ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200121

Regret-Based Elicitation for Solving Multi-Objective Knapsack Problems with Rank-Dependent Aggregators

Nawal Benabbou and **Cassandre Leroy** and **Thibaut Lust**[1]

Abstract. In this paper, we consider multi-objective knapsack problems where the decision maker's preferences are represented by a non-linear aggregation function whose parameters are initially not known. More precisely, we focus on rank-dependent aggregators such as ordered weighted averages (OWA) and Choquet integrals which are non-linear scalarizing functions that assign weights to ranks rather than to objectives in the aggregation process, so as to control the importance attached to the bottom performance or to any other order statistics; for instance, an OWA operator with decreasing weights helps promoting balanced solutions while ensuring overall efficiency. In this setting, we propose new interactive heuristic methods consisting in combining regret-based preference elicitation and heuristic search so as to quickly focus the search on the most promising solutions. For OWA operators and Choquet integrals, the proposed methods run in polynomial time and are guaranteed to generate no more than a polynomial number of queries. We perform numerical tests comparing our methods to different interactive solving methods in order to show the practical efficiency of our approach in terms of number of queries, computation time and gap to optimality.

1 INTRODUCTION

Multi-Objective Combinatorial Optimization (MOCO) is concerned with problems involving alternatives that are evaluated with respect to several conflicting viewpoints/objectives to be optimized simultaneously (e.g., cost, time, profit), with various applications in Artificial Intelligence (AI) such as investments planning, resource allocation, group configuration, recommendation systems and electronic commerce. To model the preferences of the Decision Maker (DM) over feasible solutions, an aggregation function is often used to synthesize the different evaluations into scalar values representing the overall performances of alternatives. Recently there has been a growing interest in rank-dependent aggregators such as Choquet integrals [26], which enable to model complex decision behaviors by considering possible (positive and negative) synergies between the objectives. They also make it possible to model preferences for "well-balanced" efficient solutions that cannot be obtained by optimizing a weighted sum. While most contributions focus on the elaboration of algorithms allowing the fast determination of the optimal solutions given a preference model, a key question is how to assess the preference parameters of such sophisticated models to best fit the DM's preferences. In this paper, we address both issues simultaneously by studying the integration of preference elicitation into the resolution of MOCO problems with rank-dependent aggregators.

[1] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France, email: firstname.lastname@lip6.fr

Preference elicitation on combinatorial domains is an active topic studied within the AI community that has motivated several contributions in various contexts, e.g., in multi-objective state space search [7], in multi-agents systems [8, 13], in stable matching problems [20], in constraint satisfaction problems [14], in Markov Decision Processes [38, 45] and in MOCO problems [10]. Here we focus on regret-based incremental preference elicitation, an approach recently designed by AI researchers, as it was proved to be very effective in practice [6, 14, 44]. The idea is to use the minimax regret decision criterion to select informative preference queries at each step of the elicitation process, so as to progressively reduce the set of admissible parameters until a robust recommendation can be made. This elicitation approach has been mainly studied with linear models and/or explicit sets of solutions. In this paper, we show how to efficiently extend this approach to MOCO problems with imprecise preferences represented by rank-dependent aggregation functions; the proposed approach is illustrated on the multi-objective knapsack problem.

The paper is organized as follows: first we introduce the formal framework (Section 2) and present some related works (Section 3). Then, we introduce two regret-based interactive procedures for the multi-objective knapsack problem with unknown preferences: a local search method and a greedy algorithm. We prove that both procedures run in polynomial time and ask no more than a polynomial number of preference queries to the DM (Section 3). Finally, we provide numerical tests that demonstrate the practical efficiency of the proposed methods, comparing our results to those obtained by two existing methods: a two-phase method and the state-of-the-art regret-based solving method for general MOCO problems (Section 4).

2 THE GENERAL FRAMEWORK

2.1 The Multi-Objective Knapsack Problem

We consider a decision problem where the DM has to select a set of items (e.g., candidates, projects, objects) in a set $\mathcal{I} = \{1, \dots, p\}$. In this problem, any subset of items can be represented by a solution vector $x = (x_1, \dots, x_p) \in \{0,1\}^p$ where $x_i = 1$ if item i is in the subset and $x_i = 0$ otherwise. In the standard knapsack problem, the set $\mathcal{X}$ of admissible solutions is defined by linear constraints of the form $\sum_{i=1}^{p} w_i x_i \leq W$ where $w_i \geq 0$ is the weight of item i and $W \geq 0$ is the maximum total weight. For instance, in committee election problems, cardinality constraints are usually imposed to control the size of the elected committee or to ensure gender parity. For the simplicity of the presentation, we will only consider one cardinality constraint in this paper but the proposed algorithm can be easily adapted to problems with additional feasibility constraints.

In the multi-objective knapsack problem, we consider a finite set of objectives $y_j : \mathcal{I} \to \mathbb{R}_+$, with $j \in N = \{1, \dots, n\}$, to be

maximized. Thus any item $i \in \mathcal{I}$ is associated with a performance vector $y(i) = (y_1(i), \ldots, y_n(i)) \in \mathbb{R}^n_+$ where $y_j(i)$ is the evaluation of item i on the j-th objective; for instance, $y_j(i)$ can be the utility of voter j for candidate i in committee election problems. Any solution $x \in \mathcal{X}$ is then characterized by a performance vector $y(x) = (y_1(x), \ldots, y_j(x))$ defined by $y_j(x) = \sum_{i=1}^{p} x_i y_j(i)$ for all $j \in N = \{1, \ldots, n\}$. In this framework, solutions are usually compared through their images in the objective space using the *Pareto dominance* relation: we say that performance vector $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ *Pareto dominates* performance vector $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ (denoted by $a \succ_P b$) if and only if $a_j \geq b_j$ for all $j \in N$, with at least one strict inequality. Then a solution is said to be Pareto-optimal if and only its performance vector is not Pareto-dominated by that of any other admissible solution. A standard way to model preferences over vectors consists in using an aggregation function $f_\omega : \mathbb{R}^n_+ \to \mathbb{R}_+$ that maps performance vectors to scalar values: given two solutions x, x' representing two subsets of items, x is at least as good as x' for the DM whenever $f_\omega(y(x)) \geq f_\omega(y(x'))$. In this paper, we focus on rank-dependent aggregation functions.

2.2 Rank-Dependent Aggregation Functions

Rank-dependent aggregation functions are scalarizing functions that sort the performance values by increasing order before mapping the performance vector into a scalar value, thus enabling to assign weighting coefficients to ranks rather than to objectives. To specify such aggregation functions, we will often denote by $(\cdot)$ a permutation defined on $\{1, \ldots, n\}$ such that $y_{(1)}(x) \leq \ldots \leq y_{(n)}(x)$, sorting the performance values of a given solution x from the smallest to the largest. Here we focus on two families of rank-dependent aggregators, namely ordered weighted averages and Choquet integrals.

2.2.1 Ordered Weighted Averages

Ordered Weighted Averages (OWA) is one of the simplest families of rank-dependent aggregation functions: it is simply defined as the weighted sum applied on the sorted performance vectors [46]. OWA is specified by a normalized weighting vector $\omega = (\omega_1, \ldots, \omega_n) \in [0, 1]^n$ where ω_j is the weight associated to the performance ranked in the j-th position.

Definition 1 (OWA operator) *The OWA value of solution $x \in \mathcal{X}$ is:*

$$f_\omega(y(x)) = \sum_{j=1}^{n} \omega_j y_{(j)}(x)$$

This family of aggregators includes the minimum, maximum, median and all order statistics as a special case. It is often used in social choice theory with decreasing weights in order to favor well-balanced Pareto-optimal solutions (see e.g., [29, 30, 35] in fair allocation problems or [25, 21] in voting problems).

2.2.2 Choquet Integrals

Choquet Integrals [18, 40, 26] is a more general family of aggregators which is appealing because it offers a wider flexibility due to a greater number of preference parameters. More precisely, these aggregators are specified by a capacity function $\omega : 2^N \to [0, 1]$ such that $\omega(\emptyset) = 0, \omega(N) = 1$ and $\omega(A) \leq \omega(B)$ for all $A \subset B \subseteq N$, allowing a fine control of interactions between the objectives by assigning weights to coalitions of objectives.

Definition 2 (Choquet integral) *The Choquet value of $x \in \mathcal{X}$ is:*

$$f_\omega(y(x)) = \sum_{j=1}^{n} \Big(y_{(j)}(x) - y_{(j-1)}(x)\Big)\omega(X_{(j)}) \text{ with } y_{(0)}(x) = 0$$

where $X_{(j)} = \{(j), \ldots, (n)\}$ is the set of objectives with respect to which x has a performance greater or equal to $y_{(j)}(x)$.

For illustration purposes, consider the following capacity function:

	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
ω	0	0.2	0.1	0.3	0.4	0.7	0.6	1

For a solution x with performance vector $y(x) = (3, 2, 5)$, the Choquet integral value is $f_\omega(x) = (2-0) \times \omega(\{1,2,3\}) + (3-2) \times \omega(\{1,3\}) + (5-3) \times \omega(\{3\}) = 2 \times 1 + 1 \times 0.7 + 2 \times 0.3 = 3.3$.

The family of Choquet integrals includes the weighted sums (when ω is additive), OWA aggregators (when ω is symmetric) and weighted OWA operators [43] as a special case. Moreover, using a convex capacity (e.g., a belief function [41]) enables to model preferences for well-balanced Pareto-optimal solutions (see e.g., [16]).

2.3 Regret-Based Preference Elicitation

In this work, we assume that preference parameters ω are not known initially. Instead, we are given a (possibly empty) set Θ of pairs $(a, b) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+$ such that vector a is known to be preferred to vector b; such preference statements can be obtained by asking comparison queries to the DM. Let Ω_Θ be the set of all parameters ω that are compatible with Θ, i.e. all parameters ω that satisfy the constraints $f_\omega(a) \geq f_\omega(b)$ for all $(a, b) \in \Theta$.

Minimax regret is a decision criterion that is commonly used within the AI community to make robust recommendations under preference imprecision (e.g., [8, 14]). It can be defined as follows:

Definition 3 (Pairwise Max Regret) *The Pairwise Max Regret (PMR) of solution $x \in \mathcal{X}$ with respect to solution $x' \in \mathcal{X}$ is:*

$$PMR(x, x', \Omega_\Theta) = \max_{\omega \in \Omega_\Theta} \Big\{f_\omega(y(x')) - f_\omega(y(x))\Big\}$$

In other words, $PMR(x, x', \Omega_\Theta)$ is the worst-case loss when recommending solution x instead of solution x' to the DM.

Definition 4 (Max Regret) *The Max Regret (MR) of $x \in \mathcal{X}$ is:*

$$MR(x, \mathcal{X}, \Omega_\Theta) = \max_{x' \in \mathcal{X}} PMR(x, x', \Omega_\Theta)$$

Thus $MR(x, \mathcal{X}, \Omega_\Theta)$ is the worst-case loss when choosing solution x instead of any other feasible solution $x' \in \mathcal{X}$.

Definition 5 (Minimax Regret) *The MiniMax Regret (MMR) is:*

$$MMR(\mathcal{X}, \Omega_\Theta) = \min_{x \in \mathcal{X}} MR(x, \mathcal{X}, \Omega_\Theta)$$

According to the minimax regret criterion, an optimal solution is a solution that achieves the minimax regret (i.e., any solution in $\arg\min_{x \in \mathcal{X}} MR(x, \mathcal{X}, \Omega_\Theta)$). By definition, recommending any of these solutions enables to minimize the worst-case loss.

Note that, depending on Θ the set of available preference statements, the worst-case loss ensured by the minimax regret criterion might be at unacceptable level for the DM. Note also that the inequality $PMR(\cdot, \cdot, \Omega) \leq PMR(\cdot, \cdot, \Omega')$ holds for any two sets $\Omega \subseteq \Omega'$.

Thus this worst-case loss can be decreased by collecting new preference information from the DM (inducing new constraints on Ω_Θ the set of admissible parameters). This observation has led to the following elicitation approach: progressively ask preference queries to the DM until the MMR value drops below a given threshold $\delta \geq 0$ representing the maximum allowable gap to optimality [14]; if we set $\delta = 0$, then we obtain the best solution for the DM at the end of the execution. This approach, sometimes referred to as *regret-based incremental elicitation*, was efficiently used in various contexts, such as multicriteria decision making [12, 37] and voting problems [8, 31].

3 RELATED WORKS

Recently it has been proposed to integrate regret-based incremental elicitation into constructive algorithms so as to produce efficient exact solving methods for MOCO problems. The general principle is to construct the optimal solution from optimal sub-solutions using the available preference information, and to ask preference queries only when necessary (see [11] for a synthesis). This approach has been mainly studied with linear aggregators in order to produce exact solving methods. In this paper, we extend this approach to rank-dependent aggregators by proposing a very efficient regret-based greedy method for the knapsack problem that is guaranteed to run in polynomial time with no more than a polynomial number of queries.

Local search is a popular heuristic approach for solving hard combinatorial optimization problems [1]. It can yield high-quality solutions by iteratively applying small modifications (local moves) to a solution with the goal of generating improving solutions. Thanks to its flexibility, local search has been successfully applied to a wide range of domains, including matching problems [24, 34], multi-agent problems [27] and natural language parsing [19]. A regret-based local search method called Interactive Local Search (ILS) has been recently proposed for MOCO problems with rank-dependent aggregators, requiring that a (near-)optimal solution can be efficiently determined when the aggregation function is precisely known [4]. In practice, this requirement is quite restrictive since most existing efficient algorithms for rank-dependent aggregators are designed for subclasses representing very specific decision behaviors or for problems involving very few objectives (e.g., [23, 33]). In this paper, we propose a regret-based local search procedure specially designed for the knapsack problem that does not require existing solving methods.

Another regret-based method has been recently introduced for rank-dependent aggregators [5, 13]. The idea is to identify informative queries by exploiting the extreme points of the polyhedron representing the admissible preference parameters; in the following, this exact method will be called IEEP for Incremental Elicitation based on Extreme Points. In [5], it is shown that IEEP outperforms the constructive regret-based algorithm presented in [11] for the multi-objective spanning tree problem with the weighted sum model. However, IEEP requires that (near-)optimal solutions can be determined efficiently when the preferences are known (as ILS). Our methods also differ from IEEP with respect to performance guarantees: IEEP is an exact exponential time method that may generate an exponential number of queries, whereas our heuristic methods run in polynomial time and ask no more than a polynomial number of queries. Our methods and IEEP will be compared in the numerical section.

Committee election (or multi-winner election) is another active topic studied within the AI community. The goal is to select individuals from a pool of candidates to form a committee that is consistent with the voters' preferences (e.g., [21, 22]). This multi-agent variant of the knapsack problem is mostly studied with multi-winner voting rules (e.g., the Chamberlin–Courant rule [15]) and social welfare functions (e.g., Nash welfare) from a computational complexity perspective, assuming that voters' preferences are known. There are a few notable exceptions: two regret-based incremental elicitation procedures designed for approval voting [8] and the Chamberlin–Courant rule [32] respectively. In these papers, the authors assume that the voting rule (i.e. the aggregation function) is known but not individual utilities (i.e., performance vectors); we make the opposite assumption in this paper. More recently, a regret-based branch and bound procedure has been proposed for solving the multi-agent knapsack problem with an OWA operator, but it has been shown that IEEP achieves better results on this problem [13].

4 NEW INTERACTIVE HEURISTIC METHODS

For MOCO problems, computing $MMR(\mathcal{X}, \Omega_\Theta)$ at every step of the elicitation procedure may induce prohibitive computation times due to the possibly large number of Pareto-optimal solutions; this may indeed require to compute the pairwise max regrets for all pairs of distinct Pareto-optimal solutions in $\mathcal{X}$ (see Definitions 3, 4 and 5). Therefore, we propose instead to combine heuristic search and regret-based incremental elicitation so as to reduce both computation times and number of queries by focusing on promising solutions.

4.1 A Regret-Based Local Search

We now introduce an interactive local search procedure that uses regret-based incremental elicitation techniques to select solutions from generated neighborhoods; this algorithm, called RBLS for Regret-Based Local Search, is summarized in Algorithm 1.

Initialization. First, we identify a promising feasible solution by applying the following greedy algorithm: starting from an empty set of items, we iteratively select an item i (among the remaining items) that maximizes the arithmetic mean $1/n \sum_{j=1}^{n} y_j(i)$ so as to construct a good compromise solution in polynomial time.

Interactive local search. Then, starting from the solution obtained at the initialization step (denoted by x^* hereafter), we iteratively move from solution to solution by considering local improvements:

- Firstly, we generate a set X^* of feasible solutions from solution x^* using the neighborhood function consisting in removing one item from subset x^* and adding one item to x^* (all combinations are performed). Then, any solution that is Pareto-dominated by another solution is removed from X^*. Note that for OWA operators the most discriminant Lorenz dominance can be used instead [36].
- Secondly, we select a solution for the next iteration step by applying the standard regret-based incremental elicitation approach on set X^*. More precisely, while $MMR(X^*, \Omega_\Theta) > \delta$ holds, the DM is asked to compare two performance vectors $a, b \in \mathbb{R}^n_+$ and then Ω_Θ is restricted by inserting the linear constraint $f_\omega(a) \leq f_\omega(b)$ (or $f_\omega(b) \geq f_\omega(b)$ depending on her answer). Finally, if $MR(x^*, X^*, \Omega_\Theta) \leq \delta$ holds at the end of the elicitation process, then we stop by returning x^* (no local improvement). Otherwise, for the next iteration step, x^* is replaced by a neighbor solution that minimizes the max regret in X^*.

Termination. We stop the process by returning solution x^* after at most max_it number of iteration steps, where max_it is polynomial in the problem size.

Algorithm 1 RBLS

IN $\downarrow P$: a multi-objective knapsack problem; f_ω: a family of rank-dependent operators; Θ: a set of statements; δ: a positive threshold.
--| Initialization:
$\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \geq f_\omega(b)\}$
$x^* \leftarrow$ `ComputeInitialSolution`(P)
$it \leftarrow 0$
improve $\leftarrow$ **true**
--| Local Search:
while improve and $(it < max\_it)$ **do**
  --| Generation of neighbors:
  $X^* \leftarrow$ `ComputeAllSwaps`(x^*)
  --| Standard regret-based elicitation:
  while $MMR(X^*, \Omega_\Theta) > \delta$ **do**
    --| Ask the DM to compare two solutions:
    $(a,b) \leftarrow$ `AskQuery`(X^*, Ω_Θ)
    --| Update preference information:
    $\Theta \leftarrow \Theta \cup \{(a,b)\}$
    $\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \geq f_\omega(b)\}$
  end while
  --| Move to another solution:
  if $MR(x^*, X^*, \Omega_\Theta) > \delta$ **then**
    $x^* \leftarrow$ `Select`$(\arg\min_{x \in X^*} MR(x, X^*, \Omega_\Theta))$
    $it \leftarrow it + 1$
  else
    improve $\leftarrow$ **false**
  end if
end while
return x^*

Example 1 *We now present an execution of RBLS. Consider an instance of the 3-objectives knapsack problem with 10 items (denoted by $i_1, \ldots, i_{10}$), where $\mathcal{X}$ the set of feasible solutions only includes the subsets of size at most 5. The performance vectors are as follows:*

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
y_1	4	10	3	10	4	4	10	8	1	3
y_2	4	1	7	2	6	1	3	3	4	5
y_3	3	9	7	2	9	7	8	9	10	10

Table 1. Performance vectors attached to items.

We assume here that the DM's preferences can be represented by an OWA operator f_{ω^*} with the hidden decreasing weights $\omega^* = (0.7, 0.2, 0.1)$. We start the execution of RBLS with an empty set of preference statements (i.e. $\Theta = \emptyset$) and $\delta = 0$. Hence Ω_Θ is initially the set of all weighting vectors $\omega = (\omega_1, \omega_2, \omega_3) \in [0,1]^3$ such that $\omega_1 + \omega_2 + \omega_3 = 1$ and $\omega_1 \geq \omega_2 \geq \omega_3$. Note that Ω_Θ is a convex polyhedron since $f_\omega(a)$ is linear in ω for fixed performance vector a. The extreme points of Ω_Θ are $(0.5, 0.5, 0)$, $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $(1, 0, 0)$ initially; in Figure 1, Ω_Θ is represented by the triangle ABC in the space (ω_1, ω_2), ω_3 being implicitly defined by $\omega_3 = 1 - \omega_1 - \omega_2$.

Initialization step: *First, we compute the optimal solution for the weighted sum with weights $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ by using our greedy algorithm. We obtain solution $x^* = (0, 1, 0, 0, 1, 0, 1, 1, 0, 1)$ whose performance vector is $y(x^*) = (35, 18, 45)$.*

Local Search: *At the first iteration step, only two neighbors of x^* are non-dominated, and the set X^* contains three solutions, denoted by $x^1 (= x^*)$, x^2 and x^3, which are evaluated as follows: $y(x^1) = (35, 18, 45)$, $y(x^2) = (24, 28, 43)$ and $y(x^3) = (20, 35, 42)$. Since $MMR(X^*, \Omega_\Theta) = 1.5 > 0$, we ask the DM to compare two solutions in X^*, say x^2 and x^3. We have $f_{\omega^*}(y(x^2)) = 26.7 > f_{\omega^*}(y(x^3)) = 25.2$. Therefore the DM prefers solution x^2 to solution x^3. Thus we obtain $\Theta = \{((24, 28, 43), (20, 35, 42))\}$ and Ω_Θ is restricted by the linear constraint $f_\omega(y(x^2)) \geq f_\omega(y(x^3))$, i.e. $\omega_2 \leq \frac{1}{8} + \frac{3}{8}\omega_1$ (see Figure 2 where Ω_Θ is represented by CDE). Now we have $MMR(X^*, \Omega_\Theta) = MR(x^2, X^*, \Omega_\Theta) = 0$. Therefore we stop asking queries and we move from solution $x^* = x^1$ to solution x^2 for the next step (i.e., we now set $x^* = x^2$).*

At the second step, X^* includes only 5 non-dominated solutions denoted by $x^1 (= x^*)$, x^2, x^3, x^4 and x^5 with the following evaluations: $y(x^1) = (24, 28, 43)$, $y(x^2) = (35, 18, 45)$, $y(x^3) = (30, 23, 36)$, $y(x^4) = (35, 20, 42)$ and $y(x^5) = (35, 21, 35)$. Since $MMR(X^*, \Omega_\Theta) = \frac{7}{11} \geq 0$, the DM is asked to compare two solutions, say x^* and x^5. Since $f_{\omega^*}(y(x^1)) = 26.7 > f_{\omega^*}(y(x^5)) = 25.2$, the DM prefers x^* to x^5. Then Θ is set to $\{((24, 28, 43), (20, 35, 42)), ((24, 28, 43), (35, 21, 35))\}$ and Ω_Θ is restricted by $\omega_2 \leq \frac{8}{15} - \frac{1}{3}\omega_1$ (see Figure 3 where Ω_Θ is represented by CDFG). Now we have $MR(x^*, X^*, \Omega_\Theta) = 0$. Therefore x^* is a local optimum (variable improve is set to **false** *and the while loop ends). RBLS ends by returning $x^* = (0, 0, 1, 0, 1, 0, 1, 1, 0, 1)$ which is the preferred solution. Thus only 2 queries were needed to discriminate between the 27 Pareto-optimal solutions of this instance.*

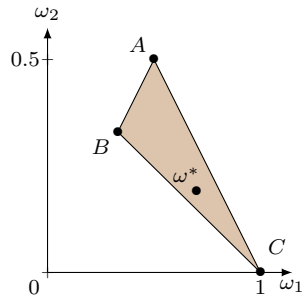

Figure 1. Initial set Ω_Θ.

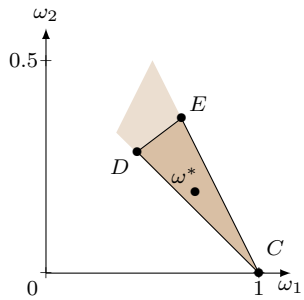

Figure 2. Ω_Θ after 1 query.

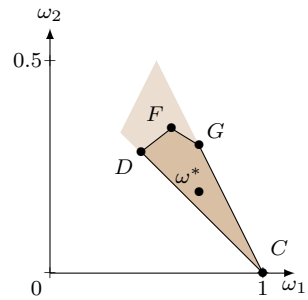

Figure 3. Ω_Θ after 2 queries.

4.2 A Regret-Based Greedy Algorithm

In this subsection, we propose a new interactive solving method that consists in integrating regret-based incremental elicitation techniques into a greedy algorithm. Our algorithm, called RBGA for Regret-Based Greedy Algorithm, starts from an empty set of items (denoted

by S hereafter) and then iterates as follows:

- First, we compute X^* the set of all (partial) solutions obtained by adding one item to set S; formally $X^* = \{S \cup \{i\} : i \in \mathcal{I} \backslash S\}$. Then, while $MMR(X^*, \Omega_\Theta) > \delta$ holds, we ask preference queries to the DM and update the set of admissible preference parameters accordingly.
- Once $MMR(X^*, \Omega_\Theta) \leq \delta$ holds, we select an item i^* from $\mathcal{I} \backslash S$ such that $S \cup \{i^*\}$ minimizes the max regret. Finally, item i^* is added to set S for the next iteration step.

Termination. RBGA stops whenever the knapsack capacity is reached, i.e. after exactly W iteration steps (see Algorithm 2).

Algorithm 2 RBGA

IN $\downarrow P$: a multi-objective knapsack problem; f_ω: a family of rank-dependent operators; Θ: a set of statements; δ: a positive threshold.
--| Initialization:
$\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \geq f_\omega(b)\}$
$S \leftarrow \emptyset$
$it \leftarrow 0$
--| Greedy Search:
while $it < W$ **do**
  $X^* \leftarrow \{S \cup \{i\} : i \in \mathcal{I} \backslash S\}$
  --| Standard Regret-Based Elicitation:
  while $MMR(X^*, \Omega_\Theta) > \delta$ **do**
    --| Ask the DM to compare two solutions:
    $(a,b) \leftarrow \texttt{Query}(X^*)$
    --| Update preference information:
    $\Theta \leftarrow \Theta \cup \{(a,b)\}$
    $\Omega_\Theta \leftarrow \{\omega : \forall (a,b) \in \Theta, f_\omega(a) \geq f_\omega(b)\}$
  end while
  --| Item Selection:
  $i^* \leftarrow \texttt{Select}(\arg\min_{i \in \mathcal{I} \backslash S} MR(S \cup \{i\}, X^*, \Omega_\Theta))$
  $S \leftarrow S \cup \{i^*\}$
  $it \leftarrow it + 1$
end while
return S

Note that, when the DM's preferences are represented by a weighted sum that is precisely known, an optimal solution can be obtained by a greedy algorithm that simply selects an item maximizing the aggregated value at each iteration step, until the knapsack capacity is reached (here after W steps). When the weighted sum is imprecisely known, one may ask preference queries at each step until identifying an item maximizing the "unknown" aggregated value (i.e., until there is an item with a max regret equal to zero); that is precisely what RBGA does when $\delta = 0$. Thus, RBGA necessarily returns an optimal solution according to the DM's preferences when f_ω is a weighted sum and $\delta = 0$ (for $\delta > 0$, one can easily prove that the worst-case loss is bounded above by $W \times \delta$). Moreover, we observed in practice that it often returns the optimal solution with OWA operators and Choquet integrals (see Section 4), and it constructs the optimal solution when applied to Example 1 with only two queries.

4.3 Performance guarantees

This subsection is devoted to the proof of the following result:

Proposition 1 *RBLS and RBGA can be implemented in such way that they run in polynomial time and ask no more than a polynomial number of queries for OWA aggregators and Choquet integrals.*

We remark first that the number of iteration steps of the outer loop is polynomial (in the problem size) for both procedures, and this is also the case for the size of X^* at each iteration step.

4.3.1 OWA operators

On the number of preference queries. Since the number of iteration steps of the outer loop is polynomial for both RBLS and RBGA, the number of generated queries is also polynomial if and only if the minimax regret of X^* drops below δ after a polynomial number of comparison queries at each iteration step of the search procedures. To do so, one can simply apply the Current Solution Strategy (CSS) [14] which consists in asking to compare a solution x achieving the minimax regret to one of its adversary's choice (i.e. a solution in $\arg\max_{x' \in X^*} PMR(x, x', \Omega_\Theta)$) at each iteration step of the elicitation process. By doing so, one can ensure that the minimax regret will be equal to zero after at most $|X^*| - 1$ queries since at least one solution is eliminated after every comparison query.

On the computation times. Recall that both the number of steps and the size of X^* at each step are polynomial in the problem size. Moreover, we have just proved that, at each step of the outer loop, the number of preference queries (hence the number of steps of the inner loop) is polynomial when applying the CSS strategy. Therefore, we only need to prove that MMR-computations can be performed in polytime. Recall that $MMR(X^*, \Omega_\Theta)$ can be obtained by solving at most $|X^*|^2$ pairwise max regrets (see Definitions 4 and 5). Therefore, it is sufficient to show that pairwise max regrets can be computed in polynomial time. For OWA aggregators, even though $f_\omega(a) = \sum_{j=1}^n \omega_j a_{(j)}$ is non-linear in a for fixed parameters ω, it is linear in ω for fixed a. Thus $PMR(x, x', \Omega_\Theta)$ can be obtained in polynomial time by solving the following linear program:

$$\begin{aligned} \max_{\omega} \quad & \sum_{j=1}^n \omega_j y_{(j)}(x') - \sum_{j=1}^n \omega_j y_{(j)}(x) \\ s.t. \quad & \sum_{j \in N} \omega_j = 1 \\ & \omega_j \geq 0, \forall j \in N \\ & \sum_{j=1}^n \omega_j a_{(j)} - \sum_{j=1}^n \omega_j b_{(j)} \geq 0, \ \forall (a,b) \in \Theta \end{aligned}$$

The constraints $\omega_j \geq \omega_{j+1}$ for all $j \in \{1, \ldots, n-1\}$ can be added so as to favor well-balanced solutions.

4.3.2 Choquet integrals

As regards the number of queries, the same reasoning as for OWA aggregators applies for Choquet integrals. This is not the case for computation times. More precisely, since $f_\omega(a)$ is linear in ω for fixed a with Choquet integrals, $PMR(x, x', \Omega_\Theta)$ can also be obtained by solving the following program:

However, in this linear program, the number of variables and constraints is exponential in the number of variables. This linear program can be simplified for some subclasses of Choquet integrals defined using an alternative formulation in terms of Möbius masses m [17]:

$$f_\omega(y(x)) = \sum_{A \subseteq N}^n m(A) \min_{j \in A} y_j(x)$$

$$\max_{\omega} \sum_{j=1}^{n} \Big(y_{(j)}(x') - y_{(j-1)}(x')\Big)\omega(X'_{(j)}) - \Big(y_{(j)}(x) - y_{(j-1)}(x)\Big)\omega(X_{(j)})$$
$$s.t.\ \omega(\emptyset) = 0, \omega(N) = 1$$
$$\omega(A) \leq \omega(B),\ \forall A \subset B \subseteq N$$
$$\sum_{j=1}^{n} \Big(a_{(j)} - a_{(j-1)}\Big)\omega(A_{(j)}) - \Big(b_{(j)} - b_{(j-1)}\Big)\omega(B_{(j)}) \geq 0,\ \forall (a,b) \in \Theta$$

where $m(A) = \sum_{B \subseteq A}(-1)^{|A \setminus B|}\omega(B)$ for all $A \subseteq N$. For instance, for the subclass of convex capacities characterized by positive Möbius masses (a.k.a. belief functions [41]), the monotonicity constraints $\omega(A) \leq \omega(B)$, with $A \subset B \subseteq N$, are naturally satisfied due to the non-negativity of Möbius masses. For the subclass of 2-additive capacities (i.e., $m(A) = 0$ for all $|A| > 2$ and $m(A) \neq 0$ for some set $|A| = 2$), it has been proposed to use the fact that these particular capacities form a convex polytope with only n^2 extreme points (see e.g., [42]), yielding a linear formulation involving a quadratic number of variables and constraints; hence for 2-additive capacities, our procedures combined with the CSS strategy run in polynomial time and generate no more than a polynomial number of queries.

For general capacities, it has been shown in [12] that PMR-optimization problems can be formulated as linear programs with only a linear number of variables and constraints, provided that the DM is only asked to compare binary alternatives to constant utility profiles. In the same paper, the authors propose a query generation strategy that is very efficient in practice but no theoretical guarantee is provided regarding the number of queries. In fact, the proposed query generation strategy can be adapted to ensure a polynomial number of queries by simply selecting the constant utility profile in the middle of the interval representing the possible capacity values, instead of selecting the query that minimizes the worst-case minimax regret (see e.g., [9]). Thus, our procedures are also polynomial for general Choquet integrals.

5 EXPERIMENTAL RESULTS

We now provide numerical results showing the practical efficiency of our algorithms in terms of computation times (given in seconds), number of queries and gap to optimality (error) expressed in terms of percentage from the optimal solution. The results are averaged over 30 runs and "/" means that the timeout (30 mins) is exceeded.

Instances. We consider instances of the multi-objective knapsack problem with $|\mathcal{I}| = 50$ and 100 items and we vary n the number of criteria from 2 to 6. Instances are generated as follows: performance vectors $y(i)$, $i \in \mathcal{I}$, are uniformly drawn in $\{1, \ldots, 1000\}^n$ and W the knapsack capacity is set to $0.5 \times |\mathcal{I}|$ so as to obtain difficult instances (i.e., with a large number of Pareto-optimal solutions).

Preferences. The DM's preferences are represented by the following aggregation models:

- OWA: OWA aggregators with decreasing positive weights to model preferences for well-balanced performance vectors.
- Choquet: Choquet integrals with 2-additive belief functions (see Section 4.3.2) to model interactions between criteria and favor well-balanced vectors.

We start the executions with an empty set of preference statements; therefore Ω_Θ is initially equal to:

- $\{\omega \in \mathbb{R}^n_+ : \forall j \in \{1, \ldots, n-1\}, \omega_{j+1} \geq \omega_j; \sum_{j=1}^n \omega_j = 1\}$ for the OWA model.
- $\{m_A \in \mathbb{R}_+, A \subseteq N, |A| \leq 2 : \exists A, |A| = 2, m_A \neq 0; \sum_A m_A = 1\}$ for Choquet integrals.

During the execution of the methods, the answers to queries are simulated using a weighting vector (ω for OWA, m for Choquet) that is randomly generated before running the methods. The weighting vector is generated using the procedure presented in [39] so as to guarantee a uniform distribution of the weights.

Regret-Based Solving Methods. In this section, we compare the results obtained by the following regret-based procedures:

- RBLS: the proposed local search method with $max\_it = 100$.
- RBGA: the proposed greedy method.
- Two-Phase$_\delta$: a two-phase method which consists in first constructing a "well-represented" Pareto set and then applying the CSS strategy on this set until the minimax regret drops below threshold $\delta \geq 0$; here we generate sets of size at most 3000.
- IEEP$_\delta$: the solving method introduced in [5] for general MOCO problems, which is based on the extreme points of Ω_Θ. This method is guaranteed to return a solution $x \in \mathcal{X}$ such that $\text{MR}(x, \mathcal{X}, \Omega_\Theta) \leq \delta$ holds at the end of the execution. To compute the optimal solutions attached to extreme points, we simply return the solutions that maximize the corresponding aggregated values in the representative Pareto set.

To generate Pareto sets for both Two-Phase$_\delta$ and IEEP$_\delta$, we use dynamic programming [2] and fast Pareto dominance checking [28] to reduce computation times (e.g., the Pareto set includes about 2×10^6 solutions for the instances with 50 items and 6 objectives). For both procedures, the reported computation times do not include the time needed to generate the Pareto set.

Implementation details. Numerical tests were performed on a Intel Core i7-950 @ 3.06 GHz with 11,7 GB of RAM, with a program written in C++, and pairwise max regret optimizations are performed by CPLEX Optimizer[2].

Results for OWA. First we compare the results obtained by the two proposed procedures (see Table 2). We observe that RBLS is more efficient that RBGA in terms of number of queries and error in all the settings (while achieving comparable computation times); for instance, when using RBLS instead of RBGA for $n = 6$ with $\delta = 0.5$, the number of queries is divided by 2 and the error is decreased by a factor of 4. Note that the error is very low in all the settings (below 1%) and that varying δ from 0 to 0.5 does not really impact on the performances of both procedures for OWA aggregators.

Now we compare RBLS to IEEP$_\delta$ and Two-Phase$_\delta$ (see Table 3); here we set $\delta = 0.01$ to allow for the same error as our method (to be as fair as possible). We observe that RBLS is drastically faster than IEEP$_{0.01}$ and generates far fewer queries in all the settings. To give an example, when using IEEP$_{0.01}$ instead of RBLS (with $\delta = 0.5$) for $n = 6$, the computation time is reduced by a factor of 260 and the number of queries is divided by 3. Finally, we see that RBLS is much faster than Two-Phase$_{0.01}$ but the former asks a few more preference queries than the latter (at most 3 queries with $\delta = 0.5$).

[2] https://www.ibm.com/analytics/cplex-optimizer

		OWA						Choquet					
		RBLS			RBGA			RBLS			RBGA		
	n	time	queries	error	time	queries	error	time	queries	error	time	queries	error
$\delta = 0$	2	0.07	3.0	0.00	0.08	6.4	0.81	0.42	5.6	0.14	0.44	10.70	0.10
	3	0.20	7.9	0.13	0.16	10.4	0.91	7.99	19.7	0.16	2.31	33.83	0.17
	4	0.02	3.0	0.00	0.19	18.2	0.63	48.04	54.0	0.15	7.27	70.13	0.63
	5	0.27	8.3	0.18	0.25	17.4	0.64	144.67	79.3	0.09	19.5	119.97	0.59
	6	0.65	17.2	0.24	0.29	17.1	0.81	975.96	154.0	0.23	38.9	179.93	0.21
$\delta = 0.5$	2	0.08	3.6	0.00	0.08	6.5	0.92	0.43	4.73	0.15	0.36	9.03	0.00
	3	0.15	5.7	0.10	0.13	8.9	0.94	3.76	9.36	1.16	1.60	24.76	0.16
	4	0.02	2.0	0.00	0.18	16.1	0.79	20.17	20.13	0.18	4.12	39.90	0.27
	5	0.23	6.4	0.19	0.22	14.8	0.65	43.56	26.40	0.12	7.12	51.83	0.54
	6	0.31	7.3	0.18	0.27	14.6	0.85	129.18	32.83	0.28	8.82	66.27	0.32

Table 2. Results obtained by RBLS and RBGA the two proposed procedures for 50 items.

	OWA						Choquet					
	Two-Phase$_{0.01}$			IEEP$_{0.01}$			Two-Phase$_{0.02}$			IEEP$_{0.02}$		
n	time	queries	error	time	queries	error	time	queries	error	time	queries	error
2	4.3	2.5	0.02	10.1	4.7	0.00	1.6	4.2	0.04	8.4	3.9	0.01
3	36.4	3.1	0.05	16.9	8.0	0.03	298.4	12.5	0.13	19.5	9.0	0.17
4	46.4	2.7	0.46	33.9	15.0	0.00	416.6	30.2	0.32	/	/	/
5	82.1	4.5	0.68	40.6	18.3	0.00	859.3	55.8	0.52	/	/	/
6	95.3	4.1	0.96	80.6	26.8	0.00	/	/	/	/	/	/

Table 3. Results obtained by IEEP$_\delta$ and Two-Phase$_\delta$ for 50 items.

	OWA						Choquet					
	RBLS			RBGA			RBLS			RBGA		
n	time	queries	error	time	queries	error	time	queries	error	time	queries	error
2	0.1	3.4	0.04	0.2	8.5	1.52	1.9	7.0	0.30	1.2	12.6	0.28
3	0.3	5.5	0.01	0.7	13.6	1.15	25.8	15.5	0.91	9.1	34.3	0.23
4	1.1	11.4	0.16	0.7	21.6	0.71	124.4	27.2	0.06	27.3	53.1	0.43
5	4.1	26.5	0.29	1.1	24.5	0.54	516.4	38.8	0.11	44.5	80.6	0.26
6	5.2	28.3	0.42	1.0	25.7	0.72	1183.5	47.8	0.10	64.3	107.0	0.23

Table 4. Results obtained by RBLS and RBGA the two proposed procedures for 100 items ($\delta = 0.5$).

Results for Choquet. In Table 2, we see that varying δ from 0 to 0.5 allows to significantly reduce the computation time and the number of preference queries of both RBLS and RBGA without increasing the error too much (at most 2%). For instance, for $n = 6$, RBLS ends after 16 minutes and 154 queries for $\delta = 0$ against 2 minutes and 32 queries for $\delta = 0.5$. This is mainly due to the fact that more preference information is needed (more preference parameters) and PMR-optimizations are more computationally demanding (a quadratic number of variables) when considering Choquet instead of OWA. Now, when comparing our procedures, we see that RBLS is better than RBGA in terms of number of queries but the latter is much faster than the former on the bigger instances. For example, for $n = 6$ and $\delta = 0.5$, RBLS generates two times less preference queries than RBGA but RBGA is about 15 times faster than RBLS.

Finally, we compare our procedures to IEEP$_\delta$ and Two-Phase$_\delta$ (see Table 3) with $\delta = 0.02$ to allow for the same error as our methods. We see that both IEEP$_{0.02}$ and Two-Phase$_{0.02}$ induce prohibitive computation times when increasing the number of criteria (even when ignoring the time required to compute the representative Pareto set). For IEEP$_{0.02}$, the problem is even more dramatic since the timeout is reached as soon as $n \geq 4$. This is mainly due to the very large number of extreme points of Ω_Θ, which increases exponentially with the number of collected preference statements; for instance, we stopped IEEP$_{0.02}$ after 3 hours for $n = 4$ and we observed that only 25 queries were asked and 10^4 extreme points were generated on average. In fact, our procedures can be applied to bigger instances, as shown in Table 4 with 100 items.

6 CONCLUSION

In this paper, we have proposed two new regret-based interactive procedures to solve multi-objective knapsack problems with unknown preferences: a local search method (called RBLS) and a greedy algorithm (called RBGA). To model the DM's preferences, we have considered the following two families of non-linear aggregators: OWA operators and Choquet integrals. We have proved that both RBLS and RBGA run in polynomial time and ask no more than a polynomial number of queries. Moreover, we have compared the performances achieved by RBLS and RBGA with that of two different regret-based solving methods. In particular, it is shown that the latter two procedures are both outperformed by RBLS and RBGA in all considered instances. We have also observed that RBLS performs better than RBGA for OWA operators, but RBGA is much faster for Choquet integrals (even if RBLS may still be preferred to minimize the number of queries). An interesting direction of research is to compare our methods to a very recent heuristic approach based on genetic algorithms, which may require to adapt the latter approach to efficiently handle rank-dependant aggregators [3].

7 Acknowledgements

This work was supported by the Paris Ile-de-France Region, the FMJH Program PGMO and EDF-THALESE-ORANGE.

REFERENCES

[1] E. Aarts and J.K. Lenstra, *Local Search in Combinatorial Optimization*, John Wiley & Sons, Inc., New York, NY, USA, 1997.

[2] C. Bazgan, H. Hugot, and D. Vanderpooten, 'Solving efficiently the 0-1 multi-objective knapsack problem', *Computers & OR*, **36**(1), 260–279, (2009).

[3] N. Benabbou, C. Leroy, and T. Lust, 'An interactive regret-based genetic algorithm for solving multi-objective combinatorial optimization problems', in *Proceedings of AAAI'20*, p. to appear, (2020).

[4] N. Benabbou, C. Leroy, T. Lust, and P. Perny, 'Combining local search and elicitation for multi-objective combinatorial optimization', in *Proceeedings of ADT 2019*, pp. 1–16, (2019).

[5] N. Benabbou and T. Lust, 'A general interactive approach for solving multi-objective combinatorial optimization problems with imprecise preferences', in *Proceedings of SOCS'19*, pp. 164–165, (2019).

[6] N. Benabbou and P. Perny, 'Combining preference elicitation and search in multiobjective state-space graphs', in *Proceedings of IJCAI'15*, pp. 297–303, (2015).

[7] N. Benabbou and P. Perny, 'Incremental weight elicitation for multiobjective state space search', in *Proceedings of AAAI'15*, pp. 1093–1098, (2015).

[8] N. Benabbou and P. Perny, 'Solving multi-agent knapsack problems using incremental approval voting', in *Proceedings of ECAI'16*, pp. 1318–1326, (2016).

[9] N. Benabbou and P. Perny, 'Adaptive elicitation of preferences under uncertainty in sequential decision making problems', in *Proceedings of IJCAI'17*, pp. 4566–4572, (2017).

[10] N. Benabbou and P. Perny, 'Interactive resolution of multiobjective combinatorial optimization problems by incremental elicitation of criteria weights', *EURO journal on decision processes*, **6**, 283–319, (2018).

[11] N. Benabbou and P. Perny, 'Interactive resolution of multiobjective combinatorial optimization problems by incremental elicitation of criteria weights', *EURO Journal on Decision Processes*, **6**(3), 283–319, (2018).

[12] N. Benabbou, P. Perny, and P. Viappiani, 'Incremental elicitation of Choquet capacities for multicriteria decision making', in *Proceedings of ECAI'14*, pp. 87–92, (2014).

[13] N. Bourdache and P. Perny, 'Active preference elicitation based on generalized Gini functions: Application to the multiagent knapsack problem', in *Proceedings of AAAI'19*, pp. 7741–7748, (2019).

[14] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans, 'Constraint-based optimization and utility elicitation using the minimax decision criterion', *Artificial Intelligence*, **170**(8–9), 686–713, (2006).

[15] J.R. Chamberlin and P.N. Courant, 'Representative deliberations and representative decisions: Proportional representation and the Borda rule', *The American Political Science Review*, **77**(3), 718–733, (1983).

[16] A. Chateauneuf, R.A. Dana, and J-M. Tallon, 'Diversification, convex preferences and non-empty core in the Choquet expected utility model', *Economic Theory*, **19**(3), 509–523, (1999).

[17] A. Chateauneuf and J-Y. Jaffray, 'Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion', *Mathematical Social Sciences*, **17**(3), 263–283, (1989).

[18] G. Choquet, 'Theory of capacities', *Annales de l'Institut Fourier*, **5**, 31–295, (1953).

[19] M. Daum and W. Menzel, 'Parsing natural language using guided local search', in *Proceedings ECAI'02*, pp. 435–439, (2002).

[20] J. Drummond and C. Boutilier, 'Preference elicitation and interview minimization in stable matchings', in *Proceedings of AAAI'14*, pp. 645–653, (2014).

[21] E. Elkind and A. Ismaili, 'OWA-based extensions of the chamberlin courant rule', in *Proceedings of ADT'15*, pp. 486–502. Springer, (2015).

[22] T. Fluschnik, P. Skowron, M. Triphaus, and K. Wilker, 'Fair knapsack', in *Proceedings of AAAI'19*, pp. 1941–1948, (2019).

[23] L. Galand and O. Spanjaard, 'Exact algorithms for owa-optimization in multiobjective spanning tree problems', *Computers & Operations Research*, **39**(7), 1540–1554, (2012).

[24] M. Gelain, M. Silvia Pini, F. Rossi, K.B. Venable, and T. Walsh, 'Local search algorithms on the stable marriage problem: Experimental studies', in *Proceedings of ECAI'10*, pp. 1085–1086, (2010).

[25] J. Goldsmith, J. Lang, N. Mattei, and P. Perny, 'Voting with rank dependent scoring rules', in *Proceedings of AAAI'14*, pp. 698–704, (2014).

[26] M. Grabisch and C. Labreuche, 'A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid', *Annals of Operations Research*, **175**(1), 247–286, (2010).

[27] A. Grubshtein, R. Zivan, and A. Meisels, 'Partial cooperation in multi-agent local search', in *Proceedings ECAI'12*, pp. 378–383, (2012).

[28] A. Jaszkiewicz and T. Lust, 'ND-Tree-based update: A fast algorithm for the dynamic nondominance problem', *IEEE Trans. Evolutionary Computation*, **22**(5), 778–791, (2018).

[29] P. Korhonen, H. Moskowitz, and J. Wallenius, 'Choice behavior in interactive multiple-criteria decision making', *Annals of Operations Research*, **23**(1), 161–179, (1990).

[30] J. Lesca and P. Perny, 'LP Solvable Models for Multiagent Fair Allocation problems', in *Proceedings of ECAI'10*, pp. 387–392, (2010).

[31] T. Lu and C. Boutilier, 'Robust approximation and incremental elicitation in voting protocols', in *Proceedings of IJCAI'11*, pp. 287–293, (2011).

[32] T. Lu and C. Boutilier, 'Multi-winner social choice with incomplete preferences', in *Proceedings of IJCAI'13*, pp. 263–270, (2013).

[33] T. Lust and A. Rolland, 'Choquet optimal set in biobjective combinatorial optimization', *Computers & OR*, **40**(10), 2260–2269, (2013).

[34] D. Munera, D. Diaz, S. Abreu, F. Rossi, V. Saraswat, and P. Codognet, 'Solving hard stable matching problems via local search and cooperative parallelization', in *Proceedings of AAAI'15*, pp. 1212–1218, (2015).

[35] W. Ogryczak, 'Inequality measures and equitable approaches to location problems', *European Journal of Operational Research*, **122**(2), 374–391, (2000).

[36] P. Perny, O. Spanjaard, and L-X. Storme, 'A decision-theoretic approach to robust optimization in multivalued graphs', *Annals of Operations Research*, **147**(1), 317–341, (2006).

[37] P. Perny, P. Viappiani, and A. Boukhatem, 'Incremental preference elicitation for decision making under risk with the rank-dependent utility model', in *Proceedings of UAI'16*, (2016).

[38] K. Regan and C. Boutilier, 'Eliciting additive reward functions for markov decision processes', in *IJCAI'11*, pp. 2159–2164, (2011).

[39] R.Y. Rubinstein, 'Generating random vectors uniformly distributed inside and on the surface of different regions', *European Journal of Operational Research*, **10**(2), 205 – 209, (1982).

[40] D. Schmeidler, 'Integral representation without additivity', *Proceedings of the American Mathematical Society*, **97**(2), 255–261, (1986).

[41] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[42] A. Fallah Tehrani, W. Cheng, K. Dembczynski, and E. Hüllermeier, 'Learning monotone nonlinear models using the Choquet integral', *Machine Learning*, **89**(1-2), 183–211, (2012).

[43] V. Torra, 'The weighted OWA operator', *International Journal of Intelligent Systems*, **12**(2), 153–166, (1997).

[44] T. Wang and C. Boutilier, 'Incremental Utility Elicitation with the Minimax Regret Decision Criterion', in *IJCAI'03*, pp. 309–316, (2003).

[45] P. Weng and B. Zanuttini, 'Interactive value iteration for markov decision processes with unknown rewards', in *IJCAI'13*, pp. 2415–2421, (2013).

[46] R.R. Yager, 'On ordered weighted averaging aggregation operators in multicriteria decisionmaking', *IEEE Trans. Syst. Man Cybern.*, **18**(1), 183–190, (January 1988).