Incremental Elicitation of Choquet Capacities for Multicriteria Decision Making

Nawal Benabbou and Patrice Perny and Paolo Viapiani

Abstract. The Choquet integral is one of the most sophisticated and expressive preference models used in decision theory for multicriteria decision making. It performs a weighted aggregation of criterion values using a capacity function assigning a weight to any coalition of criteria, thus enabling positive and/or negative interactions among criteria and covering an important range of possible decision behaviors. However, the specification of the capacity involves many parameters which raises challenging questions, both in terms of elicitation burden and guarantee on the quality of the final recommendation. In this paper, we investigate the incremental elicitation of the capacity through a sequence of preference queries selected one-by-one using a minimax regret strategy so as to progressively reduce the set of possible capacities until a decision can be made. We propose a new approach designed to efficiently compute minimax regret for the Choquet model. Numerical experiments are provided to demonstrate the practical efficiency of our approach.

1 INTRODUCTION

In the field of Multicriteria Decision Making, aggregation functions are often used to compare alternatives evaluated on multiple conflicting criteria by synthesizing their performances into overall utility values. Such functions must be sufficiently expressive to fit to Decision Maker’s (DM) preferences, allowing for instance the determination of his/her preferred alternative. Choquet integrals form a family of non-linear aggregators that are really appealing for preference modeling because they enable to model different kind of interactions between criteria and include many aggregators as special cases (e.g. linear additive models, min, max and any other order statistics, leximin and leximax, OWA and WOWA [24, 29] and Yaari’s model [28]). Choquet integral has received much attention in the last decades and is now widely used in practical decision making [9].

However, to compute overall utility values using a Choquet integral, decision support systems need to be able to assess model’s parameters according to DM’s preferences. These parameters used to capture the value system of the DM are characterized by a capacity function defining the weight attached to every subset of criteria. Therefore, they are in exponential number relatively to the number of criteria and their elicitation is a challenging issue. Most of previous works on capacity elicitation for Choquet integrals consider a static preference database as input, and focus on the determination of a set of capacity values that best fits the available preferences. For example, one can minimize a quadratic error between Choquet values and of capacity values that best fits the available preferences. For exam-ple, one can minimize a quadratic error between Choquet values and target utility values prescribed by the DM on a sample of reference alternatives. 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. [11, 18, 19, 22, 13] and [10] (Chapter 11), some of them being implemented in decision support softwares such as TOMASO [15] and MYRIAD [12].

Departing from these standard approaches, we are proposing in this paper an incremental elicitation process for Choquet integrals, in which preference queries are selected one at a time, to be as informative as possible, so as to progressively reduce the set of admissible capacities until a robust recommendation can be made. This approach relies on and extends previous works on the incremental elicitation of linear utility functions, going back to the ISMAUT method [14] and more recently, strategies developed within the artificial intelligence community for preference query selection using the minimax-regret criterion [6, 2, 27, 3]. Regret-based elicitation has been successfully demonstrated with real users in a prototype for decision support (UT- pref) and validated in a user study [5]. Adaptation of minimax regret elicitation strategies to Choquet models is not obvious, as the number of constraints required to impose that the parameters of the model are valid is exponential in the number of criteria, in the general case. In this paper we propose an efficient algorithm that avoids this issue by focusing on specific (but intuitive) types of preference statements.

2 BACKGROUND AND NOTATION

Let \( X \) be the set of alternatives (items, products, candidates...) that need to be compared in order to make a decision. Any alternative \( x \in X \) is evaluated with respect to a set of \( n \) criteria denoted \( N = \{1, \ldots, n\} \), and is characterized by a performance vector \((x_1, \ldots, x_n)\) where \( x_i \in [0, 1] \) represents the utility of \( x \) with respect to the criterion \( i \) for all \( i \in N \). For simplicity, \( x \) will indifferently denote the alternative or its performance vector.

2.1 Choquet integrals

For any alternative \( x \in X \), let \((\cdot)\) denote the permutation of \( \{1, \ldots, n\} \) which sorts the components of \( x \) by increasing order, i.e. \( x_{(i)} \leq x_{(i+1)} \) for \( i \in [1, n-1] \). Let \( X_{(i)} \) denote the subset of criteria with respect to which \( x \) has a utility greater or equal to \( x_{(i)} \), i.e. \( X_{(i)} = \{i, \ldots, n\} \); note that \( X_{(i+1)} \subseteq X_{(i)} \) for all \( i \in [1, n-1] \) by definition. In the sequel, \( X_{(i)} \) will be referred to as the \( i^{th} \) level set of \( x \) and \( Y_{(i)} \) will denote the \( i^{th} \) level set of an alternative \( y \in X \).

Let \( v \) be a Choquet capacity, i.e. a real-valued set function defined on \( 2^N \) such that \( v(\emptyset) = 0 \), \( v(N) = 1 \) and \( v(A) \leq v(B) \) for all \( A \subseteq B \subseteq N \). \( v(A) \) representing the weight attached to coalition \( A \), for any \( A \subseteq N \). The Choquet integral is defined by:

\[ I(x) = \int_{\emptyset}^{N} v(A) x_{(A)} \, dA \]
\[ C_v(x) = \sum_{i=1}^{n} [x_{(i)} - x_{(i-1)}] v(X_{(i)}) \quad \text{with} \quad x_{(0)} = 0 \]

Hence an alternative \( x \) is as least as good as \( y \) whenever \( C_v(x) \geq C_v(y) \). For example, consider a problem defined on 3 criteria, i.e. \( N = \{1, 2, 3\} \), and two performance vectors \( x = (0.7, 0.6, 1) \) and \( y = (0.8, 1, 0.6) \). The computation of their Choquet value with the following capacity gives:

\[
\begin{array}{ccccccc}
0 & 0.1 & 0.2 & 0.3 & 0.5 & 0.6 & 0.7 & 1 \\
C_v(x) = 0.6 + (0.7 - 0.6)v(\{1\}) & 1 + 0.7 - v(\{3\}) & 0.75 \\
C_v(y) = 0.6 + (0.8 - 0.6)v(\{1, 2\}) & 1 + 0.8 - v(\{2\}) & 0.74 \\
\end{array}
\]

Hence we have \( C_v(x) \geq C_v(y) \), meaning that \( x \) is strictly preferred to \( y \). In multicriteria decision making, one needs to ensure that \( C_v(x) \geq C_v(y) \) whenever \( x \) weakly Pareto-dominates \( y \) (i.e. \( x_i \geq y_i \) for all \( i \in N \)). This property holds due to the monotonicity of \( v \) with respect to set inclusion.

In many papers on multicriteria optimization with a Choquet integral, the capacity is assumed to be given [8, 23, 7, 17]. This assumes that preference elicitation methods are available to determine the capacity that best fits DM’s maker preferences. Following the line opened by Boullier in [27, 3] for simpler decision criteria, we suggest here adopting an incremental approach where capacity elicitation is seen as a game played with the DM. At every step of the elicitation process, the system generates a preference query, and then the DM reveals a piece of his/her actual preferences. The answer provides new constraints on the set of admissible capacities thus reducing the uncertainty attached to the capacity and therefore to the Choquet values. In this process, both the problem of selecting the next preference query and the one of generating a recommendation are seen as a decision problem under uncertainty, where the uncertainty is due to the imperfect knowledge of preference parameters (here the capacity). Our strategy to select the most promising alternative is based on a minimax criterion aiming at providing the best Choquet value against all admissible choices for the capacity. The selection of the query is made so that the answer will guarantee an effective regret reduction whatever the answer.

### 2.2 Minimax regret criterion

Minimax regret [21, 16] is a decision criterion classically used for optimization under uncertainty over data; it has been more recently advocated for use in decision-making where the uncertainty is over utility values [3, 20]. Let \( P \) be a set of preference statements that match with the preferences of the DM. \( P \) can include different types of information, from prior knowledge to information obtained by asking queries to the DM. Assume that DM’s preferences can be modeled by an element in the family of aggregators \( F_\Theta \), where \( \Theta \) is the set of all admissible parameters of the family (e.g. whenever \( F_\Theta \) represents the family of Choquet integrals, then \( \Theta \) is the set of all possible capacities). Let \( \Theta_\beta \) denote all parameters in \( \Theta \) compatible with \( P \).

**Definition 1.** Given a set of preferences \( P \) and assuming a model \( F_\Theta \), the pairwise max regret of the alternative \( x \) with respect to the alternative \( y \) is defined as follows:

\[
PMR(x, y; \Theta_\beta) = \max_{\theta \in \Theta_\beta} f_\theta(y) - f_\theta(x)
\]

where \( f_\theta \in F_\Theta \) is the aggregator corresponding to parameter \( \theta \).

In other words, the pairwise max regret of \( x \) with respect to \( y \) represents the worst-case loss when recommending \( x \) instead of \( y \).

**Definition 2.** Given a set of alternatives \( X \), a set of preferences \( P \) and assuming a model \( F_\Theta \), the worst-case min regret of a query \( q \in Q \) is defined as:

\[
MR(x, X; \Theta_\beta) = \max_{y \in X} \min_{\theta \in \Theta_\beta} PMR(x, y; \Theta_\beta)
\]

In other words, the max regret of \( x \) is the worst-case loss when recommending \( x \) instead of one of the adversary’s choices (i.e. \( \arg \max_{y \in X} \min_{\theta \in \Theta_\beta} PMR(x, y; \Theta_\beta) \)).

**Definition 3.** Given a set of alternatives \( X \), a set of preferences \( P \) and assuming a model \( F_\Theta \), the minmax regret is defined as:

\[
MMR(X; \Theta_\beta) = \min_{x \in X} MR(x, X; \Theta_\beta)
\]

An optimal solution for the minmax regret criterion is an alternative that achieves the minmax regret (i.e. \( \arg \min_{x \in X} MR(x; \Theta_\beta) \)). Recommending the latter alternative allows one to guarantee that the worst-case loss is minimized. In the rest of this article, \( x^* \) will denote one optimal solution for the latter criterion and \( y^* \) one of its adversary’s choice, arbitrary chosen in \( \arg \max_{y \in X} PMR(x^*, y; \Theta_\beta) \).

### 2.3 Incremental elicitation

Given a particular set of preference statements, the worst-case loss ensured by the minmax regret criterion might still be at unacceptable level. By considering additional preferences statements (inducing constraints on the set of admissible parameters), this loss may be decreased. Indeed, we know that \( \Theta_\beta \subseteq \Theta_\alpha \) for any set of preference statements \( \beta \supseteq \alpha \); then, \( PMR(x, y; \Theta_\alpha) \leq PMR(x, y; \Theta_\beta) \) for any \( x, y \in X \), and so \( MR(x, X; \Theta_\alpha) \leq MR(x, X; \Theta_\beta) \) for any \( x \in X \). Finally, \( MMR(X; \Theta_\alpha) \leq MMR(X; \Theta_\beta) \) and so the minmax regret cannot increase by adding preference statements (usually it decreases (see [4], pp. 194-202)). Therefore, the minmax regret criterion can be used within an incremental elicitation process that progressively asks preference queries to the DM until the minmax regret drops under a given threshold. At that time, recommending \( x^* \) ensures that the loss incurred by not choosing the true optimal alternative is bounded above by that threshold.

Different types of queries can be used when designing such incremental elicitation process. Comparisons queries are relatively simple, they require the DM to compare a pair of alternatives and state which one is preferred. Notice however that some queries are more informative than others (e.g. minimax regret won’t decrease when asking to compare an alternative with another that Pareto-dominates the former). Thus, it is important to make a good recommendation without asking too many queries, focusing on relevant queries. A notion of myopic value of information can be used [25] to evaluate the relevance of a query. Let \( Q \) denote the set of all considered queries.

**Definition 4.** Given a set of alternatives \( X \), a set of preferences \( P \) and assuming a model \( F_\Theta \), the worst-case minmax regret of a query \( q \in Q \) is defined as follows:

\[
WMMR(q, X; \Theta_\beta) = \max_{q \in Q} MMR(X; \Theta_\beta, \{q\})
\]

where \( T_q \) denotes the set of all possible answers to the query \( q \).

Hence the next query of the elicitation process should be chosen in \( \arg \min_{q \in Q} WMMR(q, X; \Theta_\beta) \) because any optimal solution for the WMMR criterion ensures the best reduction of minmax regret in the answer’s worst-case scenario. Note that computing the optimal query for WMMR can be computationally intensive when set \( Q \) under consideration is too large. We discuss now computational issues related to minmax regret optimization for Choquet integrals (Section 3). We will present our strategy for generating queries within an incremental elicitation process in Section 4.
3 MINIMAX REGRET OPTIMIZATION FOR CHOQUET INTEGRALS

In the procedure defined in subsection 2.2, we have to compute PMR for all ordered pairs of distinct items to determine the current optimal alternative \(x^*\); notice that \(\text{PMR}(x,x) = 0\) and that, in general, \(\text{PMR}(x, y) \neq \text{PMR}(y, x)\). Thus, the maximum regret MR can be computed for each alternative so as to determine the item having the lowest MR. However, the computational effort can be significantly reduced using standard pruning rules for min aggregators, as shown in [4]. Empirically, if we use such pruning rules, the number of PMR computations is only slightly higher than linear, but of course remains quadratic in the worst-case.

We therefore focus our discussion on the computation of PMR, assuming \(F_{\Theta}\) is the set of Choquet Integrals. In that case, \(\Theta_{\mathcal{P}}\) is the set of all capacities compatible with \(\mathcal{P}\).

3.1 A General Optimization of Pairwise Max Regret using Linear Programming

Let \(v : 2^\mathcal{N} \rightarrow \mathbb{R}\) be a set function and \(u_\Lambda\) the decision variable representing \(v(A)\) for any \(A \subseteq \mathcal{N}\). Using this notation, \(v\) will indifferently denote the set-function and the vector composed of its values. Thus, for any alternatives \(x, y \in \mathcal{X}\), \(\text{PMR}(x, y; \Theta_{\mathcal{P}})\) can be computed by solving the following linear program:

\[
\begin{align*}
\max_v & \quad C_v(y) - C_v(x) \\
\text{subject to} & \quad v_B = 0 \\
& \quad v_Y = 1 \\
& \quad v_X \leq v_{\Lambda(i)} \quad \forall i \in \mathcal{N}, \forall i \in \mathcal{N} \setminus A \\
& \quad C_v(A) \geq C_v(B) \quad \forall a, b \text{ s.t. } a \subseteq b \subseteq \mathcal{P}
\end{align*}
\]

Equations (2-4) ensure that \(v\) is indeed a capacity and Equation (5) ensures that \(v\) is compatible with \(\mathcal{P}\). Thus, for Choquet integrals, the computation of PMR involves exponentially many variables and monotonicity constraints (4). For some specific subclasses of capacities (e.g. 2-additive capacities [22]), it has been shown that the number of such constraints that are actually needed is much lower. However, these subclasses correspond to specific attitudes that do not necessarily match with the observed preferences. Hence, we investigate now the general case without any prior restriction on the admissible set of capacities.

3.2 A linear programming formulation for \(1A_0 \succsim_\Lambda \) preference statements

For any two performance vectors \(x, y\), let \(A_{(x,y)}\) be the set of all level sets of \(x, y\), i.e. \(\{X_i \mid i \in \mathcal{N}\} \cup \{Y_i \mid i \in \mathcal{N}\}\). Note that sets belonging to \(A_{(x,y)}\) are the only ones that appear in the objective function (1). This specificity can be exploited to simplify the regret optimization problem. Let us indeed consider now queries involving binary alternatives of type 1A0, where 1A0 represents a fictitious alternative with a top performance on all criteria in \(\mathcal{A} \subseteq \mathcal{N}\) and a bottom performance on all others. More precisely, the DM may be asked to compare such alternatives to constant utility profiles of type \(\Lambda = (\lambda, \ldots, \lambda)\). Note that by definition, \(C_v(1A_0) = v(A)\) and \(C_v(\Lambda) = \lambda\) for any capacity \(v\) and any set \(A \subseteq \mathcal{N}\). As a consequence, if the preference \(1A_0 \succsim_\Lambda \) is observed, then Equation (5) gives the simple constraint \(v(A) \geq \lambda\) (resp. \(v(A) \leq \lambda\)). Consequently, Equation (5) can be replaced by boundary constraints over decision variables; indeed, to ensure that the set-function \(v\) is compatible with \(\mathcal{P}\), it is sufficient to update the boundaries of an interval \([a, u_\Lambda]\) whenever a preference of type \(1A_0 \succsim_\Lambda \) or \(1A_0 \preceq_\Lambda \) is inserted in \(\mathcal{P}\).

Since \(\Theta_{\mathcal{P},1(1A_0 \succsim_\Lambda)}\) is the set of all capacities \(v \in \Theta_{\mathcal{P}}\) that satisfy \(v(A) \geq \lambda\), and keeping in mind that all capacities are monotonic by definition, then necessarily \(v(B) \geq \lambda\) for all these capacities and all \(B \supseteq A\) (i.e. \(\Theta_{\mathcal{P},1(1A_0 \preceq_\Lambda)} = \Theta_{\mathcal{P},1(1B_0 \preceq_\Lambda)} = \Theta_{\mathcal{P},1(1B_0 \preceq_\Lambda \text{ and } B \supseteq A)}\)). Thus, if the preference \(1A_0 \preceq_\Lambda \) is observed, then all preferences of type \(1B_0 \preceq_\Lambda \) for \(B \supseteq A\) can be inserted in \(\mathcal{P}\). Similarly, if the preference \(1A_0 \preceq_\Lambda \) is observed, then all preferences of type \(1B_0 \preceq_\Lambda \text{ and } B \supseteq A\) can be inserted in \(\mathcal{P}\).

Let \(P\) be a set of preferences statements obtained by inserting preferences of type \(1A_0 \succsim_\Lambda\) (resp. \(1A_0 \preceq_\Lambda\)) and, for each of them, the preference \(1B_0 \preceq_\Lambda\) for all \(B \supseteq A\) (resp. \(1B_0 \preceq_\Lambda\) for all \(B \supseteq A\)). Let \(\tilde{I}_{A, u_\Lambda}\) be the resulting lower and upper bounds obtained for all \(A \in 2^\mathcal{N}\), then the following proposition holds:

**Proposition 1.** Any function \(v : \mathcal{A} \rightarrow [0, 1] \cup \mathcal{A} \subseteq 2^\mathcal{N}\) such that \(v(A) \in [1, u_\Lambda]\) for all \(A \in \mathcal{A}\), and \(v(B) \leq v(B)\) for all \(B, A \in \mathcal{A}\) such that \(A \subseteq B\) can be completed into a capacity in \(\Theta_{\mathcal{P}}\).

**Proof.** By construction of \(\mathcal{P}\) and all its corresponding intervals, it is sufficient to complete \(v\) by setting first \(v(A) = 1\) for all \(A \in 2^\mathcal{N}\) such that \(|A| = 1\). Then, we iteratively set the value of \(v(A)\) to \(\max\{v(A), \max_{B \subseteq A} (|B| - 1) v(B)\}\) for all \(A \in 2^\mathcal{N}\), such that \(v(B)\) is known for all \(B \subseteq A\) such that \(|B| = |A| - 1\), so as to obtain a completely specified capacity.

Thus, Proposition 1 enables one to conclude that all constraints given by Equation (4) involving \(u_{\Lambda}(A)\) can be removed and so PMR \((x, y; \Theta_{\mathcal{P}})\) can be computed by solving the following simpler linear program:

\[
\begin{align*}
\max_v & \quad C_v(y) - C_v(x) \\
\text{subject to} & \quad v_{X_{i+1}} \leq v_{X_i} \quad \forall i \in [1; n - 1] \\
& \quad v_{Y_{i+1}} \leq v_{Y_i} \quad \forall i \in [1; n - 1] \\
& \quad v_{Y_i} \leq v_{X_i} \quad \forall i \in N \text{ s.t. } X_i \subseteq Y_i \\
& \quad v_{Y_i} \leq v_{Y_i} \quad \forall i \in N \text{ s.t. } Y_i \subseteq X_i \\
& \quad l_{Y_{i+1}} \leq l_{Y_i} \leq u_{Y_i} \quad \forall i \in N \\
& \quad l_{Y_i} \leq l_{Y_i} \leq u_{Y_i} \quad \forall i \in N
\end{align*}
\]

Let \(w_A\) denote the coefficient of the decision variable \(u_A\) in the objective function (6), for any set of criteria \(A \in \mathcal{A}_{(x,y)}\). Note that \(w_A = -(x_{i+1} - x_{i-1}) \leq 0\) for all \(A \in \{X_i \mid X_i \neq Y_i\}\), \(w_A = y_i - y_{i-1} \geq 0\) for all \(A \in \{Y_i \mid Y_i \neq X_i\}\) and \(w_A = y_i - y_{i-1} - (x_{i+1} - x_{i-1})\) for all \(A \in \{X_i \mid X_i = Y_i\}\). Since the objective function has to be maximized, we can deduct that \(v_A\) will be as small as possible for all \(A \in \{X_i \mid X_i \neq Y_i\}\) and as large as possible for all \(A \in \{Y_i \mid Y_i \neq X_i\}\). Thus, none of the constraints (9) are required to find the optimum. Note also that some constraints given by Equation (10) are unnecessary. Indeed, if there exists \(i, j \in N\) such that \(Y_j \subseteq X_i\), then we also have \(Y_j \subseteq X_k\) for all \(k \in [1; j]\), which creates redundant constraints added to Equation (7); thus, it is sufficient to impose \(v_{Y_j} \leq v_{Y_k}\) only if \(Y_j \subseteq X_k\) and \(Y_k \subseteq X_{j+1}\). However, if \(Y_{j+1} \subseteq X_j\) is also satisfied, there is a redundancy with Equation (8). Finally, it is sufficient to impose \(v_{Y_j} \leq v_{X_j}\) only if \(Y_j \subseteq X_j\) and \(Y_{j+1} \subseteq X_{j+1}\). Thus, the number of nonmonotonicity constraints
is now below 3(n − 1) and at most 2(n − 1) variables are used (the elements of \(A(x, y)\)).

### 3.3 Efficient Optimization for PMR

Although the numbers of constraints and variables of the linear program defined in (6-12) are polynomial in the number of criteria, the computation time required by a state-of-the-art solver to obtain minimax regret increases significantly with the number of alternatives due to the quadratic number of PMR optimizations. We show now how the PMR-optimization problem can be solved efficiently by an iterative procedure, for any \(x, y \in X\) and any set of preference statement \(P\) of type \(1A0 \succeq A\) or \(1A0 \preceq A\), obtained by doing all the insertions described in the previous subsection.

Let \(\mathcal{G} = (V, A)\) denote the constraint graph associated to Equations (7-10), \(\mathcal{G}\) is defined as follows:

- \(V\) is the set of all decision variables in the linear program; thus, \(V = \{v_{x(i)}| i \in N\} \cup \{v_{y(i)}| i \in N\}\).
- \(A\) is the set of arrows \(\{v_{A}, v_{B}\}\) such that \(v_{A} \geq v_{B}\) is given by Equations (7-8) or Equation (10) without redundancy.

Note that the \(2(n-1)\) constraints given by Equations (7 - 8) imply the existence of the two paths \(V_x = (v_{X(i)}, \ldots, v_{X(n)})\) and \(V_y = (v_{Y(i)}, \ldots, v_{Y(n)})\), which together include all the nodes in \(V\). Recall that \(w_A\) denotes the coefficient of \(v_A\) in the objective function (6).

Let \(V^{-}\) (resp. \(V^{+}\)) be the restricted sequence of \(V_x\) (resp. \(V_y\)) to all nodes \(v_A\) such that \(w_A \leq 0\) (resp. \(w_A \geq 0\) and \(v_A \not\in V^{-}\)). As already noted before, \(w_A \leq 0\) for all \(A \in \{X(i)| X(i) \neq Y(i)\}\) and \(w_A \geq 0\) for all \(A \in \{Y(i)| X(i) \neq Y(i)\}\). Thus, \(V^{-}\) and \(V^{+}\) include together all the nodes in the \(V\) and have no common node (see Figure 1 for illustration, where \(x = (1, 0.8, 0.4, 0.5, 0.1)\) and \(y = (0.8, 0.9, 0.6, 0.2, 0.4)\)).

**Algorithm 1: Iterative optimization of PMR**

**Input:** Two alternatives \(x, y \in X\)

**Output:** \(v\) defined on \(A(x, y)\) achieving PMR \((x, y; \Theta_P)\)

1. Construction of \(V^{-}\) and \(V^{+}\) from \(x\) and \(y\)
2. foreach \(v_A \in V^{+}\) do \(\varepsilon \leftarrow u_A\)
3. for \(i = 1 \ldots |V^{+}|\) do
   4. \(w \leftarrow [v_{A_i}]^{-}\)
   5. \(w \leftarrow 0\)
   6. \(j \leftarrow 1\)
   7. while \(|w| \geq w^{-}\) and \(D_j^{-}(v_{A_i})\) exists do
      8. \(v_B \leftarrow D_j^{-}(v_{A_i})\)
      9. if \(v_B < l_{A_i}\) then break
      10. \(w \leftarrow w^{-} + w_B\)
      11. \(w \leftarrow w^{-} + \sum v_{A_i} \in E^{-}(v_{A_i}) : D_j^{-}(v_{A_i}) = v_B\)
      12. \(j \leftarrow j + 1\)
   13. end
   14. if \(|w| < w^{-}\) and \(v_{A_i} < v(B)\) then \(v_{A_i} \leftarrow v(B)\) else \(v_{A_i} \leftarrow l_{A_i}\)
   15. foreach \(v_A \in D^{+}(v_{A_i})\) do \(v(A) \leftarrow \min \{v(A), v(A_i)\}\)
16. end
17. return \(v\)

### 4 AN INCREMENTAL ELICITATION METHOD FOR CHOQUET INTEGRALS

We introduce now a query strategy assuming the DM is only asked to compare binary alternatives to constant profiles. Our query selection strategy uses the WMMR criterion presented in Definition 4; thus, since the DM is only asked to compare a binary alternative \(1A0\) to a constant profile \(A = (\lambda, \ldots, \lambda)\), an optimal query is defined by a pair \((A \subseteq N, \lambda \in \{l_A, u_A\})\) that brings the smallest minimax regret in the answer's worst-case scenario. In order to find such a pair, we have to determine, for all sets \(A \subseteq N\), the value \(\lambda_A^* \in \{l_A, u_A\}\) that minimizes the WMMR criterion; thus, an optimal query is defined by a pair \((A \subseteq N, \lambda_A^*)\) that minimizes the latter criterion.

Given a set \(A \subseteq N\), determining \(\lambda_A^*\) amounts to minimizing over \(\lambda \in \{l_A, u_A\}\) the maximum between \(MRR(A; \Theta_{P_u(1A0 \preceq \lambda)})\) and \(MRR(A; \Theta_{P_u(1A0 \succeq \lambda)})\). Note that \(MRR(A; \Theta_{P_u(1A0 \succeq \lambda)})\) and
MMR(λ; Ψ, Θ) are two functions of λ and that the former is a decreasing one while the latter is an increasing one. Similarly to what is observed for utility functions over consequences [27], these two functions necessarily intersect since they have the same maximum (i.e. MMR(λ; Ψ)). This intersection gives the value of λ∗ and can easily be computed by a bisection algorithm relying on the relative positions of the two curves, observed at two distinct points. However, it may happen that the WMMR value of the optimal query is equal to MMR(λ; Ψ), which means that the latter question will not necessarily induce a regret reduction. In such cases, our procedure chooses a set A ⊆ N that minimizes, for λ = (λA + uA)/2, the expected value of minimax regret over the two possible answers with an uniform distribution hypothesis over [λA, uA].

Note that determination of the next query implies to select A within the 2n − 2 possible proper subsets of N, a number which increases significantly with the number of criteria. To make this query selection step more efficient, we propose, as a heuristic, to focus on sets directly involved in the computation of PMR(x∗, y∗; Ψ). where x∗ is an optimal solution for the MMR criterion knowing P, and y∗ is one of the adversary’s choices. These sets are those in A(x∗, y∗) = {X∗i | i ∈ N} ∪ {Y∗i | i ∈ N}, where X∗(i) and Y∗(i), respectively denote the ith level set of x∗ and y∗. Thus, the heuristic will further constrain parameters involved in the computation of the pairwise max regret of x∗ with respect to y∗ and possibly reduce the minimax regret. According to this heuristic, at most 2n−2 sets are investigated (elements of A(x∗, y∗)) instead of exactly 2n−2.

5 EXPERIMENTS

In this section, we report a number of experimental tests. The first ones aim at comparing the computation time of minimax regret calculation when using either a solver (LP) to optimize (6-12) or the iterative optimization (IO) algorithm presented in Section 3.3. To do so, we consider two datasets (“Knapsack5” and “Knapsack10”) consisting of the Pareto set of two multi-objective knapsack problems (n = 5 and n = 10), restricted to one thousand of alternatives. We can see that IO is significantly faster than LP (about five orders of magnitude) that IO is significantly faster than LP (about five orders of magnitude).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Query</th>
<th>LP</th>
<th>IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knapsack5</td>
<td>0</td>
<td>34.594</td>
<td>0.005</td>
</tr>
<tr>
<td>Knapsack5</td>
<td>10</td>
<td>27.583</td>
<td>0.005</td>
</tr>
<tr>
<td>Knapsack5</td>
<td>20</td>
<td>28.895</td>
<td>0.005</td>
</tr>
<tr>
<td>Knapsack10</td>
<td>0</td>
<td>31.120</td>
<td>0.007</td>
</tr>
<tr>
<td>Knapsack10</td>
<td>10</td>
<td>15.197</td>
<td>0.007</td>
</tr>
<tr>
<td>Knapsack10</td>
<td>20</td>
<td>13.016</td>
<td>0.006</td>
</tr>
</tbody>
</table>

The second experiments aim at evaluating the efficiency of our query strategy. Starting from an empty set of preferences P, simulated users answer to queries according to a Choquet Integral drawn at random. We implement the elicitation procedure introduced in Section 4 and compute both the minimax regret and real regret (obtained thanks to the user simulated utility model) at each iteration step. Results have been obtained by averaging over 200 runs and are given in Figure 2. We can see that the minimax regret reduces reasonably quickly as the number of preference queries increases and that the real regret is much smaller (a fact that has already been observed a number of times in regret-based elicitation [27, 25, 3]).

Figure 2. Incremental elicitation procedure (Knapsack10).

Now, we want to compare our preference elicitation procedure based on the Choquet model to the standard elicitation method based on a linear aggregation function and the Current Solution Strategy (CSS) as presented in [3] (based on the comparison of x∗ and y∗ at each step, see subsection 2.2). To do so, datasets of 100 alternatives evaluated on 10 criteria and characterized by a set of performance vectors ∆(a)n are randomly generated. They are constructed in such a way that ∑n i=1 x∗ i = 1 for all x ∈ ∆(a)n, where a ∈ {0.5, 1, 2} so as to obtain different types of Pareto sets (controlling the proportion and the location of non-supported1 Pareto-optimal solutions). We only report results with simulated users answering to queries according to a concave multattribute utility function so as to model a preference in favour of “well-balanced solutions”. In the general case (including linear utility model), after at most 20 queries, both procedures recommend an alternative with a real regret under ten percent of the maximum real regret in the dataset, on average. Experimental results are obtained by averaging over 100 runs.

Figure 3. Comparison of incremental elicitation procedures.

1 Linear optimizations are done using the Gurobi library of Java and results have been obtained by averaging over 50 runs.

2 solutions that do not belong to the boundary of the convex hull of ∆(a)n.
In Figure 3, we can see that the real regret, at any step of the elicitation process, is smaller with our procedure based on the Choquet model than with the procedure based on the linear model for any dataset $X^n$. Indeed, in most cases, there exists well-balanced Pareto-optimal alternatives (presumably very attractive for the decision maker) that cannot be obtained by optimizing a weighted sum of criterion values and consequently, reducing the space of possible weights cannot lead to the recommendation of such alternatives. On the contrary, various of these can be attained by maximizing a concave Choquet integral.

6 DISCUSSION AND CONCLUSION

In this paper we discussed the problem of interactively eliciting a Choquet capacity using a minimax regret approach. Technical difficulties are related to the number of parameters needed to characterize a given capacity and the number of constraints required to characterize the space of admissible capacity functions. We showed how, assuming that preferences are stated in a particular form, minimax regret optimization can be performed efficiently (in particular we presented both a linear programming formulation and an even faster algorithm maintaining lower and upper bounds). We presented experimental results validating both the computational efficiency of our approach in large problem instances and the quality of recommendations elaborated through our incremental process.

Our work differentiates from previous work on Choquet integrals in the focus on incremental elicitation; minimax regret provides robust recommendations. Notably, Ah-Pine et al. [1] assess a feasible capacity for a Choquet integral given some preferential information that maximize the margin of the induced constraints (in a fashion similar to SVM classifiers). This kind of “pointwise” estimation ignore however the specificity of the available items. Moreover, it does not directly provide a natural strategy for choosing the query to ask in an incremental elicitation setting.

A first direct continuation of this work is to extend the elicitation procedure for set recommendation. The approach we have proposed in this paper extends naturally to sets but is computationally more demanding. Possible future works include further experimental work and validation with real users. An interesting direction of research is demanding. Possible future works include further experimental work in this paper extends naturally to sets but is computationally more difficult recommendations. Notably, Ah-Pine et al. [1] assess a feasible capacity for a Choquet integral given some preferential information that maximize the margin of the induced constraints (in a fashion similar to SVM classifiers). This kind of “pointwise” estimation ignore however the specificity of the available items. Moreover, it does not directly provide a natural strategy for choosing the query to ask in an incremental elicitation setting.

ACKNOWLEDGEMENTS

Work supported by the French National Research Agency through the Idex Sorbonne Universités under grant ANR-11-IDEX-02.

REFERENCES
