HAL
open science

Active Preference Elicitation by Bayesian Updating on Optimality Polyhedra

Nadjet Bourdache, Patrice Perny, Olivier Spanjaard

To cite this version:

Nadjet Bourdache, Patrice Perny, Olivier Spanjaard. Active Preference Elicitation by Bayesian Updating on Optimality Polyhedra. SUM 2019 - 13th international conference on Scalable Uncertainty Management, Dec 2019, Compiègne, France. pp.93-106, 10.1007/978-3-030-35514-2_8 . hal-02419458

HAL Id: hal-02419458

https://hal.science/hal-02419458v1

Submitted on 7 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Active Preference Elicitation by Bayesian Updating on Optimality Polyhedra

Nadjet Bourdache, Patrice Perny, and Olivier Spanjaard

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
name.surname@lip6.fr

Abstract. We consider the problem of actively eliciting the preferences of a Decision Maker (DM) that may exhibit some versatility when answering preference queries. Given a set of multicriteria alternatives (choice set) and an aggregation function whose parameter values are unknown, we propose a new incremental elicitation method where the parameter space is partitioned into optimality polyhedra in the same way as in *stochastic multicriteria acceptability analysis*. Each polyhedron encompasses the subset of parameter values for which a given alternative is optimal (one optimality polyhedron, possibly empty, per alternative in the choice set). The uncertainty about the DM's judgment is modeled by a probability distribution over the polyhedra of the partition. At each step of the elicitation procedure, the distribution is revised in a Bayesian manner using preference queries whose choice is based on the *current solution strategy*, that we adapt to minimize the expected regret of the recommended alternative. We interleave the analysis of the set of alternatives with the elicitation of the parameters of the aggregation function (weighted sum or ordered weighted average). Numerical tests have been performed to evaluate the interest of the proposed approach.

Keywords: Incremental preference elicitation · Optimality polyhedra · Bayesian updating · Expected regrets

1 Introduction

Preference elicitation is an essential part of computer-aided multicriteria decision support. Indeed, criteria being often conflicting, the notion of optimality is subjective and fully depends on the Decision Maker's (DM) view on the relative importance attached to every criteria. Thus, the relevance of the recommendation depends on our ability to elicit this information and the way we model the uncertainty about the DM's preferences.

A standard way to compare feasible solutions in multicriteria decision problems is to use parameterized aggregation functions assigning a value (overall utility) to every solution. This function can be fitted to the DM preferences by eliciting the weighting coefficients that specify the importance of criteria in the aggregation. In many real cases, it is impractical but also useless to precisely specify the parameters of the aggregation function. Given a decision model,

exact choices can often be derived from a partial specification of weighting parameters. Dealing with partially specified parameters requires the development of solution methods that can determine an optimal or near optimal solution with such partial information. This is the aim of *incremental preference elicitation*, that consists on interleaving the elicitation with the exploration of the set of alternatives to adapt the elicitation process to the considered instance and to the DM's answers. Thus, the elicitation effort is focused on the useful part of the preference information. The purpose of incremental elicitation is not to learn precisely the values of the parameters of the aggregation function but to specify them sufficiently to be able to determine a relevant recommendation.

Incremental preference elicitation is the subject of several contributions in various contexts, see e.g. [3, 4, 7, 16]. Starting from the entire set of possible parameter values, incremental elicitation methods are based on the reduction of the uncertainty about the parameter values by iteratively asking the DM to provide new preference information (e.g., with pairwise comparisons between alternatives). Any new information is translated into a hard constraint that allows to reduce the parameter space. In this way, preference data are collected until a necessarily optimal or near optimal solution can be determined, i.e., a solution that is optimal or near optimal for all the possible parameter values. These methods are very efficient because they allow a fast reduction of the parameter space. Nevertheless, they are very sensitive to possible mistakes of the DM in her answers. Indeed, in case of a wrong answer, the definitive reduction of the parameter space will exclude the wrong part of the set of possible parameter values, which is likely to exclude the optimal solution from the set of *possibly optimal* solutions (i.e., solutions that are optimal for at least one possible parameter value). Consequently, the relevance of the recommendation may be significantly impacted if there is no possible backtrack. A way to overcome this drawback is to use probabilistic approaches that allow to model the uncertainty about the DM's answers, and thus to give her the opportunity to contradict herself without impacting too much the quality of the recommendation. In such methods, the parameter space remains unchanged throughout the algorithm and the uncertainty about the real parameter values (which characterize the DM's preferences) is represented by a probability density function that is updated when new preference statements are collected.

This idea has been developed in the literature. In the context of incremental elicitation of utility values, Chajewska et al. [8] proposed to update a probability distribution over the DM's utility function to represent the belief about the utility value. The probability distribution is incrementally adjusted until the expected loss of the recommendation is sufficiently small. This method does not apply in our setting because we consider that the utility values of the alternatives on every criterion are known and that we elicit the values of the weighting coefficients of the aggregation function. Sauré and Vielma [15] introduced a method based on maintaining a confidence ellipsoid region using a multivariate Gaussian distribution over the parameter space. They use mixed integer programming to select a preference query that is the most likely to reduce the volume of the

confidence region. In a recent work [5], the uncertainty about the parameter values is represented by a Gaussian distribution over the parameter space of rank-dependent aggregation functions. Preference queries are selected by minimizing expected regrets to update the density function using Bayesian linear regression. As the updating of a continuous density function is computationally cumbersome (especially when analytical results for the obtention of the posterior density function do not exist), data augmentation and sampling techniques are used to approximate the posterior density function. These methods are time consuming and require to make a tradeoff between computation time and accuracy of the approximation. In addition, the information provided by a continuous density function may be much richer than the information really needed by the algorithm to conclude. Indeed, it is generally sufficient to know that the true parameter values belong to a given restricted area of the parameter space to be able to identify an optimal solution without ambiguity. Thus, we introduce in this paper a new model-based incremental elicitation algorithm based on a discretization of the parameter space. We partition the parameter space into optimality polyhedra and we define a probability distribution over the partition. After each query, this distribution is updated using Bayes' rule.

The paper is organised as follows. Section 2 recalls some background on weighted sums and ordered weighted averages. We also introduce the optimality polyhedra we use in our method and we discuss our contribution with regard to related works relying on the optimality polyhedra. We present our incremental elicitation method in Section 3. Finally, some numerical tests showing the interest of the proposed approach are provided in Section 4.

2 Background and Notations

Let $\mathcal{X}$ be a set of n alternatives evaluated on p criteria. Any alternative of $\mathcal{X}$ is characterized by a performance vector $x = (x_1, \ldots, x_p)$, where $x_i \in [0, U]$ is the performance of the alternative on criterion i, and U is the maximum utility value. All utilities x_i are expressed on the same scale; the utility functions must be defined from the input data (criterion or attribute values), as proposed by, e.g., Grabisch and Labreuche [10]. To refine the Pareto dominance relation and to be able to better discriminate between alternatives in $\mathcal{X}$, we use a parametrized aggregation function denoted by f_w. The weighting vector w of the function defines how the components of x should be aggregated and thus makes it possible to model the decision behavior of the DM. In this paper, we consider two operators: the weighted sum (WS) and the ordered weighted average (OWA). We give some notations and recall some basic notions about this two aggregation functions in the following.

Weighted Sum. Let $x \in \mathbb{R}_+^p$ be a performance vector and $w \in \mathbb{R}_+^p$ be a weighting vector. The weighted sum is defined by:

$$\mathrm{WS}_w(x) = \sum_{i=1}^{p} w_i x_i \tag{1}$$

Ordered Weighted Average. Introduced by Yager [17], the OWA is a rank-dependent aggregation function, where the weights are not associated to the criteria but to the ranks in the ordered performance vector, giving more or less importance to good or bad performances. Let $x \in \mathbb{R}_+^p$ be a performance vector and $w \in \mathbb{R}_+^p$ be a weighting vector. The ordered weighted average is defined by:

$$\mathrm{OWA}_w(x) = \sum_{i=1}^{p} w_i x_{(i)} \tag{2}$$

where $x_{(.)}$ is a permutation of vector x such that $x_{(1)} \leq \cdots \leq x_{(p)}$.

Example 1 *Let $x = (14, 9, 10)$, $y = (10, 12, 10)$ and $z = (9, 16, 6)$ be three performance vectors to compare, and assume that the weighting vector is $w = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$. Applying equation (2), we obtain:* $\mathrm{OWA}_w(x) = 10.75 > \mathrm{OWA}_w(y) = 10.5 > \mathrm{OWA}_w(z) = 10$.

Note that OWA includes the minimum ($w_1 = 1$ and $w_i = 0, \forall i \in [\![2, p]\!]$), the maximum ($w_p = 1$ and $w_i = 0, \forall i \in [\![1, p-1]\!]$), the arithmetic mean ($w_i = \frac{1}{p}, \forall i \in [\![1, p]\!]$) and all other order statistics as special cases.

If w is chosen with decreasing components (i.e., the greatest weight is assigned to the worst performance), the OWA function is *concave* and well-balanced performance vectors are favoured. We indeed have, for all $x \in \mathcal{X}$, $\mathrm{OWA}_w((x_1, \ldots, x_i - \varepsilon, \ldots, x_j + \varepsilon, \ldots, x_p)) \geq \mathrm{OWA}_w(x)$ for all i, j and $\varepsilon > 0$ such that $x_i - x_j \geq \varepsilon$. Depending on the choice of the weighting vector w, a concave OWA function allows to define a wide range of mean type aggregation operators between the minimum and the arithmetic mean. In the remainder of the paper, we only consider *concave* OWA functions. For the sake of brevity, we will say OWA for concave OWA.

Example 2 *Consider vectors x, y and z defined in example 1 and assume that the weighting vector is now $w = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$. We have:* $\mathrm{OWA}_w(x) = \frac{61}{6}$, $\mathrm{OWA}_w(y) = \frac{62}{6}$ *and* $\mathrm{OWA}_w(z) = \frac{52}{6}$. *The alternative y, which corresponds to the most balanced performance vector, is the preferred one.*

Using f_w (defined with (1) or (2)) as an aggregation function, we call f_w-*optimal* an alternative x that maximizes $f_w(x)$. Eliciting the DM's preferences amounts to eliciting the weighting vector w. The rest of the section defines how we deal with the imprecise knowledge of the parameter values in the optimization process involved in the elicitation.

Optimality polyhedra. We denote by W the set of all feasible weighting vectors. Note that, to limit the scale of this set, one can add the additional non restrictive normalisation constraint $\sum_{i=1}^{p} w_i = 1$. Thus, W is defined by $W = \{w \in \mathbb{R}_+^p \mid \sum_{i=1}^{p} w_i = 1 \text{ and } w_i \geq 0, \forall i\}$. In the case of a concave OWA, the additional constraint $w_1 \geq \cdots \geq w_p$ is enforced.

Starting from W and the set $\mathcal{X}$ of alternatives, we partition W into *optimality polyhedra*: the optimality polyhedron associated to an alternative x is the set of

weighting vectors such that x is optimal. Note that the aggregation functions we use are linear in w (even though OWA is not linear in x because of the sorting of x before applying the aggregation operation).

This explains why the sets of the partition are convex polyhedra. Any preference statement of the form "Alternative x is preferred to alternative y" is indeed translated into a constraint $f_w(x) \geq f_w(y)$ which is linear in w.

More formally, the optimality polyhedron W_x associated to an alternative $x \in \mathcal{X}$ is defined by $W_x = \{w \in W | f_w(x) \geq f_w(y), \forall y \in \mathcal{X}\}$. Note that any empty set W_x (there is no $w \in W$ such that x is f_w-optimal) or not full dimensional set (i.e., $\forall w \in W_x, \exists y \in \mathcal{X}$ such that $f_w(x) = f_w(y)$) can be omitted. An example of such partition is given in Figure 1 for the instance of Example 1, where the aggregation function is a weighted sum. Note that w_3 can be omitted thanks to the normalization constraint ($w_3 = 1 - w_1 - w_2$).

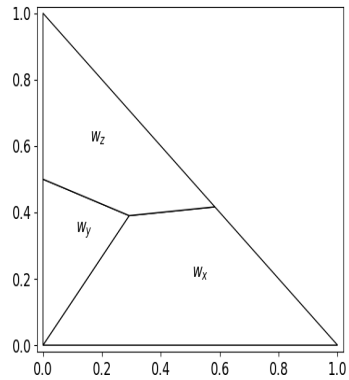

Fig. 1: Optimality polyhedra for x, y and z in Example 1 with WS.

In order to represent the uncertainty about the exact values of parameters, a probability distribution is defined over the polyhedra of the partition. This distribution is updated using an incremental elicitation approach that will be described in the next section.

Related works. The idea of partitioning the parameter space is closely related to Stochastic Multiobjective Acceptability Analysis (SMAA for short). The SMAA methodology has been introduced by Charnetski and Soland under the name of *multiple attribute decision making with partial information* [9]. Given a set of utility vectors and a set of linear constraints characterizing the feasible parameter space for a weighted sum (partial information elicited from the DM), they assume that the probability of optimality for each alternative is proportional to the hypervolume of its optimality polyhedron (the hypervolume reflects how likely an alternative is to be optimal). Lahdelma et al. [12] developed this idea in the case of imprecision or uncertainty in the input data (utilities of the alternatives according to the different criteria) by considering the criteria values as probability distributions. They defined the *acceptability index* for an alternative, that measures the variety of different valuations which allow for that alternative to be optimal, and is proportional to the *expected* volume of its optimality polyhedron. They also introduced a *confidence factor*, that measures if the input data is accurate enough for making an informed decision. The methodology has been adapted to the 2-additive Choquet integral model by Angilella et al. [2]. These works consider that the uncertainty comes from the criterion values or from the variation in the answers provided by *different*

DMs. They also consider that some prior preference information is given and that there is no opportunity to ask the DM for new preference statements. Our work differentiates from these works in the following points:

- the criterion values are accurately known and only the parameter values of the aggregation function must be elicited;
- the uncertainty comes from possible errors in the DM's answers to preference queries;
- the elicitation process is incremental.

3 Incremental Elicitation Approach

Once the parameter space W is partitioned into *optimality polyhedra* as explained above, a prior density function is associated to the partition. This distribution informs us on how likely each solution is to be optimal. In the absence of a prior information about the DM's preferences, we define the prior distribution such that the probability of any polyhedron is proportional to its volume, as suggested by Charnetski and Soland [9]. The volume of W_x gives indeed a measure on the proportion of weighting vectors for which the alternative x is ranked first. More formally, the prior probability of x to be optimal is $P(x) = \frac{vol_{W_x}}{vol_W}$ where $vol_{\mathcal{W}}$ denotes the volume of a convex polyhedron $\mathcal{W}$. We assume here a complete ignorance of the continuous probability distribution for w within each polyhedron. After each new preference statement, the probability distribution P is updated using Bayes' rule.

The choice of the next query to ask is a key point for the efficiency of the elicitation process in acquiring enough preferential information to make a recommendation with sufficient confidence.

Query selection strategy. In order to get the most informative possible query we use a strategy based on the minimization of expected regrets. Let us first introduce how we define expected regrets in our setting:

Definition 1 *Given two alternatives x and y, and a probability distribution P on $\mathcal{X}$, the pairwise expected regret PER is defined by:*

$$\mathrm{PER}(x, y, \mathcal{X}, P) = \sum_{z \in \mathcal{X}} \max\{0, \mathrm{PMR}(x, y, W_z)\} P(z)$$

where $P(z)$ represents the probability for z to be optimal and $\mathrm{PMR}(x, y, \mathcal{W})$ is the pairwise maximum regret over a polyhedron $\mathcal{W}$, defined by:

$$\mathrm{PMR}(x, y, \mathcal{W}) = \max_{w \in \mathcal{W}} \{f_w(y) - f_w(x)\}$$

In other words, the PER defines the expected worst utility loss incurred by recommending an alternative x instead of an alternative y, and $\mathrm{PMR}(x, y, \mathcal{W})$ is the worst utility loss in recommending alternative x instead of alternative y

given that w belongs to $\mathcal{W}$. The use of the PMR within a polyhedron is justified by the complete ignorance about the probability distribution in the polyhedron, thereby, the worst case is considered.

Definition 2 *Given a set $\mathcal{X}$ of alternatives, the maximum expected regret of $x \in \mathcal{X}$ and the minimax expected regret over $\mathcal{X}$ are defined by:*

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

In other words, the MER value defines the worst utility loss incurred by recommending an alternative $x \in \mathcal{X}$ and the MMER value defines the minimal MER value over $\mathcal{X}$.

The notion of regret expresses a measure of the interest of an alternative. At any step of the algorithm, the solution achieving the MMER value is a relevant recommendation because it minimizes the expected loss in the current state of knowledge. It also allows to determine an informative query to ask. Various query selection strategies based on regrets and expected regrets have indeed been introduced in the literature, see e.g. [6] in a deterministic context (*current solution strategy*) and [11] in a probabilistic context (a probability distribution is used to model the uncertainty about the parameter values). Adapting the current solution strategy to our probabilistic setting, we propose here a strategy that consists in asking the DM to compare the current recommendation $x^* = \arg\min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, P)$ to its best challenger defined by $y^* = \arg\max_{y \in \mathcal{X}} \mathrm{PER}(x^*, y, P)$. The current probability distribution is then updated according to the DM's answer, as explained hereafter. The procedure can be iterated until the MMER value drops below a predefined threshold ε.

The approach proposed in this paper consists in interleaving preference queries and Bayesian updating of the probability distribution based on the DM's answers. The elicitation procedure is detailed in Algorithm 1. At each step i of the algorithm, we ask the DM to compare two alternatives $x^{(i)}$ and $y^{(i)}$. The answer is denoted by a_i, where $a_i = 1$ if $x^{(i)}$ is preferred to $y^{(i)}$ and $a_i = 0$ otherwise. From each answer a_i, the conditional probability $P(.|a_1, \ldots, a_{i-1})$ over the set of alternatives is updated in a Bayesian manner (Line 13 of Algorithm 1).

Bayesian updating. We assume that answers a_i are independent binary random variables, i.e. $P(a_i|x^{(i)}, y^{(i)})$ only depends on the (unknown) weighting vector w and on the performance vectors of $x^{(i)}, y^{(i)}$. This is a standard assumption in Bayesian analysis of binary response data [1]. To alleviate the notations, we omit the conditioning statement in $P(a_i|x^{(i)}, y^{(i)})$, that we abbreviate by $P(a_i)$. Using Bayes' rule, the posterior probability of any alternative $z \in \mathcal{X}$ is given by:

$$P(z|a_1, \ldots, a_i) = \frac{P(a_1, \ldots, a_i|z)P(z)}{P(a_1, \ldots, a_i)} = \frac{P(a_i|z)P(a_1, \ldots, a_{i-1}|z)P(z)}{P(a_i)P(a_1, \ldots, a_{i-1})} \quad (3)$$
$$= \frac{P(a_i|z)P(z|a_1, \ldots, a_{i-1})}{P(a_i)} \quad (4)$$

Algorithm 1: Incremental Elicitation Procedure

Input: $\mathcal{X}$: set of alternatives, ε: acceptance threshold; W: parameter space.
Output: x^* : best recommendation in $\mathcal{X}$

$P(z) \leftarrow \frac{vol_{W_z}}{vol_W}, \forall z \in \mathcal{X}$
$i \leftarrow 0$
repeat
  $i \leftarrow i + 1$
  $x^{(i)} \leftarrow \arg\min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, P(.|a_1, \ldots, a_{i-1}))$
  $y^{(i)} \leftarrow \arg\max_{y \in \mathcal{X}} \mathrm{PER}(x^{(i)}, y, P(.|a_1, \ldots, a_{i-1}))$
  Ask the DM if $x^{(i)}$ is preferred to $y^{(i)}$
  if *the answer is yes* **then**
    $a_i \leftarrow 1$
  else
    $a_i \leftarrow 0$
  for $z \in \mathcal{X}$ **do**
    Compute $P(z|a_1, \ldots, a_i)$ using Bayesian updating
until $\mathrm{MMER}(\mathcal{X}, P(.|a_1, \ldots, a_i)) \leq \varepsilon$;
return x^* selected in $\arg\min_{x \in \mathcal{X}} \mathrm{MER}(x, \mathcal{X}, P(.|a_1, \ldots, a_i))$

The likelihood function $P(a_i|z)$ is the conditional probability that the answer is a_i given that z is optimal. Let us denote by $W_{x^{(i)} \succsim y^{(i)}}$ the subset of W containing all vectors w such that $f_w(x^{(i)}) \geq f_w(y^{(i)})$; the likelihood function is defined as:

$$P(a_i = 1|z) = \begin{cases} \delta & \text{if } \; W_z \subseteq W_{x^{(i)} \succsim y^{(i)}} \\ 1 - \delta & \text{if } \; W_z \cap W_{x^{(i)} \succ y^{(i)}} = \emptyset \\ P(a_i = 1) & \text{otherwise} \end{cases}$$

where $\delta \in (\frac{1}{2}, 1]$ is a constant. The corresponding update of the probability masses follows the idea used by Nowak in noisy generalized binary search [14] and its effect is simple; the probability masses of polyhedra that are compatible with the preference statement are boosted relative to those that are not compatible, while the probability masses of the other polyhedra remain unchanged. The parameter δ controls the size of the boost, and can be seen as a lower bound on the probability of a correct answer. The three cases are depicted in Figure 2.

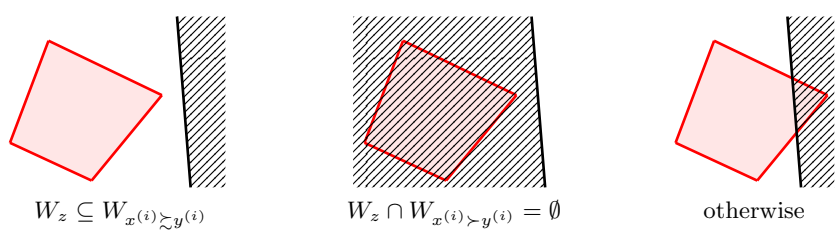

Fig. 2: The polyhedron is W_z. The non-hatched area is the half-space $W_{x^{(i)} \succsim y^{(i)}}$.

In the third case (on the right of Figure 2), due to the assumption of complete ignorance within a polyhedron, the new preference statement is not informative enough to update the probability of z to be optimal. Therefore, for all alternatives z such that W_z is cut by the constraint $f_w(x^{(i)}) \geq f_w(y^{(i)})$ no updating is performed and therefore $P(a_i|z) = P(a_i)$; consequently $P(z|a_1, \ldots, a_i) = P(z|a_1, \ldots, a_{i-1})$ by Equation 4.

Regarding Equation 4, note that, in practice, we do not need to determine $P(a_i)$. For any alternative $z \in \mathcal{X}$ such that W_z is not cut by the constraint, we have indeed $P(z|a_1, \ldots, a_i) \propto P(a_i|z)P(z|a_1, \ldots, a_{i-1})$. More precisely, $P(z|a_1, \ldots, a_i)$ is obtained by the following equation:

$$P(z|a_1, \ldots, a_i) = \sum_{y \in \mathcal{Y}} P(y|a_1, \ldots, a_{i-1}) \frac{P(a_i|z)P(z|a_1, \ldots, a_{i-1})}{\sum\limits_{y \in \mathcal{Y}} P(a_i|y)P(y|a_1, \ldots, a_{i-1})} \tag{5}$$

where $\mathcal{Y}$ is the subset of alternatives whose optimality polyhedra are not cut by the constraint. The condition $\sum_{z \in \mathcal{X}} P(z|a_1, \ldots, a_i) = 1$ obviously holds.

If the optimal alternative x^* is unique, the proposition below states that, using Algorithm 1, the probability assigned to x^* cannot decrease if the DM always answers correctly.

Proposition 1 *Let us denote by x^* a uniquely optimal alternative. At any step i of Algorithm 1, if the answer to query i is correct, then:*

$$P(x^*|a_1, \ldots, a_i) \geq P(x^*|a_1, \ldots, a_{i-1})$$

Proof. Two cases can be distinguished:

Case 1. If $W_{x^*} \not\subseteq W_{x^{(i)} \succsim y^{(i)}}$ and $W_{x^*} \cap W_{x^{(i)} \succ y^{(i)}} \neq \emptyset$, then, as mentioned above, $P(x^*|a_1, \ldots, a_i) = P(x^*|a_1, \ldots, a_{i-1})$ by Equation 4 because $P(a_i|x^*) = P(a_i)$.

Case 2. Otherwise, whatever the answer α of the DM, we have $P(a_i = \alpha|x^*) = \delta$ because the answer to query i is correct. By Equation 5, it follows that:

$$P(x^*|a_1, \ldots, a_i) = \underbrace{\frac{\delta \sum_{y \in \mathcal{Y}} P(y|a_1, \ldots, a_{i-1})}{\sum_{y \in \mathcal{Y}} P(a_i = \alpha|y)P(y|a_1, \ldots, a_{i-1})}}_{\text{ratio } \rho} P(x^*|a_1, \ldots, a_{i-1})$$

We now show that $\rho \geq 1$ for $\delta > \frac{1}{2}$. Let us denote by $\mathcal{Y}_\delta$ the subset of alternatives $y \in \mathcal{Y}$ such that $P(a_i = \alpha|y) = \delta$. We have:

$$\begin{aligned} & \sum_{y \in \mathcal{Y}} P(a_i = \alpha|y)P(y|a_1, \ldots, a_{i-1}) \\ = \; & \delta \sum_{y \in \mathcal{Y}_\delta} P(y|a_1, \ldots, a_{i-1}) + (1 - \delta) \sum_{y \in \mathcal{Y}_{1-\delta}} P(y|a_1, \ldots, a_{i-1}) \\ & \text{because } \mathcal{Y} = \mathcal{Y}_\delta \cup \mathcal{Y}_{1-\delta} \text{ and } \mathcal{Y}_\delta \cap \mathcal{Y}_{1-\delta} = \emptyset \end{aligned}$$

$$\leq \delta \sum_{y \in \mathcal{Y}_\delta} P(y|a_1, \ldots, a_{i-1}) + \delta \sum_{y \in \mathcal{Y}_{1-\delta}} P(y|a_1, \ldots, a_{i-1})$$

$$\text{because } \delta > \frac{1}{2} \text{ (the only case of equality is when } \mathcal{Y}_{1-\delta} = \emptyset)$$

$$= \delta \sum_{y \in \mathcal{Y}} P(y|a_1, \ldots, a_{i-1})$$

Consequently, $\rho \geq 1$ and thus $P(x^*|a_1, \ldots, a_i) \geq P(x^*|a_1, \ldots, a_{i-1})$. □

Toward an efficient implementation. As mentioned above, in order to update the probability of an alternative z, we need to know the relative position of its optimality polyhedron W_z compared to the constraint induced by the new preference statement $f_w(x^{(i)}) \geq f_w(y^{(i)})$. In this purpose, we can consider the Linear Programs (LPs) $\text{opt}\{f_w(x^{(i)}) - f_w(y^{(i)}) | w \in W_z\}$, where opt = min or max.

If the optimal values of both LPs share the same sign, then we can conclude that the polyhedron is not cut by the constraint, otherwise it is cut. To limit the number of LPs that need to be solved (determining the positions of all the polyhedra would indeed require to solve $2n$ LPs), and thereby speed up the Bayesian updating, we propose to approximate the polyhedra by their outer Chebyshev balls (i.e., the smallest ball that contains the polyhedron). Let us denote by r the radius of the Chebyshev ball and by d the distance between the center of the ball and the hyperplane induced by the preference statement:

- if $d \geq r$ then the polyhedron is not cut by the constraint (see Figure 3.a). In order to know whether the polyhedron verifies the constraint or not, we just need to check whether the center of the ball verifies it or not. Thus, in this case, only two scalar products are required.
- if $d < r$ then an exact computation is required because the polyhedron can either be cut by the constraint (Figure 3.b) or not (Figure 3.c). In this way, the use of Chebyshev balls does not impact the results of the Bayesian updating but only speeds up the computations.

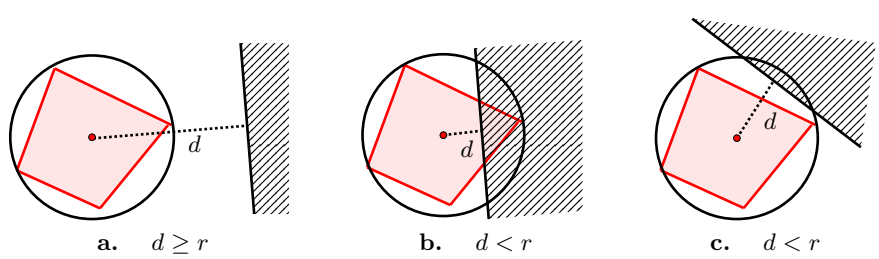

Fig. 3: Example of an approximation of a polyhedron by an outer Chebyshev ball.

4 Experimental Results

Algorithm 1 has been implemented in Python using the *polytope* library to manage optimality polyhedra, and tested on randomly generated instances. We performed the tests on an Intel(R) Core(TM) i7-4790 CPU with 15GB of RAM.

Random generation of instances. To evaluate the performances of Algorithm 1, we generated instances with 100 alternatives evaluated on 5 criteria, all possibly f_w-optimal (i.e., $W_x \neq \emptyset \ \forall x \in \mathcal{X}$). The generation of the performance vectors depends on the aggregation function (WS or OWA) that is considered:

- **WS instances.** An alternative x of the instance is generated as follows: a vector y of size 4 is uniformly drawn in $[0,1]^4$, then x is obtained by setting $x_i = y_i - y_{i-1}$ for $i = 1, \ldots, 5$, where $y_0 = 0$ and $y_5 = 1$. The vectors thus generated all belong to the same hyperplane (because $\sum_{i=1}^{5} x_i = 1$ for all $x \in \mathcal{X}$) and the set of possibly *unique* WS-optimal alternatives is therefore significantly reduced (because the optimality polyhedra of many alternatives are not full dimensional). To avoid this issue, as suggested by Li [13], we apply the square root function on all components x_i for all $x \in \mathcal{X}$ in order to *concavify* the Pareto front. The set of performance vectors obtained is illustrated on the left of Figure 4 in the bicriteria case.
- **OWA instances.** An alternative x is possibly OWA-optimal in a set $\mathcal{X}$ if its *Lorenz curve* $L(x)$ defined by $L_k(x) = \sum_{i=1}^{k} x_{(i)} (k \in [\![1,5]\!])$ is possibly WS-optimal in $\{L(x) : x \in \mathcal{X}\}$. We say that a vector z is *Lorenz* if there exists a vector x such that $z = L(x)$. Given a Lorenz vector z, we denote by $L^{-1}(z)$ any vector x such that $L(x) = z$. For such a vector x, we have $x_{(i)} = z_i - z_{i-1}$ for all $i = 1, \ldots, 5$, where $z_0 = 0$. An alternative x of the instance is generated as follows: we first generate a point y in the polyhedron defined by the following linear constraints:

$$(\mathcal{P}) \begin{cases} y_{i+1} \geq y_i & \forall i \in [\![0,4]\!] \quad (1) \\ (i+1)^2 y_{i+1} - i^2 y_i \geq i^2 y_i - (i-1)^2 y_{i-1} & \forall i \in [\![1,4]\!] \quad (2) \\ \sum_{i=1}^{5} i^2 y_i = \sum_{i=1}^{5} i^2 & (3) \\ y_0 = 0 \end{cases}$$

The set $\mathcal{L} = \{(i^2 y_i)_{i \in [\![1,5]\!]} : y \in \mathcal{P}\}$ contains vectors that are all Lorenz thanks to constraints (1) and (2). Furthermore, they belong to the same hyperplane due to constraint (3), and therefore they are all possibly WS-optimal. Consequently, all the alternatives in the set $\{L^{-1}(z) : z \in \mathcal{L}\}$ are possibly OWA-optimal. As above, to make them all possibly *unique* OWA-optimal, the square root function is applied on each component of vectors z in $\mathcal{L}$. The obtained set is $\mathcal{L}' = \{(i\sqrt{y_i})_{i \in [\![1,5]\!]} : y \in \mathcal{P}\}$. All the vectors in $\mathcal{X}' = \{L^{-1}(z) : z \in \mathcal{L}'\}$ are possibly unique OWA-optimal. Finally, to generate an alternative x in $\mathcal{X}'$, we randomly draw a convex combination $y = \sum_{j=1}^{m} \alpha_j \hat{y}^j$ of vertices $\hat{y}^1, \ldots, \hat{y}^m$ of $\mathcal{P}$. The obtained alternative is then defined by $x = L^{-1}((i\sqrt{y_i})_{i \in [\![1,5]\!]})$. The set of performance vectors obtained is illustrated on the right of Figure 4 in the bicriteria case.

Finally, for both types of instances, a hidden vector w is generated to simulate the preferences of the DM.

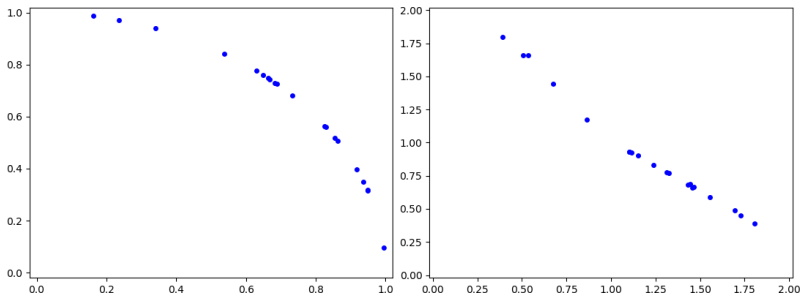

Fig. 4: Example of WS (left) and OWA (right) instances with $n = 20$ and $p = 2$

Simulation of the interactions with the DM. To simulate the DM's answer to query i, we represent the intensity of preference between alternatives $x^{(i)}$ and $y^{(i)}$ by the variable $u^{(i)} = f_w(x^{(i)}) - f_w(y^{(i)}) + \varepsilon^{(i)}$ where $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian noise modelling the possible DM's error, with σ determining how wrong the DM can be. The DM states that $x^{(i)} \succsim y^{(i)}$ if and only if $u^{(i)} \geq 0$.

Analysis of the results. We evaluated the efficiency of Algorithm 1 in terms of the actual rank of the recommended alternative. We considered different values for σ in order to test the tolerance to possible errors. More precisely, $\sigma = 0$ gives an error free model while $\sigma \in \{0.1, 0.2, 0.3\}$ models different rates of errors in the answers to queries. In the considered instances, these values led to, respectively, $3.6\%, 10\%$ and 22% of wrong answers for WS and to $3.2\%, 16\%$ and 25% of wrong answers for OWA. We set $\delta = 0.8$, which corresponds to a prior assumption of an error rate of 20%. Thus, the value of δ we used in the experiments is uncorrelated to the ones of σ. The computation time between two queries is less than 1 second in all cases. Results are averaged over 40 instances.

We first observed the evolution of the actual rank of the MMER alternative over queries (actual rank according to a hidden weighting vector representing the DM's preferences). Figure 5 (resp. 6) shows the curves obtained for WS (resp. OWA). We observe that the mean rank drops below 2 (out of 100 alternatives) after about 14 queries for WS with $\sigma < 0.3$, while the same happens for OWA whatever value of σ. We see that, in practice, the efficiency of the approach can be significantly impacted only when the error rate becomes greater than 20%.

We next compared the performance of Algorithm 1 with a deterministic approach described in [4], that consists in reducing the parameter space after each query (assuming that all answers are correct). The results are illustrated by the boxplots in Figure 7 for WS, and in Figure 8 for OWA. We can see that our probabilistic approach is more tolerant to errors than the deterministic approach. As the value of σ increases, the deterministic approach makes less and

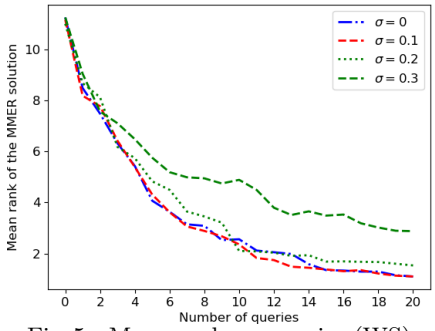

Fig. 5: Mean rank vs. queries (WS)

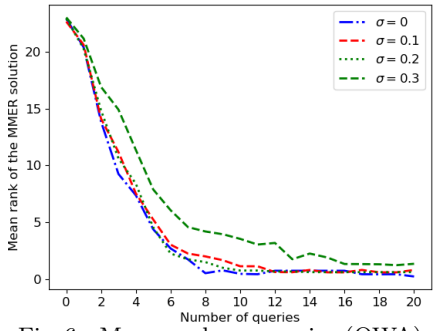

Fig. 6: Mean rank vs. queries (OWA)

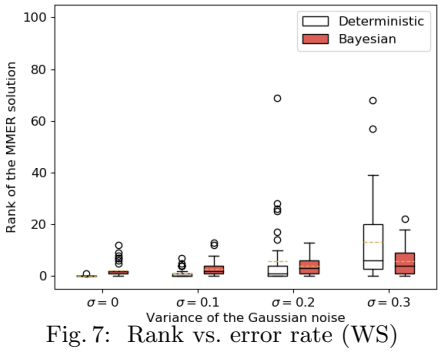

Fig. 7: Rank vs. error rate (WS)

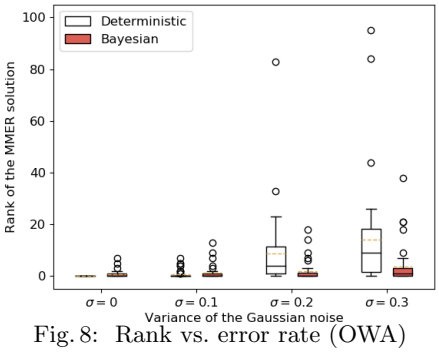

Fig. 8: Rank vs. error rate (OWA)

less relevant recommendations. The deterministic approach indeed recommends, in the worst case, alternatives that are ranked around 90 while it is less than 40 for Algorithm 1. More precisely, when $\sigma = 0.3$ (for both WS and OWA), in more than 75% of instances, Algorithm 1 recommends an alternative with a better rank than the mean rank obtained in the deterministic case.

5 Conclusion

We introduced in this paper a new model based incremental multicriteria elicitation method relying on a partition of the parameter space. The elements of the partition are the optimality polyhedra of the different alternatives, relatively to a weighted sum or an ordered weighted average. A probability distribution is defined over this partition, where each probability represents the likelihood that the true weighting vector belongs to the polyhedron. The approach is robust to possible mistakes in the DM's answers thanks to the incremental revision of probabilities in a Bayesian setting. We provide numerical tests showing the efficiency of the proposed algorithm in terms of number of queries, as well as the interest of using such a probabilistic approach compared to a deterministic approach. A short term research direction is to investigate if it is possible to further speed up the Bayesian updating by using outer Löwner-John ellipsoids instead of Chebyshev balls. The answer is not straightforward because, on the one hand,

the use of ellipsoids indeed refines the approximation of the polyhedra, but on the other hand, this requires the use of matrix calculations to establish whether or not an ellipsoid is cut by the constraint induced by a preference statement. Another natural research direction is to extend our approach to more sophisticated aggregation functions admitting a linear representation, such as Weighted OWAs and other Choquet integrals, to improve our descriptive possibilities.

References

1. Albert, J.H., Chib, S.: Bayesian analysis of binary and polychotomous response data. J. Am. Stat. Assoc. **88**(422), 669–679 (1993)
2. Angilella, S., Corrente, S., Greco, S.: Stochastic multiobjective acceptability analysis for the choquet integral preference model and the scale construction problem. European Journal of Operational Research **240**(1), 172–182 (2015)
3. Benabbou, N., Perny, P.: Incremental weight elicitation for multiobjective state space search. In: AAAI-15. pp. 1093–1099 (2015)
4. Bourdache, N., Perny, P.: Active preference elicitation based on generalized gini functions: Application to the multiagent knapsack problem. In: AAAI 2019. pp. 7741–7748 (2019)
5. Bourdache, N., Perny, P., Spanjaard, O.: Incremental elicitation of rank-dependent aggregation functions based on bayesian linear regression. In: IJCAI 2019. pp. 2023–2029 (2019)
6. Boutilier, C., Patrascu, R., Poupart, P., Schuurmans, D.: Constraint-based optimization and utility elicitation using the minimax decision criterion. Artificial Intelligence **170**(8-9), 686–713 (2006)
7. Braziunas, D., Boutilier, C.: Minimax regret based elicitation of generalized additive utilities. In: Proc. of UAI-07. pp. 25–32 (2007)
8. Chajewska, U., Koller, D., Parr, R.: Making rational decisions using adaptive utility elicitation. In: Proceedings of AAAI-00. pp. 363–369 (2000)
9. Charnetski, J.R., Soland, R.M.: Multiple-attribute decision making with partial information: The comparative hypervolume criterion. Naval Research Logistics Quarterly **25**(2), 279–288 (1978)
10. Grabisch, M., Labreuche, C.: A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. Annals of OR **175**(1), 247–286 (2010)
11. Guo, S., Sanner, S.: Multiattribute bayesian preference elicitation with pairwise comparison queries. In: NIPS. pp. 396–403 (2010)
12. Lahdelma, R., Hokkanen, J., Salminen, P.: SMAA - Stochastic Multiobjective Acceptability Analysis. European J. of Operational Research **106**(1), 137 – 143 (1998)
13. Li, D.: Convexification of a noninferior frontier. Journal of Optimization Theory and Applications **88**(1), 177–196 (1996)
14. Nowak, R.: Noisy generalized binary search. In: Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 1366–1374. Curran Associates, Inc. (2009)
15. Sauré, D., Vielma, J.P.: Ellipsoidal methods for adaptive choice-based conjoint analysis. Operations Research (2019)
16. Wang, T., Boutilier, C.: Incremental utility elicitation with the minimax regret decision criterion. In: IJCAI. vol. 3, pp. 309–316 (2003)
17. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. on Systems, Man, and Cyber. **18**(1), 183–190 (1988)