



HAL
open science

Minimax Regret Approaches for Preference Elicitation with Rank-Dependent Aggregators

Nawal Benabbou, Christophe Gonzales, Patrice Perny, Paolo Viappiani

► **To cite this version:**

Nawal Benabbou, Christophe Gonzales, Patrice Perny, Paolo Viappiani. Minimax Regret Approaches for Preference Elicitation with Rank-Dependent Aggregators. *EURO Journal on Decision Processes*, 2015, 3 (1-2), pp.29-64. <10.1007/s40070-015-0040-6>. <hal-01170030>

HAL Id: hal-01170030

<https://hal.science/hal-01170030v1>

Submitted on 11 Jul 2016

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.



HAL Authorization

Minimax Regret Approaches for Preference Elicitation with Rank-Dependent Aggregators

Nawal Benabbou, Christophe Gonzales, Patrice Perny, Paolo Viappiani
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6
CNRS, UMR 7606, LIP6, 4 Place Jussieu, 75005 Paris, France
email: name.surname@lip6.fr

Abstract

Recently there has been a growing interest in non-linear aggregation models to represent the preferences of a decision maker in a multicriteria decision problem. Such models are expressive as they are able to represent synergies (positive and negative) between attributes or criteria, thus modeling different decision behaviors. They also make it possible to generate Pareto-optimal solutions that cannot be obtained by optimizing a linear combination of criteria. This is the case of rank-dependent aggregation functions such as Ordered Weighted Averages and their weighted extensions, but more generally of Choquet integrals. A key question is how to assess the parameters of such models to best fit decision maker's behaviors or preferences. In this work, adopting a principled decision-theoretic approach, we consider the optimization problem induced by adaptive elicitation using the minimax regret criterion.

1 Introduction

Human decision support and automated decision processes often rely on a mathematical decision model allowing the comparison of the possible alternatives and the selection of a proper solution. This is typically the case in Multicriteria Decision Making where an aggregation function is often used to synthesize different viewpoints into aggregated values representing the overall utilities of the alternatives for the Decision Maker (DM). The aggregation function must be sufficiently expressive to fit DM's preferences in human decision support, but also to model and simulate a prescribed decision behavior in automated decision systems. This explains the diversity of decision models available in the literature but also the increasing interest for sophisticated parameterized models such as the Choquet integral model [18] that can encompass, in the same general formula, various preference value systems, depending on the choice of its preference parameters. The Choquet integral model is indeed very flexible as it subsumes many different aggregation models, including linear models, leximin and lexicographic aggregators, the Ordered Weighted Average operator and its weighted extensions such as the Weighted Ordered Weighted Average aggregator.

A key question is, however, how to assess the preference parameters of such aggregation functions in order to model DM's preferences. For the Choquet integral model, in particular, one needs to assess a capacity, a function defined on the power set of the criteria, so as to control the importance attached to all subsets of criteria, and possibly positive or negative synergies between criteria; thus the number of preference parameters is exponential in the number of criteria. Even if further assumptions are made

(e.g. k -additive capacity) to reduce the number of parameters in the model, parameter learning is still challenging; in general, only limited information about the DM’s preferences will be available (e.g. in the form of partial ranking of alternatives or qualitative statements on the quality of concrete alternatives). Preference handling and elicitation with limited available information is acknowledged to be a crucial task in many application domains, including recommender systems, interface customization and personal assistants [36].

Most of previous works on the elicitation of Choquet integral parameters consider a “static” preference database as input, and focus on the determination of a set of preference parameters that best fits the available preferences. For example, one can minimize a quadratic error between Choquet values and target utility values prescribed by the DM. Alternatively, one can impose some constraints on Choquet values to enforce the decision model to be compatible with a partial or total order available on a subset of alternatives. These approaches are illustrated in many papers see, e.g. [20, 17, 31, 32, 1] and [19] (Chapter 11), and implemented in decision support softwares such as TOMASO [30], MYRIAD [22] and KAPPALAB [17].

Departing from these standard approaches, we are advocating in this paper an incremental elicitation process in which preference queries are selected one by one, to be as informative as possible, so as to progressively reduce the set of possible parameters of a rank-dependent aggregation function until a robust recommendation can be made. This approach relies on previous works on the incremental elicitation of linear aggregation functions (see e.g. ISMAUT method [23]) and exploits recent advances in Artificial Intelligence techniques such as adaptive utility elicitation [9] based on a principled decision-theoretic approach. The system maintains an explicit representation of the possible aggregation functions (sometimes called “belief”) that are consistent with the DM’s behavior, and that is updated whenever new information about the DM is available. Such representation can be useful in order to identify informative queries that can be asked to the DM with the goal of reducing uncertainty as much as possible with a limited number of queries (limiting the DM’s effort). The second central aspect is the robustness in face of utility uncertainty, that means the capability of offering guarantees on the quality of the recommended alternative, given the current available information.

A solution to this is represented by the approach of Boutilier et. al. [6] based on the minimax regret criterion allowing robust recommendations under utility uncertainty with guarantees with respect to the worst case loss. Furthermore, minimax regret computation can also be used to generate effective queries asking to choose the preferred alternative in a subset of alternatives (for a set of two elements, they are called comparison queries) [46]; regret-based elicitation has been successfully demonstrated with real users (DMs) in a prototype for decision support (UT-pref) and validated in a user study [10].

However, adapting minimax regret optimization to non-linear utility models is not trivial. The large parameter space associated to Choquet integrals, as well as the necessity of handling the induced constraints (set monotonicity of the underlying capacity measure), means that particular care has to be given to the optimization. Furthermore, the decision space might be very large, this is often the case in recommender spaces where the possible alternatives can be thousands. We therefore need methodologies that can optimize the proposed criterion in an effective way. We will focus on the case where the possible decisions are given explicitly (as in a database of products) given. This paper aims at proposing some algorithmic methods for assessing rank-dependent aggregation functions by a progressive collect of preference statements using the minimax regret criterion.

The paper is organized as follows: in Section 2, we recall some useful notions related to rank-dependent aggregation functions. In Section 3, we recall the principle of incremental elicitation methods in which queries are selected using the minimax regret criterion and we explain how this approach can be carried out in the case of rank-dependent aggregation functions. Section 4 discusses some methodological points

so as to compute efficiently minimax regret values during the elicitation process, and provides numerical tests. Section 5 provides and discusses experimental results concerning incremental elicitation sessions for the aggregation functions considered in the paper.

2 Rank-dependent Aggregation Functions

Let \mathcal{X} be the set of alternatives (items, products, candidates...) that must be evaluated to make a decision. We consider here a multicriteria evaluation problem and denote $N = \{1, \dots, n\}$ the set of criteria. Every alternative $x \in \mathcal{X}$ is characterized by a performance vector (x_1, \dots, x_n) where $x_i \in [0, 1]$ represents the utility of alternative x with respect to criterion i , with $i \in N$. For simplicity, x will indifferently denote the alternative or its performance vector. In this framework, rank-dependent aggregation functions are scalarizing functions that sort criterion utility values by increasing order before mapping the performance vector into a scalar value (or overall utility value). This allows to assign weights to ranks rather than to components in the aggregation process, so as to control the importance attached to the bottom performance, to the top performance or to any other order statistics. In order to specify such rank-dependent aggregation functions, we will often make use of the permutation (\cdot) defined on N and such that $x_{(1)} \leq \dots \leq x_{(n)}$ (sorting the components of x from the smallest to the largest).

2.1 Ordered Weighted Average

Ordered Weighted Averages (OWA) [50] is one of the simplest families of rank-dependent aggregation functions used in multicriteria aggregation. It is defined as a weighted sum of the components of x , once reordered from the worst to the best. OWA is specified by a vector of weights $w = (w_1, \dots, w_n) \in [0, 1]^n$ such that $\sum_{i \in N} w_i = 1$, where w_k is the weight associated to the component of x ranked in the k -th position (i.e. $x_{(k)}$). More formally, for any alternative $x \in \mathcal{X}$, the OWA value is defined as follows:

Definition 1.

$$\text{OWA}(x; w) = \sum_{i=1}^n w_i x_{(i)}$$

In other words, OWA is the scalar product between the vector of weights w and the sorted vector $(x_{(1)}, \dots, x_{(n)})$. Note that OWA are symmetric functions as the weights do not address the components of x , but those of the sorted vector. This family of rank-dependent aggregators includes the minimum, maximum, median and all order statistics as a special case (just use a vector of weights whose all components except one are null). OWA is often used with decreasing weights in Social Choice theory as a measure of inequalities incorporated in Social Evaluation functions [49] due to the following property:

Proposition 1.

$$\begin{aligned} \forall x \in \mathbb{R}^n, \forall i, k \in N, i < k \text{ s.t. } x_k < x_i, \forall \varepsilon \in (0, x_i - x_k), \\ [\forall i \in \{1, \dots, n-1\}, w_i > w_{i+1}] \Rightarrow \text{OWA}(x; w) < \text{OWA}(x^\varepsilon; w) \end{aligned}$$

where $x^\varepsilon = (x_1, \dots, x_i - \varepsilon, \dots, x_k + \varepsilon, \dots, x_n)$.

Thus, if weights w_i are strictly decreasing as i increases, any move from an alternative to another, resulting in a utility transfer of size ε that reduces inequality between satisfaction of criteria i and k , will

increase the OWA value. Such transfers are known as Pigou-Dalton transfers in Economics where they are used to express the idea of fairness, see e.g.[41, 49]. Besides this fairness property, OWA operators are monotonically increasing with respect to every component. The conjunction of these two properties shows that solutions maximizing an OWA function are those Pareto-optimal solutions that cannot be improved in terms of Pigou-Dalton transfer. Hence OWA maximization using strictly decreasing weights helps promoting balanced solutions while ensuring overall efficiency. The tradeoff between efficiency and fairness is controlled by the weights. Playing with these decreasing weights leaves room for various attitudes. Another way to see the treatment of inequalities by OWA is to remark that choosing decreasing weights gives more importance to the least satisfied criterion, then a less emphasis is given to the second least satisfied criterion and so on. To give an example, if one wishes to compare vectors $x = (0.25, 0.5, 0.75)$ and $y = (0.5, 0.6, 0.4)$ with OWA using the vector of weights $w = (0.5, 0.33, 0.17)$, one gets $\text{OWA}(x; w) = 0.42 < 0.47 = \text{OWA}(y; w)$, and so y is preferred to x . OWA aggregators often appear in fair allocation problems, see e.g. [34, 26] but, beside the fact that we should use decreasing weights, very few information is given on the determination of these weights to model a specific attitude of the DM towards fairness.

2.2 Hybrid Weighted Average

As we noted in the previous subsection, a feature of OWA is to be a family of symmetric aggregation functions. This property seems natural when the criteria are individual points of views in a collective decision problem but it may not be desired in Multicriteria Decision Making, especially when certain criteria are considered more important than others. It is then interesting to consider weighted extensions of OWA, the initial weights of OWA only serving to control the importance that we give to good and bad performances. A solution for that is resorting to two vectors of weights, one for the OWA weights, the other to weight criteria. A first solution in this direction is given by the weighted OWA also known as Weighted Ordered Weighted Average [44]. However, although this solution extends the descriptive possibilities of OWA, it does not allow a sufficient flexibility and control for exploring easily the Pareto set. The impact of criterion weights is quite complex and may be sometimes counter-intuitive [35]. Hence we consider here another weighted variant of OWA named the Hybrid Weighted Average (HWA) [27].

HWA is an operator that makes use of two different sets of weights. The first vector of weights, denoted $p = (p_1, \dots, p_n) \in [0, 1]^n$, is a normalized vector impacting on the importance attached to criteria. The second, denoted $w = (w_1, \dots, w_n)$, is a vector of rank-dependent weights like in the OWA operator. However, unlike in the original OWA, the ranked components are obtained by multiplication with the weights of p . More precisely, for any alternative $x \in \mathcal{X}$, the HWA value is given by:

Definition 2.

$$\text{HWA}(x; p, w) = \text{OWA}(\eta \circ p \circ x; w)$$

where \circ denotes element-by-element multiplication (a.k.a. Hadamard product) and $\eta = (n, \dots, n)$.

Operators HWA and OWA are exactly equivalent when $p = (1/n, \dots, 1/n)$, while HWA is equivalent to the weighted sum when $w = (1/n, \dots, 1/n)$. The insertion of weighting vector p enables to scale up or down components, depending on their importance, before performing the aggregation with OWA. For instance, let us come back to the example provided at the end of section 2.1 to compare $x = (0.25, 0.5, 0.75)$ and $y = (0.5, 0.6, 0.4)$ with OWA using the vector of weights $w = (0.5, 0.33, 0.17)$. If $p = (0.6, 0.3, 0.1)$, we get $\eta \circ p \circ x = (0.45, 0.45, 0.225)$ and $\eta \circ p \circ y = (0.9, 0.54, 0.12)$. Then, we obtain $\text{HWA}(x; p, w) = \text{OWA}((0.45, 0.45, 0.225); w) = 0.3375 < 0.3912 = \text{OWA}((0.9, 0.54, 0.12); w) =$

HWA($y; p, w$) and so y is better than x . This is mainly due to the importance attached to the first component. Conversely, if we use another weighting vector $p = (0.1, 0.3, 0.6)$, putting more importance on the third component, we obtain the reversed preference: HWA($x; p, w$) = OWA($(0.075, 0.45, 1.35); w$) = $0.4155 > 0.3756 =$ OWA($(0.15, 0.54, 0.72); w$) = HWA($y; p, w$). The above example shows that the use of weighting vector p in HWA provides some additional flexibility compared to OWA so as to control the optimal profile.

2.3 Choquet Integral

We consider now a more general family of aggregators known as Choquet integrals [39, 18] which is appealing because it offers a wider flexibility due to a greater number of preference parameters. These parameters are weights attached to every subset of criteria. They allow a fine control of interactions between criteria and take the form of a capacity function on the power set 2^N .

Definition 3. $v : 2^N \rightarrow \mathbb{R}$ is a normalized capacity if $v(\emptyset) = 0$, $v(N) = 1$ and $v(A) \leq v(B)$ whenever $A \subset B$ (monotonicity).

A capacity v is said to be convex or supermodular when $v(A \cup B) + v(A \cap B) \geq v(A) + v(B)$ for all $A, B \subseteq N$, additive when $v(A \cup B) + v(A \cap B) = v(A) + v(B)$ for all $A, B \subseteq N$, and concave or submodular when $v(A \cup B) + v(A \cap B) \leq v(A) + v(B)$ for all $A, B \subseteq N$.

We now use the notion of capacity to define the Choquet integral model. Recall that (\cdot) denotes the permutation of N that sorts the components of x in increasing order. Let $X_{(i)}$ denote the subset of criteria with respect to which x has a utility greater or equal to $x_{(i)}$ (i.e. $X_{(i)} = \{(i), \dots, (n)\}$). Note that $X_{(i+1)} \subset X_{(i)}$ for all $i \in \{1, \dots, n-1\}$. In the following, the nested sequence of sets $X_{(i)}$, $i = 1, \dots, n$, are said to be the level sets of x . Similarly, $Y_{(i)}$ will denote the i -th level set of an alternative y . Using these notations, for any alternative $x \in \mathcal{X}$, the Choquet integral value is defined as follows:

Definition 4.

$$C_v(x) = \sum_{i=1}^n [x_{(i)} - x_{(i-1)}] v(X_{(i)}) \quad \text{with } x_{(0)} = 0.$$

In the definition of the Choquet integral model, the use of a capacity v instead of an arbitrary set-function enforces compatibility with Pareto-dominance (x Pareto-dominates y if $x_i \geq y_i$ for all $i \in N$ and $x_j > y_j$ for some $j \in N$) due to the monotonicity of v with respect to set inclusion. In other words, it ensures that $C_v(x) \geq C_v(y)$ whenever x Pareto-dominates y . Function C_v is known to be convex whenever v is concave (submodular). Conversely, C_v is concave whenever v is convex (supermodular) [28]. The concavity of C_v and therefore the use of a convex capacity has an interpretation in terms of preferences. More precisely, it is shown (see [12]) that if v is convex, then $\forall x^1, x^2, \dots, x^m \in \mathbb{R}^n$ and $\forall \lambda = (\lambda_1, \dots, \lambda_m) \in [0, 1]^m$ such that $\sum_{i=1}^m \lambda_i = 1$:

$$C_v(x^1) = C_v(x^2) = \dots = C_v(x^m) \Rightarrow \forall k \in \{1, 2, \dots, m\}, C_v\left(\sum_{i=1}^m \lambda_i x^i\right) \geq C_v(x^k)$$

For example, when using a convex capacity, if one is indifferent between performance vectors $(0, 1)$ and $(1, 0)$, one will prefer solution $(0.5, 0.5)$, which corresponds to the average of the two vectors ($\lambda_1 = 0.5$, $\lambda_2 = 0.5$), to any of the two initial vectors. Hence using a convex capacity is a way of promoting equity within the Choquet integral model. Obviously, we would obtain reverse preferences with a concave capacity since the associated Choquet integral is convex. Capacity v may both be concave and convex

(i.e. it is additive) and in this case, the Choquet integral boils down to a weighted sum. Alternatively, v could be neither convex nor concave to model more complex preferences (for more details see [19]). Note also that if v is defined by $v(A) = \varphi(\sum_{i \in A} p_i)$ for some positive weights p_i and a non-decreasing function φ such that $\varphi(0) = 0$ and $\varphi(1) = 1$, then the Choquet integral is a WOWA operator (see [44]). Whenever all p_i are equal, we obtain an OWA operator as introduced before, and more generally, any Choquet integral used with a symmetric capacity (i.e. $v(A) = f(|A|)$ for all $A \subseteq N$) is an OWA and vice-versa.

Another useful formulation of the Choquet integral exists, using Möbius masses attached to the capacity [13]. Any set-function v admits indeed an alternative representation in terms of the Möbius inverse:

Definition 5. To any set-function $v : 2^N \rightarrow \mathbb{R}$ is associated $m : 2^N \rightarrow \mathbb{R}$ a mapping called Möbius inverse, defined by:

$$\forall A \subseteq N, m(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} v(B).$$

The capacity v can be reconstructed from its Möbius inverse as follows:

$$\forall A \subseteq N, v(A) = \sum_{B \subseteq A} m(B).$$

Coefficients $m(B)$ for $B \subseteq A$ are called Möbius masses. Interestingly, a set-function whose Möbius masses are non-negative (also known as belief function) is necessarily convex [40]. Using the Möbius inverse, we can define the notion of k -additive capacities as follows [19]:

Definition 6. A capacity is said to be k -additive when its Möbius inverse vanishes for any $A \subseteq N$ such that $|A| > k$, and there exists at least one subset A of exactly k elements such that $m(A) \neq 0$. More formally:

$$\begin{aligned} \forall A \subseteq N, |A| > k &\Rightarrow m(A) = 0 \\ \exists A \subseteq N, |A| = k &\text{ and } m(A) \neq 0. \end{aligned}$$

If $k = 1$ we get an additive capacity. For small values of k greater than 1, k -additive capacities are very useful because in practical situations, they offer a sufficient expressivity to model positive or negative interactions among criteria with a reduced number of parameters. For example, when $k = 2$ the capacity is completely characterized by $(n^2 + n)/2$ coefficients (one Möbius mass for every singleton and every pair). For any alternative $x \in \mathcal{X}$, the Choquet integral value can be rewritten in function of Möbius masses as follows:

$$C_v(x) = \sum_{B \subseteq N} m(B) \min_{i \in B} x_i = m \cdot \tilde{x} \quad (1)$$

where m is the vector of all Möbius masses $m(B)$, $B \subseteq N$, and \tilde{x} is the vector whose components are all $\tilde{x}_B = \min_{i \in B} x_i$, $B \subseteq N$. This makes explicit another interpretation of C_v : it is a linear aggregator in a feature space including 2^n components. This weighted linear combination is much more compact when v is k -additive because terms corresponding to null Möbius masses vanish. For 1-additive capacities, it boils down to a simple weighted sum in the initial space.

3 Regret-based Elicitation of Rank-dependent Aggregators

In this section, we first review the idea of minimax regret for decision-making under utility uncertainty (traditionally used only in linear models) and then, we explain how it can be used within an incremental elicitation procedure to make robust recommendation for rank-dependent aggregators. In the following, we assume the DM needs to make a choice among the dataset of possible alternatives \mathcal{X} .

3.1 Minimax Regret

Minimax regret [38] is a decision criterion that has been advocated as a mean for robust optimization in the presence of data uncertainty [25] and has been used for decision making with utility function uncertainty [5, 37, 6].

Consider a set of preference statements obtained from the DM. These statements could be of different kinds; a common form are answers to comparison queries or prior knowledge. Let \mathcal{P} be the set gathering all pairs of alternatives (x, y) such that x is known to be preferred to y . Let \mathcal{F} be the family of aggregation functions considered as potential candidates to approximate the DM's preferences. Let $\mathcal{F}_{\mathcal{P}}$ be the subset of \mathcal{F} containing all aggregation functions f consistent with the set of observed preference statements \mathcal{P} , i.e. such that $f(x) \geq f(y)$ for all $(x, y) \in \mathcal{P}$. The problem is now to determine the most promising alternative under uncertainty over the DM's aggregation function represented by the set $\mathcal{F}_{\mathcal{P}}$. To this end, the minimax regret approach is based on the following definitions.

Definition 7. *The Pairwise Max Regret (PMR) of alternative $x \in \mathcal{X}$ with respect to $y \in \mathcal{X}$ is:*

$$\text{PMR}(x, y; \mathcal{P}) = \max_{f \in \mathcal{F}_{\mathcal{P}}} f(y) - f(x)$$

The pairwise max regret $\text{PMR}(x, y; \mathcal{P})$ is the worst case regret that the recommender system (RS) imposes to the DM when recommending the alternative x instead of the alternative y . Note that the PMR concept bears some relations with the notion of possible and necessary preferences as introduced in robust ordinal regression [21]. More precisely, when $\text{PMR}(y, x; \mathcal{P})$ is strictly positive, then x is possibly strictly preferred to y (i.e. there exists $f \in \mathcal{F}_{\mathcal{P}}$ such that $f(x) > f(y)$); if $\text{PMR}(y, x; \mathcal{P}) = 0$, then x is possibly equally preferred to y (i.e. there exists $f \in \mathcal{F}_{\mathcal{P}}$ such that $f(x) = f(y)$). If $\text{PMR}(x, y; \mathcal{P})$ is strictly negative, then x is necessarily strictly preferred to y (i.e. for all $f \in \mathcal{F}_{\mathcal{P}}$, $f(x) > f(y)$). When $\text{PMR}(x, y; \mathcal{P}) = 0$, x is necessarily weakly preferred to y (i.e. for all $f \in \mathcal{F}_{\mathcal{P}}$, $f(x) \geq f(y)$). Finally, if $\text{PMR}(x, y; \mathcal{P}) = \text{PMR}(y, x; \mathcal{P}) = 0$ then x is necessarily equally preferred to y , (i.e. for all $f \in \mathcal{F}_{\mathcal{P}}$, $f(x) = f(y)$).

Definition 8. *The Max Regret (MR) of alternative $x \in \mathcal{X}$ is:*

$$\text{MR}(x, \mathcal{X}; \mathcal{P}) = \max_{y \in \mathcal{X}} \text{PMR}(x, y; \mathcal{P}) = \max_{y \in \mathcal{X}} \max_{f \in \mathcal{F}_{\mathcal{P}}} f(y) - f(x)$$

The max regret $\text{MR}(x, \mathcal{X}; \mathcal{P})$ is the worst case regret associated with recommending alternative x instead of any other alternative in \mathcal{X} .

Remark that the use of rank-dependent aggregation functions like OWA, HWA and Choquet assumes that utilities x_i are expressed on a common interval scale. This scale is unique up to any positive affine transformation (of type $\alpha x + \beta$, $\alpha > 0$). As a consequence, the order induced by max regret (MR) over alternatives is meaningful since pairwise max regret (PMR) values are independent of β and the order of regret values is independent on α as well.

Now, the MR-optimal value (called Minimax Regret) is defined as follows:

Definition 9. *The Minimax Regret (MMR) over \mathcal{X} is:*

$$\text{MMR}(\mathcal{X}; \mathcal{P}) = \min_{x \in \mathcal{X}} \text{MR}(x, \mathcal{X}; \mathcal{P})$$

Recommending any MR-optimal alternative (i.e. any $x \in \mathcal{X}$ such that $\text{MR}(x, \mathcal{X}; \mathcal{P}) = \text{MMR}(\mathcal{X}; \mathcal{P})$) allows the RS to guarantee that the worst case regret of not choosing the true optimal alternative for the DM is minimized. The minimax regret criterion is natural in this decision context where the RS has to select an alternative while the aggregation function of the DM is still imprecisely known. Given an alternative x chosen by the RS, using this decision criterion amounts to considering a fictitious DM (the ‘‘adversary’’) that will always choose the function $f \in \mathcal{F}_{\mathcal{P}}$ that maximizes the utility gap $\max_{y \in \mathcal{X}} f(y) - f(x)$ over $\mathcal{F}_{\mathcal{P}}$. In the sequel, MR-optimal solutions will be referred to as MMR-alternatives.

When \mathcal{F} is the set of linear or piece-wise linear aggregators, $\mathcal{F}_{\mathcal{P}}$ is defined by a number of constraints approximately equal to the size of \mathcal{P} and so PMR can be computed exactly quite efficiently by solving a simple linear program. This approach is more tractable than probabilistic models of utility that rely on computationally expensive Bayes updates [4, 11]. In our setting, we assume that alternatives are evaluated according to n different criteria, and the performance vectors are known and available to the system. When instead alternatives are constructed by solving a configuration problem, optimization over product space \mathcal{X} can be formulated as a CSP or a mixed integer program (MIP) [6, 8].

Remark that MMR cannot increase by adding preference constraints (i.e. by asking new preference queries to the DM) and it usually decreases (see [7], pages 194-202).

Observation 1. *Given two sets of statements $\mathcal{P}, \mathcal{P}'$ s.t. $\mathcal{P} \subseteq \mathcal{P}'$, the following inequalities hold:*

- $\text{PMR}(x, y; \mathcal{P}') \leq \text{PMR}(x, y; \mathcal{P}), \forall x, y \in \mathcal{X}$
- $\text{MR}(x, \mathcal{X}; \mathcal{P}') \leq \text{MR}(x, \mathcal{X}; \mathcal{P}), \forall x \in \mathcal{X}$
- $\text{MMR}(\mathcal{X}; \mathcal{P}') \leq \text{MMR}(\mathcal{X}; \mathcal{P})$

We present now the general principle of incremental elicitation procedures based on this observation.

3.2 Incremental Elicitation

As proposed in [48, 6, 10], the minimax regret criterion can be used within an incremental elicitation process that progressively asks queries to the DM in order to assess preference parameters until the MMR (or worst case regret when recommending any MMR-alternative) drops below a given threshold; thanks to Observation 1, when the DM answers an additional query, we know that MMR cannot increase (and it will usually decrease), so asking new queries will make possible to make better recommendations by reducing the regret due to not choosing the true optimal alternative for the DM; therefore, regret-based elicitation supports a principled termination condition.

In order to better interpret regret values, MR can be normalized so as to belong to the [0,1] interval, 1 corresponding to the initial MR obtained before asking any query, and 0 being the ultimate goal when the uncertainty on parameters is sufficiently reduced to be able to identify the preferred alternative with certainty. With such normalized values, a minimax regret (MMR) value of 0.1, at a given step of the incremental elicitation process, simply means that the initial utility gap due to uncertainty has been reduced to 10% of its initial value. An interpretation of regret with practical use can be made in domains

where utility can be expressed in equivalent monetary values. The monetary equivalent of MMR can be displayed to the DM, together with an explanation of the kind “if you buy the recommended product now, instead of continuing the elicitation process, the most you can lose is this dollar amount”. This idea has been demonstrated in a prototype of a rental recommendation systems and validated with user experiments in [10].

Queries can be of different types. For instance, bound queries ask if the overall utility (aggregate value) of an alternative is higher or lower than a given reference value (they might be cognitively hard to answer). Comparison queries are relatively easy to ask as they require the DM to compare a pair of alternatives and state which one is preferred among the two. Even assuming that the type of the query is fixed, there are of course several queries that one can ask at each step. Notice that some queries will be more informative than others. For instance, asking to compare an alternative with one that is Pareto-dominated by the former will provide no value (MMR will not change); instead asking to compare two potentially good alternatives will possibly constitute a good query. Choosing good queries is therefore essential in order to be able to provide good recommendations reasonably quickly (asking only few queries).

Assume that, at a given point in the interaction, \mathcal{P} is the set gathering all pairs of alternatives (x, y) such that x is known to be preferred to y and $\mathcal{F}_{\mathcal{P}}$ is the set of aggregation functions consistent with \mathcal{P} . Assuming a query q and all its possible responses \mathcal{R}_q , we can consider the space $\mathcal{F}_{\mathcal{P} \cup \{r\}}$ in the scenario that the DM answers r , for any $r \in \mathcal{R}_q$. One can judge that a good query will significantly reduce MMR in each of the scenarios (recall that, for any $r \in \mathcal{R}_q$, $\text{MMR}(\mathcal{X}; \mathcal{P} \cup \{r\})$ can only be lower or equal to $\text{MMR}(\mathcal{X}; \mathcal{P})$ from Observation 1). Thus, a (non probabilistic) notion of myopic value of information [46] when the aggregation function must lie in $\mathcal{F}_{\mathcal{P}}$ can be defined as follows:

Definition 10. *The Worst Case Minimax Regret (WMMR) one step ahead of a query q is:*

$$\text{WMMR}(q; \mathcal{P}) = \max_{r \in \mathcal{R}_q} \text{MMR}(\mathcal{X}; \mathcal{P} \cup \{r\}).$$

The optimal query according to this measure is then $q_{\mathcal{P}}^* = \arg \min_q \text{WMMR}(q; \mathcal{P})$. Notice that a query can have significantly different WMMR values depending on $\mathcal{F}_{\mathcal{P}}$ (i.e. the currently known preferences \mathcal{P} and the assumption about the family of aggregation functions that can model the DM’s preferences \mathcal{F} may have a strong impact on the value of the query). Thus, when considering comparison queries, in order to determine $q_{\mathcal{P}}^*$, we have to compute for any two alternatives x and y the MMR value in the two spaces corresponding to possible answers (in one x has higher utility than y , in the other the reverse is true). One can equivalently look for the query maximizing the worst case regret reduction, i.e. $\min_{r \in \mathcal{R}_q} [\text{MMR}(\mathcal{X}; \mathcal{P}) - \text{MMR}(\mathcal{X}; \mathcal{P} \cup \{r\})]$. The optimization of comparison queries (and choice queries, that extend comparison queries to a set of elements) is thoroughly discussed in [46].

Since computing the myopically optimal query $q_{\mathcal{P}}^*$ is computationally intensive, we mainly focus here on a very efficient query strategy (though not optimal in general) called the Current Solution Strategy (CSS) [6]. At a given step of the interaction, as next query, the CSS asks the DM to compare a MMR-alternative $x_{\mathcal{P}}^*$ (arbitrarily chosen in $\arg \min_{x \in \mathcal{X}} \text{MR}(x, \mathcal{X}; \mathcal{P})$) to an alternative $y_{\mathcal{P}}^*$ arbitrarily chosen in $\arg \max_{y \in \mathcal{X}} \text{PMR}(x_{\mathcal{P}}^*, y; \mathcal{P})$. If the DM states that $x_{\mathcal{P}}^*$ is preferred to $y_{\mathcal{P}}^*$, then $(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)$ is inserted in \mathcal{P} . Otherwise, $(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)$ is inserted instead. In both cases, $\mathcal{F}_{\mathcal{P}}$ is updated accordingly. This elicitation scheme is very frequent in practice, see e.g. [6, 10, 29, 14, 2].

The CSS is often a good heuristic for the following reason: if $x_{\mathcal{P}}^*$ is preferred to $y_{\mathcal{P}}^*$, when computing $\text{MMR}(\mathcal{X}; \mathcal{P} \cup \{(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)\})$ in the next cycle of the incremental elicitation procedure, $y_{\mathcal{P}}^*$ will not maximize $\text{PMR}(x_{\mathcal{P}}^*, y; \mathcal{P} \cup \{(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)\})$ over \mathcal{X} (unless the maximum is zero, but in that case, $x_{\mathcal{P}}^*$ is necessarily the preferred alternative for the DM). Therefore, if $y_{\mathcal{P}}^*$ was the unique alternative achieving $\text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P})$, we will necessarily have $\text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P} \cup \{(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)\}) < \text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P})$. Hence, if $x_{\mathcal{P}}^*$

was the unique MMR-alternative given \mathcal{P} , then we will have $\text{MMR}(\mathcal{X}; \mathcal{P} \cup \{(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)\}) < \text{MMR}(\mathcal{X}; \mathcal{P})$. Conversely, if $y_{\mathcal{P}}^*$ is stated to be preferred to $x_{\mathcal{P}}^*$, now the latter is dominated by the former and so $\text{MR}(y_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P} \cup \{(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)\}) \leq \text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P} \cup \{(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)\})$. Moreover, if $y_{\mathcal{P}}^*$ was the unique alternative achieving $\text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P})$, then the expression above is necessarily strict: $\text{MR}(y_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P} \cup \{(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)\}) < \text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P} \cup \{(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)\})$. Therefore, in practice, the MMR will often decrease significantly.

Proposition 2. *Asking queries according to the CSS cannot lead to inconsistencies; $\mathcal{F}_{\mathcal{P}}$ cannot become empty during the incremental elicitation procedure.*

Proof. Let \mathcal{P} be a set of preference statements such that $\mathcal{F}_{\mathcal{P}} \neq \emptyset$. We want to show that $\mathcal{F}_{\mathcal{P}}$ will not vanish after adding the preference statement $(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)$ or $(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)$ to \mathcal{P} . We assume here that $\text{MMR}(\mathcal{X}; \mathcal{P}) > 0$, otherwise $x_{\mathcal{P}}^*$ is necessarily one of the preferred alternatives and there is no need to ask further queries. Since $\text{MMR}(\mathcal{X}; \mathcal{P}) > 0$, we can deduce that $\text{PMR}(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*; \mathcal{P}) > 0$, and so we know that there exists an aggregation function in $\mathcal{F}_{\mathcal{P}}$ such that $y_{\mathcal{P}}^*$ is preferred to $x_{\mathcal{P}}^*$ (see Definition 7), i.e. $\mathcal{F}_{\mathcal{P} \cup \{(y_{\mathcal{P}}^*, x_{\mathcal{P}}^*)\}} \neq \emptyset$. Assume now that there is no aggregation function in $\mathcal{F}_{\mathcal{P}}$ such that $x_{\mathcal{P}}^*$ is preferred to $y_{\mathcal{P}}^*$. In that case, we have $f(x_{\mathcal{P}}^*) < f(y_{\mathcal{P}}^*)$ for all $f \in \mathcal{F}_{\mathcal{P}}$. Then, we can deduce that $\text{MR}(y_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P}) < \text{MR}(x_{\mathcal{P}}^*, \mathcal{X}; \mathcal{P})$, i.e. $x_{\mathcal{P}}^* \notin \arg \min_{x \in \mathcal{X}} \text{MR}(x, \mathcal{X}; \mathcal{P})$ which contradicts the fact that $x_{\mathcal{P}}^*$ is among MMR-alternatives. Thus, we can conclude that there exists an aggregation function in $\mathcal{F}_{\mathcal{P}}$ such that $x_{\mathcal{P}}^*$ is preferred to $y_{\mathcal{P}}^*$, i.e. $\mathcal{F}_{\mathcal{P} \cup \{(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*)\}} \neq \emptyset$. Therefore, asking to compare $x_{\mathcal{P}}^*$ to $y_{\mathcal{P}}^*$ cannot lead to inconsistencies. \square

In the following subsections, we address how regret-based optimization can handle rank-dependent operators. We focus on PMR computation (as in this optimization lie the specific difficulties of treating rank-dependent operators) and we delay computation of the MMR to section 4.1 (as the latter is computed as a series of PMR and can be presented in a general way for all models). To do so, we will assume to be at a given point in the incremental elicitation procedure, and \mathcal{P} is the current set gathering all pairs of alternatives (x, y) such that x is known to be preferred to y . We will also denote $\Theta_{\mathcal{P}}$ the set of parameters consistent with \mathcal{P} (defining the space of aggregation functions $\mathcal{F}_{\mathcal{P}}$); for instance, whenever \mathcal{F} is the set of Choquet integrals, then $\Theta_{\mathcal{P}}$ is the set of capacities consistent with \mathcal{P} .

3.3 Application to OWA and HWA

To address how regret-based optimization can handle the OWA and HWA operators, we start by quickly reminding how PMR is computed in the (simpler) case of a linear utility model, where criteria are aggregated using a weighted sum.

Weighted Sum In the case of linear utility models, for a given vector of weights w , the utility (aggregate value) of $x \in \mathcal{X}$ is simply the scalar product $w \cdot x$. Consequently, for any two alternatives $x, y \in \mathcal{X}$, the pairwise max regret of x with respect to y with a linear utility model assumption is then:

$$\text{PMR}_{WS}(x, y; \mathcal{P}) = \max_{w \in \Theta_{\mathcal{P}}} [y \cdot w - x \cdot w]$$

Note that we make now the assumption about the utility model (or family of aggregators) explicit and so $\Theta_{\mathcal{P}}$ is here the set of vectors of weights compatible with the known preferences \mathcal{P} .

The computation of $\text{PMR}_{WS}(x, y; \mathcal{P})$ can be done efficiently with a linear program; constraints given by DM's preference statements (such as alternative x is preferred to alternative y) are encoded by linear constraints ($w \cdot x \geq w \cdot y$) and do not pose particular problems.

We are now interested in the cases where the DM's aggregation function is represented by either OWA or HWA. The uncertainty is on the weights w (for OWA) or for both the weights w and p (for HWA). In particular, the computation of PMR between option x and option y for the OWA and HWA operators consists in the following maximizations:

- $\text{PMR}_{\text{OWA}}(x, y; \mathcal{P}) = \max_{w \in \Theta_{\mathcal{P}}} \text{OWA}(y; w) - \text{OWA}(x; w)$ where $\Theta_{\mathcal{P}}$ is for OWA the set of weight vectors compatible with \mathcal{P} ,
- $\text{PMR}_{\text{HWA}}(x, y; \mathcal{P}) = \max_{(p, w) \in \Theta_{\mathcal{P}}} \text{HWA}(y; p, w) - \text{HWA}(x; p, w)$ where $\Theta_{\mathcal{P}}$ is for HWA the Cartesian product of the set of criterion weights and the set of rank-dependent weights compatible with \mathcal{P} .

OWA PMR_{OWA} can be easily optimized by using the standard methods by a simple manipulation of the dataset. Let x^\uparrow denote the vector whose components are those of the alternative x rearranged from the worst to the best (i.e. $x^\uparrow = (x_{(1)}, \dots, x_{(n)})$) and let \mathcal{X}^\uparrow be the set $\{x^\uparrow \mid x \in \mathcal{X}\}$. Thus, we have the following equality:

$$\text{PMR}_{\text{OWA}}(x, y; \mathcal{P}) = \text{PMR}_{\text{WS}}(x^\uparrow, y^\uparrow; \mathcal{P})$$

The preceding observation enables us to solve $\text{PMR}_{\text{OWA}}(x, y; \mathcal{P})$ by using linear programming by simply sorting the components of x and y :

$$\begin{aligned} \max_w \quad & w \cdot y^\uparrow - w \cdot x^\uparrow \\ \text{s.t.} \quad & w_i \geq 0 \quad \forall i \in N \\ & w_i \geq w_{i+1} \quad \forall i \in \{1, \dots, n-1\} \\ & \sum_{i \in N} w_i = 1 \\ & w \cdot a^\uparrow \geq w \cdot b^\uparrow \quad \forall (a, b) \in \mathcal{P} \end{aligned} \tag{2}$$

Note that constraints (2) are used to obtain strictly decreasing weights so as to favor well-balanced solutions. This equation can be dropped if we do not want to require decreasing weights.

HWA In the case of HWA, pairwise max regret computation consists in maximizing over $(p, w) \in \Theta_{\mathcal{P}}$ the value $\sum_{i=1}^n w_i n[(p \circ y)_{(i)} - (p \circ x)_{(i)}]$. This is not easily solvable using linear programming, neither it is easy to represent the constraints arising from the set of preference statements \mathcal{P} .

Our solution for PMR_{HWA} maximization relies on a hybrid technique involving sampling and linear programming. Notice that the difficulty relies on not knowing the order of the elements induced by the multiplication with the components of the p vector: if p was known, then the elements could be sorted and PMR_{HWA} could have been computed with a linear program like for OWA (see Definition 2).

Given two performance vectors x and y , we aim at estimating $\text{PMR}_{\text{HWA}}(x, y; \mathcal{P})$. To do so, we sample (uniformly at random) a set of l normalized weighting vectors denoted p^i , $i \in \{1, \dots, l\}$, representing possible values for p . For all $i \in \{1, \dots, l\}$, we construct the vectors $\eta \circ p^i \circ x$ and $\eta \circ p^i \circ y$, where $\eta = (n, \dots, n)$, and cast the problem as one of PMR_{OWA} computation where the set $\Theta_{\mathcal{P}}$ of feasible rank-dependent weights for OWA is constructed from \mathcal{P} and p^i . The estimated value of $\text{PMR}_{\text{HWA}}(x, y; \mathcal{P})$ is

then the maximum obtained with each of these surrogate optimizations. More formally:

$$p^i \sim U[0, 1]^n \text{ s.t. } \sum_{j=1}^n p_j^i = 1, \quad i \in \{1, \dots, l\}$$

$$\text{PMR}_{\text{HWA}}(x, y; \mathcal{P}) \approx \max_{i \in \{1, \dots, l\}} \text{PMR}_{\text{OWA}}(\eta \circ p^i \circ x, \eta \circ p^i \circ y; \mathcal{P})$$

where $\eta = (n, \dots, n)$.

Obviously, the quality of the approximation relies on the number of samples (and with a large enough number of samples, the estimated value will converge to the true value of $\text{PMR}_{\text{HWA}}(x, y; \mathcal{P})$).

3.4 Application to Choquet integrals

For the Choquet integral model, the uncertainty is on the capacity v and so $\Theta_{\mathcal{P}}$ is the set of all capacities compatible with the known preferences \mathcal{P} . Letting v_B be the variable used for the capacity $v(B)$ for any set $B \subseteq N$, for any two alternatives x and y in \mathcal{X} , the computation of $\text{PMR}_{C_v}(x, y; \mathcal{P})$ (i.e. $\max_{v \in \Theta_{\mathcal{P}}} C_v(y) - C_v(x)$) can be done as follows:

$$\max_v \sum_{i=1}^n [y_{(i)} - y_{(i-1)}] v_{Y_{(i)}} - \sum_{i=1}^n [x_{(i)} - x_{(i-1)}] v_{X_{(i)}} \quad (3)$$

$$\text{s.t. } v_{\emptyset} = 0, v_N = 1$$

$$v_A \leq v_{A \cup \{i\}} \quad \forall i \in N, \forall A \subseteq N \setminus \{i\} \quad (4)$$

$$\sum_{i=1}^n [a_{(i)} - a_{(i-1)}] v_{A_{(i)}} \geq \sum_{i=1}^n [b_{(i)} - b_{(i-1)}] v_{B_{(i)}} \quad \forall (a, b) \in \mathcal{P} \quad (5)$$

where $X_{(i)}$, $Y_{(i)}$, $A_{(i)}$ and $B_{(i)}$ respectively denote the i -th level set of x , y , a and b (see Definition 4). Note that x and y being known, $C_v(x)$ and $C_v(y)$ are linear functions of capacities values $v_B, B \subseteq N$, hence the objective function is linear. Constraints of type (4) are used to enforce monotonicity with respect to set inclusion. However, the number of such constraints is exponential in the number of criteria. Moreover the number of variables is also exponential in the number of criteria. This linear problem remains solvable easily for any two alternatives x and y as long as the number of criteria is kept reasonably small; then, $\text{MR}_{C_v}(x, \mathcal{X}; \mathcal{P})$ is derived from Definition 8. Note that another approach to compute $\text{MR}_{C_v}(x, \mathcal{X}; \mathcal{P})$ is proposed in [43]. It relies on the fact that $\text{MR}_{C_v}(x, \mathcal{X}; \mathcal{P}) = \max_{v^i \in \Theta_{\mathcal{P}}^*} G(v^i)$ where $G(v) = \max_{y \in \mathcal{X}} C_v(y) - C_v(x)$ and $\Theta_{\mathcal{P}}^*$ are the extreme points of $\Theta_{\mathcal{P}}$. However, using this approach within an incremental elicitation procedure seems prohibitive for general capacities due to the large size of $\Theta_{\mathcal{P}}^*$.

Another formulation of interest is the one obtained using the vector of Möbius masses m , whose components being decision variables instead of capacity values, taking advantage of the fact that $C_v(x) = m \cdot \tilde{x}$ (recall that \tilde{x} is the vector whose components are $\tilde{x}_B = \min_{i \in B} x_i$, for all $B \subseteq N$, see (1)). Letting m_B be the variable used for $m(B)$ for any set $B \subseteq N$, the computation of $\text{PMR}_{C_v}(x, y; \mathcal{P})$ can be achieved

by solving the following linear program:

$$\begin{aligned}
& \max_m \quad m \cdot \tilde{y} - m \cdot \tilde{x} \\
& \text{s.t.} \quad m_\emptyset = 0, \quad \sum_{B \subseteq N} m_B = 1 \\
& \quad \quad \sum_{B: \{i\} \subseteq B \subseteq A} m_B \geq 0 \quad \forall A \subseteq N, \forall i \in A \\
& \quad \quad m \cdot \tilde{a} \geq m \cdot \tilde{b} \quad \forall (a, b) \in \mathcal{P}
\end{aligned} \tag{6}$$

Here, monotonicity with respect to set inclusion is guaranteed by the constraints given in Equation (6) that are the direct translation of constraints (4), as already observed in previous works [13, 32]. Their number still grows exponentially with the number of criteria. The number of variables is also exponential in the number of criteria. Fortunately, whenever the number of criteria is large, more compact LP formulations can be derived for some particular subclasses of capacities. For example, we consider now the case of 2-additive capacities and the case of belief functions before considering general capacities.

2-Additive Capacities Recall that a 2-additive capacity has non-null Möbius masses only on subsets having at most two elements. Hence we have only $(n^2 + n)/2$ variables, one Möbius mass $m_{\{i\}}$ for every singleton and one denoted indifferently $m_{\{i,j\}}$ or $m_{\{j,i\}}$ for every subset $\{i, j\}$ of size two. Using these notations, the previous linear program simplifies as follows:

$$\begin{aligned}
& \max_m \quad \sum_{B \subseteq N: |B| \leq 2} m_B (\tilde{y}_B - \tilde{x}_B) \\
& \text{s.t.} \quad m_\emptyset = 0, \quad \sum_{i \in N} m_{\{i\}} + \sum_{i, j \in N: i < j} m_{\{i, j\}} = 1
\end{aligned} \tag{7}$$

$$\begin{aligned}
& \sum_{j \in B} m_{\{i, j\}} \geq 0 \quad \forall i \in N, \forall B \subseteq N: \{i\} \subseteq B \\
& \sum_{B \subseteq N: |B| \leq 2} m_B (\tilde{a}_B - \tilde{b}_B) \geq 0 \quad \forall (a, b) \in \mathcal{P}
\end{aligned} \tag{8}$$

As stated above, the general exact method to optimize $\text{PMR}_{\text{Cv}}(x, y; \mathcal{P})$ requires handling the monotonicity constraints. This requires $n2^{n-1}$ constraints (due to (8)) and two constraints for ensuring a normalized capacity (7). This results in a number of constraints which is exponential in the number of criteria, even assuming a 2-additive capacity with a quadratic number of variables! In the 2-additive case, Hüllermeier et al. [42] proposed to reduce the number of constraints by using the fact that normalized (monotonic) capacities form a convex polytope. As a consequence, all normalized (monotonic) capacities can be written as convex combinations of its n^2 extreme points given below:

$$m_1^A(X) = \begin{cases} 1 & \text{if } X = A \\ 0 & \text{otherwise} \end{cases} \quad A \in Z_1 \quad \text{and} \quad m_2^A(X) = \begin{cases} 1 & \text{if } X \neq \emptyset \text{ and } X \subset A \\ -1 & \text{if } X = A \\ 0 & \text{otherwise} \end{cases} \quad A \in Z_2$$

where $Z_1 = \{A \subseteq N \mid 1 \leq |A| \leq 2\}$ and $Z_2 = \{A \subseteq N \mid |A| = 2\}$. Thus, at the expense of adding a quadratic number of variables, which are the weights of the convex combination, the monotonicity (8) and normalization (7) constraints can be replaced by the following $(3n^2 + n + 2)/2$ constraints:

$$\begin{aligned}
m_B &= \sum_{A \in Z_1} \alpha_1^A m_1^A(B) + \sum_{A \in Z_2} \alpha_2^A m_2^A(B) \quad \forall B \subseteq N \text{ s.t. } |B| \leq 2 \\
\alpha_1^A &\geq 0 \quad \forall A \in Z_1 \\
\alpha_2^A &\geq 0 \quad \forall A \in Z_2 \\
\sum_{A \in Z_1} \alpha_1^A + \sum_{A \in Z_2} \alpha_2^A &= 1
\end{aligned}$$

Therefore, for 2-additive capacities, this LP-formulation based on Möbius masses is more compact and can be solved efficiently to compute $\text{PMR}_{\text{Cv}}(x, y; \mathcal{P})$ and therefore $\text{MR}_{\text{Cv}}(x, \mathcal{X}; \mathcal{P})$. Moreover the approach proposed in [43] to compute $\text{MR}_{\text{Cv}}(x, \mathcal{X}; \mathcal{P})$ may also be practically feasible, as observed by the author, since the number of extreme points of 2-additive capacities is quadratic in the number of criteria. However, using this approach within an incremental elicitation procedure may be prohibitive, since the size of $\Theta_{\mathcal{P}}^*$ becomes larger as the size of \mathcal{P} increases.

Belief Functions Belief functions are a subclass of convex capacities characterized by positive Möbius masses. We have seen in the previous section that convex capacities might be a natural choice to model the preference for balanced solutions within the Choquet integral model. Hence, we may want to use belief functions for that purpose. Whenever we restrict the set of capacities to belief functions, monotonicity constraints are naturally satisfied due to the non-negativity of Möbius masses. In this case, constraints in (6) are no longer necessary for monotonicity and can be replaced by all non-negativity constraints $m_B \geq 0$, $B \subseteq N$. Moreover, if we further restrict the analysis to 2-additive set-functions, we get:

$$\begin{aligned}
\max_m \quad & \sum_{B \subseteq N: |B| \leq 2} m_B (\tilde{y}_B - \tilde{x}_B) \\
\text{s.t.} \quad & m_{\{i\}} \geq 0 \quad \forall i \in N \\
& m_{\{i,j\}} \geq 0 \quad \forall i, j \in N : i < j \\
& \sum_{i \in N} m_{\{i\}} + \sum_{i,j \in N: i < j} m_{\{i,j\}} = 1 \\
& \sum_{B \subseteq N: |B| \leq 2} m_B (\tilde{a}_B - \tilde{b}_B) \geq 0 \quad \forall (a, b) \in \mathcal{P}
\end{aligned}$$

For 3-additive capacities, one needs to account for masses $m_{\{i,j,k\}}$ but the principle is the same.

General capacities As mentioned above, in the general case, the computation of any $\text{PMR}_{\text{Cv}}(x, y; \mathcal{P})$ involves exponentially many constraints so as to guarantee the monotonicity of capacities considered in the elicitation process. We have shown above that, for some subclasses of capacities (i.e. 2-additive, belief functions), the number of such constraints that are actually needed to formulate the PMR optimization problem, is much lower. Unfortunately, these subclasses correspond to specific attitudes that do not necessarily match with the DM's preferences. We present now another incremental elicitation procedure without prior assumption on the capacity (for more details, the reader is referred to [3]). To do so, we first focus on the PMR-optimization problem and then, we present the associated query strategy.

This procedure is based on a different type of queries involving binary alternatives of type 1A0 where 1A0 represents, for any subset of criteria A , a fictitious alternative with a top utility on all criteria in A

and a bottom utility on all other criteria. Precisely, the DM will be asked to compare such alternatives to constant utility profiles of type $\Lambda = (\lambda, \dots, \lambda)$. Let \mathcal{P} be a set of preference statements obtained by only asking this type of queries to the DM.

Note that with the Choquet integral model, for any $v \in \Theta_{\mathcal{P}}$, we have $C_v(\Lambda) = \lambda$ for any $\Lambda \in [0, 1]^n$ and $C_v(1A0) = v(A)$ for any $A \subseteq N$. Consequently, if the DM states that $1A0$ (resp. Λ) is preferred to Λ (resp. $1A0$), then adding the pair $(1A0, \Lambda)$ (resp. $(\Lambda, 1A0)$) to \mathcal{P} amounts to imposing the constraint $v(A) \geq \lambda$ (resp. $v(A) \leq \lambda$) to all capacities $v \in \Theta_{\mathcal{P}}$. Thus, all constraints given by Equation (5) can be replaced by boundary constraints over variables by updating the boundaries of an interval $[l_A, u_A]$ whenever a preference statement involving $1A0$ is observed, $A \subseteq N$ (i.e. at any step of the elicitation procedure, $[l_A, u_A]$ defines the set of admissible values of $v(A)$ according to the observed preferences). Moreover, for all $v \in \Theta_{\mathcal{P}}$, we know that v is monotonic by Definition 3, and so $v(A) \geq \lambda$ (resp. $v(A) \leq \lambda$) implies that $v(B) \geq \lambda$ (resp. $v(B) \leq \lambda$) for all $B \supseteq A$ (resp. $B \subseteq A$). Assume now that, whenever a pair of type $(1A0, \Lambda)$ (resp. $(\Lambda, 1A0)$) is inserted in \mathcal{P} , the boundaries of all intervals $[l_B, u_B]$ are updated accordingly, $B \supseteq A$ (resp. $B \subseteq A$). Then, the following proposition holds:

Proposition 3. *Any set-function $v : \mathcal{A} \rightarrow [0, 1]$, with $\mathcal{A} \subset 2^N$, such that:*

1. $v(A) \in [l_A, u_A]$ for all $A \in \mathcal{A}$ (boundary constraints)
2. $v(A) \leq v(B)$ for all $A, B \in \mathcal{A}$ s.t. $A \subset B$ (monotonicity constraints)

can be completed into a capacity in $\Theta_{\mathcal{P}}$.

Proof. We can construct a complete capacity by setting first $v(\{i\})$ to $l_{\{i\}}$ for all $i \in N$ such that $\{i\} \notin \mathcal{A}$; then, we can set iteratively the value of $v(A)$ to $\max\{l_A, \max_{\{i \in A\}} v(A \setminus \{i\})\}$, for all A proper subset of N such that $A \notin \mathcal{A}$ and $v(A \setminus \{i\})$ is known for all $i \in A$. Thus, the constructed set-function defined on 2^N satisfies monotonicity and preference constraints by construction of all boundary intervals. \square

Finally, thanks to the latter proposition, we can remove from Equation (4) and Equation (5) all monotonicity and preference constraints involving any subset of criteria that is not in the objective function (3). Thus, recalling that $X_{(i)}$ (resp. $Y_{(i)}$) denotes the i -th level set of x (resp. y) used in the computation of its Choquet value (see Definition 4), the optimization of $\text{PMR}_{C_v}(x, y; \mathcal{P})$ for any two alternatives $x, y \in \mathcal{X}$ can be achieved by solving the following simpler linear program:

$$\max_v \sum_{i=1}^n [y_{(i)} - y_{(i-1)}] v_{Y_{(i)}} - \sum_{i=1}^n [x_{(i)} - x_{(i-1)}] v_{X_{(i)}} \quad (9)$$

$$\text{s.t. } v_{X_{(i+1)}} \leq v_{X_{(i)}} \quad \forall i \in \{1, \dots, n-1\} \quad (10)$$

$$v_{Y_{(i+1)}} \leq v_{Y_{(i)}} \quad \forall i \in \{1, \dots, n-1\} \quad (11)$$

$$v_{X_{(i)}} \leq v_{Y_{(j)}} \quad \forall i, j \in N \text{ s.t. } X_{(i)} \subset Y_{(j)} \quad (12)$$

$$v_{Y_{(i)}} \leq v_{X_{(j)}} \quad \forall i, j \in N \text{ s.t. } Y_{(i)} \subset X_{(j)} \quad (13)$$

$$l_{X_{(i)}} \leq v_{X_{(i)}} \leq u_{X_{(i)}} \quad \forall i \in N \quad (14)$$

$$l_{Y_{(i)}} \leq v_{Y_{(i)}} \leq u_{Y_{(i)}} \quad \forall i \in N \quad (15)$$

where Equations (10-13) and Equations (14-15) are respectively the new monotonicity and preference constraints. In this linear program, there are at most $2n - 1$ variables (all level sets of x and y) instead of 2^n (all subsets of criteria). However, some monotonicity constraints are redundant or unnecessary, meaning that this linear program can be further simplified. More precisely, for all decision variables v_A , let w_A denote its coefficient in the objective function (9) given below:

- $\forall i \in N, [X_{(i)} \neq Y_{(i)}] \Rightarrow [w_{X_{(i)}} = -(x_{(i)} - x_{(i-1)}) \leq 0 \text{ and } w_{Y_{(i)}} = y_{(i)} - y_{(i-1)} \geq 0]$
- $\forall i \in N, [X_{(i)} = Y_{(i)}] \Rightarrow [w_{X_{(i)}} = y_{(i)} - y_{(i-1)} - (x_{(i)} - x_{(i-1)})]$

Consequently, since the objective function has to be maximized, we know that v_A will be as small (resp. large) as possible for all $A \in \{X_{(i)} \mid i \in N, X_{(i)} \neq Y_{(i)}\}$ (resp. $A \in \{Y_{(i)} \mid i \in N, Y_{(i)} \neq X_{(i)}\}$) and so Equation (12) is not necessary to solve the PMR_{Cv} optimization problem. Besides, the number of constraints given by Equation (13) can be reduced by noting that if there exists $i, j \in N$ such that $Y_{(i)} \subset X_{(j)}$, then we also have $Y_{(i)} \subset X_{(k)}$ for all $k \in \llbracket 1; j \rrbracket$, which imposes redundant constraints with Equation (10). Consequently, the constraint $v_{Y_{(i)}} \leq v_{X_{(j)}}$ should be imposed only if $Y_{(i)} \subset X_{(j)}$ and $Y_{(i)} \not\subset X_{(j+1)}$. However, if $Y_{(i+1)} \subseteq X_{(j)}$ is also satisfied, then there is still a redundancy but with Equation (11) this time. Thus, the constraint $v_{Y_{(i)}} \leq v_{X_{(j)}}$ should be imposed only if $Y_{(i)} \subset X_{(j)}$, $Y_{(i)} \not\subset X_{(j+1)}$ and $Y_{(i-1)} \not\subset X_{(j)}$. Finally, the number of monotonicity constraints that are actually needed to solve this optimization problem is now below $3(n-1)$. This simpler linear program can be solved using an iterative procedure (see [3] for further details).

Example 1. Consider a problem defined on 5 criteria, i.e. $N = \{1, 2, 3, 4, 5\}$, and two performance vectors $x = (0.3, 0.1, 0.6, 0.5, 0.8)$ and $y = (0.1, 0.2, 0.3, 0.5, 0.4)$. The linear program associated with $\text{PMR}_{\text{Cv}}(x, y; \mathcal{P})$ computation is the following:

$$\begin{aligned} \max_v \quad & 0.1v_{\{2,3,4,5\}} + 0.1v_{\{4,5\}} + 0.1v_{\{4\}} - 0.2v_{\{1,3,4,5\}} - 0.1v_{\{3,4,5\}} - 0.1v_{\{3,5\}} - 0.2v_{\{5\}} \\ \text{s.t.} \quad & v_{\{5\}} \leq v_{\{3,5\}}, v_{\{3,5\}} \leq v_{\{3,4,5\}}, v_{\{3,4,5\}} \leq v_{\{1,3,4,5\}}, v_{\{1,3,4,5\}} \leq v_N & (16) \\ & v_{\{4\}} \leq v_{\{4,5\}}, v_{\{4,5\}} \leq v_{\{3,4,5\}}, v_{\{3,4,5\}} \leq v_{\{2,3,4,5\}}, v_{\{2,3,4,5\}} \leq v_N & (17) \\ & l_{\{5\}} \leq v_{\{5\}} \leq u_{\{5\}}, l_{\{3,5\}} \leq v_{\{3,5\}} \leq u_{\{3,5\}}, l_{\{3,4,5\}} \leq v_{\{3,4,5\}} \leq u_{\{3,4,5\}} \\ & l_{\{1,3,4,5\}} \leq v_{\{1,3,4,5\}} \leq u_{\{1,3,4,5\}}, l_N \leq v_N \leq u_N, l_{\{4\}} \leq v_{\{4\}} \leq u_{\{4\}} \\ & l_{\{4,5\}} \leq v_{\{4,5\}} \leq u_{\{4,5\}}, l_{\{2,3,4,5\}} \leq v_{\{2,3,4,5\}} \leq u_{\{2,3,4,5\}} \end{aligned}$$

where the constraints in (16) are those given by Equation (10) and the constraints in (17) correspond to Equation (11); the other constraints are those associated with Equations (14-15). Thus, there are only eight monotonicity constraints (16-17).

Since the procedure focuses on a specific type of queries, we cannot use the CSS to choose the next query in the elicitation procedure. The WMMR criterion (see Definition 10) is used instead, i.e. we are looking for a query involving a pair $(A^* \subseteq N, \lambda^* \in [l_{A^*}, u_{A^*}])$ that minimizes the minimax regret in the worst scenario of answers. More formally, the pair $(A^* \subseteq N, \lambda^* \in [l_{A^*}, u_{A^*}])$ minimizes $\max\{\text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P} \cup \{(1A0, \Lambda)\}), \text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P} \cup \{(\Lambda, 1A0)\})\}$. In order to determine such a pair, we have to compute, for all $A \subseteq N$:

$$\min_{\lambda \in [l_A, u_A]} \max\{\text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P} \cup \{(1A0, \Lambda)\}), \text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P} \cup \{(\Lambda, 1A0)\})\} \quad (18)$$

Then, A^* is the set minimizing (18) and λ^* is the value in $[l_{A^*}, u_{A^*}]$ achieving this minimum. Similarly to what is observed for utility functions over consequences [48], we observe here that, given a set $A \subseteq N$, the two functions $\text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P} \cup \{(1A0, \Lambda)\})$ and $\text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P} \cup \{(\Lambda, 1A0)\})$ necessarily intersect because the former is a decreasing function of λ , the latter is an increasing one, and they have the same maximum ($\text{MMR}_{\text{Cv}}(\mathcal{X}; \mathcal{P})$). Thus, for any set $A \subseteq N$, the intersection of these functions gives the value of λ achieving the minimum (18); it can be easily computed by a bisection algorithm relying on the relative positions of the two curves at two distinct points. However, it may happen that the value defined in (18) decreases below $\text{MMR}(\mathcal{X}; \mathcal{P})$ for no set $A \subseteq N$, which means that the WMMR optimal query will not necessarily induce a minimax regret reduction. In such cases, to decide between all possible queries, a less conservative criterion than WMMR can be used (for example, see [3]).

Note however that the determination of the next query implies to select A within the $2^n - 2$ possible proper subsets of N , which becomes cumbersome as the number of criteria increases. To make this selection step more efficient, we focus on sets directly involved in the computation of $\text{MMR}_{C_V}(\mathcal{X}; \mathcal{P})$ as a heuristic and so of $\text{PMR}_{C_V}(x_{\mathcal{P}}^*, y_{\mathcal{P}}^*; \mathcal{P})$, where $x_{\mathcal{P}}^*$ is an arbitrarily chosen MMR-alternative and $y_{\mathcal{P}}^*$ is arbitrarily chosen in $\arg \max_{y \in \mathcal{X}} \text{PMR}(x_{\mathcal{P}}^*, y; \mathcal{P})$. These sets are all level sets of $x_{\mathcal{P}}^*$ and $y_{\mathcal{P}}^*$. By doing so, the heuristic will further constrain utility parameters involved in $\text{MMR}_{C_V}(\mathcal{X}; \mathcal{P})$, thus intuitively reducing MMR_{C_V} (a fact that has already been observed in [3]). Thus, at most $2n - 2$ sets are investigated (all level sets of $x_{\mathcal{P}}^*$ and $y_{\mathcal{P}}^*$, excluding N) instead of $2^n - 2$.

3.5 Observations

We are interested in comparing the behavior of MMR under the different family of aggregators considered here. We have noted above that constraining the set of feasible aggregation functions by asking queries leads to lower regrets (see Observation 1). The same observation can be done whenever constraining the set of aggregation functions by assuming a more restricted family of aggregation functions.

Observation 2. *Given the same set of preference statements \mathcal{P} , OWA with monotone weights (OWA_m) is associated with lower minimax regret than OWA with (arbitrary) positive weights (OWA_p).*

$$\text{MMR}_{\text{OWA}_m}(\mathcal{X}; \mathcal{P}) \leq \text{MMR}_{\text{OWA}_p}(\mathcal{X}; \mathcal{P})$$

Similarly, a linear utility model is associated with lower regret than a model based on Choquet (the latter having more flexibility and including linear utility as a special case), when considering the same set of preference statements. The same can be stated comparing OWA and Choquet integral model.

Observation 3. *Given the same set of preference statements \mathcal{P} , the linear utility model is associated with lower minimax regret than the Choquet integral model.*

$$\text{MMR}_{\text{WS}}(\mathcal{X}; \mathcal{P}) \leq \text{MMR}_{C_V}(\mathcal{X}; \mathcal{P})$$

Given the same set of preference statements \mathcal{P} , OWA is associated with lower minimax regret than the Choquet integral model.

$$\text{MMR}_{\text{OWA}}(\mathcal{X}; \mathcal{P}) \leq \text{MMR}_{C_V}(\mathcal{X}; \mathcal{P})$$

The set of linear utility models and OWA models are disjoint (excluding degenerate cases), therefore there is no strong relation between regrets in these two cases. However, as we shall see in the experimental section, we observe that minimax regret is often lower in practice when assuming OWA rather than the linear utility model. One intuition of why this happens is the following. The rearrangement of criteria for computing minimax regret under the assumption of a OWA utility model, induces a mapping into a more compact space. In particular, notice that the dominance relation \succ_P according to Pareto will still hold if we re-arrange the criteria of each, sorting them in increasing order of satisfaction.

Observation 4. *For any two alternatives $x, y \in \mathcal{X}$:*

$$x \succ_P y \Rightarrow x^\uparrow \succ_P y^\uparrow$$

where vector x^\uparrow (resp. y^\uparrow) is obtained by sorting the components of x (resp. y) from the worst to the best.

Therefore it follows that $\text{ND}(\mathcal{X}^\uparrow)$, the non dominated set (all alternatives for which there is no other alternative that \succ_P -dominates them) of the data set $\mathcal{X}^\uparrow = \{x^\uparrow \mid x \in \mathcal{X}\}$, is a subset of $\text{ND}(\mathcal{X})$, the non dominated set of \mathcal{X} .

Observation 5. $\text{ND}(\mathcal{X}^\dagger) \subseteq \text{ND}(\mathcal{X})$

Since a dominated alternative cannot be among MMR-alternatives and neither will be chosen by the DM, they can be pruned (we need to maintain only alternatives in $\text{ND}(\mathcal{X}^\dagger)$). This means that, in many instances, the RS and the DM will have to choose among a more restricted set of choices.

4 Minimax Regret Optimization

In this section, we consider how to address the computation of MMR. In Section 4.1, we describe how a MMR-alternative can be found using search methods, solving a reduced number of PMR optimizations (thoroughly discussed in the previous section). Then, we perform a number of numerical tests and report computation times in a number of different settings in Section 4.2.

4.1 Implementation with Search Methods

PMR computation constitutes the main building block for regret-based optimization; in order to determine a MMR-alternative, a series of PMR computations has to be performed to identify the alternative that has the smallest PMR against all other possible alternatives (i.e. the smallest MR). The minimax regret criterion is essentially a game between the recommender system (RS) and an adversary, a fictitious decision maker, whose most preferred choice is always the alternative that maximizes the pairwise max regret (PMR) of the RS's choice.

In order to compute MMR, the naive approach would test all $n(n-1)$ combinations of choices for the RS and for the adversary by computing PMR for each of these combinations. Notice that only non dominated alternatives ($\text{ND}(\mathcal{X})$) need to be checked; indeed, Pareto-dominated alternatives can be removed as they will never be chosen by either the RS (alternatives that Pareto-dominate them always have smaller PMR) or the adversary (alternatives that Pareto-dominate them will induce larger regrets).

A better idea is to implement a search problem [7]. The optimization is a search process in an extremely broad but shallow tree; the root corresponds to the RS's choice of the recommendation and for each possible choice, the adversary can respond with an alternative (aimed at maximizing PMR). The leafs of the tree correspond to PMR computations between the corresponding RS's choice and the adversary's. During the search, the following bounds are maintained:

- $U(\mathcal{X})$: an upper bound on MMR, consisting in the MR of the best option found so far for the RS's choice, initialized to positive infinity.
- $L(x^i)$: a lower bound on MR associated with $x^i \in \mathcal{X}$ initialized to negative infinity.

After computing $\text{PMR}(x^i, x^j; \mathcal{P})$, the pairwise max regret of x^i against x^j , the lower bound $L(x^i)$ is updated to $\max\{L(x^i), \text{PMR}(x^i, x^j; \mathcal{P})\}$. Whenever it happens that $L(x^i) \geq U(\mathcal{X})$, the RS can prune option x^i (because its MR can only be higher than $L(x^i)$ and the RS has already found an alternative with lower MR).

This method results in a significant reduction of PMR computations. It is however important to carefully choose the evaluation order of RS's choices in order to prune as much as possible. This can result that (empirically) only almost a linear number of PMR optimizations is needed instead of a quadratic number. In our implementation, following [45], we augment the tree search optimization of [7] considering the following strategy that appears to be very efficient. Before starting the search, we improve the lower bound $L(x^i)$ for all $x^i \in \mathcal{X}$ as follows. We consider k instances of weights (these represent a standard

weight vector for WS, OWA and HWA models and a vector of Möbius masses for Choquet integral model; however the treatment is identical in all these cases), that we call reference vectors and we denote them with w^1, \dots, w^k . These are obtained by finding, for each component j of these vectors, the compatible vector of weights that maximizes the component j (i.e. $w^j = \arg \max_{w \in \Theta_{\mathcal{P}}} w_j$ where $\Theta_{\mathcal{P}}$ is the set of parameters consistent with \mathcal{P}); k is therefore the number of parameters of our problem (e.g. n for WS and OWA, $2n$ for HWA and $n \frac{(n+1)}{2}$ for 2-additive Choquet). For each alternative $x^i \in \mathcal{X}$, we update the value of $L(x^i)$ (initialized to negative infinity) k times. More precisely, for each reference vector w^j , we first evaluate the regret, when assuming w^j represents DM's preferences, of choosing x^i instead of the best item according to w^j (i.e. $x_{w^j}^* = \arg \max_{x \in \mathcal{X}} w^j \cdot x$). Then, we update as follows:

$$L(x^i) = \max\{w^j \cdot x_{w^j}^* - w^j \cdot x^i, L(x^i)\} \quad (19)$$

Using this method (that only involves standard vector multiplications), it is not unusual that a large number of alternatives are immediately discarded from RS's choices without any PMR computation ($L(x^i)$ being larger than the current value of $U(\mathcal{X})$).

We also consider the following enhancements in the search evaluation:

- To determine the order for evaluating RS's choices, we sort alternatives by $L(x^i)$ values computed thanks to reference vectors (19).
- Whenever we evaluate a new candidate x^i for the RS's choice, if $L(x^i) < U(\mathcal{X})$, we solve the optimization problem $w^- = \min_{w \in \Theta_{\mathcal{P}}} w \cdot x^i$ before starting the series of PMR computations. The weights w^- are the one minimizing the value of x^i over the set of aggregation functions consistent with \mathcal{P} . We then use w^- to check if option x^i can be pruned: that happens if the regret between the best alternative according to w^- and x^i , when assuming w^- represents DM's preferences, exceeds the current bound $U(\mathcal{X})$. Since w^- is the less favorable instance to x^i , it is often the case.
- Moreover, if x^i has not been pruned after considering the previous pruning rule, we use w^- for establishing in which order to evaluate the possible choices of the adversary. More precisely, we sort the alternatives according to w^- from the best to the worst. The intuition is that alternatives that have high aggregate value with w^- have the potential of achieving high PMR against the chosen x^i (and hence x^i will be pruned sooner when comparing $L(x^i)$ to $U(\mathcal{X})$).

4.2 Experiments

In the following experiments, we aim at quantifying computation time for optimizing MMR, for the different aggregation models, in different settings (varying dataset size, the number of criteria n and the number of preference statements). In these experiments, we consider the following datasets \mathcal{X} :

- two datasets of a hundred random alternatives ($n = 10$ and $n = 20$) sampled uniformly in $[0, 1]^n$.
- datasets consisting of the Pareto set obtained by solving randomly generated instances of the multiobjective Knapsack problem:

$$\begin{aligned} \max_x \quad & \left(\sum_{i=1}^m x_i v_i^1, \dots, \sum_{i=1}^m x_i v_i^n \right) \\ \text{s.t.} \quad & \sum_{i=1}^m w_i x_i \leq W \\ & x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

where the max operator represents the Pareto-dominance, m is the number of items, w_i is the mass of item i , x_i is the decision variable associated with the item i , v_i^j is the value of item i on the criterion j and W is the knapsack's capacity.

And we consider the following different aggregation models:

- OWA_m : OWA model with decreasing positive weights.
- OWA_p : OWA model with positive weights.
- Linear: Weighted sum model without any restriction on the set of weights.
- C2A_p : 2-additive Choquet integral model with positive Möbius masses.
- HWA: HWA model with positive p weights and decreasing positive w weights.
- Cv : Choquet integral without any restriction on the capacity.

Note that we will consider HWA and Cv separately as PMR_{HWA} can not be optimized exactly using mathematical programming and PMR_{Cv} is not based on comparison queries (preference statements must compare fictitious alternatives).

Table 1: Comparison of MMR optimizations under different aggregation models.

Dataset	n	size	preferences	OWA_m		OWA_p		Linear		C2A_p	
				time	checks	time	checks	time	checks	time	checks
Uniform	10	100	0	0.16	101	0.13	101	0.13	101	0.21	101
Uniform	10	102	5	0.19	101	0.16	101	0.21	101	0.36	202
Uniform	10	100	10	0.21	101	0.19	101	0.24	101	0.48	101
Uniform	10	102	20	0.25	101	0.25	101	0.28	102	0.61	101
Uniform	20	100	0	0.19	99	0.15	99	0.15	99	0.59	99
Uniform	20	100	5	0.21	99	0.18	99	0.21	198	0.92	99
Uniform	20	100	10	0.24	99	0.21	99	0.26	100	1.07	206
Uniform	20	100	20	0.28	99	0.26	99	0.29	99	1.49	99
Knapsack	5	1000	0	1.47	999	1.21	999	1.19	999	1.21	999
Knapsack	5	1000	5	1.71	999	1.72	999	2.61	999	2.65	999
Knapsack	5	1000	10	1.88	999	2.09	999	2.61	2997	3.61	2997
Knapsack	5	1000	20	2.26	999	2.55	999	2.97	2040	4.74	1191
Knapsack	5	5353	0	8.04	5352	6.38	5352	6.32	5352	6.57	5352
Knapsack	5	5353	5	9.17	5352	8.52	5352	13.01	5352	15.71	5352
Knapsack	5	5353	10	10.68	5352	12.59	5352	15.35	5352	20.56	5352
Knapsack	5	5353	20	12.30	5352	13.28	5352	17.41	5352	32.42	14401
Knapsack	10	1000	0	1.58	999	1.21	999	1.21	999	1.46	999
Knapsack	10	1000	5	2.02	999	2.49	1998	2.97	999	3.24	1266
Knapsack	10	1000	10	2.03	999	2.40	999	3.27	999	3.93	999
Knapsack	10	1000	20	2.44	999	3.21	999	3.09	1864	4.67	2085
Knapsack	10	6518	0	10.14	6517	7.77	6517	7.71	6517	9.74	6517
Knapsack	10	6518	5	12.52	6517	15.75	13034	35.90	13034	28.47	6517
Knapsack	10	6518	10	13.00	6517	20.12	6517	30.94	6517	54.19	6517
Knapsack	10	6518	20	15.89	6517	21.45	11067	29.47	6517	50.17	25413

In Table 1, we report computation times¹ in seconds (time) for the optimization of MMR and the number of PMR computations (checks) to show the effectiveness of the pruning techniques described in Section 4.1. In order to evaluate how the number of preference statements (preferences) impacts on the two previous quantities, we report computational results at different steps of the incremental elicitation procedure (varying the number of preference statements). Results have been obtained by averaging over 30 runs and preference statements have been generated by randomly choosing two alternatives in the datasets.

From Table 1, we can notice that the number of PMR computations is usually much lower than $n(n-1)$, thanks to the very efficient pruning technique. In fact, when no preference statement is present, the alternative that is tested first is the minimax optimal alternative (thanks to the efficient ordering produced by the reference vectors). Moreover, MMR optimization with the $C2A_p$ model is more computationally intense (due to the larger number of parameters) but all computations are relatively fast and require less than a minute almost all the time. Finally, as expected, the computation time increases with the number of preference statements (however this is not a general rule) for all considered models. This is due to the fact that adding a preference statement amounts to adding a constraint in the linear program optimizing PMR.

Table 2: MMR optimizations under Choquet integral model.

Dataset	n	size	preferences	time	Dataset	n	size	preferences	time
Knapsack	5	150	0	3.72	Knapsack	5	1000	0	34.59
Knapsack	5	150	5	2.79	Knapsack	5	1000	5	34.37
Knapsack	5	150	10	3.00	Knapsack	5	1000	10	27.58
Knapsack	5	150	20	2.68	Knapsack	5	1000	20	28.89
Knapsack	6	150	0	3.27	Knapsack	6	1000	0	26.22
Knapsack	6	150	5	2.37	Knapsack	6	1000	5	29.93
Knapsack	6	150	10	2.36	Knapsack	6	1000	10	28.64
Knapsack	6	150	20	2.13	Knapsack	6	1000	20	28.42
Knapsack	7	150	0	5.07	Knapsack	7	1000	0	87.09
Knapsack	7	150	5	3.70	Knapsack	7	1000	5	34.73
Knapsack	7	150	10	2.97	Knapsack	7	1000	10	25.19
Knapsack	7	150	20	2.67	Knapsack	7	1000	20	22.37
Knapsack	8	150	0	4.89	Knapsack	8	1000	0	28.59
Knapsack	8	150	5	4.36	Knapsack	8	1000	5	34.98
Knapsack	8	150	10	4.14	Knapsack	8	1000	10	26.24
Knapsack	8	150	20	3.98	Knapsack	8	1000	20	22.05
Knapsack	9	150	0	8.46	Knapsack	9	1000	0	59.54
Knapsack	9	150	5	5.18	Knapsack	9	1000	5	38.87
Knapsack	9	150	10	3.69	Knapsack	9	1000	10	29.40
Knapsack	9	150	20	2.76	Knapsack	9	1000	20	23.42
Knapsack	10	150	0	2.67	Knapsack	10	1000	0	31.12
Knapsack	10	150	5	1.82	Knapsack	10	1000	5	18.19
Knapsack	10	150	10	1.95	Knapsack	10	1000	10	15.19
Knapsack	10	150	20	1.62	Knapsack	10	1000	20	13.01

We now focus on the optimization method that can elicit a Choquet integral with a general capacity (see Section 3.4). In the experiments, we consider the Pareto set of multiobjective Knapsack problems

¹Linear optimizations are done under MatLab using the Gurobi solver on a machine with an Intel Core i7 CPU 3.60GHz with 16 GB of memory.

restricted to 150 and 1000 alternatives. Results have been obtained by averaging over 50 runs² and random preferences are imposed of the type $1A0 \succsim \Lambda$ or $\Lambda \succsim 1A0$.

From Table 2, we can see that the higher the number of preference statements, the lower the computation time of MMR_{C_V} optimization, unlike the previous models; in this case, a new preference statement involving a set of criteria $A \subseteq N$ reduces the space of feasible values of the decision variable $v(A)$.

Table 3: MMR optimizations under HWA model.

Dataset	n	size	preferences	HWA 100 particles		HWA 1000 particles	
				time	checks	time	checks
Knapsack	2	40	0	22.56	40600	266.63	482000
Knapsack	2	40	20	9.12	3900	80.21	39000
Knapsack	3	150	0	113.04	152700	1253.19	702000
Knapsack	3	150	20	148.55	14900	1578.94	149000

Finally, we consider computation time of MMR_{HWA} optimization. To do so, we consider the Pareto set of multiobjective Knapsack problems (restricted to 40 and 150 alternatives); here again, considering preference statements amounts to adding constraints in the linear program used for computing PMR. Results are given in Table 3 and have been obtained by averaging over 30 runs³.

Our technique for PMR_{HWA} optimization relies on sampling a set of criterion weights p^i , computing the corresponding PMR_{OWA} for each of them and finally taking the highest value (see Section 3.3). In our simulations, as expected, computation time of heavily depends on the number of samples used; the larger computation time is due to repeating PMR_{OWA} optimizations with different p^i , multiplying the running time by a large constant factor.

5 Incremental Elicitation Experiments

In a series of experiments, we consider how our regret optimization methods can be used for incremental elicitation; we make use of these optimizations in order to decide the next query to ask (e.g. for the CSS, an arbitrarily chosen MMR-alternative $x_{\mathcal{P}}^*$ has to be compared to an alternative arbitrarily chosen in $\arg \max_{y \in \mathcal{X}} \text{PMR}(x_{\mathcal{P}}^*, y; \mathcal{P})$) and measure MMR in function of the number of queries asked.

First, we aim at comparing the incremental elicitation based on the following preference models: OWA model with decreasing positive weights (OWA_m), OWA model with positive weights (OWA_p), HWA model with positive p weights and decreasing positive w weights (HWA), linear model using a weighted sum (Linear) and 2-additive Choquet with positive Möbius masses (C2A_p). While there are different types of queries that could be asked to the DM, such as comparisons and bound queries (local/global), we consider here comparison queries dictated by the very efficient CCS (see Section 3.2), that are holistic comparisons between two alternatives.

For each of these preference model assumptions (that is the assumption made by the RS when computing the MMR-alternative), we experiment with simulated DMs that answer queries according to an aggregation functions from the same family of aggregators with random parameters (the weights of Linear,

²Linear optimizations are done using the Gurobi library of Java.

³Linear optimizations are done under MatLab using the Gurobi solver on a machine with an Intel Core i7 CPU 3.60GHz with 16 GB of memory.

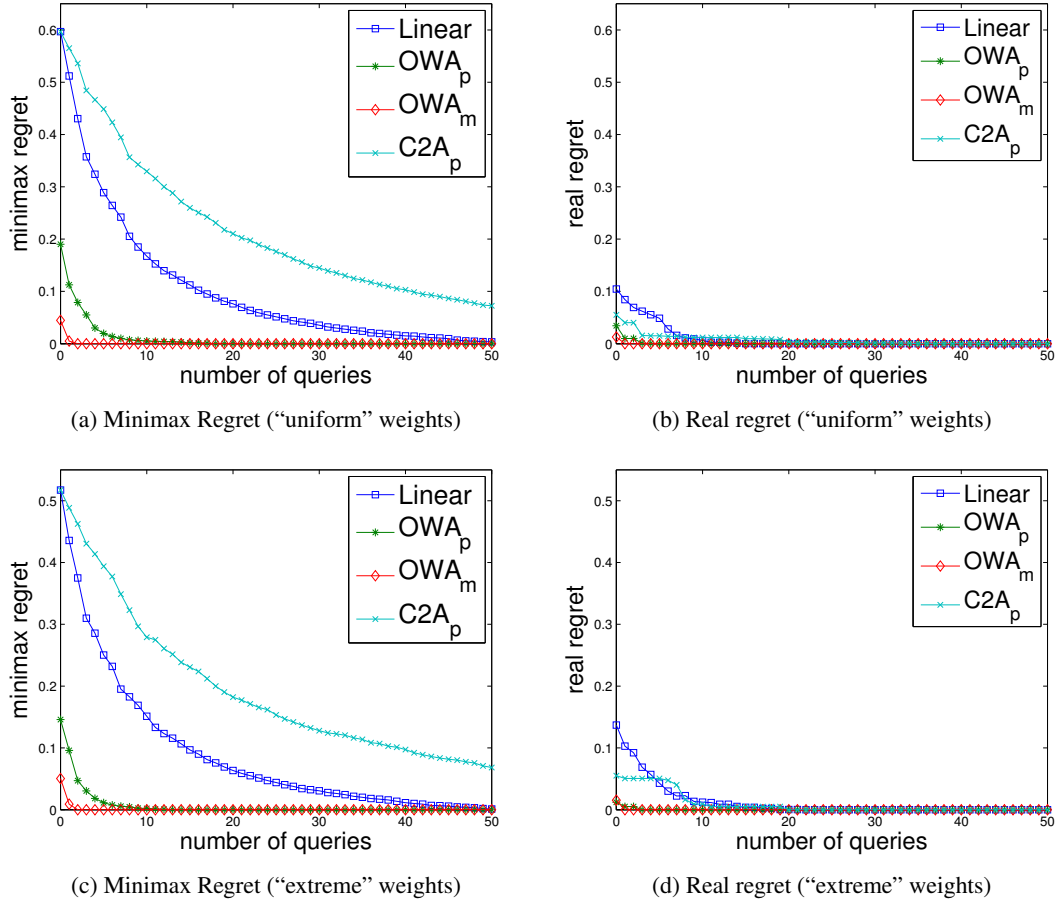


Figure 1: Elicitation with the CSS: uniformly randomly generated datasets (150 alternatives, $n = 10$).

OWA_m and OWA_p , the weights p and w for HWA and the Möbius masses for $C2A_p$). To randomly generate the weights of simulated DMs, we consider different generative models:

- a set of "uniform" weights that are randomly sampled (for each criterion) from an uniform distribution between 0 and 1, and then normalized so that each vector sums up to 1.
- a set of "extreme" weights that are generated by sampling, for each criterion, a value 0 with probability 0.4, a value 1 with probability 0.1, a value sampled from $U[0, 0.3]$ with probability 0.3 and a value sampled $U[0, 1]$ with probability 0.2; each vector of weights is then normalized.

The second generative model has been considered in order to avoid experimenting only with aggregation functions that score too similarly the different criteria when considering a large number of criteria. We consider a simulated setting where the DM answers comparison queries by indicating which alternative is preferred according to his/her sampled aggregation function. We start from an empty set of preferences and we compute MMR at each step of the incremental elicitation process. We plot on the same graphs the results obtained with different models in order to explicitly compare the difference in the values of regret. Results have been obtained by averaging over 30 runs.

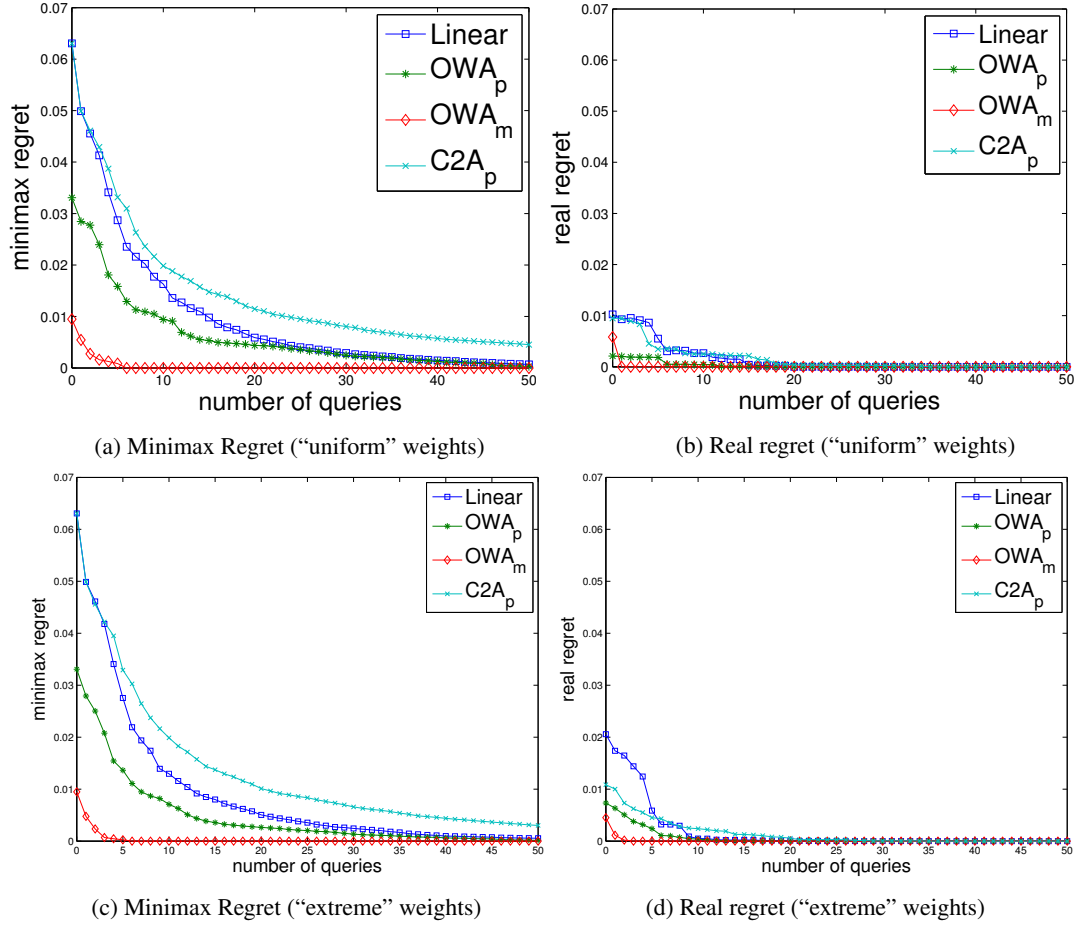


Figure 2: Elicitation with the CSS: multiobjective Knapsack problem (150 alternatives, $n = 10$).

Figure 1 shows the reduction of MMR and the real regret (computed with the DM’s sampled aggregation function) for a simulated incremental elicitation process with a random dataset composed of 150 alternatives and 10 criteria (analogously to the tests described in Section 4.2). Figures 1a, 1b refer to a DM with an “uniform” aggregation function, while Figure 1c, 1d report results for an “extreme” aggregation function. Figure 2 shows the same kind of experiments but for the Knapsack dataset with 10 criteria described in Section 4.2 and restricted to 150 alternatives.

From these two figures, we can see first how the MR of the MMR-alternative (or MMR) reduces reasonably quickly during the interaction procedure. The real regret, computed with the DM simulated aggregation function, is significantly lower than MMR; in fact, the true optimal alternatives are always (in all runs of the experiments) found in a number of iterations much lower than 50.

Then, we empirically observe that MMR_{OWA_p} and MMR_{OWA_m} are lower than MMR_{WS} and that the incremental elicitation procedure based on OWA requires fewer queries to identify the true best alternative (null real regret). When assuming monotonicity for the weights (MMR_{OWA_m}), favoring well-balanced alternatives, this is even more striking. Finally, the greater flexibility of Choquet models means that MMR_{C2A_p} is higher (than MMR_{WS} , MMR_{OWA_m} and MMR_{OWA_p}), but still, the real regret plot shows

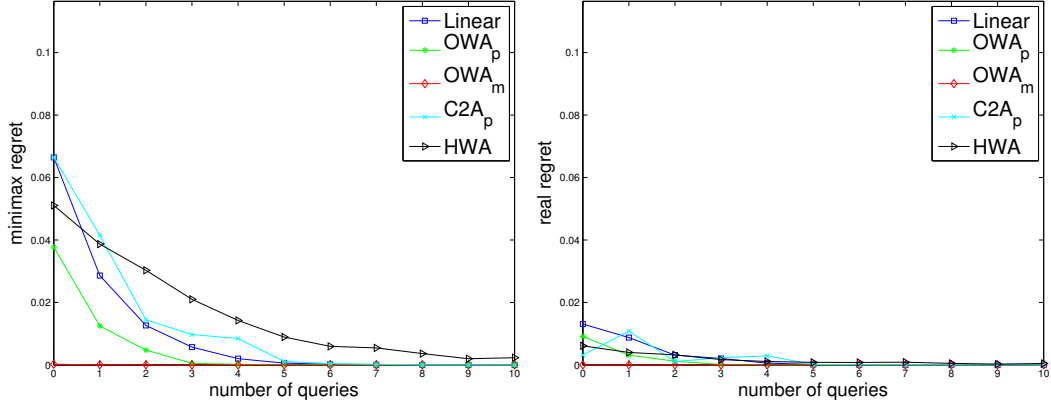


Figure 3: Elicitation with CSS; Knapsack problem (40 alternatives, $n = 2$, “uniform” weights).

that the incremental elicitation process converges to the best alternative reasonably fast.

Figure 3 includes in the comparison the more demanding model HWA in reduced instances of the bi-criteria Knapsack problem with only 40 alternatives. As we only consider two criteria, there is no need to consider the case of “extreme” weights. Moreover, in this setting, 100 particles are sufficient to well approximate the MMR value. From Figure 3, we can notice that, in all cases, the recommended item very quickly converges to the optimal choice in a couple of queries, even if HWA is associated with highest minimax regret.

Table 4: Cross-evaluation of elicitation processes - 150 alternatives (Knapsack $n = 3$).

		RS model assumption			
DM true model		OWA _m	OWA _p	Linear	C2A _p
<i>No query</i>					
	OWA _m	0.0213	0.3234	0.0804	0.0804
	OWA _p	0.2633	0.2260	0.2326	0.2326
	Linear	0.3081	0.2844	0.2857	0.2857
	C2A _p	0.1131	0.1896	0.1197	0.1197
<i>After 5 queries</i>					
	OWA _m	0.0000	0.0014	0.1666	0.1214
	OWA _p	0.1265	0.0030	0.1546	0.1280
	Linear	0.2000	0.0772	0.0277	0.0283
	C2A _p	0.0592	0.0219	0.0609	0.0427
<i>After 10 queries</i>					
	OWA _m	0.0000	0.0000	0.0949	0.0254
	OWA _p	0.1265	0.0000	0.0963	0.0767
	Linear	0.2000	0.0759	0.0012	0.0019
	C2A _p	0.0592	0.0203	0.0216	0.0062
<i>After 15 queries</i>					
	OWA _m	0.0000	0.0000	0.0944	0.0235
	OWA _p	0.1265	0.0000	0.0955	0.0723
	Linear	0.2000	0.0759	0.0000	0.0000
	C2A _p	0.0592	0.0203	0.0107	0.0026

Table 5: Cross-evaluation of elicitation processes - 150 alternatives (Knapsack $n = 6$).

	DM true model	RS model assumption			
		OWA _m	OWA _p	Linear	C2A _p
<i>No query</i>					
	OWA _m	0.0000	0.2269	0.1993	0.1993
	OWA _p	0.1676	0.1370	0.1737	0.1737
	Linear	0.2811	0.2079	0.2363	0.2363
	C2A _p	0.0459	0.1393	0.1603	0.1603
<i>After 5 queries</i>					
	OWA _m	0.0000	0.0711	0.1588	0.0727
	OWA _p	0.0592	0.0118	0.1325	0.0724
	Linear	0.1148	0.0865	0.0733	0.1119
	C2A _p	0.0186	0.0535	0.1221	0.0870
<i>After 10 queries</i>					
	OWA _m	0.0000	0.0000	0.0036	0.0727
	OWA _p	0.0592	0.0008	0.0617	0.0576
	Linear	0.1148	0.0373	0.0132	0.0266
	C2A _p	0.0186	0.0065	0.0391	0.0595
<i>After 15 queries</i>					
	OWA _m	0.0000	0.0000	0.0000	0.0000
	OWA _p	0.0592	0.0000	0.0420	0.0338
	Linear	0.1148	0.0328	0.0000	0.0063
	C2A _p	0.0186	0.0000	0.0018	0.0088

In the second experiments, we perform a kind of “cross-evaluation” of interactive elicitation with different preference model assumptions, meaning that MMR is calculated according to a model (Linear, OWA_m, OWA_p and C2A_p) that might be different from the DM underlying model that dictates his/her answers. The results of this experimentation setting are presented in Table 4 and Table 5 (the numbers correspond to the real regrets after 0, 5, 10 and 15 queries). The first table considers 150 alternatives of the Knapsack problem with 3 criteria (with “extreme” weights) and the second one considers 150 alternatives of the Knapsack problem with 6 criteria (with “extreme” weights). Results are averaged over 100 runs.

Not surprisingly, the real regret is generally lowest when the preference model assumed by the RS coincides with the “true” model of the DM (lowest numbers are around the diagonal). However, we notice a certain degree of robustness in using a more general model (C2A_p) even when the DM is making decision using a more specialized model as OWA_m, OWA_p or the weighted sum. The situation is severely bad (high real regret) when assuming OWA_m or OWA_p but the DM is making decisions using a linear model. For the kind of weights used in the experiments, we noticed a (surprising) relatively good performance in some settings of assuming OWA_p when tested against C2A_p; however in other settings, this assumption results in a very significant real regret. Further studies with plausible weights learned from real users (DMs) are needed to make stronger conclusions about the robustness of the models in practical contexts.

Finally, we consider the method that is able to elicit a general capacity by asking queries about fictitious alternatives (see paragraph “General capacities” in Section 3.4). These last experiments allow elicitation without making prior assumptions on the type of the capacity; in the simulation we consider here, different preference models (Linear, OWA, Choquet integral) are used to represent the true underlying model of DM preferences (but the RS assumes a general Choquet integral as preference model). The simulated

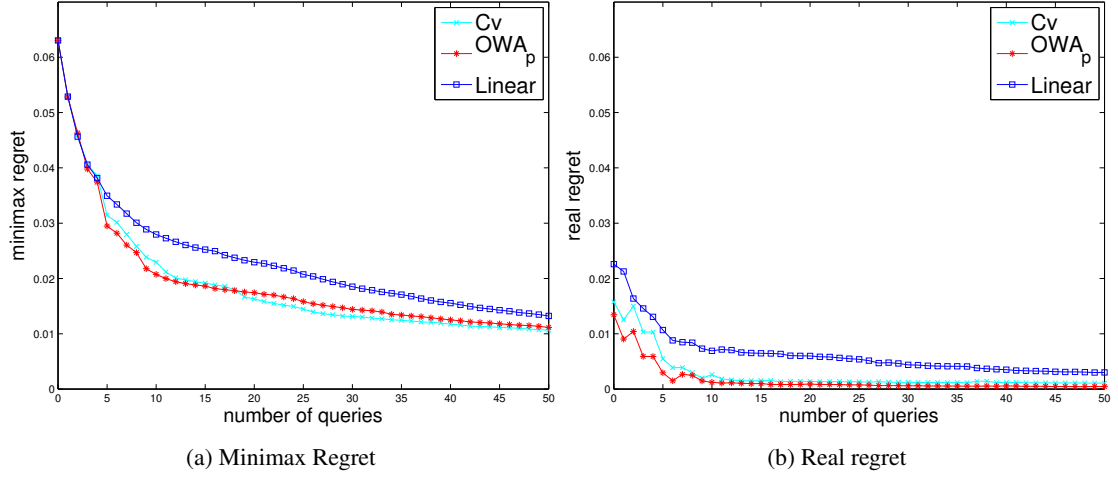


Figure 4: Elicitation with queries of type “1A0 preferred to Λ ?”: multiobjective Knapsack problem with 150 alternatives and $n = 10$ (simulated DMs answering according to a linear, OWA or general Choquet).

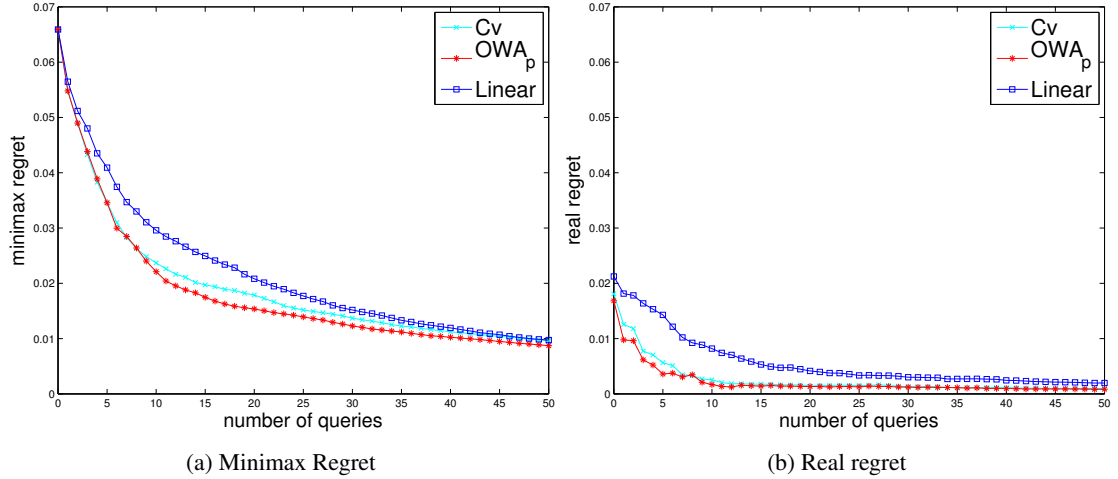


Figure 5: Elicitation with queries of type “1A0 preferred to Λ ?”: multiobjective Knapsack problem with 1000 alternatives and $n = 10$ (simulated DMs answering according to a linear, OWA or general Choquet).

DM is asked to answer queries of type $1A0 \succsim \Lambda$ chosen in order to ensure the highest myopic reduction of MMR_{Cv} in the worst case of answer. In order to randomly generate general capacities satisfying the set monotonicity constraints to model DM’s preferences, we considered a recursive method where capacity values are sampled recursively starting from the bottom of the set-inclusion lattice. The capacity of singletons are sampled from $U[0, 0.5]$; then, iteratively, for the set of criteria A of size $d > 1$, we sample from $U[\max_{i \in A} v(A \setminus \{i\}), 1]$, the uniform distribution between the maximum of capacity values of its subsets of size $d-1$, and the value 1; the capacity of the set containing all criteria is imposed to be 1. By construction, the resulting capacity is monotone and normalized. We also consider an alternative method where the capacity value of a set of size d is sampled between the maximum capacity of its subsets of size $d-1$ and the latter quantity plus 0.1. The second preference simulation has been used to

model preferences somewhat more difficult to elicit. The results on the multiobjective Knapsack problem with 10 criteria (restricted to 150 alternatives for Figure 4 and 1000 alternatives for Figure 5) have been obtained by averaging over 200 runs (100 runs for each different ways to simulate DM).

Here again, from these two figures, we can see that the MMR reduces reasonably quickly during the incremental elicitation process. Since the RS is given no information about the type of the capacity, MMR_{C_V} necessarily decreases more slowly (than when making a specific assumption about the type of the capacity); this can be seen by comparing Figure 4a with Figures 2a and 2c. Figure 4b and Figure 5b show that the recommended alternative is approximately as good as the true optimal alternative; the real regret is much smaller than MMR_{C_V} .

6 Discussion and Conclusions

A large variety of systems need to handle preferences [36]; these include recommender systems and systems for decision support [47]. Preference or utility assessment is therefore critical for these systems. Preference elicitation and preference learning are currently active areas of research, as shown by a recently edited book [15].

A key idea, adopted by a number of researchers, for acquiring decision maker’s preferences is that of eliciting in an adaptive way, meaning to focus on asking relevant queries for the given decision problem. An introduction to some fundamental ideas of learning the preferences adaptively can be found in [9]; minimax regret criterion has been proposed as a tool for utility elicitation and recommendation by [6]. The problem of choosing the “best” query selection strategy (limiting the number of queries) for regret-based elicitation is addressed in [46].

Additive utility functions suffer from limited flexibility. In order to overcome the limitations of additive models, generalized additivity has been considered [16]. Alternatively, non linear aggregation functions such as Choquet have been advocated; a thorough review of the use of Choquet integrals can be found in [18].

This article contributes to the state of the art by proposing methods for incrementally learning complex preference models from decision maker’s responses. In our work, we adopted the minimax regret approach of Boutilier et. al. [6] allowing robust recommendations under uncertainty with guarantees with respect to the worst case loss. Furthermore, minimax regret can be used as a driver for choosing the next query to ask [46] allowing efficient interactive elicitation; regret-based elicitation has been validated in a user study [10].

We considered regret-based optimization assuming different utility models: OWA, HWA and Choquet integrals. The optimization is encoded as a search tree as in [7], where at each step, a pairwise max regret maximization is performed; computation of pairwise max regret values differs for each model. We provided efficient methods for regret optimization when considering OWA and restricted Choquet models assuming positive Möbius masses, and a method for HWA using sampling. We have also presented a very efficient method to elicit a Choquet integral without making any restrictive assumption on the capacity. Note that efficiency and generality are obtained at the expense of restricting queries to the comparison of fictitious alternatives rather than using actual alternatives.

However, adapting minimax regret optimization to non linear utility models proved to be not trivial, due to the large parameter space associated with the Choquet integral, as well as the necessity of handling the induced constraints (set monotonicity of the underlying capacity measure, see [42]). This means that particular care has to be given to the optimization part as future work. For instance, in [3], the authors

overcome this issue by asking a specific type of queries to the DM and propose an algorithm in $O(n^2)$ to solve the PMR_{CV} optimization problem. Another challenge is computational efficiency when computing MMR, as the decision space might be very large (this is often the case in recommender spaces, where the possible alternatives can be of several thousands.); we proposed here an efficient search method and provided experimental results.

In order to facilitate the use of our methodologies in practical contexts, a number of research questions still needs to be addressed: how to identify the type of utility model in the first place? How to choose the type of query (comparison, bound query, etc) appropriately? Our hope is that our contribution is a step forward a wider adoption of these aggregation methods. We also stress that there are also challenges related to the conceptual design of the interface of an interactive system from the point of view of user interaction, and behavioral issues related to preference assessment [47, 33, 24].

7 Acknowledgments

This work is part of the ELICIT project supported by the French National Research Agency through the Idex Sorbonne Universités under grant ANR-11-IDEX-0004-02.

References

- [1] Silvia Angilella, Salvatore Greco, and Benedetto Matarazzo. Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral. *European Journal of Operational Research*, 201(1):277–288, 2010.
- [2] Nikolaos Argyris, Alec Morton, and José Rui Figueira. CUT: A multicriteria approach for concavifiable preferences. *Operations Research*, 62(3):633–642, 2014.
- [3] Nawal Benabbou, Patrice Perny, and Paolo Viappiani. Incremental elicitation of Choquet capacities for multicriteria decision making. In *European Conference on Artificial Intelligence*, pages 87–92, 2014.
- [4] Craig Boutilier. A POMDP Formulation of Preference Elicitation Problems. In *Proc. of AAAI-02*, pages 239–246, 2002.
- [5] Craig Boutilier, Fahiem Bacchus, and Ronen I. Brafman. UCP-Networks: A directed graphical representation of conditional utilities. In *Proc. of UAI-01*, pages 56–64, 2001.
- [6] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based Optimization and Utility Elicitation using the Minimax Decision Criterion. *Artificial Intelligence*, 170(8–9):686–713, 2006.
- [7] Darius Braziunas. *Decision-theoretic elicitation of generalized additive utilities*. PhD thesis, University of Toronto, 2011.
- [8] Darius Braziunas and Craig Boutilier. Minimax Regret-based Elicitation of Generalized Additive Utilities. In *Proc. of UAI-07*, pages 25–32, 2007.
- [9] Darius Braziunas and Craig Boutilier. Elicitation of factored utilities. *AI Magazine*, 29(4):79–92, 2008.

- [10] Darius Braziunas and Craig Boutilier. Assessing regret-based preference elicitation with the UT-PREF recommendation system. In *Proceedings 11th ACM Conference on Electronic Commerce (EC-2010)*, pages 219–228, 2010.
- [11] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making Rational Decisions Using Adaptive Utility Elicitation. In *Proc. of AAAI-2000*, pages 363–369, 2000.
- [12] Alain Chateauneuf, Rose Anne Dana, and Jean-Marc Tallon. Diversification, convex preferences and non-empty core in the Choquet expected utility model. *Economic Theory*, 19(3):509–523, 1999.
- [13] Alain Chateauneuf and Jean-Yves Jaffray. Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion. *Mathematical Social Sciences*, 17(3):263–283, 1989.
- [14] Joanna Drummond and Craig Boutilier. Elicitation and approximately stable matching with partial preferences. In *Proceedings of IJCAI*, pages 97–105, 2013.
- [15] Johannes Fürnkranz and Eyke Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2010.
- [16] Christophe Gonzales and Patrice Perny. GAI networks for utility elicitation. In *Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 224–234, 2004.
- [17] Michel Grabisch, Ivan Kojadinovic, and Patrick Meyer. A review of methods for capacity identification in Choquet integral based multi-attribute utility theory. *European Journal of Operational Research*, 186(2):766–785, 2008.
- [18] Michel Grabisch and Christophe Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1):247–286, 2010.
- [19] Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. *Aggregation Functions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, New-York, 2009.
- [20] Michel Grabisch, Hunt T. Nguyen, and Elbert A. Walker. *Fundamentals of Uncertainty Calculi, with Applications*. Encyclopedia of Mathematics and its Applications. Kluwer Academic Publishers, 1995.
- [21] Salvatore Greco, Vincent Mousseau, and Roman Slowinski. Ordinal regression revisited: Multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research*, 191(2):416–436, 2008.
- [22] Fabien Le Huédé, Michel Grabisch, Christophe Labreuche, and Pierre Savéant. Integration and propagation of a multi-criteria decision making model in constraint programming. *Journal of Heuristics*, 12(4-5):329–346, 2006.
- [23] Chelsea C. White III, Andrew P. Sage, and Shigeru Dozono. A model of multiattribute decision-making and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(2):223–229, 1984.
- [24] Pekka Korhonen, Herbert Moskowitz, and Jyrki Wallenius. Choice behavior in interactive multiple-criteria decision making. *Annals of Operations Research*, 23(1):161–179, 1990.
- [25] Kouvelis, Panos, and Gang Yu. *Robust Discrete Optimization and Its Applications*. Kluwer, 1997.
- [26] Julien Lesca and Patrice Perny. LP Solvable Models for Multiagent Fair Allocation problems. In *European Conference on Artificial Intelligence*, pages 387–392, 2010.

- [27] Bonifacio Llamazares. On generalizations of weighted means and OWA operators. In *EUSFLAT Conf.*, pages 9–14, 2011.
- [28] László Lovász. Submodular functions and convexity. In *Mathematical Programming, the State of the Art*, pages 235–257. A. Bachem and M. Grötschel and B. Korte, 1983.
- [29] Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proceedings of IJCAI*, pages 287–293, 2011.
- [30] Jean-Luc Marichal, Patrick Meyer, and Marc Roubens. Sorting multi-attribute alternatives: the TOMASO method. *Computers & Operations Research*, 32(2):861–877, 2005.
- [31] Jean-Luc Marichal and Marc Roubens. Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research*, 124(3):641–650, 2000.
- [32] Patrick Meyer and Marc Roubens. On the use of the Choquet integral with fuzzy numbers in multiple criteria decision support. *Fuzzy Sets and Systems*, 157(7):927–938, 2006.
- [33] Alec Morton and Barbara Fasolo. Behavioural decision theory for multi-criteria decision analysis: a guided tour. *Journal of the Operational Research Society*, 60(2):268–275, 2009.
- [34] Włodzimierz Ogryczak. Inequality measures and equitable approaches to location problems. *European Journal of Operational Research*, 122(2):374–391, 2000.
- [35] Włodzimierz Ogryczak, Patrice Perny, and Paul Weng. On WOWA Rank Reversal. In *International Conference on Modelling Decisions for Artificial Intelligence*, volume 7647 of *LNAI*, pages 66–77, 2012.
- [36] Bart Peintner, Paolo Viappiani, and Neil Yorke-Smith. Preferences in interactive systems: Technical challenges and case studies. *AI Magazine*, 29(4):13–24, 2008.
- [37] Ahti Salo and Raimo P. Hämmäläinen. Preference ratios in multiattribute evaluation (PRIME)–elicitation and decision procedures under incomplete information. *IEEE Trans. on Systems, Man and Cybernetics*, 31(6):533–545, 2001.
- [38] Leonard J. Savage. *The Foundations of Statistics*. Wiley, 1954.
- [39] David Schmeidler. Integral representation without additivity. *Proceedings of the American Mathematical Society*, 97(2):255–261, 1986.
- [40] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [41] Anthony F. Shorrocks. Ranking income distributions. *Economica*, 50(197):3–17, 1983.
- [42] Ali Fallah Tehrani, Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1-2):183–211, 2012.
- [43] Mikhail Timonin. Robust optimization of the Choquet integral. *Fuzzy Sets and Systems*, 213(1):27–46, 2013.
- [44] Vincenç Torra. The weighted OWA operator. *International Journal of Intelligent Systems*, 12(2):153–166, 1997.
- [45] Paolo Viappiani and Craig Boutilier. Optimal set recommendations based on regret. In *The 7th International Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP)*, 2009.

- [46] Paolo Viappiani and Craig Boutilier. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys09)*, pages 101–108, 2009.
- [47] Paolo Viappiani, Boi Faltings, and Pearl Pu. Preference-based search using example-critiquing with suggestions. *J. Artif. Intell. Res. (JAIR)*, 27(1):465–503, 2006.
- [48] Tianhan Wang and Craig Boutilier. Incremental Utility Elicitation with the Minimax Regret Decision Criterion. In *Proc. of IJCAI-03*, pages 309–316, 2003.
- [49] John A. Weymark. Generalized Gini inequality indices. *Mathematical Social Sciences*, 1(4):409–430, 1981.
- [50] Ronald R. Yager. On Ordered Weighted Averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems, Man and Cybern.*, 18(1):183–190, 1998.