (1, \dots, 1)$. This way we get a hypergrid of in total $\prod_{j=1}^{D} \lceil \frac{M^j - m^j}{\epsilon} \rceil$ disjoint D-rectangles. The D-rectangle $\prod_{j=1}^{D} [m^j + o^j \epsilon, m^j + (o^j + 1)\epsilon[$ corresponding with index $\boldsymbol{o}$ is denoted as $R_{\boldsymbol{o}}$.

```
Require: $\delta > 0$
  $\mathcal{C} \to \emptyset$
  for all arms $i \in I$ do
    Play it for $n_i = \frac{4}{\boldsymbol{\xi}_i^2} \ln \frac{2DK}{\delta}$ times
    Let $\tilde{\boldsymbol{\mu}}_i$ be its estimated mean reward vector
    Assign $i$ it to the $D$-rectangle $R_i$ that contains $\tilde{\boldsymbol{\mu}}_i$
    If $R_i \notin \mathcal{C}$, then $\mathcal{C} \to \mathcal{C} \cup \{R_i\}$
  end for
  Remove the dominated $D$-rectangles from $\mathcal{C}$
  return The set of non-dominated $D$-rectangles $\mathcal{C}$
```

Algorithm 3: $\boldsymbol{\epsilon}$-approximate Pareto Front Identification ($\boldsymbol{\epsilon}$-PFI)

Pareto dominance for D-rectangles. Let $\boldsymbol{o}_1 = (o_1^1, \dots, o_1^D)$ and $\boldsymbol{o}_2 = (o_2^1, \dots, o_2^D)$ be indices. Then D-rectangle $R_{\boldsymbol{o}_1}$ is *non-dominated* by D-rectangle $R_{\boldsymbol{o}_2}$ iff there exists an objective j for which $o_1^j \geq o_2^j$. And D-rectangle $R_{\boldsymbol{o}_1}$ is *dominated* by D-rectangle $R_{\boldsymbol{o}_2}$ iff for all objectives j we have $o_1^j < o_2^j$. The Pareto front of non-dominated D-rectangles is denoted with $\mathcal{I}^*$. Note that by definition the non-dominated relation between D-rectangles is more relaxed than the non-dominated relation between arms.

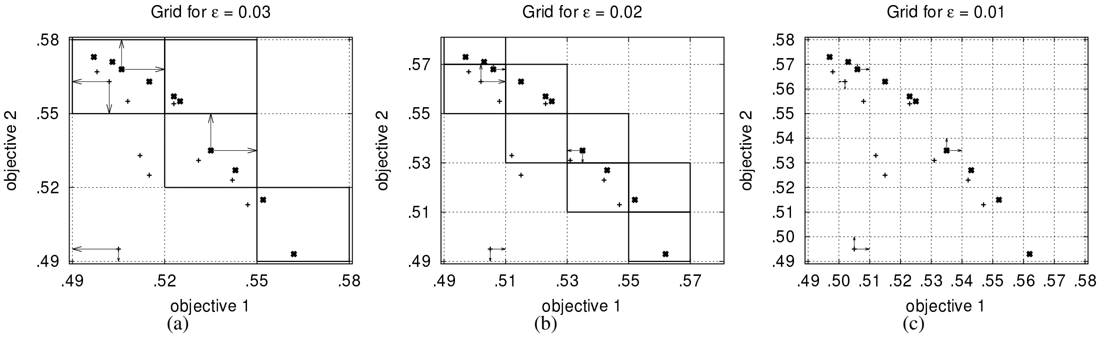
The algorithm. Each arm is individually assigned to a D-rectangle. One arm is assigned to a single D-rectangle but one D-rectangle can contain several arms. The estimation $\tilde{\boldsymbol{\mu}}_i$, based on a number of samples, of the true mean reward vector $\boldsymbol{\mu}_i$ is used to assign arm i to a D-rectangle. We want to bound the probability that arm i is assigned to a wrong D-rectangle, i.e. a D-rectangle that does not contain its estimated mean reward vector $\tilde{\boldsymbol{\mu}}_i$. The confidence in assignment will be higher when the estimation $\tilde{\boldsymbol{\mu}}_i$ is near the center of a D-rectangle while it will be lower when it is close to the border.

The pseudo-code for the $\boldsymbol{\epsilon}$-PFI algorithm is given in Algorithm 3. Let $\delta > 0$ be the confidence for this algorithm. Let $\mathcal{C}$ be the set of D-rectangles containing at least one arm. Initially, $\mathcal{C}$ is the empty set. Consider and arm i and the D-rectangle R_i corresponding to the Cartesian product $\prod_{j=1}^{D} [i^j \epsilon^j, (i^j + 1)\epsilon^j[$ and containing the estimated mean of the arm i, $\tilde{\boldsymbol{\mu}}_i \in R_i$. We consider that an arm i that is at a certain distance $\xi_i \cdot \mathbf{1}$ from the bounds of the D-rectangle R_i can be assigned to R_i. Thus, $\exists j$, such that $\tilde{\mu}_i^j > i^j \epsilon^j + \xi_i$ and $\tilde{\mu}_i^j < (i^j + 1)\epsilon^j - \xi$.

Each arm is pulled for $n_i = \frac{4}{\xi_i^2} \ln \frac{2DK}{\delta}$ times and the resulting estimation $\tilde{\boldsymbol{\mu}}_i$ of its true mean reward vector $\boldsymbol{\mu}_i$ is used to assign the arm to one D-rectangle. Let $\xi_i > 0$ be the accuracy for the arm i, where ξ_i is the minimal distance to the corresponding D-rectangle. Thus, if the D-rectangle is defined by the coordinates $[k^j \epsilon^j, (k^j + 1)\epsilon^j[$, then

$$\xi_i \leq \min_{1 \leq j \leq D} \min(\tilde{\mu}_i^j - k^j \epsilon^j, \tilde{\mu}_i^j - (k^j + 1)\epsilon^j)$$

Finally, the D-rectangles that are dominated in all objectives by at least one other D-rectangle from $\mathcal{C}$ are deleted. The computational complexity of determining the relation between the non-empty D-rectangles is the same as that of sorting, i.e. $O(c \log c)$ where $c = |C|$. One arm is selected at



(a) (b) (c)

Fig. 1. Twenty reward vectors of which ten belong to a convex Pareto front in a grid for different values of ϵ: a) $\epsilon = 0.03$, b) $\epsilon = 0.02$, and c) $\epsilon = 0.01$.

random from each non-dominated D-rectangle. These arms can be at most a distance ϵ away from the Pareto front.

Example. Let us consider again the mean reward vectors from Section II. In Figure 1 a), the size of the hypergrid segment is $\epsilon = 0.03$. There are five non-empty non-dominated D-rectangles and two non-empty dominated D-rectangles. Note that there is one non-dominated D-rectangle that does not contain any non-dominated arm. Three of the non-dominated D-rectangles contain two Pareto optimal arms and one D-rectangle contains four Pareto optimal arms.

In Figure 1 b), the size of the hypergrid segment is $\epsilon = 0.02$. Now, there are eight non-empty non-dominated D-rectangles and two non-empty dominated D-rectangles. Again, one non-dominated D-rectangle does contain only one dominated arm but it is now non-dominated since it is non-empty and it *could have* contained a Pareto optimal arm. There is one non-dominated D-rectangle containing three Pareto optimal arms, one D-rectangle containing two Pareto optimal arms and the rest of the non-dominated D-rectangles contain each one Pareto optimal arm.

In Figure 1 c), the size of the hypergrid segment is $\epsilon = 0.01$. There are ten non-empty non-dominated D-rectangles and five non-empty dominated D-rectangles. Again, one non-dominated D-rectangle does contain only a dominated arm. There is one non-dominated D-rectangle containing two Pareto optimal arms, and the rest of the non-dominated D-rectangles contain each one Pareto optimal arm.

Theoretical analysis. ϵ-PFI, cf. Algorithm 3, is a variant of the naive $(\epsilon, \delta)-$PAC algorithm [20]. We want to bound the probability that the true expected reward vector for an arm i, $\boldsymbol{\mu}_i$, does not belong to the same D-rectangle like its estimated reward vector, $\tilde{\boldsymbol{\mu}}_i$, is bounded with accuracy ξ_i and the confidence interval δ.

We want to bound the probability of the event $|\mu_i^j - \tilde{\mu}_i^j| > \xi_i$ for all objectives j. If the arm i is close to the center of the D-rectangle R_i, then it can be assigned easier to that D-rectangle that an arm that is closer to the border of the D-rectangle. The following theorem states that ϵ-PFI, cf. Algorithm 3, is a naive (ϵ, δ)-PAC algorithm. The proof follows directly from the prove of Theorem 1.

Theorem 3: Let ϵ-PFI, cf. Algorithm 3, having $K > 1$ arms with arbitrary reward distributions $\mathbf{P}_1, \dots \mathbf{P}_K$ with support in $[0,1]^D$.

Then ϵ-PFI, cf. Algorithm 3, is a naive (ϵ, δ)-PAC with sample complexity $\sum_{i=1}^{K} \frac{N}{\xi_i^2} \log \frac{2DK}{\delta}$.

We will always consider the case where $\boldsymbol{\xi} \ll \boldsymbol{\epsilon}$. Then, the mean reward vector $\tilde{\boldsymbol{\mu}}_i$ and its estimation $\tilde{\boldsymbol{\mu}}_i$ are in the same D-rectangle with high probability.

Note that the derived bound does not depend on the number of optimal arms but only on the number of dimensions D. As a result, an arm should be pulled much longer in ϵ-PFI, cf. Algorithm 3, than in the standard naive (ϵ, δ)-PAC algorithm introduced in [20].

The closer an arm is to the border of a D-rectangle, the larger the probability to select the wrong D-rectangle. The Pareto partial order between arms in the same D-rectangle is independent from the process of assigning these arms to the right D-rectangle.

$$N_P = \sum_{i \in I} n_i = \sum_{i \in I} \frac{4}{{\xi_i}^2} \ln \frac{2DK}{\delta} \tag{4}$$

Note that the total budget N_P increases when δ decreases, i.e. when the confidence that a certain arm is assigned to the right D-rectangle increases.

V. Fixed budget Pareto Subfront Identification

In this section, we combine the two approaches presented in the last two sections: the ϵ-Parento Front Identification algorithm that selects a representative subset of non-dominated D-rectangles with the best arm identification algorithm to identify a non-dominated arm in each of these non-dominated D-rectangles. We call the resulting algorithm fixed budget *Pareto Subfront Identification* or the ϵ-PSI algorithm. We show that the upper confidence bound for this hybrid algorithm depends only on the number of dimensions D and the number of trials N but it does not depend on the number of arms K as was the case for *Pareto Front Identification.*

The pseudo-code for ϵ-PSI is given in Algorithm 4. First, each arm is assigned to a D-rectangle according to ϵ-Parento Front Identification algorithm, cf. Algorithm 3. The ϵ-PSI algorithm deletes the dominated D-rectangles together

```
Require: $\delta > 0$
  $\mathcal{C} \leftarrow$ $\epsilon$-PFI is the not everywhere dominated set of $D$-rectangles
  $\tilde{\mathcal{I}}^* \leftarrow \emptyset$
  for all $D$-rectangles $c = 1, \dots, |\mathcal{C}|$ do
    $i^* \leftarrow$ ParetoSuccessiveReject$(1)$
    $\tilde{\mathcal{I}}^* \leftarrow \tilde{\mathcal{I}}^* \cup \{i^*\}$
    Delete dominated arms in $\tilde{\mathcal{I}}^*$
  end for
  return The empirical Pareto front $\tilde{\mathcal{I}}^*$
```

Algorithm 4: Fixed budget Pareto Subfront Identification (ϵ-PSI)

with the arms inside these D-rectangles since these arms are dominated. The resulting list of non-empty non-dominated D-rectangles is $\mathcal{C}$. Second, the algorithm selects a single representative optimal arm in each of the non-dominated D-rectangles using a best arm identification algorithm. Thus, there is no need to know in advance the number of non-dominated arms in each D-rectangle. For each D-rectangle in $\mathcal{C}$, we run PFI, cf. Algorithm 2, on the assumption that only one non-dominated arm represents that D-rectangle. In case of arms of equal quality, we remove this time the arm with the lowest confidence. This way, Pareto optimal arms near the center of a D-rectangle will more often be selected than Pareto optimal arms near the border. This is important since arms near the border have a higher probability of being assigned to a wrong D-rectangle. The selected optimal arm in each D-rectangle is then added to the list of non-dominated arms $\tilde{\mathcal{I}}^*$. If there are dominated arms in $\tilde{\mathcal{I}}^*$, they will be deleted. The algorithm returns a list of arms that are uniformly spread and approximate well the original Pareto front $\tilde{\mathcal{I}}^*$.

The ϵ-PSI-algorithm, cf. Algorithm 4, has an upper confidence bound on the probability that *all* Pareto optimal arms in a D-rectangle are deleted and that the selected arm is suboptimal instead.

Corollary 1: The probability of wrongly deleting *all* Pareto optimal arms in a D-rectangle after N plays is at most

$$e_N \le |\mathcal{C}| \cdot D \binom{K-1}{2} \cdot e^{-\frac{(N/|\mathcal{C}|-K)}{H_2(\log(K)+1)}} \qquad (5)$$

This results immediately from the proof of Theorem 2 if we take into account that a single best arm is identified in each of the in total $|\mathcal{C}|$ D-rectangles.

Note that the error of wrongly deleting *all* Pareto optimal arms in the ϵ-PSI algorithm, cf. Equation 5, is smaller than the error of deleting *any* Pareto optimal arm in ϵ-PFI, cf. Equation 3. Furthermore, the error in the ϵ-PSI algorithm depends on the number $|\mathcal{C}|$ of non-dominated D-rectangles. This in turn depends on the accuracy $\epsilon > 0$ that can be controlled by the user.

In the limit, when all K arms are in a few D-rectangles, the hypergrid is considered to be too coarse. If $|\mathcal{C}| \approx 1$, then the ϵ-PSI algorithm selects only one non-dominated arm and it is equivalent to a best arm identification algorithm. If (almost) all non-empty non-dominated D-rectangles have only one arm, then the hypergrid is considered too fine. When $|\mathcal{C}| \approx |\mathcal{I}^*|$, ϵ-PFI has a performance similar to PFI, cf. Algorithm 2.

To compare Pareto MAB-algorithms, we need to allocate the same computational budget to each algorithm. All algorithms have a fixed budget of N pulls. ϵ-PFI spends N_P pulls to assign arms to D-rectangles, and ϵ-PSI algorithm plays N_I times to identify *one* Pareto optimal arm in each D-rectangle. Thus, the total budget

$$N = N_P + N_I \qquad (6)$$

and the number of pulls for ϵ-PSI algorithm, cf. Algorithm 4, is

$$N_I = N - N_P = N - \frac{4K}{\epsilon^2} \ln \frac{K}{\delta} \qquad (7)$$

This can be tuned also with different values for ϵ and δ. In general, the amount of pulls needed for each of the two algorithms depends on the characteristics of the environment. When there are many Pareto optimal arms, we want to spend less arm pulls on assigning arms to the hypergrid than on removing arms from a D-rectangle. In case of many suboptimal arms, it is important to correctly assign arms to the corresponding D-rectangle.

Example. Let us consider again the mean reward vectors from Section II. In Figure 1 a), where $\epsilon = 0.03$, we select in each non-dominated D-rectangle a Pareto optimal arm. By design, the arm selected is both Pareto optimal and has the largest confidence from all Pareto optimal arms in that D-rectangle. In this example, the resulting Pareto optimal set of arms has four arms that are part of the Pareto front $\mathcal{I}^*$.

In Figure 1 b), where $\epsilon = 0.02$, only five Pareto optimal arms are selected however. That is because the Pareto optimal arm with the largest confidence in a D-rectangle might be dominated given the Pareto front $\mathcal{I}^*$.

Remark. It is interesting to note that the ϵ-PSI algorithm can be parallelized. The arms are assigned independently to the hypergrid and the D-rectangles are also processed independently. This is an important side effect that allows to exploit the performance of supercomputers.

VI. Experiments

In this section, we consider two bi-objective Pareto MAB-problems where the rewards vectors are drawn according to a multivariate Bernoulli distribution with diagonal covariance matrix. And we want to identify the Pareto front $\mathcal{I}^*$.

The goal is to compare experimentally the behavior of the four introduced Pareto MAB-algorithms and this for different parameter settings in order to monitor the performance and the robustness of the proposed algorithms.

The four compared Pareto MAB-algorithms are:

PUCB1 The Pareto UCB1 algorithm introduced in [2];
PFI The Pareto successive rejects algorithm introduced in Section III;
ϵ-PSI the ϵ-Pareto subfront identification algorithm introduced in Section IV;

Hoef As a base-line, we consider an adaptation of the Hoeffding race algorithm [21] to the multi-objective spaces. All arms are pulled equally often. The arms with non-dominated empirical mean vectors are selected.

Each algorithm run 100 times with a fixed budged of $N = 10^5$, and a small confidence interval $\delta = 0.01$. The number of pulls, i.e. the budget, for assigning arms to D-rectangles is set to $N_P \approx 32 \cdot 10^3$, and the budget for identifying a Pareto optimal arm in each D-rectangle is $N_I \approx 67 \cdot 10^3$, almost the double of N_P. To calculate N_I and N_P, we have considered Equation 7 and 6. These parameters are set based on trial runs. Automatically tuning these parameters remains to be done as future work.

We consider the convex problem introduced in Section II. In Figure 2, we compare the performance of ϵ-PSI for different values of the coarseness of the grid $\epsilon \in \{0.005, 0.01, 0.02, 0.03\}$ in terms of number of pulls of Pareto optimal arms. The finest hypergrid has the best performance in terms of the Pareto projection regret. The coarsest hypergrid has the worst performance since it identifies only three to four of the Pareto optimal arms. It is interesting to note that the best performing ϵ-PSI algorithms have the finest hypergrids for which either only one or at most two Pareto optimal arm are located in a D-rectangle. This means that the ϵ-PSI algorithm is efficient because it deletes dominated D-rectangles rather than dominated arms. The disadvantage of using a finer hypergrid is the larger number of arms pulls required to assign an arm to a D-rectangle for a given confidence value $\delta = 0.1$.

In Figure 3, we compare the performance of the four Pareto MAB-algorithms listed above. In Figure 3 a), the algorithm with the smallest Pareto projection regret is ϵ-PSI. PUCB1 has the second smallest Pareto projection regret, followed by PFI. Note the peaks in the beginning of the runs of ϵ-PSI. These peaks corresponds to the peaks in the beginning of the runs for the same algorithm for the criteria percentage of Pareto optimal arm pulls' shown in Figure 2. This is because the ϵ-PSI algorithm has two phases. In the first phase all the arms including the Pareto optimal arms are pulled for a fixed budget in order to be assigned to the right D-hypercube.

All three algorithms perform better than the baseline algorithm Hoef meaning that the designed features are meaningful. The best performing algorithm is ϵ-PSI, due to a good exploitation of the entire Pareto front. The size of the hypergrid has a slight influence on the performance of ϵ-PSI. The good performance of ϵ-PSI is explained by the fair and intensive use of Pareto optimal arms. The performance of the convex Pareto front vary more with the choice of ϵ than the performance of the non-convex Pareto front. The second best algorithm is PUCB1 and PFI is third.

VII. Conclusions

We combined techniques from both the multi-armed bandit problem and multi-objective optimization to design multiple arm identification algorithms. The Pareto front identification algorithm deletes suboptimal arms using a fixed budget multiple arm identification algorithm, but the size of the Pareto front needs to be known beforehand.

We incorporated techniques from multi-objective evolutionary algorithms like Pareto ϵ-dominance to deal with large Pareto fronts. The ϵ-approximate Pareto Front Identification algorithm is a fix confidence multiple arms identification algorithm that assign first all the arms to D-rectangles of a hypergrid. The performance of this algorithm is independent of the size of the Pareto front, instead it depends on the coarseness of the hypergrid that is tuned by a user.

The last proposed algorithm combines the fixed budget and the fixed confidence Pareto front identification algorithms. The ϵ-approximate Pareto Subfront Identification algorithm identifies a proper subset of the Pareto front that is uniformly spread over that front.

Finally, we compared the algorithms introduced in this paper on an artificially two bi-objective generated problem with convex Pareto front. ϵ-approximate Pareto Subfront Identification was the most efficient algorithm, as was to be expected from the theoretical analysis.

VIII. Acknowledgements

Madalina M. Drugan was supported by ITEA2 project M2Mgrid.

References

[1] 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.

[2] M. Drugan and A. Nowe, "Designing multi-objective multi-armed bandits: an analysis," in *Proc of International Joint Conference of Neural Networks (IJCNN)*, 2013.

[3] D. Lizotte, M. Bowling, and S. Murphy, "Efficient reinforcement learning with multiple reward functions for randomized clinical trial analysis," in *Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML)*, 2010.

[4] M. Wiering and E. de Jong, "Computing optimal stationary policies for multi-objective markov decision processes," in *Proc of Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, 2007, pp. 158–165.

[5] K. van Moffaert, M. Drugan, and A. Nowe, "Hypervolume-based multi-objective reinforcement learning," in *Proc of Evolutionary Multi-objective Optimization (EMO)*. Springer, 2013.

[6] W. Wang and M. Sebag, "Multi-objective Monte Carlo tree search," in *Asian conference on Machine Learning*, 2012, pp. 1–16.

[7] D. Roijers, S. Whiteson, and F. Oliehoek, "Computing convex coverage sets for multi-objective coordination graphs," in *ADT 2013: Proceedings of the Third International Conference on Algorithmic Decision Theory*, November 2013.

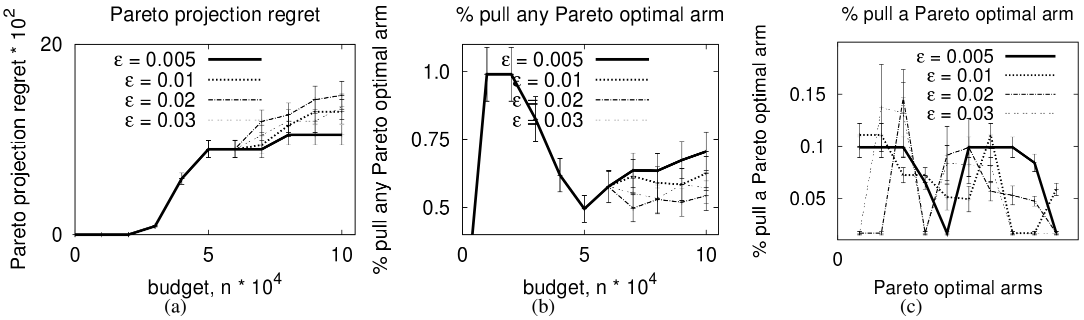
[8] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE T. on Evol. Comput.*, vol. 7, pp. 117–132, 2003.

[9] E. Kaufmann and S. Kalyanakrishnan, "Information complexity in bandit subset selection," in *Proc of COLT*, 2013, pp. 228–251.

[10] V. Gabillon, M. Ghavamzadeh, and A. Lazaric, "Best arm identification: A unified approach to fixed budget and fixed confidence," in *NIPS*, 2012, pp. 3221–3229.

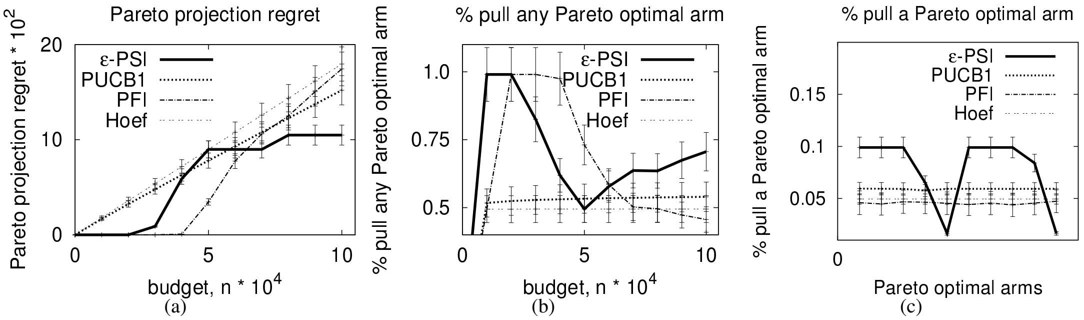
[11] J.-Y. Audibert, S. Bubeck, and R. Munos, "Best arm identification in multi-armed bandits," in *Proc of Conference on Learning Theory (COLT'10)*, 2010.

[12] M. Drugan and A. Nowe, "Scalarization based pareto optimal set of arms identification algorithms," in *Proc of International Joint Conference of Neural Networks (IJCNN)*, 2014.



(a) (b) (c)

Fig. 2. The performance of the four ϵ-PSI algorithms on the convex bi-objective problem for different values of ϵ, where $\epsilon \in \{0.005, 0.01, 0.02, 0.03\}$.



(a) (b) (c)

Fig. 3. Performance of the four Pareto MAB-algorithms on the bi-objective convex problem: i) Pareto UCB1 (PUCB1), ii) Pareto successive rejects that is an instance of Pareto front identification (PFI), iii) ϵ-Pareto subfront identification algorithm (ϵ-PSI) for $\epsilon = 0.005$, and iv) the Hoeffding race algorithm (Hoef).

[13] M. M. Drugan, A. Nowe, and B. Manderick, “Pareto upper confidence bounds algorithms: an empirical study,” in *Proc of IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2014.

[14] P. Auer and R. Ortner, “UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem,” *Periodica Mathematica Hungarica*, vol. 61, no. 1-2, pp. 55–65, 2010.

[15] S. Bubeck, T. Wang, and N. Viswanathan, “Multiple identifications in multi-armed bandits,” in *Proc of International Conference on Machine Learning (ICML’13)*, 2013.

[16] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[17] S. Q. Yahyaa, M. M. Drugan, and B. Manderick, “Exploration vs exploitation in the multi-objective multi-armed bandit problem,” in *Proc of International Joint Conference on Neural Networks (IJCNN)*, 2014.

[18] T. Voß, H. Trautmann, and C. Igel, “New uncertainty handling strategies in multi-objective evolutionary optimization,” in *PPSN*, 2010, pp. 260–269.

[19] Z. Karnin, T. Koren, and O. Somekh, “Almost optimal exploration in multi-armed bandits,” in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, vol. 28. JMLR Workshop and Conference Proceedings, 2013, pp. 1238–1246.

[20] E. Even-Dar, S. Mannor, and Y. Mansour, “Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems,” *The Journal of Machine Learning Research*, vol. 7, pp. 1079–1105, 2006.

[21] O. Maron and A. Moore, “Hoeffding races: Accelerating model selection search for classification and function approximation,” in *Advances in Neural Information Processing Systems*, vol. 6. Morgan Kaufmann, 1994, pp. 59–66.