Bayesian preference elicitation for multiobjective combinatorial optimization

Nadjet Bourdache, Patrice Perny, Olivier Spanjaard

Bayesian preference elicitation for multiobjective combinatorial optimization

Nadjet Bourdache[1] and **Patrice Perny**[1] and **Olivier Spanjaard**[1]

Abstract. We introduce a new incremental preference elicitation procedure able to deal with noisy responses of a Decision Maker (DM). The originality of the contribution is to propose a Bayesian approach for determining a preferred solution in a multiobjective decision problem involving a *combinatorial* set of alternatives. The preferences of the DM are represented by an aggregation function whose parameters are unknown. The uncertainty about these parameters is represented by a density function on the parameter space. Pairwise comparison queries are used to reduce this uncertainty by Bayesian revision. The query selection strategy is based on the solution of a mixed integer linear program with a combinatorial set of variables and constraints, which requires to use columns and constraints generation methods. Numerical tests are provided to show the practicability of the approach.

1 Introduction

The increasing complexity of problems encountered in applications is a permanent motivation for the development of intelligent systems for human decision support. Among the various difficulties to overcome for decision making in complex environments we consider here three sources of complexity that often coexist in a decision problem: 1) the combinatorial nature of the set of feasible alternatives 2) the fact that multiple points of view, possibly conflicting, about the value of solutions may coexist, 3) the need of formulating recommendations that are tailored to the objectives and preferences of users and that take into account the uncertainty in preference elicitation (due to possible mistakes in the responses of users to preference queries).

The first difficulty occurs as soon as the solutions to be compared are characterized by the combinations of elementary decisions. This is the case for instance for the selection problem of an optimal subset within a reference set under a budget constraint (a.k.a. knapsack problem), where a solution is characterized by elementary decisions concerning items of the reference set. This difficulty prevents the explicit evaluation of all solutions and the determination of the best option requires implicit enumeration techniques. The second difficulty appears in multiagent decision contexts when the agents have different individual value systems or objectives leading to possibly conflicting preferences. It also appears in single-agent decision contexts when the alternatives are assessed w.r.t. different criteria. Finally, it appears in decision under uncertainty when several scenarios that have different impacts on the outcomes of the alternatives are considered. In all these situations, preference modeling requires the definition of multiple objectives to be optimized simultaneously. The combination of difficulties 1 and 2 is at the core of multiobjective combinatorial optimization [12].

Let us now come to the third difficulty. The coexistence of multiple objectives makes the notion of optimality subjective and requires additional preference information to be collected from the users in order to discriminate between Pareto-optimal solutions. In multiobjective decision problems, the "optimal" solution fully depends on the relative importance attached to the different objectives under consideration and on how performances are aggregated. A standard tool used to generate compromise solutions tailored to the decision maker (DM) value system is to optimize a parameterized aggregation function summarizing the performance vector of any solution into a scalar value. This makes it possible to reformulate the initial problem as a single-objective optimization problem (see e.g., [22]). However, a precise specification of the preference parameters (e.g., weighting coefficients), prior to the exploration of the set of alternatives, may be cumbersome because it requires a significant amount of preference information. To overcome this problem, incremental decision procedures aiming to integrate and combine the elicitation of preference parameters and the exploration of the set of feasible solutions are appealing. They make it possible to focus the elicitation burden on the information that is really useful to separate competing solutions during the optimization process, and this significantly reduces the number of queries asked to the user.

In the fields of operations research and artificial intelligence, numerous contributions have addressed the problem of incrementally eliciting preferences. A first stream of research concerns preference elicitation for decision making in explicit sets (i.e., non-combinatorial problems), to assess multiattribute utility functions [28], weights of criteria in aggregation functions [4], utility functions for decision making under risk [10, 26, 16, 19], or individual utilities in collective decision making [17]. Preference elicitation for decision support on combinatorial domains is a challenging issue that has also been studied in various contexts such as constraint satisfaction [13], matching under preferences [11], sequential decision making under risk [20, 27, 14, 2], and multiobjective combinatorial optimization [9, 3, 6].

Almost all incremental elicitation procedures mentioned above proceed by progressive reduction of the parameter space until an optimal decision can be identified. At every step of the elicitation process, a preference query is asked to the DM and the answer induces a constraint on the parameter space, thus a polyhedron including all parameter values compatible with the DM's responses is updated after each answer (*polyhedral method* [24]). Queries are selected to obtain a fast reduction of the parameter space, in order to enforce a fast determination of the optimal solution. However, such procedures do not offer any opportunity to the DM to revise her opinion about

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

alternatives and the final result may be sensitive to errors in preference statements.

A notable exception in the list of contributions mentioned above is the approach proposed by Chajewska et al. [10]. The approach relies on a prior probabilistic distribution over the parameter space and uses preference queries over gambles to update the initial distribution using Bayesian methods. It is more tolerant to errors and inconsistencies over time in answering preference queries. The difficulties with this approach may lie in the choice of a prior distribution and in the computation of Bayesian updates at any step of the procedure. A variant, proposed in [15], relies on simpler questions under certainty, so as to reduce the cognitive load.

Motivation of the paper As far as we know, the works mentioned in the last paragraph has not been extended for decision making on combinatorial domains. Our goal here is to fill the gap and to propose a Bayesian approach for determining a preferred solution in a multiobjective combinatorial optimization problem. The main issue in this setting is the determination of the next query to ask to the DM, as there is an exponential number of possible queries (due to the combinatorial nature of the set of feasible solutions).

Related work Several recently proposed Bayesian preference elicitation methods may be related to our work.

– Sauré and Vielma [21] proposed an error tolerant variant of the polyhedral method, where the polyhedron is replaced by an ellipsoidal credibility region computed from a multivariate normal distribution on the parameter space. This distribution, and thus the ellipsoidal credibility region, is updated in a Bayesian manner after each query. In contrast with their work, where the set of alternatives is explicitly defined, our method applies on implicit sets of alternatives. Besides, although our method also involves a multivariate normal density function on the parameter space, our query selection strategy is based on the whole density function and not only on a credibility region.

– Vendrov et al. [25] proposed a query selection procedure able to deal with large sets of alternatives (up to hundreds of thousands) based on *Expected Value Of Information* (EVOI). The EVOI criterion consists in determining a query maximizing the expected utility of the recommended alternative conditioned on the DM's answer (where the probability of each answer depends on a response model, e.g. the logistic response model). However, the subsequent optimization problem becomes computationally intractable with a large set of alternatives. The authors consider a continuous relaxation of the space of alternatives that allows a gradient-based approach. Once a query is determined in the relaxed space, the corresponding pair of fictive alternatives is projected back into the space of feasible alternatives. In addition, a second contribution of the paper is to propose an elicitation strategy based on *partial comparison queries*, i.e. queries involving partially specified multi-attribute alternatives, which limits the cognitive burden when the number of attributes is large. We tackle here another state-of-the-art query selection strategy that aims at minimizing the max expected regret criterion (instead of maximizing the EVOI criterion), a popular measure of recommendation quality.

– In a previous work [7], we introduced an incremental elicitation method based on Bayesian linear regression for assessing the weights of rank-dependent aggregation functions used in decision theory (typically OWA and Choquet integrals). The query selection strategy we proposed is based on the min max expected regret criterion, similarly to the one we use in the present work. However, the method can only be applied to *explicit* sets of alternatives and does not scale to combinatorial domains. The computation of expected regrets in the provided procedure (in order to determine the next query) requires indeed the enumeration of all possible pairs of solutions for each query, which is impractical if the set of solutions is combinatorial in nature. In order to scale to such large problems, we propose a method based on mixed integer linear programming that allows us to efficiently compute expected regrets on combinatorial domains.

Organization of the paper In Section 2, we describe the incremental elicitation procedure proposed in [7] for the determination of an optimal choice over *explicit* sets of solutions and we point out the main issues to overcome to extend the approach to combinatorial domains. Section 3 is devoted to the method we propose to compute expected regrets on combinatorial domains. Numerical tests are presented in Section 4 to show the efficiency of the proposed procedure.

2 Incremental elicitation

We first provide a general overview of the incremental Bayesian elicitation procedure on an *explicit set* and then discuss the extension to cope with *combinatorial optimization* problems.

Let $\mathcal{X}$ denote the set of possible solutions. Since we are in the context of multiobjective optimization, we assume that a utility vector $u(x) \in \mathbb{R}^n$ is assigned to any solution $x \in \mathcal{X}$. Then we consider the problem of maximizing over $\mathcal{X}$ a scalarizing function of the form $f_w(x) = \sum_{k=1}^{n} w_k g_k(u(x))$ where w_k are positive weights and $g_k : \mathbb{R}^n \to \mathbb{R}$ are *basis functions* [5] (introduced to extend the class of linear models to nonlinear ones). For simplicity, the reader can assume that $g_k(u(x)) = u_k(x)$, i.e., the k-th component of $u(x)$, and w_k is the (imperfectly known) weight of criterion k in the following. At the start of the procedure, a prior density function p is associated to the parameter space $W = \{w \in [0,1]^n | \sum_k w_k = 1\}$, where the unknown weighting vector w takes value. Then, at each step, the DM responds to a pairwise comparison query and, based on this new preference information, the density function is updated in a Bayesian manner. The aim is, in a minimum number of queries, to acquire enough information about the weighting vector w to be able to recommend a solution $x \in \mathcal{X}$ that is near optimal. We present here the three main parts of the decision process: query selection strategy, Bayesian updating after each query and stopping condition.

2.1 Query selection strategy

At each step of the algorithm, a new preference statement is needed to update the density function on W. In order to select an informative query, we use an adaptation of the *Current Solution Strategy* (CSS) introduced in [8] and based on regret minimization. In our probabilistic setting, regrets are replaced by *expected* regrets. Before describing more precisely the query selection strategy, we recall some definitions about expected regrets [7].

Definition 1. *Given a set $\mathcal{X}$ of solutions and a density function p on W, the pairwise expected regret of the pair $(x, y) \in \mathcal{X}^2$, the max expected regret of $x \in \mathcal{X}$ and the minimax expected regret over $\mathcal{X}$ are defined by:*

$$\mathrm{PER}(x, y, p) = \int_{w \in W} \max\{0, f_w(y) - f_w(x)\} p(w) dw,$$

$$\mathrm{MER}(x, \mathcal{X}, p) = \max_{y \in \mathcal{X}} \mathrm{PER}(x, y, p),$$

$$\mathrm{MMER}(\mathcal{X}, p) = \min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, p).$$

In other words, the Pairwise Expected Regret (PER) of x with respect to y represents the expected utility loss when recommending solution x instead of solution y, the Max Expected Regret (MER) of x is the maximum expected utility loss incurred in selecting x in $\mathcal{X}$, while the MiniMax Expected Regret (MMER) is the minimal max expected regret value of a solution in $\mathcal{X}$. In practice, PER, MER and MMER values are approximated using a sample S of weighting vectors drawn from p. This discretization of W enables us to convert the integral into an arithmetic mean:

$$\mathrm{PER}(x, y, S) = \frac{1}{|S|} \sum_{w \in S} \max\{0, f_w(y) - f_w(x)\} \quad (1)$$

$$\mathrm{MER}(x, \mathcal{X}, S) = \max_{y \in \mathcal{X}} \mathrm{PER}(x, y, S) \quad (2)$$

$$\mathrm{MMER}(\mathcal{X}, S) = \min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, S) \quad (3)$$

We can now describe the adaptation of CSS to the probabilistic setting. The max expected regret of a solution is used to determine which solution to recommend in $\mathcal{X}$ (the lower, the better) in the current state of knowledge characterized by p. At a step i of the elicitation procedure, if the stopping condition (that will be defined below) is met, then a solution $x^{(i)} \in \arg\min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, S)$ is recommended. Otherwise, if the knowledge about the value of w needs to be better specified to make a recommendation, the DM is asked to compare $x^{(i)}$ to its best challenger $y^{(i)} \in \arg\max_{y \in \mathcal{X}} \mathrm{PER}(x^{(i)}, y, S)$ (best challenger in the current state of knowledge). In the next subsection, we describe how one uses the DM's answer to update the density function p.

Note that we could swap the expectation and the maximization in the max regret computation, i.e., define an expected max regret by $\mathrm{EMR}(x, \mathcal{X}, p) = \int_{w \in W} \max\{f_w(y) - f_w(x) : y \in \mathcal{X}\} p(w) dw$ instead of the proposed MER. However, EMR has two main disadvantages: 1) minimizing EMR amounts to maximize the expected value (and thus does not actually define a regret) while minimizing MER allows to determine a minimal worst case loss, 2) $\mathrm{EMR}(x, \mathcal{X}, p)$ does not allow to identify a *feasible* best challenger (and thus to define a strategy to select the next preference query) because it compares x to a fictive solution $\hat{y}$ maximizing f_w for any value of w.

2.2 Bayesian updating

At step i of the procedure, a new query of the form "$x^{(i)} \succsim y^{(i)}$?" is asked to the DM. Her answer is translated into a binary variable $a^{(i)}$ that takes value 1 if the answer is yes and 0 otherwise. Using Bayes' rule, the posterior density function $p(w|a^{(i)})$ satisfies:

$$p(w|a^{(i)}) \propto p(w) p(a^{(i)}|w) \quad (4)$$

where $p(w)$ is assumed to be multivariate Gaussian (the initialization used for $p(w)$ will be specified in the numerical tests section). The posterior density function $p(w|a^{(i)})$ is hard to compute analytically using Equation 4. Indeed, the likelihood $p(a^{(i)}|w)$ follows a Bernoulli distribution and no conjugate prior is known for this likelihood function in the multivariate case. To overcome this difficulty, one uses a data augmentation method [1] that consists in introducing a latent variable $z^{(i)} = w^T(u(x^{(i)}) - u(y^{(i)})) + \varepsilon^{(i)}$ that represents a noisy scalarized difference between the two compared solutions, where $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma)$ is a Gaussian noise accounting for the uncertainty about the DM's answer. Using this latent variable, the posterior distribution $p(w|a^{(i)})$ is formulated as:

$$p(w|a^{(i)}) = \int p(w, z|a^{(i)}) dz = \int p(w|z) p(z|a^{(i)}) dz \quad (5)$$

If the prior density $p(w)$ is multivariate Gaussian then $p(w|a^{(i)})$ is also Gaussian and can be approximated using an iterative procedure using sampling techniques [23]. As the updating method is not the main concern of this paper, we will not elaborate more on this topic here. We refer the reader to a previous work [7] for a more detailed presentation of the Bayesian updating procedure.

2.3 Stopping condition

The principle of the incremental elicitation procedure is to alternate queries and update operations on the density $p(w)$ until the uncertainty about the weighting vector w is sufficiently reduced to be able to make a recommendation with a satisfactory confidence level. A stopping condition that satisfies this specification consists in waiting for the $\mathrm{MMER}(\mathcal{X}, S)$ value to drop below a predefined threshold, which can be defined as a percentage of the initial MMER value.

2.4 Main obstacles for extending the approach

The main obstacles encountered while managing to extend the approach to a combinatorial setting are related to the computation of MER and MMER values as they are defined in Equations 2 and 3:

- both values require an exponential number of pairwise comparisons to be computed (because there is an exponential number of feasible solutions);
- the use of linear programming to compute these values is not straightforward because the constraint $\max\{0, .\}$ in Equation 1 is not linear.

These issues are all the more critical given that the MER and MMER values are computed at every step of the incremental elicitation procedure to determine whether it should be stopped or not, and to select the next query.

3 Computation of regrets

While the use of mathematical programming is standard in minmax regret optimization, the framework of minmax *expected* regret optimization is more novel. We propose here a new method to compute $\mathrm{MER}(x, \mathcal{X}, S)$ and $\mathrm{MMER}(\mathcal{X}, S)$ by mixed integer linear programming, where $\mathcal{X}$ is implicitly defined by a set of linear constraints and S is a sample drawn from the current density $p(w)$. We consider in this section that $f_w(x)$ is linear in $u(x)$, but the presented approach is adaptable to non-linear aggregation functions if there exist appropriate linear formulations (e.g., the linear formulation of the ordered weighted average [18]). We also assume that $f_w(x) \in [0, 1]$.

3.1 Linear programming for MER computation

To obtain a linear expression for $\mathrm{MER}(x, \mathcal{X}, S)$, we replace the function $\max\{0, f_w(y) - f_w(x)\}$ in Equation 1 by $b_w[f_w(y) - f_w(x)]$ for each weighting vector $w \in S$, where b_w is a binary variable such that $b_w = 1$ if $f_w(y) - f_w(x) > 0$ and $b_w = 0$ if $f_w(y) - f_w(x) < 0$ (the value of b_w does not matter if $f_w(y) - f_w(x) = 0$, because $b_w[f_w(y) - f_w(x)] = 0$ anyway). For this purpose, we need the following additional constraints:

$$\begin{cases} b_w \leq f_w(y) - f_w(x) + 1 & \forall w \in S \quad (c_{\leq}) \\ b_w \geq f_w(y) - f_w(x) & \forall w \in S \quad (c_{\geq}) \end{cases}$$

Proposition 1. *Given $w \in S$, $x \in \mathcal{X}$ and $y \in \mathcal{X}$, if f_w is an aggregation function defined such that $f_w(z) \in [0,1]$ for any $z \in \mathcal{X}$ and $w \in S$, and if b_w satisfies the constraints $(c_{\geq})$ and $(c_{\leq})$, then:*

$$\max\{0, f_w(y) - f_w(x)\} = b_w[f_w(y) - f_w(x)].$$

Proof. Let us denote by d_w the value $f_w(y) - f_w(x)$ for any $w \in S$. First note that $d_w \in [-1,1], \forall w \in S$, because f_w is such that $f_w(z) \in [0,1], \forall z \in \mathcal{X}$. For any $w \in S$, three cases are possible: *Case 1.* w is such that $d_w > 0$: $(c_{\geq})$ becomes $b_w \geq d_w > 0$, thus $b_w = 1$ and we indeed have $b_w d_w = d_w \geq 0$. *Case 2.* w is such that $d_w < 0$: $(c_{\leq})$ becomes $b_w \leq d_w + 1 < 1$ and implies $b_w = 0$ and thus $b_w d_w = 0$. *Case 3.* w is such that $d_w = 0$ then $b_w d_w = 0, \forall b_w \in \{0,1\}$. In the three cases we have thus $b_w d_w = \max\{0, d_w\}$. □

The constraints $(c_{\leq})$ and $(c_{\geq})$ are linear as $f_w(x)$ is linear in $u(x) = (u_1(x), \ldots, u_n(x))$. Nevertheless, using variables b_w and their constraints in the formulation of $\mathrm{MER}(x, \mathcal{X}, S)$ gives a system of linear constraints with a quadratic objective function:

$$\begin{array}{ll} \max \frac{1}{|S|} \sum_{w \in S} b_w[f_w(y) - f_w(x)] & \\ b_w \leq f_w(y) - f_w(x) + 1 & \forall w \in S \\ b_w \geq f_w(y) - f_w(x) & \forall w \in S \\ b_w \in \{0,1\} & \forall w \in S \\ y \in \mathcal{X} & \end{array}$$

The objective function is quadratic because the term $b_w f_w(y)$ is quadratic in variables b_w and y. To linearize the program, we introduce a positive real variable p_w for each $w \in S$, that replace the product term $b_w f_w(y)$. Note that the term $b_w f_w(x)$ does not need linearization because solution x is fixed in the MER computation. The obtained linear program is:

$$(P_{\mathrm{MER}}): \quad \begin{array}{ll} \max \frac{1}{|S|} \sum_{w \in S} [p_w - b_w f_w(x)] & \\ b_w \leq f_w(y) - f_w(x) + 1 & \forall w \in S \\ b_w \geq f_w(y) - f_w(x) & \forall w \in S \\ p_w \leq b_w & \forall w \in S \\ p_w \leq f_w(y) & \forall w \in S \\ p_w \geq b_w + f_w(y) - 1 & \forall w \in S \\ b_w \in \{0,1\} & \forall w \in S \\ p_w \in \mathbb{R}^+ & \forall w \in S \\ y \in \mathcal{X} & \end{array}$$

It is easy to see that $p_w = b_w f_w(y)$ for all $w \in S$ thanks to the constraints on p_w. We indeed have $p_w = 0$ when $b_w = 0$ thanks to the constraint $p_w \leq b_w$, and $p_w = f_w(y)$ when $b_w = 1$ thanks to constraints $p_w \leq f_w(y)$ and $p_w \geq b_w + f_w(y) - 1 = f_w(y)$.

Overall, $2|S|$ variables are involved in the linearization of the expression $\frac{1}{|S|} \sum_{w \in S} \max\{0, [f_w(y) - f_w(x)\}$: $|S|$ binary variables b_w are used to linearize the $\max\{0, .\}$ function, and $|S|$ real variables p_w are used to linearize the product term $b_w f_w(y)$.

3.2 Linear programming for MMER computation

For computing $\mathrm{MMER}(\mathcal{X}, S)$, the objective function $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{X}} \frac{1}{|S|} \sum_{w \in S} \max\{0, f_w(y) - f_w(x)\}$ can be linearized by using $|\mathcal{X}|$ constraints (standard linearization of a $\min\max$ objective function, where the max is taken over a finite set):

$$\begin{array}{ll} \min t & \\ t \geq \frac{1}{|S|} \sum_{w \in S} \max\{0, f_w(y) - f_w(x)\} & \forall y \in \mathcal{X} \quad (*) \\ t \in \mathbb{R} & \end{array}$$

Note that computing the minmax expected regret over $\mathcal{X}$ requires the introduction of one binary variable b_w^y for *each* solution $y \in \mathcal{X}$, so that $\max\{0, f_w(y) - f_w(x)\} = b_w^y(f_w(y) - f_w(x))$ for all $y \in \mathcal{X}$ (while computing the max expected regret of a *given* solution x only required the introduction of a *single* binary variable b_w such that $\max\{0, f_w(\hat{y}) - f_w(x)\} = b_w(f_w(\hat{y}) - f_w(x))$ for $\hat{y} \in \arg\max_{y \in \mathcal{X}} \mathrm{PER}(x, y, S)$).

Consider the following program that involves quadratic constraints:

$$(P_{\mathrm{MMER}}) \quad \begin{array}{ll} \min t & \\ t \geq \frac{1}{|S|} \sum_{w \in S} b_w^y [f_w(y) - f_w(x)] & \forall y \in \mathcal{X} \\ b_w^y \leq f_w(y) - f_w(x) + 1 & \forall w, y \in S \times \mathcal{X} \\ b_w^y \geq f_w(y) - f_w(x) & \forall w, y \in S \times \mathcal{X} \\ b_w^y \in \{0,1\} & \forall w, y \in S \times \mathcal{X} \\ x \in \mathcal{X} & \\ t \in \mathbb{R} & \end{array}$$

Proposition 2. *A solution $x^* \in \mathcal{X}$ optimizing P_{MMER} is such that $\mathrm{MER}(x^*, \mathcal{X}, S) = \mathrm{MMER}(\mathcal{X}, S)$.*

Proof. We denote by t^* the optimal value of P_{MMER}. We prove that t^* is equal to $\mathrm{MMER}(\mathcal{X}, S)$. For a given instance of x, constraint $(*)$ must be satisfied for *any* possible instance of y. Thus, by Proposition 1, we have that $t \geq \mathrm{PER}(x, y, S)$ for all $y \in \mathcal{X}$ because $\frac{1}{|S|} \sum_{w \in S} b_w^y [f_w(y) - f_w(x)] = \mathrm{PER}(x, y, S)$. It implies that $t \geq \max_y \mathrm{PER}(x, y, S) = \mathrm{MER}(x, \mathcal{X}, S)$. As the objective function is $\min t$, for each instance of x, the variable t takes value $\mathrm{MER}(x, \mathcal{X}, S)$. The $\min$ objective function implies that (1) $t = \mathrm{MER}(x, \mathcal{X}, S)$ for a given x. Finally, varying x over $\mathcal{X}$, we can easily see that $t^* \leq \mathrm{MER}(x, \mathcal{X}, S)\ \forall x \in \mathcal{X}$, and thus (2) $t^* = \mathrm{MMER}(\mathcal{X}, S)$. The result follows from (1) and (2). □

The quadratic terms $b_w^y f_w(x)$ are linearized by introducing $|S| \times |\mathcal{X}|$ positive real variables p_w^y:

$$(P_{\mathcal{X}}): \quad \begin{array}{ll} \min t & \\ t \geq \frac{1}{|S|} \sum_{w \in S} b_w^y [f_w(y) - p_w^y] & \forall y \in \mathcal{X} \\ b_w^y \leq f_w(y) - f_w(x) + 1 & \forall w, y \in S \times \mathcal{X} \\ b_w^y \geq f_w(y) - f_w(x) & \forall w, y \in S \times \mathcal{X} \\ p_w^y \leq b_w^y & \forall w, y \in S \times \mathcal{X} \\ p_w^y \leq f_w(x) & \forall w, y \in S \times \mathcal{X} \\ p_w^y \geq b_w^y + f_w(x) - 1 & \forall w, y \in S \times \mathcal{X} \\ b_w^y \in \{0,1\} & \forall w, y \in S \times \mathcal{X} \\ p_w^y \in \mathbb{R}^+ & \forall w, y \in S \times \mathcal{X} \\ x \in \mathcal{X} & \\ t \in \mathbb{R} & \end{array}$$

One comes up with a mixed integer linear program $P_{\mathcal{X}}$ involving $|S| \times |\mathcal{X}|$ binary variables b_w^y, $|S| \times |\mathcal{X}|$ positive real variables p_w^y and $|\mathcal{X}| + 6 \times |S| \times |\mathcal{X}|$ constraints, hence an exponential number of variables and constraints due to the combinatorial nature of $\mathcal{X}$. In the remainder of the section, we propose a method to overcome this issue.

3.3 MMER computation method

The proposed method is based on mixed integer linear programming with dynamic generation of variables and constraints to compute $\mathrm{MMER}(\mathcal{X}, S)$, an optimal solution $x_S^* \in \arg\min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, S)$ and its best challenger $\hat{y}_S \in \arg\max_{y \in \mathcal{X}} \mathrm{PER}(x_S^*, y, S)$.

Let us first define a mixed integer linear program P_A that contains only a subset of variables b_w^y and p_w^y, and a subset of constraints

of type $(*)$. Given a subset $A \subseteq \mathcal{X}$ of solutions, P_A computes the minimax expected regret $\mathrm{MMER}_A(\mathcal{X}, S)$ defined by:

$$\min_{x \in \mathcal{X}} \mathrm{MER}(x, A, S) = \min_{x \in \mathcal{X}} \max_{y \in A} \mathrm{PER}(x, y, S).$$

Put another way, $\mathrm{MER}(x, A, S)$ is the max expected regret of a solution $x \in \mathcal{X}$ w.r.t. solutions in A. More formally, P_A is written:

$$
\begin{array}{ll}
 & \min t \\
 & t \geq \frac{1}{|S|} \sum_{w \in S} b_w^y [f_w(y) - p_w^y] \qquad \forall y \in A \\
 & b_w^y \leq f_w(y) - f_w(x) + 1 \qquad \forall w, y \in S \times A \\
 & b_w^y \geq f_w(y) - f_w(x) \qquad \forall w, y \in S \times A \\
 & p_w^y \leq b_w^y \qquad \forall w, y \in S \times A \\
(P_A): & p_w^y \leq f_w(x) \qquad \forall w, y \in S \times A \\
 & p_w^y \geq b_w^y + f_w(x) - 1 \qquad \forall w, y \in S \times A \\
 & b_w^y \in \{0, 1\} \qquad \forall w, y \in S \times A \\
 & p_w^y \in \mathbb{R}^+ \qquad \forall w, y \in S \times A \\
 & x \in \mathcal{X} \\
 & t \in \mathbb{R}
\end{array}
$$

Note that P_A only involves $|S| \times |A|$ variables b_w^y, $|S| \times |A|$ variables p_w^y and $|A| + 6 \times |S| \times |A|$ constraints.

The algorithm we propose consists in alternatively solving P_A and P_{MER}. Let x_A (resp. $\hat{y}$) denote the optimal solution returned by solving P_A (resp. P_{MER} for $x = x_A$). The algorithm starts with a small set A of feasible solutions (see Section 3.5 for initialization details), and then iteratively grows it by adding the best challenger $\hat{y}$ of x_A to A. Convergence is achieved when P_{MER} returns a solution $\hat{y}$ that already belongs to A, which implies that $\mathrm{MMER}_A(\mathcal{X}, S) = \mathrm{MMER}(\mathcal{X}, S)$. Algorithm 1 describes the procedure.

By abuse of notation, $\mathrm{MMER}_A(\mathcal{X}, S)$ is viewed in the algorithm as a procedure returning the optimal value *mmer*$_A$ of P_A and the corresponding optimal solution x_A. Similarly, $\mathrm{MER}(x_A, \mathcal{X}, S)$ is viewed as a procedure returning the optimal value *mer*$\_x_A$ of P_{MER} and the corresponding optimal solution $\hat{y}$. At the termination of the algorithm, *mmer*$_A$ corresponds to $\mathrm{MMER}(\mathcal{X}, S)$, x_A is the MMER solution and $\hat{y}$ its best challenger.

Proposition 3. *Algorithm 1 terminates and returns a minmax expected regret solution and its best challenger.*

Proof. First, it is easy to see that Algorithm 1 always terminates. Indeed, at every step of the algorithm, if the stopping condition is not satisfied then a new solution $\hat{y} \notin A$ is added to A and a new iteration is performed. In the worst case, all the solutions of $\mathcal{X}$ are added to the set A and the stopping condition is trivially satisfied.

Algorithm 1: $\mathrm{MMER}(\mathcal{X}, A, S)$

Input: $\mathcal{X}$: combinatorial set of feasible solutions;
$A \subseteq \mathcal{X}$: subset of challengers;
S: sample of weighting vectors.
Output: MMER value, MMER solution and its best challenger for the considered sample

$\hat{y} \leftarrow null$
repeat
  if $\hat{y} \neq null$ **then** $A \leftarrow A \cup \{\hat{y}\}$;
  $(mmer_A, x_A) \leftarrow \mathrm{MMER}_A(\mathcal{X}, S)$ (using P_A)
  $(mer\_x_A, \hat{y}) \leftarrow \mathrm{MER}(x_A, \mathcal{X}, S)$ (using P_{MER})
until $\hat{y} \in A$;
return *mmer*$_A$, x_A, $\hat{y}$

We now prove the validity of Algorithm 1, i.e. $\mathrm{MMER}_A(\mathcal{X}, S) = \mathrm{MMER}(\mathcal{X}, S)$ if $\hat{y} \in A$. Assume that $A \subsetneq \mathcal{X}$ (if $A = \mathcal{X}$ the equality is trivially true). On the one hand, at any step of the algorithm, we have (1) *mmer*$_A \leq$ *mer*$\_x_A$ because $\mathrm{MER}(x_A, A, S) \leq \mathrm{MER}(x_A, \mathcal{X}, S)$. On the other hand, if $\hat{y} \in A$ then the constraint $t \geq \frac{1}{|S|} \sum_{w \in S} [f_w(\hat{y}) - f_w(x_A)]$ is satisfied for $t =$ *mmer*$_A$, i.e., *mmer*$_A \geq \mathrm{PER}(x_A, \hat{y}, S)$. As $\mathrm{PER}(x_A, \hat{y}, S) = \mathrm{MER}(x_A, \mathcal{X}, S)$ by definition of $\hat{y}$, it implies that (2) *mmer*$_A \geq \mathrm{MER}(x_A, \mathcal{X}, S) =$ *mer*$\_x_A$. By (1) and (2), we conclude that *mmer*$_A =$ *mer*$\_x_A$.

Finally, *mmer*$_A$ minimizes $\frac{1}{|S|} \sum_{w \in S} b_w [f_w(\hat{y}) - f_w(x)]$ for $x \in \mathcal{X}$, thus *mmer*$_A$ minimizes $\mathrm{PER}(x, \hat{y}, S)$ over $\mathcal{X}$. Consequently, *mer*$\_x_A \leq \mathrm{PER}(x, \hat{y}, S)$ for all $x \in \mathcal{X}$ and then *mer*$\_x_A \leq \mathrm{MER}(x, \mathcal{X}, S), \forall x \in \mathcal{X}$. Thus, by definition of the MMER, we have *mer*$\_x_A = \mathrm{MMER}(\mathcal{X}, S)$ and thereby *mmer*$_A =$ *mer*$\_x_A = \mathrm{MMER}(\mathcal{X}, S)$. □

3.4 Clustering the samples

To decrease the computation times between two queries, we propose to reduce the number of variables and constraints in P_A by applying a clustering method on each sample S drawn from $p(w)$. Let C denote the set of cluster centers. The idea is to replace the $|S|$ weights by the $|C|$ centers, the formula for the pairwise expected regret becoming:

$$\mathrm{PER}(x, y, C) = \sum_{c \in C} \rho_c \max\{0, f_c(y) - f_c(x)\} \qquad (6)$$

where ρ_c is the weight of the cluster center $c \in C$ and represents the proportion of weighting vectors of S that are in the cluster of center c. The formulas for $\mathrm{MER}(x, \mathcal{X}, C)$ and $\mathrm{MMER}(\mathcal{X}, C)$ are adapted in the same way.

3.5 Incremental decision making approach

As detailed in section 2, the MMER computation is used to determine which query to ask at each step as well as to trigger the stopping condition. The whole incremental decision making procedure is summarized in Algorithm 2. The set A is heuristically defined as the set of f_w-optimal solutions for w in C (Line 5) but can be defined otherwise without any impact on the result in proposition 3. The variable *mmer* (Line 6) represents the current minmax expected regret value and is computed using Algorithm 1 by replacing the sample S by the set of cluster centers C.

4 Experimental results

Algorithm 2 has been implemented in Python using the *SciPy*, *Scikit-Learn* and *gurobipy* libraries for, respectively, Gaussian sampling, clustering[2] and solving the mixed integer linear programs. The numerical tests have been carried out on 50 randomly generated instances of the multi-objective *knapsack* problem. For all tests we used an Intel(R) Core(TM) i7-4790 CPU with 15GB of RAM.

Multi-objective Knapsack Problem (MKP) This vector optimization problem is formulated as $\max z = Ux$ subject to $\sum_{i=1}^{p} \alpha_i x_i \leq \gamma$, where U is an $n \times p$ matrix of general term u_{ki} representing the utility of item $i \in \{1, \ldots, p\}$ w.r.t objective $k \in \{1, \ldots, n\}$, $x = (x_1, \ldots, x_p)^T$ is a vector of binary decision variables such that $x_i = 1$ if item i is selected and $x_i = 0$ otherwise, α_i is the weight of

[2] We used *k-means* clustering.

Algorithm 2: Incremental Decision Making

Input: $\mathcal{X}$: combinatorial set of feasible solutions;
$p_0(w)$: prior density function.
Output: x^* : recommended solution.

1 $p(w) \leftarrow p_0(w); i \leftarrow 1$
2 **repeat**
3 $S \leftarrow$ sample drawn from $p(w)$
4 $C \leftarrow$ cluster centers of S
5 $A \leftarrow \{\arg\max_{x\in\mathcal{X}} f_w(x) | w \in C\}$
6 $(mmer, x^{(i)}, y^{(i)}) \leftarrow \text{MMER}(\mathcal{X}, A, C)$
7 Ask the DM if $x^{(i)}$ is preferred to $y^{(i)}$
8 $a^{(i)} \leftarrow 1$ if the answer is yes and 0 otherwise
9 $p(w) \leftarrow p(w|a^{(i)})$ (see Algorithm 2 in [7])
10 $i \leftarrow i + 1$
11 **until** *mmer* stabilizes
12 **return** x^* selected in $\arg\min_{x\in\mathcal{X}} \text{MER}(x, \mathcal{X}, C)$

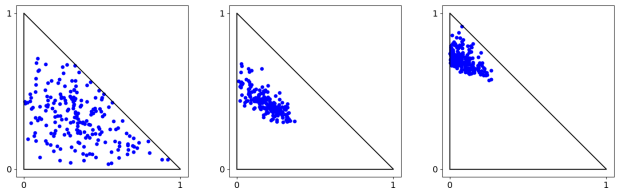
item i and γ is the knapsack's capacity. The set of feasible knapsacks is $\mathcal{X} = \{x \in \{0,1\}^p | \sum_{i=1}^p \alpha_i x_i \leq \gamma\}$, and the performance vector $z \in \mathbb{R}^n$ associated to a solution x is $z = Ux$.

To simulate elicitation sessions, we consider the problem $\max_{x\in\mathcal{X}} f_w(x)$ where $f_w(x) = \sum_k w_k \sum_i u_{ki} x_i$. The weighting vector w in $W = \{w \in [0,1]^n : \sum_k w_k = 1\}$ is initially unknown. We generated instances of MKP for $n = 5$ objectives and $p = 100$ items. Every item i has a positive weight α_i uniformly drawn in $\{1, \ldots, 20\}$, and $\gamma = \frac{1}{2}\sum_{k=1}^{100} \alpha_k$. Utilities u_{ki} are uniformly drawn in $[0, \frac{1}{p}]$ to make sure that $f_w(x) \in [0,1], \forall x \in \mathcal{X}$.

Simulation of the DM's answers In order to simulate the interactions with the DM, for each instance, the hidden weighting vectors w are uniformly drawn in the canonical basis of $\mathbb{R}^n$ (the more vector w is unbalanced, the worse the initial recommendation). At each query, the answer is obtained using the response model given in Section 2.2, i.e., for query i, the answer depends on the sign of $z^{(i)} = w^T d^{(i)} + \varepsilon^{(i)}$, where $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$. We used different values of σ to evaluate the tolerance of the approach to wrong answers. We set $\sigma = 0$ to simulate a DM that is perfectly reliable in her answers. The strictly positive values are used to simulate a DM that may be inconsistent in her answers. Setting $\sigma = 0.01$ led to 16% of wrong answers, while $\sigma = 0.02$ led to 24% of wrong answers.

Parameter settings in algorithms The prior density in Algorithm 1 is set to $\mathcal{N}((10, \ldots 10)^T, 100 I_5)$, where I_5 is the identity matrix 5×5, so that the distribution is rather flat. At each step of Algorithm 2, a new sample S of 100 weighting vectors is generated; the vectors $w \in S$ are normalized and partitioned into 20 clusters. This number of clusters has been chosen empirically after preliminary numerical tests: considering the entire sample or using more than 20 clusters led to higher computation times and did not offer a significant improvement on the quality of the recommendations. Last but not least, we stopped the algorithm after 15 queries if the termination condition was not fulfilled before.

Illustrative example Before coming to the presentation of the numerical results, let us first illustrate the progress of the elicitation procedure on the following example: we applied Algorithm 2 on a randomly generated instance of MKP with 3 objectives, 100 items, a hidden weighting vector $w = (0, 1, 0)$, and we set $\sigma = 0.02$, which led to an error rate of 20%. Figure 1 illustrates the convergence of



Query 0 Query 3 Query 10

Figure 1: Evolution of the samples toward the hidden weight.

the generated samples of weighting vectors (Line 3 of Algorithm 2) toward the hidden weight during the execution of the algorithm. As the weighting vectors are normalized, two components are enough to characterize them. Every graph shows the sample drawn at a given step of the algorithm.

Analysis of the results We first evaluated the efficiency of Algorithm 2 according to the value of σ. We observed the evolution of the quality of the recommendation (the minimax expected regret solution) after every query. The quality of a recommendation x^* is defined by the score $s_{w_h}(x^*) = f_{w_h}(x^*)/f_{w_h}(x_h)$, where w_h is the hidden weighting vector and x_h is an optimal solution for w_h. The obtained curves are given in Figure 2. We observe that the quality of the recommendation (measured by the score function s_{w_h}) is of course negatively impacted when σ increases. However, the score of the recommendation at the termination of Algorithm 2 is ≥ 0.98 for $\sigma \in \{0, 0.01\}$, and ≥ 0.96 for $\sigma = 0.02$. Regarding the computation times, the mean time between two queries over the 50 instances was around 4 seconds.

We next compared the performances of Algorithm 2 to the performance of a deterministic approach that does not take into account the possible errors in responses [6] (approach based on the systematic reduction of the parameter space by minimizing the minimax regret at each step). The aim was to evaluate how much the DM's inconsistencies in her answers impact the two procedures. In this purpose, we set $\sigma = 0.02$. The obtained results are given in the box plots of Figure 3 for Algorithm 2, and of Figure 4 for the deterministic algorithm. In these figures, the box plots give, for any given question, the score of the recommendation for every considered instance (the bottom and top bands of the whiskers are the minimum and maximum scores over the 50 instances, the bottom and top bands of the boxes are the first and third quartiles, the band in the box is the median, the dotted band is the average, and the circles are isolated values). The histogram gives the number of observed values for every query; the bin i indeed gives the number of instances for which query i is reached before the stopping condition is fulfilled.

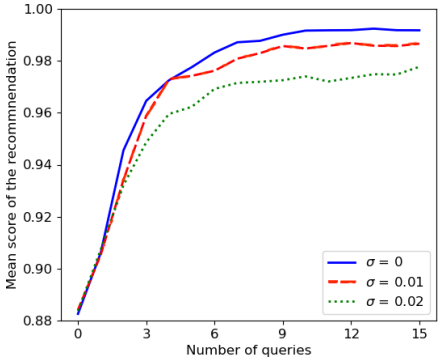


Figure 2: Mean score vs. queries

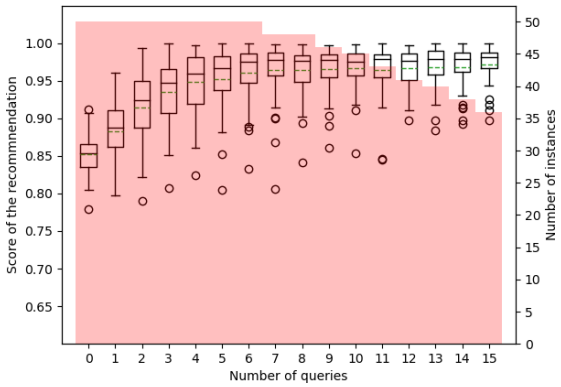


Figure 3: Algorithm 2 - Score vs. queries

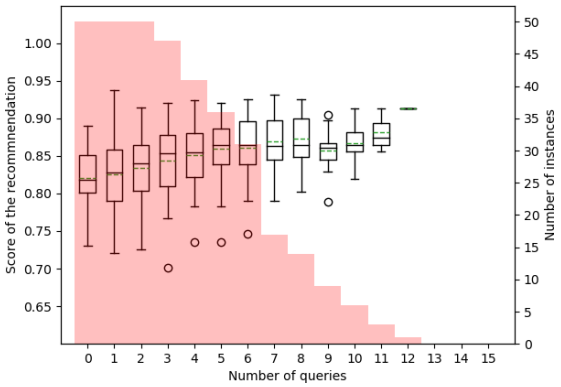


Figure 4: Deterministic algorithm [6] - Score vs. queries

Figures 3 and 4 show the interest of considering our Bayesian elicitation procedure in comparison with a deterministic approach. Indeed, the deterministic approach converges quickly and requires less queries than Algorithm 2; however, the score of the current recommendation at every step of the algorithm does not exceed 0.94 for any considered instance and is ≤ 0.9 for 75% of the instances. In contrast, for Algorithm 2, the score of the current recommendation is ≥ 0.95 in 75% of the instances from query 6.

5 Conclusion

We introduced in this paper a Bayesian incremental preference elicitation approach for solving multiobjective combinatorial optimization problems when the preferences of the decision maker are represented by an aggregation function whose parameters are initially unknown. The proposed approach deals with the possibility of inconsistencies in the decision maker's answers to pairwise preference queries. Our approach uses a columns and constraints generation solution method for the computation of expected regrets. The approach is general and can be applied to any problem having an efficient mixed integer linear programming formulation. An interesting research direction would be to refine the approach in the case of non-linear aggregation functions. The approach is indeed compatible with such functions provided they can be linearized (e.g., the linearization of ordered weighted averages [18]), but the subsequent linear formulations often involve many additional variables and constraints, thus the need for an optimization.

REFERENCES

[1] J. H. Albert and S. Chib, 'Bayesian analysis of binary and polychotomous response data', *J. Am. Stat. Assoc.*, **88**(422), 669–679, (1993).

[2] Nawal Benabbou and Patrice Perny, 'Adaptive elicitation of preferences under uncertainty in sequential decision making problems', in *IJCAI-17*, pp. 4566–4572, (2017).

[3] Nawal Benabbou and Patrice Perny, 'Interactive resolution of multiobjective combinatorial optimization problems by incremental elicitation of criteria weights', *EURO J. on Decision Proc.*, **6**(3-4), 283–319, (2018).

[4] Nawal Benabbou, Patrice Perny, and Paolo Viappiani, 'Incremental elicitation of Choquet capacities for multicriteria choice, ranking and sorting problems', *Artificial Intelligence*, **246**, 152–180, (2017).

[5] Christopher M Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[6] Nadjet Bourdache and Patrice Perny, 'Active preference elicitation based on generalized Gini functions: Application to the multiagent knapsack problem', in *AAAI 2019*, pp. 7741–7748, (2019).

[7] Nadjet Bourdache, Patrice Perny, and Olivier Spanjaard, 'Incremental elicitation of rank-dependent aggregation functions based on Bayesian linear regression', in *IJCAI-19*, pp. 2023–2029, (2019).

[8] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans, 'Constraint-based optimization and utility elicitation using the minimax decision criterion', *Artif. Intelligence*, **170**(8-9), 686–713, (2006).

[9] Juergen Branke, Salvatore Corrente, Salvatore Greco, Roman Słowiński, and Piotr Zielniewicz, 'Using Choquet integral as preference model in interactive evolutionary multiobjective optimization', *EJOR*, **250**(3), 884–901, (2016).

[10] Urszula Chajewska, Daphne Koller, and Ronald Parr, 'Making rational decisions using adaptive utility elicitation', in *AAAI-00*, pp. 363–369, (2000).

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

[12] Matthias Ehrgott, *Multicriteria optimization*, Springer Science & Business Media, 2005.

[13] Mirco Gelain, Maria Silvia Pini, Francesca Rossi, K Brent Venable, and Toby Walsh, 'Elicitation strategies for soft constraint problems with missing preferences: properties, algorithms and experimental studies', *Artif. Intelligence*, **174**(3), 270–294, (2010).

[14] Hugo Gilbert, Olivier Spanjaard, Paolo Viappiani, and Paul Weng, 'Reducing the number of queries in interactive value iteration', in *ADT-15*, pp. 139–152, (2015).

[15] Shengbo Guo and Scott Sanner, 'Multiattribute Bayesian preference elicitation with pairwise comparison queries', in *NIPS-10*, pp. 396–403, (2010).

[16] Greg Hines and Kate Larson, 'Preference elicitation for risky prospects', in *AAMAS-10*, pp. 889–896, (2010).

[17] Tyler Lu and Craig Boutilier, 'Robust approximation and incremental elicitation in voting protocols', in *IJCAI-11*, pp. 287–293, (2011).

[18] Włodzimierz Ogryczak and Tomasz Śliwiński, 'On solving linear programs with the ordered weighted averaging objective', *EJOR*, **148**(1), 80–91, (2003).

[19] Patrice Perny, Paolo Viappiani, and Abdellah Boukhatem, 'Incremental preference elicitation for decision making under risk with the rank-dependent utility model', in *UAI-16*, pp. 597–606, (2016).

[20] Kevin Regan and Craig Boutilier, 'Eliciting additive reward functions for Markov decision processes', in *IJCAI-11*, pp. 2159–2164, (2011).

[21] Denis Sauré and Juan Pablo Vielma, 'Ellipsoidal methods for adaptive choice-based conjoint analysis', *Oper. Res.*, **67**(2), 315–338, (2019).

[22] Ralph E Steuer, *Multiple criteria optimization: theory, computation, and application*, volume 233, Wiley New York, 1986.

[23] Martin A Tanner and Wing Hung Wong, 'The calculation of posterior distributions by data augmentation', *J. Am. Stat. Asso.*, **82**(398), 528–540, (1987).

[24] Olivier Toubia, John R. Hauser, and Duncan I. Simester, 'Polyhedral methods for adaptive choice-based conjoint analysis', *Journal of Marketing Research*, **41**(1), 116–131, (2004).

[25] I. Vendrov, T. Lu, Q. Huang, and C. Boutilier, 'Gradient-based optimization for Bayesian preference elicitation', in *AAAI-20*, (2020).

[26] T. Wang and C. Boutilier, 'Incremental utility elicitation with the minimax regret decision criterion', in *IJCAI-03*, pp. 309–316, (2003).

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

[28] C. C. White III, A. P. Sage, and S. Dozono, 'A model of multiattribute decision making and trade-off weight determination under uncertainty', *IEEE Trans. on Systems, Man, and Cyb.*, **14**(2), 223–229, (1984).