An Interactive Polyhedral Approach for Multi-Objective Combinatorial Optimization with Incomplete Preference Information

Nawal Benabbou, Thibaut Lust

An Interactive Polyhedral Approach for Multi-Objective Combinatorial Optimization with Incomplete Preference Information

Nawal Benabbou and Thibaut Lust

Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6
LIP6 F-75005 Paris, France
nawal.benabbou@lip6.fr, thibaut.lust@lip6.fr

Abstract. In this paper, we develop a general interactive polyhedral approach to solve multi-objective combinatorial optimization problems with incomplete preference information. Assuming that preferences can be represented by a parameterized scalarizing function, we iteratively ask preferences queries to the decision maker in order to reduce the imprecision over the preference parameters until being able to determine her preferred solution. To produce informative preference queries at each step, we generate promising solutions using the extreme points of the polyhedron representing the admissible preference parameters and then we ask the decision maker to compare two of these solutions (we propose different selection strategies). These extreme points are also used to provide a stopping criterion guaranteeing that the returned solution is optimal (or near-optimal) according to the decision maker's preferences. We provide numerical results for the multi-objective spanning tree and traveling salesman problems with preferences represented by a weighted sum to demonstrate the practical efficiency of our approach. We compare our results to a recent approach based on minimax regret, where preference queries are generated during the construction of an optimal solution. We show that better results are achieved by our method both in terms of running time and number of questions.

Keywords: Multi-objective Combinatorial Optimization · Minimum Spanning Tree Problem · Traveling Salesman Problem · Incremental Preference Elicitation · Minimax Regret

1 Introduction

The increasing complexity of applications encountered in Computer Science significantly complicates the task of decision makers who need to find the best solution among a very large number of options. Multi-objective optimization is concerned with optimization problems involving several (conflicting) objectives/criteria to be optimized simultaneously (e.g., minimizing costs while maximizing profits). Without preference information, we only know that the best solution for the decision maker (DM) is among the Pareto-optimal solutions (a

solution is called Pareto-optimal if there exists no other solution that is better on all objectives while being strictly better on at least one of them). The main problem with this kind of approach is that the number of Pareto-optimal solutions can be intractable, that is exponential in the size of the problem (e.g. [14] for the multicriteria spanning tree problem). One way to address this issue is to restrict the size of the Pareto set in order to obtain a "well-represented" Pareto set; this approach is often based on a division of the objective space into different regions (e.g., [16]) or on ϵ-dominance (e.g., [19]). However, whenever the DM needs to identify the best solution, it seems more appropriate to refine the Pareto dominance relation with preferences to determine a single solution satisfying the subjective preferences of the DM. Of course, this implies the participation of the DM who has to give us some insights and share her preferences.

In this work, we assume that the DM's preferences can be represented by a parameterized scalarizing function (e.g., a weighted sum), allowing some trade-off between the objectives, but the corresponding preference parameters (e.g., the weights) are initially not known; hence, we have to consider the set of all parameters compatible with the collected preference information. An interesting approach to deal with preference imprecision has been recently developed [20, 22, 31] and consists in determining the *possibly* optimal solutions, that is the solutions that are optimal for at least one instance of the preference parameters. The main drawback of this approach, though, is that the number of possibly optimal solutions may still be very large compared to the number of Pareto-optimal solutions; therefore there is a need for elicitation methods aiming to specify the preference model by asking preference queries to the DM.

In this paper, we study the potential of incremental preference elicitation (e.g., [24, 28]) in the framework of multi-objective combinatorial optimization. Preference elicitation on combinatorial domains is an active topic that has been recently studied in various contexts, e.g. in multi-agents systems [1, 4, 7], in stable matching problems [10], in constraint satisfaction problems [8], in Markov Decision Processes [12, 25, 29] and in multi-objective optimization problems [2, 5, 15, 17]. Our aim here is to propose a general interactive approach for multi-objective optimization with imprecise preference parameters. Our approach identifies informative preference queries by exploiting the extreme points of the polyhedron representing the admissible preference parameters. Moreover, these extreme points are also used to provide a stopping criterion which guarantees the determination of the (near-)optimal solution. Our approach is general in the sense that it can be applied to any multi-objective optimization problem, providing that the scalarizing function is linear in its preference parameters (e.g., weighted sums, Choquet integrals [9, 13]) and that there exists an efficient algorithm to solve the problem when preferences are precisely known (e.g., [18, 23] for the minimum spanning tree problem with a weighted sum).

The paper is organized as follows: We first give general notations and recall the basic principles of regret-based incremental elicitation. We then propose a new interactive method based on the minimax regret decision criterion and extreme points generation. Finally, to show the efficiency of our method, we

provide numerical results for two well-known problems, namely the multicriteria traveling salesman and multicriteria spanning tree problems; for the latter, we compare our results with those obtained by the state-of-the-art method.

2 Multi-objective Combinatorial Optimization

In this paper, we consider a general multi-objective combinatorial optimization (MOCO) problem with n objective functions $y_i, i \in \{1, \dots, n\}$, to be minimized. This problem can be defined as follows:

$$\underset{x \in \mathcal{X}}{\text{minimize}} \big(y_1(x), \dots, y_n(x)\big)$$

In this definition, $\mathcal{X}$ is the feasible set in the decision space, typically defined by some constraint functions (e.g., for the multicriteria spanning tree problem, $\mathcal{X}$ is the set of all spanning trees of the graph). In this problem, any solution $x \in \mathcal{X}$ is associated with a cost vector $y(x) = (y_1(x), \dots, y_n(x)) \in \mathbb{R}^n$ where $y_i(x)$ is the evaluation of x on the i-th criterion/objective. Thus the image of the feasible set in the objective space is defined by $\{y(x) : x \in \mathcal{X}\} \subset \mathbb{R}^n$.

Solutions are usually compared through their images in the objective space (also called points) using the *Pareto dominance* relation: we say that point $u = (u_1, \dots, u_n) \in \mathbb{R}^n$ *Pareto dominates* point $v = (v_1, \dots, v_n) \in \mathbb{R}^n$ (denoted by $u \prec_P v$) if and only if $u_i \leq v_i$ for all $i \in \{1, \dots, n\}$, with at least one strict inequality. Solution $x^* \in \mathcal{X}$ is called *efficient* if there does not exist any other feasible solution $x \in \mathcal{X}$ such that $y(x) \prec_P y(x^*)$; its image in objective space is then called a non-dominated point.

3 Minimax Regret Criterion

We assume here that the DM's preferences over solutions can be represented by a parameterized scalarizing function f_ω that is linear in its parameters ω. Solution $x \in \mathcal{X}$ is preferred to solution $x' \in \mathcal{X}$ if and only if $f_\omega(y(x)) \leq f_\omega(y(x'))$. To give a few examples, function f_ω can be a weighted sum (i.e. $f_\omega(y(x)) = \sum_{i=1}^{n} \omega_i y_i(x)$) or a Choquet integral with capacity ω [9, 13]. We also assume that parameters ω are not known initially. Instead, we consider a (possibly empty) set Θ of pairs $(u, v) \in \mathbb{R}^n \times \mathbb{R}^n$ such that u is known to be preferred to v; this set can be obtained by asking preference 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 $f_\omega(u) \leq f_\omega(v)$ for all $(u, v) \in \Theta$. Thus, since f_ω is linear in ω, we can assume that Ω_Θ is a convex polyhedron throughout the paper. The problem is now to determine the most promising solution under the preference imprecision (defined by Ω_Θ). To do so, we use the minimax regret approach (e.g., [8]) which is based on the following definitions:

Definition 1 (Pairwise Max Regret) *The Pairwise Max Regret (PMR) of solution $x \in \mathcal{X}$ with respect to solution $x' \in \mathcal{X}$ is:*

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

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

Definition 2 (Max Regret) *The Max Regret (MR) of solution $x \in \mathcal{X}$ 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 selecting solution x instead of any other feasible solution $x' \in \mathcal{X}$. We can now define the minimax reget:

Definition 3 (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 $\arg\min_{x \in \mathcal{X}} MR(x, \mathcal{X}, \Omega_\Theta)$), allowing to minimize the worst-case loss. Note that if $MMR(\mathcal{X}, \Omega_\Theta) = 0$, then any optimal solution for the minimax regret criterion is necessarily optimal according to the DM's preferences.

4 An Interactive Polyhedral Method

Our aim is to produce an efficient regret-based interactive method for the determination of a (near-)optimal solution according to the DM's preferences. Note that the value $MMR(\mathcal{X}, \Omega_\Theta)$ can only decrease when inserting new preference information in Θ, as observed in previous works (see e.g., [6]). Therefore, the general idea of regret-based incremental elicitation is to ask preference queries to the DM in an iterative way, until the value $MMR(\mathcal{X}, \Omega_\Theta)$ drops below a given threshold $\delta \geq 0$ representing the maximum allowable gap to optimality; one can simply set $\delta = 0$ to obtain the preferred solution (i.e., the optimal solution according to the DM's preferences).

At each iteration step, the minimax regret $MMR(\mathcal{X}, \Omega_\Theta)$ could be obtained by computing the pairwise max regrets $PMR(x, x', \Omega_\Theta)$ for all pairs (x, x') of distinct solutions in $\mathcal{X}$ (see Definitions 2 and 3). However, this would not be very efficient in practice due to the large size of $\mathcal{X}$ (recall that $\mathcal{X}$ is the feasible set of a MOCO problem). This observation has led a group of researchers to propose a new approach consisting in combining preference elicitation and search by asking preference queries during the construction of the (near-)optimal solution (e.g., [3]). In this work, we propose to combine incremental elicitation and search in a different way: at each iteration step, we generate a set of promising solutions using the extreme points of Ω_Θ (the set of admissible parameters), we ask the DM to compare two of these solutions, we update Ω_Θ according to her answer and we stop the process whenever a (near-)optimal solution is detected (i.e. a solution $x \in \mathcal{X}$ such that $MR(x, \mathcal{X}, \Omega_\Theta) \leq \delta$ holds). More precisely, taking as input a MOCO problem P, a tolerance threshold $\delta \geq 0$, a scalarizing function f_ω with unknown parameters ω and an initial set of preference statements Θ, our algorithm iterates as follows:

1. First, the set of all extreme points of polyhedron Ω_Θ are generated. This set is denoted by EP_Θ and its kth element is denoted by ω^k.
2. Then, for every point $\omega^k \in EP_\Theta$, P is solved considering the *precise* scalarizing function f_{ω^k} (the corresponding optimal solution is denoted by x^k).
3. Finally $MMR(X_\Theta, \Omega_\Theta)$ is computed, where $X_\Theta = \{x^k : k \in \{1, \ldots, |EP_\Theta|\}\}$. If this value is strictly larger than δ, then the DM is asked to compare two solutions $x, x' \in X_\Theta$ and Ω_Θ is updated by imposing the linear constraint $f_\omega(x) \leq f_\omega(x')$ (or $f_\omega(x) \geq f_\omega(x')$ depending on her answer); the algorithm stops otherwise.

Our algorithm, called IEEP (for Incremental Elicitation based on Extreme Points), is summarized in Algorithm 1. The implementation details of `Select`, `Optimizing` and `ExtremePoints` procedures are given in the numerical section. Note however that `Optimizing` is a procedure that depends on the optimization problem (e.g., Prim algorithm could be used for the spanning tree problem). The following proposition establishes the validity of our interactive method:

Proposition 1 *For any positive tolerance threshold δ, algorithm IEEP returns a solution $x^* \in \mathcal{X}$ such that the inequality $MR(x^*, \mathcal{X}, \Omega_\Theta) \leq \delta$ holds.*

Proof. Let x^* be the returned solution and let K be the number of extreme points of Ω_Θ at the end of the execution. For all $k \in \{1, \ldots, K\}$, let ω^k be the kth extreme point of Ω_Θ and let x^k be a solution minimizing function f_{ω^k}. Let $X_\Theta = \{x^k : k \in \{1, \ldots, K\}\}$. We know that $MR(x^*, X_\Theta, \Omega_\Theta) \leq \delta$ holds at the end of the while loop (see the loop condition); hence we have $f_\omega(x^*) - f_\omega(x^k) \leq \delta$ for all solutions $x^k \in X_\Theta$ and all parameters $\omega \in \Omega_\Theta$ (see Definition 2).

We want to prove that $MR(x^*, \mathcal{X}, \Omega_\Theta) \leq \delta$ holds at the end of execution. To do so, it is sufficient to prove that $f_\omega(x^*) - f_\omega(x) \leq \delta$ holds for all $x \in \mathcal{X}$ and all $\omega \in \Omega_\Theta$. Since Ω_Θ is a convex polyhedron, for any $\omega \in \Omega_\Theta$, there exists a vector $\lambda = (\lambda_1, \ldots, \lambda_K) \in [0,1]^K$ such that $\sum_{k=1}^K \lambda^k = 1$ and $\omega = \sum_{k=1}^K \lambda_k \omega^k$. Therefore, for all solutions $x \in \mathcal{X}$ and for all parameters $\omega \in \Omega_\Theta$, we have:

$$
\begin{aligned}
f_\omega(x^*) - f_\omega(x) &= \sum_{k=1}^{K} \Big[\lambda^k (f_{\omega^k}(x^*) - f_{\omega^k}(x))\Big] \text{ by linearity} \\
&\leq \sum_{k=1}^{K} \Big[\lambda^k (f_{\omega^k}(x^*) - f_{\omega^k}(x^k))\Big] \text{ since } x^k \text{ is } f_{\omega^k}\text{-optimal} \\
&\leq \sum_{k=1}^{K} \Big[\lambda^k \times \delta\Big] \text{ since } f_\omega(x^*) - f_\omega(x^k) \leq \delta \\
&= \delta \times \sum_{k=1}^{K} \lambda^k \\
&= \delta. \qquad \square
\end{aligned}
$$

For illustration proposes, we now present the execution of our algorithm on a small instance of the multicriteria spanning tree problem.

Algorithm 1 `IEEP`

IN $\downarrow$ P: a MOCO problem; δ: a threshold; f_ω: a scalarizing function with unknown parameters ω; Θ: a set of preference statements.
OUT $\uparrow$: a solution x^* with a max regret smaller than δ.

--| Initialization of the convex polyhedron:
$\Omega_\Theta \leftarrow \{\omega : \forall (u,v) \in \Theta, f_\omega(u) \leq f_\omega(v)\}$
--| Generation of the extreme points of the polyhedron:
$EP_\Theta \leftarrow$`ExtremePoints`(Ω_Θ)
--| Generation of the optimal solutions attached to EP_Θ:
$X_\Theta \leftarrow$ `Optimizing(`P`,`EP_Θ`)`
while $MMR(X_\Theta, \Omega_\Theta) > \delta$ **do**
  --| Selection of two solutions to compare:
  $(x, x') \leftarrow$ `Select`(X_Θ)
  --| Question:
  query(x, x')
  --| Update preference information:
  if x is preferred to x' **then**
    $\Theta \leftarrow \Theta \cup \{(y(x), y(x'))\}$
  else
    $\Theta \leftarrow \Theta \cup \{(y(x'), y(x))\}$
  end
  $\Omega_\Theta \leftarrow \{\omega : \forall (u,v) \in \Theta, f_\omega(u) \leq f_\omega(v)\}$
  --| Generation of the extreme points of the polyhedron:
  $EP_\Theta \leftarrow$`ExtremePoints`(Ω_Θ)
  --| Generation of the optimal solutions attached to EP_Θ:
  $X_\Theta \leftarrow$ `Optimizing(`P`,`EP_Θ`)`
end
return a solution $x^* \in X_\Theta$ minimizing $MR(x, X_\Theta, \Omega_\Theta)$

Example 1. Consider the multicriteria spanning tree problem with 5 nodes and 7 edges given in Figure 1. Each edge is evaluated with respect to 3 criteria. Assume that the DM's preferences can be represented by a weighted sum f_ω with unknown parameters ω. Our goal is to determine an optimal spanning tree for the DM ($\delta = 0$), i.e. a connected acyclic sub-graph with 5 nodes that is f_ω-optimal. We now apply algorithm IEEP on this instance, starting with an empty set of preference statements (i.e. $\Theta = \emptyset$).

Initialization: As $\Theta = \emptyset$, Ω_Θ is initialized to the set of all weighting vectors $\omega = (\omega_1, \omega_2, \omega_3) \in [0,1]^3$ such that $\omega_1 + \omega_2 + \omega_3 = 1$. In Figure 2, Ω_Θ is represented by the triangle ABC in the space (ω_1, ω_2); value ω_3 is implicitly defined by $\omega_3 = 1 - \omega_1 - \omega_2$. Hence the initial extreme points are the vectors of the natural basis of the Euclidean space, corresponding to Pareto dominance [30]; in other words, we have $EP_\Theta = \{\omega^1, \omega^2, \omega^3\}$ with $\omega^1 = (1,0,0)$, $\omega^2 = (0,1,0)$ and $\omega^3 = (0,0,1)$. We then optimize according to all weighting vectors in EP_Θ using Prim algorithm [23], and we obtain the following three solutions: for ω^1, we have a spanning tree x^1 evaluated by $y(x^1) = (15, 17, 14)$; for ω^2, we obtain a spanning tree x^2 with $y(x^2) = (23, 8, 16)$; for ω^3, we find a spanning tree x^3 such that $y(x^3) = (17, 16, 11)$. Hence we have $X_\Theta = \{x^1, x^2, x^3\}$.

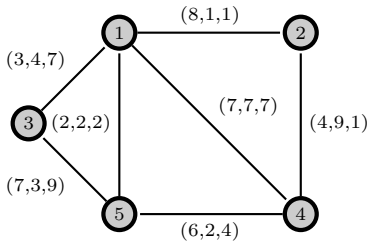

Fig. 1. A three-criteria minimum spanning tree problem.

Iteration step 1: Since $MMR(X_\Theta, \Omega_\Theta) = 8 > \delta = 0$, we ask the DM to compare two solutions in X_Θ, say x^1 and x^2. Assume that the DM prefers x^2. In that case, we perform the following updates: $\Theta = \{((23, 8, 16), (15, 17, 14))\}$ and $\Omega_\Theta = \{\omega : f_\omega(23, 8, 16) \leq f_\omega(15, 17, 14)\}$; in Figure 3, Ω_Θ is represented by triangle BFE. We then compute the set EP_Θ of its extreme points (by applying the algorithm in [11] for example) and we obtain $EP_\Theta = \{\omega^1, \omega^2, \omega^3\}$ with $\omega^1 = (0.53, 0.47, 0)$, $\omega^2 = (0, 0.18, 0.82)$ and $\omega^3 = (0, 1, 0)$. We optimize according to these weights and we obtain three spanning trees: $X_\Theta = \{x^1, x^2, x^3\}$ with $y(x^1) = (23, 8, 16)$, $y(x^2) = (17, 16, 11)$ and $y(x^3) = (19, 9, 14)$.

Iteration step 2: Here $MMR(X_\Theta, \Omega_\Theta) = 1.18 > \delta = 0$. Therefore, we ask the DM to compare two solutions in X_Θ, say x^1 and x^2. Assume she prefers x^2. We then obtain $\Theta = \{((23, 8, 16), (15, 17, 14)), ((17, 16, 11), (23, 8, 16))\}$ and we set $\Omega_\Theta = \{\omega : f_\omega(23, 8, 16) \leq f_\omega(15, 17, 14) \wedge f_\omega(17, 16, 11) \leq f_\omega(23, 8, 16)\}$. We compute the corresponding extreme points which are given by $EP_\Theta = \{(0.43, 0.42, 0.15), (0, 0.18, 0.82), (0, 0.38, 0.62)\}$ (see triangle HGE in Figure 4); finally we have $X_\Theta = \{x^1, x^2\}$ with $y(x^1) = (17, 16, 11)$ and $y(x^2) = (19, 9, 14)$.

Iteration step 3: Now $MMR(X_\Theta, \Omega_\Theta) = 1.18 > \delta = 0$. Therefore we ask the DM to compare x^1 and x^2. Assuming that she prefers x^2, we update Θ by inserting the preference statement $((19, 9, 14), (17, 16, 11))$ and we update Ω_Θ by imposing the following additional constraint: $f_\omega(19, 9, 14) \leq f_\omega(17, 16, 11)$ (see Figure 5); the corresponding extreme points are given by $EP_\Theta = \{(0.18, 0.28, 0.54), (0, 0.3, 0.7), (0, 0.38, 0.62), (0.43, 0.42, 0.15)\}$. Now the set X_Θ only includes one spanning tree x^1 and $y(x^1) = (19, 9, 14)$. Finally, the algorithm stops (since we have $MMR(X_\Theta, \Omega_\Theta) = 0 \leq \delta = 0$) and it returns solution x^1 (which is guaranteed to be the optimal solution for the DM).

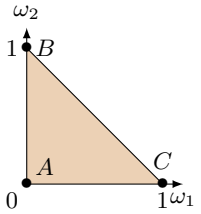

Fig. 2. Initial set.

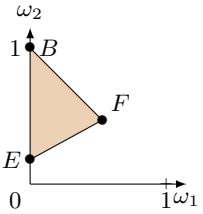

Fig. 3. After step 1.

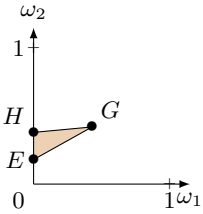

Fig. 4. After step 2.

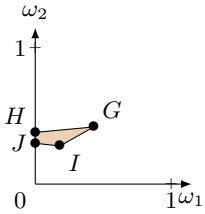

Fig. 5. After step 3.

5 Experimental Results

We now provide numerical results aiming to evaluate the performance of our interactive approach. At each iteration step of our procedure, the DM is asked to compare two solutions selected from the set X_Θ until $MR(X_\Theta, \Omega_\Theta) \leq \delta$. Therefore, we need to estimate the impact of procedure `Select` on the performances of our algorithm. Here we consider the following query generation strategies:

- **Random:** The two solutions are randomly chosen in X_Θ.
- **Max-Dist:** We compute the Euclidean distance between all solutions in the objective space and we choose a pair of solutions maximizing the distance.
- **CSS:** The Current Solution Strategy (CSS) consists in selecting a solution that minimizes the max regret and one of its adversary's choice [8][1].

These strategies are compared using the following indicators:

- **time:** The running time given in seconds.
- **eval:** The number of evaluations, i.e. the number of problems with known preferences that are solved during the execution; recall that we solve one optimization problem per extreme point at each iteration step (see `Optimizing`).
- **queries:** The number of preference queries generated during the execution.
- **qOpt:** The number of preference queries generated until the determination of the preferred solution (but not yet proved optimal).

We assume here that the DM's preferences can be represented by a weighted sum f_ω but the weights $\omega = (\omega_1, \ldots, \omega_n)$ are not known initially. More precisely, we start the execution with an empty set of preference statements (i.e. $\Theta = \emptyset$ and $\Omega_\Theta = \{\omega \in \mathbb{R}^n_+ : \sum_{i=1}^n \omega_i = 1\}$) and then any new preference statement $(u, v) \in \mathbb{R}^2$ obtained from the DM induces the following linear constraint over the weights: $\sum_{i=1}^n \omega_i u_i \leq \sum_{i=1}^n \omega_i v_i$. Hence Ω_Θ is a convex polyhedron. In our experiments, the answers to queries are simulated using a weighting vector ω randomly generated before running the algorithm, using the procedure presented in [26], to guarantee a uniform distribution of the weights.

Implementation Details. Numerical tests were performed on a Intel Core i7-7700, at 3.60GHz, with a program written in C. At each iteration step of our algorithm, the extreme points associated to the convex polyhedron Ω_Θ are generated using the `polymake` library[2]. Moreover, at each step, we do not compute PMR values using a linear programming solver. Instead, we only compute score differences since the maximum value is always obtained for an extreme point of the convex polyhedron. Furthermore, to reduce the number of PMR computations, we use Pareto dominance tests between the extreme points to eliminate dominated solutions, as proposed in [21].

[1] Note that these three strategies are equivalent when only considering two objectives since the number of extreme points is always equal to two in this particular case.

[2] https://polymake.org

5.1 Multicriteria spanning tree

In these experiments, we consider instances of the multicriteria spanning tree (MST) problem, which is defined by a connected graph $G = (V, E)$ where each edge $e \in E$ is valued by a cost vector giving its cost with respect to different criteria/objectives (every criterion is assumed to be additive over the edges). A spanning tree of G is a connected sub-graph of G which includes every vertex $v \in V$ while containing no cycle. In this problem, $\mathcal{X}$ is the set of all spanning trees of G. We generate instances of $G = (V, E)$ with a number of vertices $|V|$ varying between 50 and 100 and a number of objectives n ranging from 2 to 6. The edge costs are drawn within $\{1, \ldots, 1000\}^n$ uniformly at random. For the MST problem, procedure `Optimizing`(P, EP_Θ) proceeds as follows: First, for all extreme points $\omega^k \in EP_\Theta$, an instance of the spanning tree problem with a single objective is created by simply aggregating the edge costs of G using weights ω^k. Then, Prim algorithm is applied on the resulting graphs. The results obtained by averaging over 30 runs are given in Table 1 for $\delta = 0$.

		IEEP - Random				IEEP - Max-Dist				IEEP - CSS			
n	$\|V\|$	time(s)	queries	eval	qOpt	time(s)	queries	eval	qOpt	time(s)	queries	eval	qOpt
2	50	8.6	**7.4**	**9.4**	**4.6**	8.0	**7.4**	**9.4**	**4.6**	**7.7**	**7.4**	**9.4**	**4.6**
3	50	16.9	16.2	34.9	10.9	**16.5**	**15.2**	**33.1**	**10.2**	17.9	16.9	35.9	12.0
4	50	27.5	25.7	117.3	19.7	**26.4**	**24.6**	**112.3**	**17.2**	30.7	28.9	130.8	20.1
5	50	37.7	35.0	363.2	27.2	**36.2**	**34.3**	**358.4**	**23.3**	42.3	39.8	404.7	30.6
6	50	46.1	43.3	**1056.3**	35.3	**45.5**	**42.7**	1075.2	**32.8**	62.6	57.6	1537.9	43.6
2	100	10.0	**8.6**	**10.6**	**5.7**	**8.9**	**8.6**	**10.6**	**5.7**	9.2	**8.6**	**10.6**	**5.7**
3	100	**18.7**	17.6	37.8	14.0	19.0	**17.4**	**37.2**	13.9	19.0	17.7	37.7	**13.0**
4	100	32.0	29.9	134.0	23.3	**30.1**	**28.4**	**129.9**	**22.3**	34.8	32.5	147.0	24.1
5	100	**41.8**	39.8	**404.4**	31.3	42.1	**39.2**	411.5	**31.0**	55.9	51.7	564.8	40.6
6	100	55.9	51.5	1306.1	40.0	**52.3**	**49.1**	**1259.3**	**38.7**	84.0	75.7	2329.6	62.1

Table 1. MST: comparison of the different query strategies (best values in bold).

Running time and number of evaluations. We observe that Random and Max-Dist strategies are much faster than CSS strategy; for instance, for $n = 6$ and $|V| = 100$, Random and Max-Dist strategies end before one minute whereas CSS needs almost a minute and a half. Note that time is mostly consumed by the generation of extreme points, given that the evaluations are performed by Prim algorithm which is very efficient. Since the number of evaluations with CSS drastically increases with the size of the problem, we may expect the performance gap between CSS and the two other strategies to be much larger for MOCO problems with a less efficient solving method.

Number of generated preference queries. We can see that Max-Dist is the best strategy for minimizing the number of generated preference queries. More precisely, for all instances, the preferred solution is detected with less than 40 queries and the optimality is established after at most 50 queries. In fact, we can reduce even further the number of preference queries by considering a strictly positive tolerance threshold; to give an example, if we set $\delta = 0.1$ (i.e. 10% of the "maximum" error computed using the ideal point and the worst objective vector), then our algorithm combined with Max-Dist strategy generates at most 20 queries in all considered instances. In Table 1, we also observe that CSS strategy generates many more queries than Random, which is quite surprising since CSS strategy

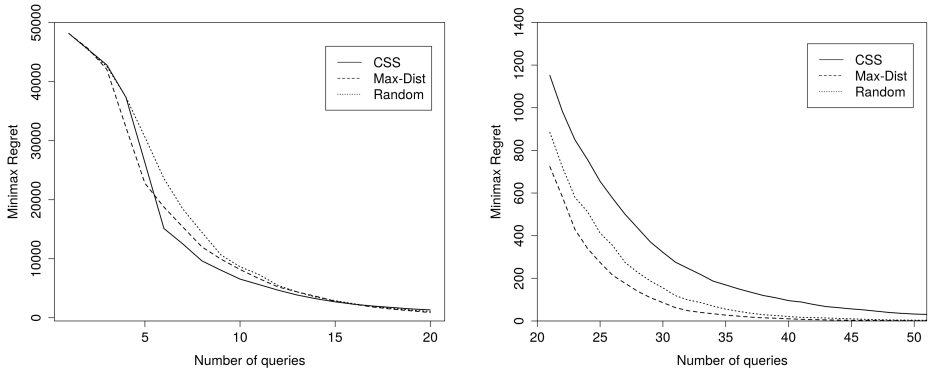

Fig. 6. MST problem with $n = 6$ and $|V| = 100$: evolution of the minimax regret between 1 and 20 queries (left) and between 21 and 50 queries (right).

is intensively used in incremental elicitation (e.g., [5, 8]). To better understand this result, we have plotted the evolution of minimax regret with respect to the number of queries for the bigger instance of our set ($|V| = 100$, $n = 6$). We have divided the figure in two parts: the first part is when the number of queries is between 1 and 20 and the other part is when the number of queries is between 20 and 50 (see Figure 6). In the first figure, we observe that there is almost no difference between the three strategies, and the minimax regret is already close to 0 after only 20 questions (showing that we are very close to the optimum relatively quickly). However, there is a significant difference between the three strategies in the second figure: the minimax regret with CSS starts to reduce less quickly after 30 queries, remaining strictly positive after 50 queries, whereas the optimal solution is found after about 40 queries with the other strategies. Thus, queries generated with CSS gradually becomes less and less informative than those generated by the two other strategies. This can be explained by the following: CSS always selects the minimax regret optimal solution and one of its worst adversary. Therefore, when the minimax regret optimal solution does not change after asking a query, the same solution is used for the next preference query. This can be less informative than asking the DM to compare two solutions for which we have no preference information at all; Random and Max-Dist strategies select the two solutions to compare in a more diverse way.

Comparison with the State-of-the-Art Method. In this subsection, we compare our interactive method with the state-of-the-art method proposed in [3]. The latter consists essentially in integrating incremental elicitation into Prim algorithm [23]; therefore, this method will be called IE-Prim hereafter. The main difference between IE-Prim and IEEP is that IE-Prim is constructive: queries are not asked on complete solutions but on partial solutions (edges of the graph). We have implemented ourselves IE-Prim, using the same programming language and data structures than IEEP, in order to allow a fair comparison between these

methods. Although IE-Prim was only proposed and tested with CSS in [3], we have integrated the two other strategies (i.e., Max-Dist and Random) in IE-Prim.

		IEEP - Max-Dist		IE-Prim - Random		IE-Prim - Max-Dist		IE-Prim - CSS	
n	$\|V\|$	time(s)	queries	time(s)	queries	time(s)	queries	time(s)	queries
2	50	**8.0**	**7.4**	13.3	12.3	12.1	11.2	13.0	12.3
3	50	**16.5**	**15.2**	28.6	26.7	26.1	24.5	31.9	29.6
4	50	**26.4**	**24.6**	45.0	42.1	42.5	39.7	55.6	50.8
5	50	**36.2**	**34.3**	59.7	55.5	56.9	53.2	80.4	73.4
6	50	**45.5**	**42.7**	78.7	73.4	79.4	73.5	117.8	108.1
2	100	**8.9**	**8.6**	15.9	15.1	14.6	13.6	16.1	15.0
3	100	**19.0**	**17.4**	34.6	32.4	33.6	31.1	36.9	35.3
4	100	**30.1**	**28.4**	55.6	51.6	54.7	51.2	66.6	61.6
5	100	**42.1**	**39.2**	75.4	70.7	76.4	71.7	103.7	95.3
6	100	**52.3**	**49.1**	103.7	96.0	100.3	93.5	162.3	146.2

Table 2. MST: comparison between IEEP and IE-Prim (best values in bold).

In Table 2, we compare IEEP with Max-Dist and IE-Prim in terms of running times and number of queries[3]. We see that IEEP outperforms IE-Prim in all settings, allowing the running time and the number of queries to be divided by three in our biggest instances. Note that Max-Dist and Random strategies improve the performances of IE-Prim (compared to CSS), but it is still not enough to achieve results comparable to IEEP. This shows that asking queries during the construction of the solutions is less informative than asking queries using the extreme points of the polyhedron representing the preference uncertainty.

Now we want to estimate the performances of our algorithm seen as an anytime algorithm (see Figure 7). For each iteration step i, we compute the error obtained when deciding to return the solution that is optimal for the minimax regret criterion at step i (i.e., after i queries); this error is here expressed in terms of percentage from the optimal solution. For the sake of comparison, we also include the results obtained with IE-Prim. However IE-Prim cannot be seen as an anytime algorithm since it is constructive. Therefore, to vary the number of queries, we used different tolerance thresholds: $\delta = 0.3$, 0.2, 0.1, 0.05 and 0.01.

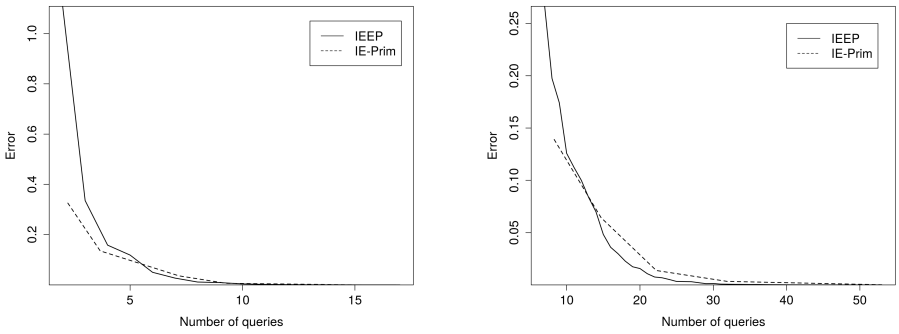

Fig. 7. MST problem with $|V| = 100$: Comparison of the errors with respect to the number of queries for $n = 3$ (left) and for $n = 6$ (right).

[3] Note that we cannot compute qOpt and eval for IE-Prim since it is constructive and makes no evaluation.

In Figure 7, we observe that the error drops relatively quickly for both procedures. Note however that the error obtained with IE-Prim is smaller than with IEEP when the number of queries is very low. This may suggest to favor IE-Prim over IEEP whenever the interactions are very limited and time is not an issue.

5.2 Multicriteria traveling salesman problem

We now provide numerical results for the multicriteria traveling salesman problem (MTSP). In our tests, we consider existing Euclidean instances of the MTSP with 50 and 100 cities, and $n = 2$ to 6 objectives[4]. Moreover, we use the exact solver Concorde[5] to perform the optimization part of IEEP algorithm (see procedure `Optimizing`). Contrary to the MST, there exist no interactive constructive algorithms to solve the MTSP. Therefore, we only provide the results obtained by our algorithm IEEP with the three proposed query generation strategies (namely Random, Max-Dist and CSS). The results obtained by averaging over 30 runs are given in Table 3 for $\delta = 0$.

In this table, we see that Max-Dist remains the best strategy for minimizing the number of generated preference queries. Note that the running times are much higher for the MTSP than for the MST (see Table 1), as the traveling salesman problem is much more difficult to solve exactly with known preferences.

		IEEP - Random				IEEP - Max-Dist				IEEP - CSS			
n	$\|V\|$	time(s)	queries	eval	qOpt	time(s)	queries	eval	qOpt	time(s)	queries	eval	qOpt
2	50	**8.0**	**6.3**	**8.3**	**3.7**	8.8	**6.3**	**8.3**	**3.7**	10.0	**6.3**	**8.3**	**3.7**
3	50	**21.2**	14.3	31.3	10.0	23.5	**13.3**	**29.5**	**9.5**	24.5	14.9	32.4	10.6
4	50	**38.7**	22.6	101.5	**16.0**	50.2	**20.7**	**93.6**	16.2	67.7	24.2	109.1	16.9
5	50	210.9	31.2	331.1	22.7	**95.1**	**28.6**	**304.8**	**19.2**	137.1	38.5	387.7	23.9
6	50	390.8	41.0	1044.5	26.2	**238.8**	**37.3**	**949.3**	**24.4**	584.9	58.4	1531.0	28.9
2	100	12.2	**7.6**	**9.6**	**4.3**	**11.3**	**7.6**	**9.6**	**4.3**	19.1	**7.6**	**9.6**	**4.3**
3	100	28.3	15.9	34.7	12.4	**27.3**	**15.4**	**33.7**	12.1	42.2	16.5	35.6	**11.7**
4	100	73.1	26.7	121.1	20.0	**69.9**	**25.4**	**115.8**	**18.1**	94.8	28.4	124.9	19.6
5	100	241.9	36.4	**380.6**	27.3	**237.0**	**35.5**	383.0	**24.4**	361.8	44.7	481.3	31.2
6	100	981.2	45.0	1106.8	32.8	**586.3**	**41.7**	**1014.5**	**30.2**	1618.3	68.8	1865.3	39.2

Table 3. MTSP: comparison of the different query strategies (best values in bold)

6 Conclusion and perspectives

In this paper, we have proposed a general method for solving multi-objective combinatorial optimization problems with unknown preference parameters. The method is based on a sharp combination of 1) regret-based incremental preference elicitation and 2) the generation of promising solutions using the extreme points of the polyhedron representing the admissible preference parameters; several query generation strategies have been proposed in order to improve its performances. We have shown that our method returns the optimal solution according to the DM's preferences. Our method has been tested on the multicriteria spanning tree and multicriteria traveling salesman problems until 6 criteria

[4] https://eden.dei.uc.pt/ paquete/tsp/
[5] http://www.math.uwaterloo.ca/tsp/concorde

and 100 vertices. We have provided numerical results showing that our method achieves better results than IE-Prim (the state-of-the-art method for the MST problem) both in terms of number of preference queries and running times.

Thus, in practice, our algorithm outperforms IE-Prim which is an algorithm that runs in polynomial time and generates no more than a polynomial number of queries. However, our algorithm does not have these performance guarantees. More precisely, the performances of our interactive method strongly depend on the number of extreme points at each iteration step, which can be exponential in the number of criteria (see e.g., [27]). Therefore, the next step could be to identify an approximate representation of the polyhedron which guarantees that the number of extreme points is always polynomial, while still being able to determine a (near-)optimal solution according to the DM's preferences.

References

1. Benabbou, N., Di Sabatino Di Diodoro, S., Perny, P., Viappiani, P.: Incremental preference elicitation in multi-attribute domains for choice and ranking with the Borda count. In: Proceedings of SUM'16. pp. 81–95 (2016)
2. Benabbou, N., Perny, P.: Combining preference elicitation and search in multiobjective state-space graphs. In: Proceedings of IJCAI'15. pp. 297–303 (2015)
3. Benabbou, N., Perny, P.: On possibly optimal tradeoffs in multicriteria spanning tree problems. In: Proceedings of ADT'15. pp. 322–337 (2015)
4. Benabbou, N., Perny, P.: Solving multi-agent knapsack problems using incremental approval voting. In: Proceedings of ECAI'16. pp. 1318–1326 (2016)
5. Benabbou, N., Perny, P.: Interactive resolution of multiobjective combinatorial optimization problems by incremental elicitation of criteria weights. EURO journal on decision processes (2018)
6. Benabbou, N., Perny, P., Viappiani, P.: Incremental elicitation of Choquet capacities for multicriteria choice, ranking and sorting problems. Artificial Intelligence **246**, 152180 (2017)
7. Bourdache, N., Perny, P.: Active preference elicitation based on generalized Gini functions: Application to the multiagent knapsack problem. In: Proceedings of AAAI'19 (2019)
8. Boutilier, C., Patrascu, R., Poupart, P., Schuurmans, D.: Constraint-based Optimization and Utility Elicitation using the Minimax Decision Criterion. Artifical Intelligence **170**(8–9), 686–713 (2006)
9. Choquet, G.: Theory of capacities. Annales de lInstitut Fourier **5**, 31–295 (1953)
10. Drummond, J., Boutilier, C.: Preference elicitation and interview minimization in stable matchings. In: Proceedings of AAAI'14. pp. 645–653 (2014)
11. Dyer, M., Proll, L.: An algorithm for determining all extreme points of a convex polytope. Mathematical Programming pp. 12–81 (1977)
12. Gilbert, H., Spanjaard, O., Viappiani, P., Weng, P.: Reducing the number of queries in interactive value iteration. In: Proceedings of ADT'15. pp. 139–152 (2015)
13. Grabisch, M., Labreuche, C.: A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. Annals of Operations Research **175**(1), 247–286 (2010)
14. Hamacher, H., Ruhe, G.: On spanning tree problems with multiple objectives. Annals of Operations Research **52**, 209–230 (1994)

15. Kaddani, S., Vanderpooten, D., Vanpeperstraete, J.M., Aissi, H.: Weighted sum model with partial preference information: application to multi-objective optimization. European Journal of Operational Research **260**, 665–679 (2017)
16. Karasakal, E., Köksalan, M.: Generating a representative subset of the nondominated frontier in multiple criteria. Operations Research **57**(1), 187–199 (2009)
17. Korhonen, P.: Multiple criteria decision analysis. chap. Interactive Methods. Greco, Salvatore and Figueira, J and Ehrgott, M. Springer (2005)
18. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society **7**, 48–50 (1956)
19. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evol. Comput. **10**(3), 263–282 (2002)
20. Lust, T., Rolland, A.: Choquet optimal set in biobjective combinatorial optimization. Computers & OR **40**(10), 2260–2269 (2013)
21. Marinescu, R., Razak, A., Wilson, N.: Multi-objective constraint optimization with tradeoffs. In: Proceedings of CP'13. pp. 497–512. Springer (2013)
22. Marinescu, R., Razak, A., Wilson, N.: Multi-objective influence diagrams with possibly optimal policies. In: Proceedings of AAAI'17. pp. 3783–3789 (2017)
23. Prim, R.C.: Shortest connection networks and some generalizations. Bell System Technical Journal **36**, 1389–1401 (1957)
24. White III, C.C., Sage, A.P., Dozono, S.: A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. IEEE Transactions on Systems, Man, and Cybernetics **14**(2), 223–229 (1984)
25. Regan, K., Boutilier, C.: Eliciting additive reward functions for markov decision processes. In: Proceedings of IJCAI'11. pp. 2159–2164 (2011)
26. Rubinstein, R.: Generating random vectors uniformly distributed inside and on the surface of different regions. European Journal of Operational Research **10**(2), 205 – 209 (1982)
27. Schrijver, A.: Combinatorial Optimization - Polyhedra and Efficiency. Springer (2003)
28. Wang, T., Boutilier, C.: Incremental Utility Elicitation with the Minimax Regret Decision Criterion. pp. 309–316 (2003)
29. Weng, P., Zanuttini, B.: Interactive Value Iteration for Markov Decision Processes with Unknown Rewards. In: Proceedings of IJCAI'13. pp. 2415–2421 (2013)
30. Wiecek, M.M.: Advances in cone-based preference modeling for decision making with multiple criteria. Decision Making in Manufacturing and Services **Vol. 1, no. 1-2**, 153–173 (2007)
31. Wilson, N., Razak, A., Marinescu, R.: Computing possibly optimal solutions for multi-objective constraint optimisation with tradeoffs. In: Proceedings of IJCAI'15. pp. 815–822 (2015)