Scalarization based Pareto optimal set of arms identification algorithms

Madalina M. Drugan[1], Ann Nowe[1]

Artificial Intelligence Lab of Computer Science Department,
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium
mdrugan@vub.ac.be, anowe@vub.ac.be

Résumé : *Multi-objective multi-armed bandits* (MOMAB) is an extension of the multi-armed bandits framework that considers reward vectors instead of scalar reward values. Scalarization functions transform the reward vectors into reward values in order to use the standard multi-armed bandits (MAB) algorithms. However for many applications it is not obvious to come up with a good scalarization set and therefore there is needed to develop MAB that discover the whole Pareto set of arms. Our approach to this multi-objective MAB problem is two folded : i) identify the set of Pareto optimal arms and ii) identify the minimum subset of scalarization functions that optimize the set of Pareto optimal arms. We experimentally compare the proposed MOMAB algorithms on a multi-objective Bernoulli problem.

1 Introduction

Multi-armed bandits (MAB) is a machine learning paradigm used to study and analyze resource allocation in stochastic and noisy environments. We consider the definition for a multi-armed bandit algorithm where only one arm is played at a time and there are fixed equal range stochastic reward vectors for each arm. The multi-armed bandit framework considers multi-objective, or multi dimensional, rewards and imports techniques from multi-objective optimization into the multi-armed bandits algorithms. This framework is called *multi-objective multi-armed bandits* (MOMABs) Drugan & Nowe (2013).

Let's consider an initial set of arms $\mathcal{I}$ with cardinality K, where and $K \geq 2$ and the vector reward space is defined as a D-dimensional hypercube $[0, 1]^D$. When arm i is played a random vector of rewards is received, one component per objective, when one of the arms is pulled. The random vectors have a stationary distribution with support in the D-dimensional hypercube $[0, 1]^D$ but the vector of true expected rewards $\boldsymbol{\mu}_i = (\mu_i^1, \ldots, \mu_i^D)$, where D is the number of objectives, is unknown. At time steps t_1,t_2,. . ., the corresponding reward vectors $\mathbf{X}_i^{t_1}$, $\mathbf{X}_i^{t_2}$, . . . are independently and identically distributed according to an unknown law with unknown expectation vector $\boldsymbol{\mu}_i = (\mu_i^1, \ldots, \mu_i^D)$. Reward values obtained from different arms are also assumed to be independent. A scalarized MOMAB algorithm chooses the next machine to play based on the sequence of past plays and obtained reward values.

MOMABs lead to important differences compared to the standard MABs. A reward vector can optimize one objective and be sub-optimal in the other objectives, leading with many vector rewards qualitatively incomparable. Thus, there could be several arms considered to be the best according to their reward vectors.

A common approach is to transform the multi-objective environment into a single-objective environment using *scalarization functions*. Single-objective environments, in general, results in a single optimum, therefore we need a set of scalarization functions to generate a variety of elements belonging to the Pareto optimal set.

In this paper, we consider three types of scalarization functions that weight the values of the reward vector : i) linear, ii) Chebyshev and iii) L_p norm scalarization functions. *We consider that each set of weights generate a scalarization function.* Linear scalarization is a popular choice in designing multi-objective reinforcement learning algorithms Lizotte *et al.* (2010); van Moffaert *et al.* (2013); Wang & Sebag (2012) as well as multi-armed bandits Drugan & Nowe (2013).

Main contributions. In this paper, we introduce several scalarization based MOMAB algorithms that identify the entire Pareto set of optimal arms. In Section 3, we report a first contribution. We propose an algorithm that considers a fixed set of scalarization functions and that deletes suboptimal arms, i.e. arms that are not optimum for any scalarization function, in order to identify the Pareto optimal set of arms. The proposed algorithm is an extension to scalarized reward vectors of the successive rejects algorithm Audibert *et al.* (2010). We name this algorithm the *scalarized successive rejects* algorithm and we assume that each scalarization function identifies a single arm as optimal.

In Section 4, we generalize the scalarized successive rejects algorithm to allow that p arms are identified from the Pareto optimal front when the L_p norm is used as the scalarization function. The proposed algorithm is an extension to the reward vectors of multi-arm identification algorithm Bubeck *et al.* (2013). Because of the usage of the L_p scalarization function, this algorithm can identify arms on a non-convex Pareto front when $p > 2$. In Section 5, we compare runs of the proposed multi-objective algorithms on the bi-objective stochastic environment generated with the control problem mentioned above.

A common approach in optimization is to select a number of linear scalarization functions that are uniform randomly spread in the weighted space. However, most of the multi-objective environments do not have the isomorphism property meaning that the scalarized values are not uniform randomly spread in the objective space. A solution to this problem could be an increasing the number of scalarization functions in order to identify all arms in the Pareto front. However, in stochastic environments optimizing in a scalarized direction is more costly than in deterministic environments because arms are pulled for each scalarization in order to compute confidence intervals to decide which is the best arm for a particular scalarization function. Note that it is not straightforward to reuse arm pulls from one scalarization function to another since the ranking and the values of arms in each of them can be different. Therefore, using the right amount of scalarization functions is from a computational point of view important in a multi-armed bandits setting. This bring us to the third contribution of this paper, which is to find a minimum set of scalarization functions such that all the Pareto optimal arms are identified.

In Section 6, for each Pareto optimal arm, the linear scalarization function with the smallest variance around the mean of the arm is selected as the best scalarization function.

Section 7 concludes the paper.

2 Background : the partial scalarized order relationship

There are mainly two partial order relationships used in multi-objective optimization. Pareto partial order relationship Zitzler *et al.* (2003) is the natural order for these environments allowing to order the reward vectors directly in the multi-objective reward space. *Scalarization functions* Eichfelder (2008) transform the reward vectors into scalar rewards using e.g. linear or non-linear weighted sums.

A reward vector μ is considered better than, or *dominating*, another reward vector ν, $\nu \prec \mu$, iff 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 *non-dominated* by ν, $\nu \not\succ \mu$, iff there exists at least one dimension j for which $\nu^j < \mu^j$.

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* $\mathcal{I}^*$ be the set of arms whose reward vectors belong to $\mathcal{O}^*$. Then, $\forall \mu^*_\ell \in \mathcal{O}^*$, and $\forall \mu_o$, we have $\mu^*_\ell \not\prec \mu_o$.

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

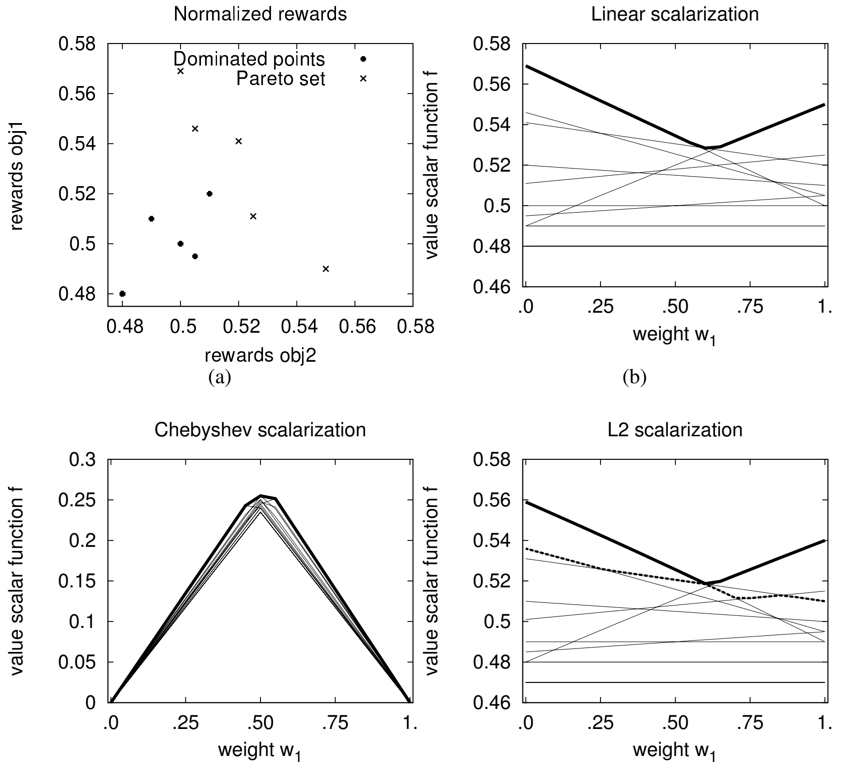
Optimization and learning algorithms using Pareto partial order relationships Wiering & de Jong (2007); Drugan & Nowe (2013) often have computational problems because of the large sets of best arms that need to be explored and stored.

2.1 Scalarized multi-objective multi-armed bandits

Due to its simplicity, *the linear scalarization* is the most popular scalarization function in designing multi-objective optimization problems. This maps the problem into a single-objective environment whose optimization results in a single optimum. Therefore, we need a set of scalarization functions to generate a variety of elements belonging to the Pareto optimal set. It weighs each value of the reward vector and the result is the sum of these weighted values.

The linear scalarized reward is

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



(a)

(b)

(c)

(d)

FIGURE 1 – a) Ten reward vectors, where five are optimal : μ_1^*, μ_2^*, μ_3^*, μ_4^* and μ_5^*. Compute the maximal reward values for b) the linear scalarization function, and c) the Chebyshev scalarization function. d) Compute the best (the straight upper line) and the second best (the doted line) reward values for the L_2 scalarization function.

where $(w^1, \ldots, w^D)$ is a predefined weight vector 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, even by varying the weights.

L_p **norm** is a scalarization function that generalizes the linear scalarized function, and it is defined as

$$f_p(\mu_i) = \sqrt[p]{\sum_{j=1}^{D} \omega^j \cdot (\mu_i^j - z^j)^p}$$

where p is a positive integer, $\mathbf{z} = (z^1, \ldots, z^D)$ is a reference point chosen such 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 sampled so far minus a small positive value, $\epsilon^j > 0$,

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

An L_p scalarization function can find p solutions on a non-convex Pareto front, being a polynomial function with p solutions.

The Chebyshev scalarization has the advantage that in certain conditions it can find all the points in a non-convex Pareto set. The *Chebyshev scalarization reward* is

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

The Chebyshev metric can be considered as a particular case of the L_p norm with $p \leftarrow \infty$.

2.2 Comparing the partial order relationships

Consider the 10 bi-dimensional reward vectors from Figure 1 a) and the Pareto partial order. Let there be five optimal reward vectors, $\mu_1^* = (0.55, 0.49)$, $\mu_2^* = (0.53, 0.511)$, $\mu_3^* = (0.52, 0.541)$, $\mu_4^* = (0.505, 0.563)$ and $\mu_5^* = (0.5, 0.569)$ in $\mathcal{O}^*$. Consider five different suboptimal reward vectors, $\mu_6 = (0.51, 0.52)$, $\mu_7 = (0.5, 0.5)$, $\mu_8 = (0.505, 0.495)$, $\mu_9 = (0.5, 0.5)$, and $\mu_{10} = (0.49, 0.51)$. Note that the suboptimal reward μ_6 is non-dominated by three optimal reward vectors from $\mathcal{O}^*$, μ_1^*, μ_4^* and μ_5^*, but it is dominated by μ_2^* and μ_3^*. μ_7 and μ_8 are dominated by all the other reward vectors.

In Figure 1 b), for any set of weights, the optimum reward vectors are μ_1^*, μ_3^*, and μ_5^*, but never μ_2^* or μ_4^* that are non-convex points. $\mathcal{O}_L^*$ therefore does not include μ_2^* and μ_4^* from the non-convex Pareto optimal reward set $\mathcal{O}^*$.

In Figure 1 c), the optimum reward vectors are μ_1^*, μ_3^*, μ_4^* and μ_5^*, but never μ_2^*. Thus, $\mathcal{O}_C^*$ with a fixed reference point does not include all from the arms in $\mathcal{O}^*$.

In Figure 1 d), the best and the second best reward values for L_p scalarization function are plotted. The best arms are only two, which are the extremities μ_1^* and μ_5^*. The second best arms are the rest of the three Pareto optimal arms μ_2^*, μ_3^* and μ_4^*. Note that all the Pareto points are identified if we consider the two lines combined, a single line identifies a complementary subset of the Pareto front.

2.3 The regret metric

The *scalarized regret* for a particular scalarized function f^j and for the arm i is

$$\Delta_i^j =^{def} f^j(\mu^*) - f^j(\mu_i) \tag{2}$$

where the optimum reward value μ^* is the reward for which the function f (linear, L_p, or Chebyshev) attains its maximum value

$$f^j(\mu^*) = argmax_{k \in \mathcal{I}} f^j(\mu_k)$$

It is straightforward to show that the maximum value for any set of weights is a Pareto optimal arm. We denote the Pareto optimal set of arms identifiable by : i) the linear scalarization with $\mathcal{I}_L^*$, ii) the L_p-norm scalarization with $\mathcal{I}_p^*$ and iii) the Chebyshev scalarization with $\mathcal{I}_C^*$. The corresponding set of Pareto optimal reward sets are : i) $\mathcal{O}_L^*$ for linear scalarization, ii) $\mathcal{O}_p^*$ for L_p norm, and iii) $\mathcal{O}_C^*$ for Chebyshev scalarization.

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

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. Therefore, we need other mechanisms to ensure the global performance of the multi-armed bandits that use scalarization functions.

Empirical estimates. The expected value of each arm is computed by averaging the samples observed over time. Let $T_i(n)$ be the number of times that arm i is pulled during a total number of pulls of n. The mean of arm i is estimated as $\widehat{\mu}_i(n) = \sum_{s=1}^{T_i(n)} \mathbf{X}_k(s)/T_k(n)$, is the s-th sample observed for arm i.

3 Scalarization based Pareto optimal set of arms identification algorithms

In this section, we introduce a scalarization based multi-objective MAB algorithm with the goal of identifying the Pareto optimal set of arms. This class of algorithm deletes suboptimal arms in such a way that the probability to delete the optimal arm is bounded. There are two classes of algorithms : i) with fixed budget, and ii) with fixed confidence. We focus on the fixed budged best arm identification algorithms, more exactly the successive reject algorithm Audibert *et al.* (2010).

3.1 The scalarized successive rejects algorithm (sSR)

We introduce the *scalarized successive rejects* (sSR) policy which is an extension of the successive rejects algorithm for single objective such that for each set of weights, or scalarization functions, we identify the best arm that optimizes that scalarization function. Intuitively, this algorithm is a set of parallel best arm

Algorithm 1 The scalarized successive rejects sSR($\mathcal{I}$, S, n)

```
Require: A set of arms $\mathcal{I}$
Require: A set of scalarization functions $S$
Require: The budget $n * |S|$
  $\mathcal{I}^*_S \leftarrow \emptyset$, $A^j_1 \leftarrow \mathcal{I}, \forall f^j \in S$
  Let $n_0 = 0$, and $n_k = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K+1-k} \right\rceil$
  for all $f^j \in S$ do
    for all rounds $k = 1, 2, \dots, K-1$ do
      Pull arm $i \in A^j_k$ for $n_k - n_{k-1}$ rounds
      Let $i \leftarrow argmin_{i \in A^j_k} f^j(\widehat{\mu}_i)$ the arm to dismiss
      $A^j_{k+1} \leftarrow A^j_k \setminus argmin_{i \in A^j_k} f^j(\widehat{\mu}_i)$
    end for
    if $i^* \notin \mathcal{I}^*_S$, where $i^* \in A^j_{K-1}$ then
      $\mathcal{I}^*_S \leftarrow \mathcal{I}^*_S \cup \{i^*\}$
    end if
  end for
  return $\mathcal{I}^*_S$
```

identification algorithms Audibert *et al.* (2010), where each scalarization function identifies a single Pareto optimal arm, not necessary distinct. The pseudo-code for our scalarized successive rejects algorithm is given in Algorithm 1.

Let's consider a set of fixed scalarization functions $S = \{f^1, f^2, \dots, f^{|S|}\}$ and to each scalarization function f^j let a set of active arms A^j_1 be associated. At initialization, each active arms set contains the whole set of arms, $A^j_1 \leftarrow \mathcal{I}$. The algorithm considers $K-1$ phases of increasing length. In each phase, we pull an arm iff it is *active* for that scalarization function. Thus, in the k-th phase, all the active arms, i.e. not dismissed yet, are equally pulled for $n_k - n_{k-1}$ times. The length of a phase is chosen to obtain a logarithmic convergence rate, and as for the single objective successive reject algorithm. We have

$$n_k = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K+1-k} \right\rceil$$

where, by definition, we have $\overline{\log}(K) = \frac{1}{2} + \sum_{j=2}^{K} \frac{1}{j}$. For each scalarization function $f^j \in S$, we successively remove the arm with the minimum scalarized reward from the list of active arms. The algorithm stops when, for each scalarization function, all the arms but the best arm are deleted from the list of active arms. The surviving arm is recommended as the best arms for the function f^j and, in the same time a Pareto optimal arm in $\mathcal{I}^*$.

The worst arm is pulled $n_1 = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K} \right\rceil$ times, the second worst arms is pulled for $n_2 = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K-1} \right\rceil$ times, and the number of times an arm is pulled increases with its quality. The best arm is pulled $n_{K-1} = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{2} \right\rceil$ times. This algorithm has a fixed budget $n \cdot |S|$ which it is not exceed because

$$\begin{aligned} \sum_{k=1}^{K-1} n_k + n_{K-1} &\leq K + \frac{n-K}{\overline{\log}(K)+1} \left(\frac{1}{2} + \sum_{k=1}^{K-1} \frac{1}{K+1-k} \right) \\ &= n \end{aligned}$$

Let's consider the complexity measure

$$H_2 = \max_{i \in \mathcal{I}} \max_{f^j \in S} \frac{i}{(\Delta^j_i)^2} \tag{3}$$

where Δ^j_i is defined as in Equation 2. H_2 quantifies the hardness of the scalarized successive rejects algorithm (SSR).

Algorithm 2 Efficient Scalarized Successive Rejects esSR($\mathcal{I}$,S,n)

```
Require: A set of arms $\mathcal{I}$
Require: A set of scalarization functions $S$
Require: The budget $n \cdot |S|$
  $\mathcal{I}_S^* \leftarrow \emptyset$, $A_1^j \leftarrow \mathcal{I}$, $\forall f^j \in S$
  Let $n_0 = 0$, and $n_k = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K+1-k} \right\rceil$
  for all rounds $k = 1, 2, \ldots, K-1$ do
    for all arm $i$ for which $\exists f^j$, $i \in A_k^j$ do
      Select arm $i$ for $n_k - n_{k-1}$ rounds
    end for
    for all $f^j \in S$ do
      Let $i \leftarrow argmin_{i \in A_k^j} f^j(\widehat{\mu}_i)$ the arm to dismiss
      Set $A_{k+1}^j \leftarrow A_k^j \setminus \{i\}$
    end for
  end for
  if $i^* \notin \mathcal{I}_S^*$, where $i^* \in A_{K-1}^j$ then
    $\mathcal{I}_S^* \leftarrow \mathcal{I}_S^* \cup \{i^*\}$
  end if
  return $\mathcal{I}_S^*$
```

Theorem 1
The probability of wrongly deleting an optimal arm given a scalarization function, $f^j \in S$, after any number of n plays is at most

$$e_n \leq \binom{K-1}{2} \cdot e^{-\frac{(n \cdot |S| - K)}{H_2 \overline{\log}(K)}}$$

Proof 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

$$\widehat{\mathbf{X}}_{i,n} = 1/n \cdot \sum_{t=1}^{n} \mathbf{X}_{i,t} \quad \leftarrow \quad \forall j, \;\; \widehat{X}_{i,n}^j = 1/n \cdot \sum_{t=1}^{n} X_{i,t}^j$$

The proof follows directly from the proof in Audibert et al. *(2010) when the reward vectors $\mathbf{X}_{i,t}$ are transformed into reward values using the scalarization function f^j. Then, $Y_{i,t} \leftarrow f^j(\mathbf{X}_{i,t})$. We obtain the inequality from the theorem by using the union bound over all the scalarization functions.*

Remark 3.1 The output of the previous algorithm will include a Pareto optimal arm for each scalarization function. But Algorithm 1 finds only the Pareto optimal arms that are optimal given a set of scalarization functions S. Thus, not all the arms in $\mathcal{I}^*$ are also included in $\mathcal{I}_S^*$. Furthermore, Algorithm 1 does not guaranty that the optimal solutions of the scalarization functions are different. As it could be more scalarization functions that correspond to same Pareto optimal arm. Moreover, it is well known that linear scalarization functions cannot find all the points on a non-convex Pareto front.

Remark 3.2 Note that, for a set of weights in a linear scalarization function, there is a single optimal arm. This is an important assumption in Algorithm 1 which deletes all but one arm for a given set of weight vectors. In case we would use the Chebyshev function, there could be more than one optimal arm for a set of weights. We could be to consider more than one optimal arm for a single set of weight vectors when the Chebyshev function is used.

Remark 3.3 Algorithm 1 is a naive variant of the scalarized SR because it runs for each set of weights in S the single objective successive rejects algorithm from Audibert *et al.* (2010). To improve its performance, samples of the active arms are used by all the scalarization functions. The pseudo-code is given in Algorithm 2. Unlike in Algorithm 1, all the scalarization functions have common rounds. This means that an arm active for *at least a scalarization function* in the round k is pulled only for $n_k - n_{k-1}$ times. This is the main difference between Algorithm 1 and Algorithm 2.

Note that the complexity measure from Equation 3 is given by the worst scalarized value of an arm when all scalarization functions are considered. Similarly, the budget of Algorithm 1 is also given by the worst case, i.e. all arms are active for at least a scalarization function.

In the best case where all functions agree on the arm ranking, Algorithm 2 is $|S|$ times more efficient then its sequential variant Algorithm 1 where the arms are independently pulled for each scalarization function. In the worst case, where the scalarization functions do not agree at all on the ranking of the arms, Algorithm 2 is as efficient as Algorithm 1 because all the arms needs to be played through all the rounds.

4 The scalarized multiple arm successive accepts and rejects algorithm

The algorithm presented in the previous section can use any type of scalarization function. When linear scalarization is used, a single arm will be optimal, w.r.t. this scalarization function, so it make sense to delete all but one arm in $K-1$ rounds. In case the L_p scalarization is used, p arms can be optimal. Therefore, we also present an algorithm which allows to identify p best arms per scalarization. The proposed algorithm is an extension of multiple best arm identification Bubeck *et al.* (2013) to the scalarized multi-objective MAB algorithms. The main difference between the scalarized SR from Algorithm 2 and the scalarized successive accepts and rejects is the latter accepts arms when they are identified within a certain confidence as being one of the p best arms.

The goal of this algorithm is to minimize the probability of misclassification, i.e. of deleting one of the p-optimal arms. Let $\{J_1,\ldots,J_p\}$ be the set of p optimal arms. We want to upper bound the probability

$$e_n = \mathbb{P}(\{J_1,\ldots,J_p\} \neq \{1,\ldots,p\})$$

Similar to Bubeck *et al.* (2013), we define two performance measures. The *reward gap* between mean reward of two arms given a scalarization function f^j is defined as

$$\Delta_{(i)j}^{<p>} = \begin{cases} f^j(\widehat{\mu}_i) - f^j(\widehat{\mu}_{p+1}) & \text{if } i \leq p \\ f^j(\widehat{\mu}_p) - f^j(\widehat{\mu}_i) & \text{if } i > p \end{cases} \tag{4}$$

where the notation $(i) \in \{1,\ldots,K\}$ is defined such that $\forall j$

$$\Delta_{(1)j}^{<p>} \leq \ldots \leq \Delta_{(K)j}^{<p>}$$

The *complexity measure* that quantifies the hardness of the scalarized successive accepts and rejects algorithm is defined as

$$H_2^{<p>} = \max_{i\in\mathcal{I}} \max_{f^j\in S} \frac{i}{(\Delta_{(i)j}^{<p>})^2} \tag{5}$$

Given $X_i^0,\ldots,X_i^t$ samples from the distribution of the arm i, the *empirical* reward gap when a learning algorithm using the scalarization f^j is

$$\widehat{\Delta}_{(i)j}^{<p(k)>} = \begin{cases} f^j(\widehat{\mu}_i) - f^j(\widehat{\mu}_{p+1}) & \text{if } i \leq p(k) \\ f^j(\widehat{\mu}_{p(k)}) - f^j(\widehat{\mu}_i) & \text{if } i \geq p(k)+1 \end{cases} \tag{6}$$

where the notation $(i) \in \mathcal{I}$ is defined such that

$$f^j(\widehat{\mu}_1) \leq \ldots \leq f^j(\widehat{\mu}_K)$$

and $p(k+1) \leftarrow p(k)-1$. Relying on the estimates of the reward gap from Equation 6, we decide whenever to accept or reject an arm during the k-th phase.

The pseudo-code for the *scalarized successive accepts and rejects* is given in Algorithm 3. For each scalarization function f^j, we consider a set of active arms in the round k, A_k^j, and a set of accepted arms J_p^j. At initialization, the set of active arms coincide with the whole set of arms, $A_1^j \leftarrow \mathcal{I}$. The set of accepted arms is empty, $p^j(1) \leftarrow p$. Like in Algorithm 2, for each arm that is active for at least a scalarization function is pulled for $n_k - n_{k-1}$ rounds, where n_k is chosen to ensure a logarithmic regret.

For each function $f^j \in S$, compute the maximum empirical reward gap $\widehat{\Delta}_{(i)j}^{<p(k)>}$ for this round. For the best $p(k)$ arms, this is the distance with the best $p(k)+1$ th empirical mean. For the other active arms in

Algorithm 3 The scalarized successive accepts and rejects algorithm *sSAR*(p, $\mathcal{I}$,S,n)

```
Require: A set of arms $\mathcal{I}$
Require: A set of scalarization functions $S$
Require: The budget $n \cdot |S|$
Require: The number of arms returned $p$
  $\mathcal{I}_S^* \leftarrow \emptyset$, $p^j(1) \leftarrow p$, $A_1^j \leftarrow \mathcal{I}, \forall f^j \in S$
  Let $n_0 = 0$, and $n_k = \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K+1-k} \right\rceil$
  for all rounds $k = 1, 2, \ldots, K-1$ do
    for all arm $i$ for which $\exists f^j$, $i \in A_k^j$ do
      Select arm $i$ for $n_k - n_{k-1}$ rounds
    end for
    for all $f^j \in S$ do
      Let $i \leftarrow argmax_{i \in A_k^j} \widehat{\Delta}_{(i)j}^{<p^j(k)>}$ the arm to dismiss
      $A_{k+1}^j \leftarrow A_k^j \setminus \{i\}$
      if the arm $i$ among the best $p^j(k)$ arms for the scalarization $f^j$, $f^j(\widehat{\mu}_i) > f^j(\widehat{\mu}_{p^j(k)+1})$ then
        Accept arm $i$, $J_{p-p^j(k)}^j \leftarrow i$
        Set $p^j(k+1) \leftarrow p^j(k) - 1$ the remaining number of arms to be accepted
      end if
    end for
  end for
  return $\mathcal{I}_S^*$ the set of non-dominated arms, $\mathcal{I}_S^* \leftarrow \cup_{1 \leq i \leq p} \cup_{j \in S} J_i^j$
```

A_k^j, $\widehat{\Delta}_{(i)j}^{<p(k)>}$ represents the distance with the best $p(k)$ th arm. The arm corresponding with the largest gap is deleted. In addition to Algorithm 2, the rejected arm in round k is accepted as among the top p arms when its fitness function is larger than the top $p^j(k)+1$ arms. The algorithm stops after $K-1$ rounds. The output for Algorithm 3 is a list with the set of arms accepted by at least a scalarization function.

Theorem 2
The probability of error with Algorithm 3 is at most

$$e_n \leq 2K^2 \cdot e^{-\frac{(n-K)}{8H_2^{<p>}\overline{\log}(K)}}$$

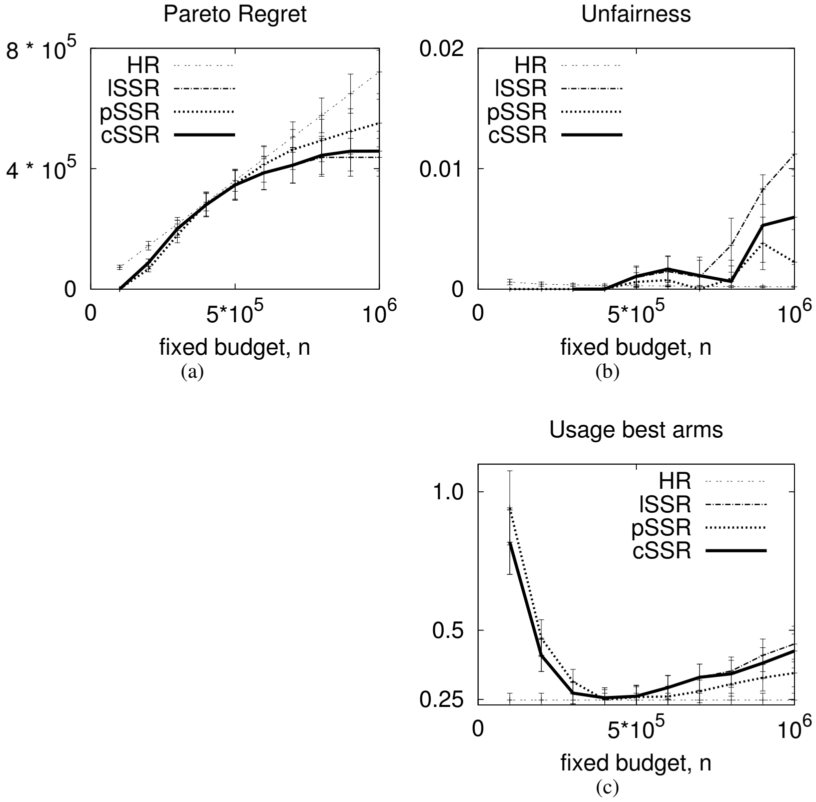
Proof 2
The proof follows directly from the proof in Bubeck et al. *(2013) when the reward vectors $\mathbf{X}_{i,t}$ are transformed into reward values using the scalarization function f^j. Then, $Y_{i,t} \leftarrow f^j(\mathbf{X}_{i,t})$. We obtain the inequality from the theorem by using the union bound over all the scalarization functions.*

Remark 4.1 Like in the single objective case, Algorithm 3 is a generalization of Algorithm 1. On the other hand, Algorithm 2 can be considered with all type of scalarization functions, whereas Algorithm 3 is designed for the L_p scalarization functions. Note that if p is larger than the size of the Pareto optimal set of arms, arms that do not belong to $\mathcal{I}^*$ will be returned as optimal. A solution is to verify that all arms in $\mathcal{I}_A^*$ are non-dominated and to delete the non-dominated arms.

5 Numerical example

We compare the performance of three scalarization multi-objective MAB algorithms proposed here with an adaptation to the multi-objective spaces of the Hoeffding race algorithm Maron & Moore (1994) as a baseline algorithm. In this version of the Hoeffding race algorithm Maron & Moore (1994), all the arms are pulled equally often and, in the end, the arms with the highest empirical scalarized mean is chosen.

As a test problem, we consider multi-objective Bernoulli distributions on the means from the example in Section 2.2. We assume sets of weight vectors of size 11 and that the corresponding weight vectors are



(a)

(b)

(c)

FIGURE 2 – The Pareto regret, b) the unfairness, and c) the usage of the optimal arms for four algorithms, e.g. Hoeffding race (HR), linear scalarized successive rejects (lSSR), Chebyshev scalarized successive rejects (cSSR), and L_2 scalarized successive rejects.

uniformly randomly spread in the bi-objective space $[0,1] \times [0,1]$. Then $w^1 \in \{0.0, 0.1, 0.2, \ldots, 1.0\}$ in the first objective, and $1 - w^1$ in the second objective.

Each algorithm is run 30 times. For the Chebyshev scalarization function, we uniform randomly generate the parameters $\epsilon^j \propto [0, 0.1]$ from Equation 1 for the reference point $\mathbf{z}$.

The scalarized successive rejects algorithm. In the first experiment, we compare the performance of Algorithm 1 for the three scalarization functions presented in Section 2.1 with the Hoeffding race algorithm. The test algorithms are

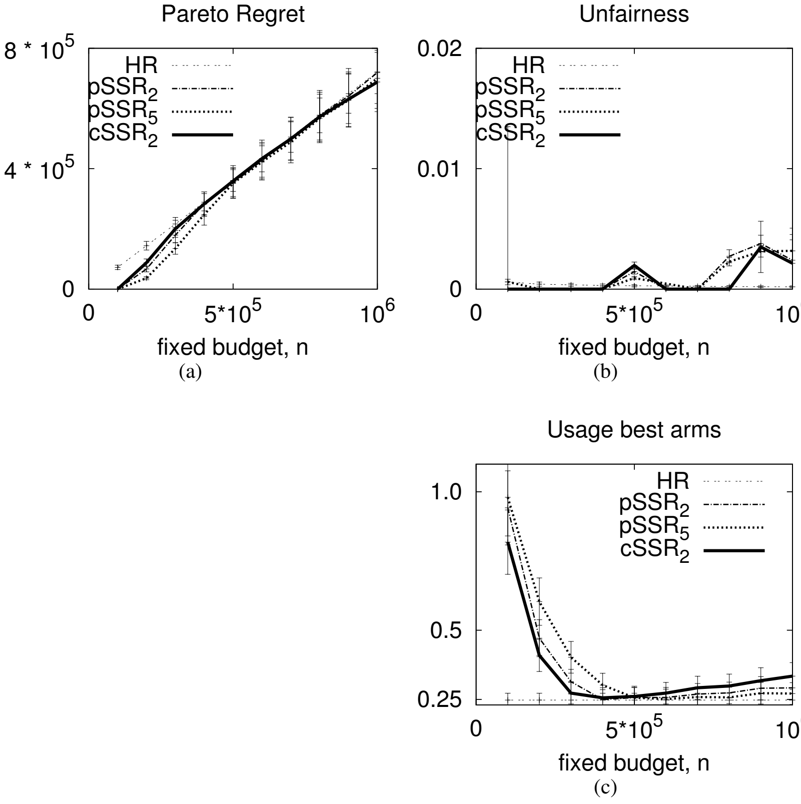
1. HR : The Hoeffding race algorithm
2. lSSR : The linear scalarized successive rejects algorithm
3. cSSR : The Chebyshev scalarized successive rejects algorithm
4. pSSR : The L_2 scalarized successive rejects algorithm, where $p = 2$

We evaluate the performance of the four multi-objective MAB algorithms, lSSR, cSSR, pSSR and HR, based on three performance measures as indicated in Figure 2.

Figure 2 a) gives the instantaneously Pareto regret metric for each of the three algorithms. According with this measure, cSSR and pSSR are the best performing algorithms and the Hoeffding race is the worst algorithm. lSSR is using linear scalarization and has a Pareto regret that is larger than of the same algorithm using cSSR.

Figure 2 b) shows the unfairness in using the optimal arms, where we have used the definition of unfairness as a regret metric from Drugan & Nowe (2013) briefly presented in the following. The *unfairness* of a multi-objective multi-armed bandits algorithm is defined as the *variance* of the arms in $\mathcal{I}^*$,

$$\frac{1}{N \cdot |\mathcal{I}^*|} \cdot \sum_{i \in \mathcal{I}^*} (T_i^*(N) - I\!E[T^*(N)])^2$$



(a)

(b)

(c)

FIGURE 3 – The Pareto regret, b) the fairness, and c) the usage of the optimal arms for four algorithms, e.g. Hoeffding race (HR), Chebyshev scalarized successive rejects and accepts (cSSAR), L_2 scalarized successive rejects and accepts (pSSAR), and L_5 scalarized successive rejects and accepts (pSSAR_5).

where $T_i^*(N)$ is the number of times an optimal arm i is pulled, and $\mathbb{E}[T^*(N)]$ the expected number of times optimal arms are selected. For a perfectly fair usage of optimal arms, we have that the unfairness is about 0. When a multi-objective strategy uses only samples a subset of $\mathcal{I}^*$, then the variance is large. In other words, a well performing multi-objective MAB algorithm has low risk of unevenly choosing between optimal arms.

According with this measure, the best performing algorithm is cSSR and pSSR where the optimal arms are fairly played. Note that this is an indication that the error probability to eliminate a (Pareto) optimal arms in a preliminary phase (i.e., the performance metric for the Pareto successive rejects algorithm) is small. The worst algorithm is the Hoeffding race algorithm.

Figure 2 c) shows the percentage of times all the Pareto optimal arms from $\mathcal{I}^*$ are used. The worst algorithm is again Hoeffding race that uses all the arms equal time, and linear scalarization has a slightly better performance than the Hoeffding race. The other two algorithms perform similarly, being the best algorithms.

The scalarized successive rejects and accepts algorithm. In the second experiment, we compare the performance of Algorithm 3 for the Chebysev and the L_p scalarization functions presented in Section 2.1 with the Hoeffding race algorithm.

The test algorithms are

1. HR : The Hoeffding race algorithm
2. cSSAR : The Chebyshev scalarized successive rejects and accepts algorithm, where $p = 2$
3. pSSAR : The L_2 scalarized successive rejects and accepts algorithm, where $p = 2$
4. pSSAR_5 : The L_5 scalarized successive rejects and accepts algorithm, where $p = 5$

The performance of the scalarized successive rejects and accepts algorithm is similar. That might be because the low number of Pareto optimal arms in the given example.

Discussion. An important note is that the performance of the algorithms does not change with increasing number of scalarization functions. Note that all the scalarized successive rejects algorithms perform better than the base line algorithm, i.e. Hoeffding race. The linear scalarization successive rejects algorithm is the second worst algorithm because, as suggested in Section 2.2, the linear scalarization function cannot identify all the arms that are in the non-convex Pareto set of arms. The successive rejects algorithm using Chebysev and L_p scalarization functions have similar performance and are the best algorithms.

Overall, the Pareto regret of the successive rejects and accepts algorithms is lower than the Pareto regret of the algorithms using successive rejects algorithms. This could be because of the usage of the gap function to evaluate when to delete an arm that can be quite small and thus prone to errors. In contrast, the fairness of the successive rejects and accepts algorithms is lower than of the successive rejects algorithms, which shows that indeed the Pareto optimal arms are the arms identified as optimal by the successive rejects and accepts algorithms.

6 The minimal set of scalarization functions identification algorithm

The most popular method in multi-objective optimization Drugan (2013b,a) is to consider a large *fixed* number of scalarization functions uniform randomly spread into the space of weight vectors. In this section, we propose a variant of the naive Probable Approximately Correct (PAC) algorithm applied to identify the best scalarization function for a Pareto optimal arm. We assume here that the set of best arms was a-priori identified, for example, with Algorithm 1.

6.1 The performance metric

Let's consider the *mean-variance risk measure* Sani *et al.* (2013) from portfolio analysis as a performance measure of the goodness of a scalarization function indicating a particular Pareto optimal arm. Let i be a Pareto optimal arm, $i \in \mathcal{I}^*$, and let the mean variance of a scalarization function f^j with respect to the optimum arm i be

$$\phi_{ji} = f^j(\sigma_i^2) - \rho \cdot f^j(\mu_i)$$

where σ_i^2 is the variance on the optimal arm i and μ_i is its mean. The best scalarization function for a Pareto optimal arm is the scalarization function with the minimal mean-variance for the Pareto optimal arm.

Given $X_i^0, \dots, X_i^t$ samples from the distribution of the arm i, the *empirical* mean-variance when a learning algorithm using the scalarization f^j over t rounds is considered is

$$\widehat{\phi}_{ij}^t = f((\widehat{\sigma}_{ij}^t)^2) - \rho \cdot f(\widehat{\mu}_{ij}^t)$$

where $\widehat{\mu}_{ij}^t = \frac{1}{t}\sum_{s=1}^t X_{ij}^s$ and $(\widehat{\sigma}_{ij}^t)^2 = \frac{1}{t}\sum_{s=1}^t (X_{ij}^s - \widehat{\mu}_{ij}^t)^2$.

The regret of a suboptimal scalarization function f^j is

$$\Delta_{ij} = \widehat{\phi}_{ij}^t - \widehat{\phi}_{i*} \tag{7}$$

where $\widehat{\phi}_{i*}$ is the mean-variance of the best scalarization function for the Pareto optimal arm i, and ρ is a positive number.

Each optimal arm identified, e.g. with Algorithm 1, is pulled a number of times and then the scalarization function with the smallest variance around the mean is chosen. Let S_i be the set of scalarization functions that optimize the same arm i, where $i \in \mathcal{I}^*$. More formally, the best scalarization function is defined by

$$f_i^* \leftarrow argmin_{j \in S_i} \widehat{\phi}_{ij}$$

For a perfect fair usage of scalarization functions, we have that $\widehat{\phi}_{i*} \leftarrow 0$ which means that the minimal set of scalarization functions identifies the whole set of Pareto optimal arms. S^* is the set of optimal scalarization functions.

6.2 The algorithm

The pseudo-code for this algorithm is given in Algorithm 4. For each optimal arm $i \in \mathcal{I}^*$, we pull i for $\ell_i = \frac{4\cdot(5+\rho)^2}{\varepsilon^2} \ln \frac{2|\mathcal{I}^*|\cdot n}{\delta}$ rounds. Afterwards, we select the scalarization function that minimizes the mean variance function.

Algorithm 4 Minimal set of Scalarization Identification algorithm $\mathsf{MSI}(\mathcal{I}^*)$

```
Require: $\forall i \in \mathcal{I}^*$, $S_i$ a set of scalarization functions with the same optimal arm $i$, where $S = \cup_{i \in \mathcal{I}^*} S_i$
Require: $\varepsilon > 0, \delta > 0$
Require: $n$ fixed budget
  $S^* \leftarrow \emptyset$
  for all $i \in \mathcal{I}^*$ do
    Pull arm $i$ for $\ell_i = \frac{4 \cdot (5+\rho)^2}{\varepsilon^2} \ln \frac{2|\mathcal{I}^*| \cdot n}{\delta}$ times
    Select the scalarization function $f_i^* \leftarrow argmin_{j \in S_i} \widehat{\phi}_{ji}$
    $S^* \leftarrow S^* \cup \{f_i^*\}$
  end for
  return $S^*$
```

Theorem 3
Algorithm 4 is a naive (ε, δ)-PAC algorithm with arm sample complexity of $\mathcal{O}(\frac{(5+\rho)^2 \cdot n}{\varepsilon^2} \log \frac{2|\mathcal{I}^*| \cdot n}{\delta})$.

Proof 3
Let f^j be a scalarization function for which $\phi_{ij} - \varepsilon > \phi_{i*}$. Thus, the probability of erroneously choosing a suboptimal scalarization function is

$$\begin{aligned} I\!\!P(\widehat{\phi}_{ij} < \widehat{\phi}_{i*}) &\leq I\!\!P(\widehat{\phi}_{ij} < \phi_{ji} - \varepsilon/2 \quad \text{or} \quad \widehat{\phi}_{i*} > \phi_{i*} + \varepsilon/2) \\ &\leq I\!\!P(\widehat{\phi}_{ij} < \phi_{ij} - \varepsilon/2) + I\!\!P(\widehat{\phi}_{i*} > \phi_{i*} + \varepsilon/2) \\ &\leq 2(5+\rho) \cdot e^{-\varepsilon^2 \ell_i / 2} = \frac{\delta}{|\mathcal{I}^*| \cdot n} \end{aligned}$$

where the last inequality uses the Hoeffding inequality for mean-variance measure

$$I\!\!P(\widehat{\phi}_{ij} < \phi_{ij} - \varepsilon/2) \leq (5+\rho) \cdot e^{-\varepsilon^2 \ell_i / 2}$$

$$I\!\!P(\widehat{\phi}_{i*} > \phi_{i*} + \varepsilon/2) \leq (5+\rho) \cdot e^{-\varepsilon^2 \ell_i / 2}$$

Choosing $\ell_i = \frac{2 \cdot (5+\rho)}{\varepsilon^2} \ln \frac{2n}{|\mathcal{I}^*| \cdot \delta}$ assures that $I\!\!P(\widehat{\phi}_{ij} < \widehat{\phi}_{i*}) \leq \frac{\delta}{|\mathcal{I}^*| \cdot n}$. Summing over all the arms, we have that the error probability is at most δ.

Remark 6.1 For an efficient scalarization-based algorithm we need a one to one correspondence between optimal arms in the Pareto set and a set of weight vectors that defines a scalarization function. Algorithm 4 deletes all but one scalarization functions that optimize the same Pareto optimal arm.

Remark 6.2 There could be Pareto optimal arms that do not correspond to any scalarization function even when the Pareto optimal is convex because, for example : i) an insufficient large number of scalarization function considered, and ii) the non-uniform spread of optimal arms on the Pareto front. It should be noted that the method of identifying the minimal set of weights from Lizzote et al Lizotte *et al.* (2010) is limited to Markov Decision Process (MDP) settings, and therefore not applicable to the MOMABs we consider here.

Remark 6.3 Algorithm 4 assumes that there is a single optimal arm for each tuple of weights as for the linear scalarization. For the Chebyshev and L_p scalarization function this assumption does not hold since it could be more than one optimal solution that is identified by a single scalarization function.

An adapted version for the Chebyshev and L_p scalarization deletes a scalarization function only if it is not optimal for *any* optimal arm. Another solution would be to use another variant of the algorithm from Drugan & Nowe (2013) where the performance of Chebyshev scalarization function is measured also by the number of optimal arms identified as such.

7 Conclusions

In this paper, we consider that the scalarized based multi-objective multi-armed bandits algorithms have to solve two problems : i) to identify the set of Pareto optimal arms, and ii) to identify the minimal set of scalarization functions that can identify the set of Pareto optimal arms. We introduced three variants of the scalarized based multi-objective multi-armed bandits algorithms. We propose a scalarized based variant of successive reject algorithm that can be used with any type of scalarization function. The scalarized successive rejects algorithm shares the sample of the pulled arms between all the scalarized functions, increasing the efficiency of the algorithm. A generalization of the successive accepts and rejects in the scalarized context designed for L_p norm like scalarization functions where there are p best arms selected for each scalarization function. The scalarized successive accepts and rejects algorithm can identify Pareto optimal arms on a non-convex front. We show some experimental results where the efficiency of these algorithms are tested and compared. The third variant of algorithms assumes a large set of scalarization functions uniform randomly spread in the weight vector space identifies the minimal set of scalarization functions.

Références

AUDIBERT J.-Y., BUBECK S. & MUNOS R. (2010). Best arm identification in multi-armed bandits. In *Proc of Conference on Learning Theory (COLT'10)*.

BUBECK S., WANG T. & VISWANATHAN N. (2013). Multiple identifications in multi-armed bandits. In *Proc of International Conference on Machine Learning (ICML'13)*.

DRUGAN M. & NOWE A. (2013). Designing multi-objective multi-armed bandits : a study. In *Proc of International Joint Conference of Neural Networks (IJCNN)*.

DRUGAN M. M. (2013a). Cartesian products of scalarization functions for many-objective qap instances with correlated flow matrices. In *Genetic and Evolutionary Computation Conference (GECCO'13)* : ACM.

DRUGAN M. M. (2013b). Sets of interacting scalarization functions in local search for multi-objective combinatorial optimization problems. In *Proc of IEEE Symposium Series on Computational Intelligence* : IEEE.

EICHFELDER G. (2008). *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer.

LIZOTTE D., BOWLING M. & MURPHY S. (2010). Efficient reinforcement learning with multiple reward functions for randomized clinical trial analysis. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML)*.

MARON O. & MOORE A. (1994). Hoeffding races : Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems*, volume 6, p. 59–66 : Morgan Kaufmann.

SANI A., LAZARIC A. & MUNOS R. (2013). Risk-aversion in multi-armed bandits. *CoRR*, **abs/1301.1936**.

VAN MOFFAERT K., DRUGAN M. & NOWE A. (2013). Hypervolume-based multi-objective reinforcement learning. In *Proc of Evolutionary Multi-objective Optimization (EMO)* : Springer.

WANG W. & SEBAG M. (2012). Multi-objective Monte Carlo tree search. In *Asian conference on Machine Learning*, p. 1–16.

WIERING M. & DE JONG E. (2007). Computing optimal stationary policies for multi-objective markov decision processes. In *Proc of Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, p. 158–165 : IEEE.

ZITZLER E., THIELE L., LAUMANNS M., FONSECA C. M. & DA FONSECA V. (2003). Performance assessment of multiobjective optimizers : An analysis and review. *IEEE T. on Evol. Comput.*, **7**, 117–132.