



HAL
open science

Incremental Preference Elicitation in Multi-Attribute Domains for Choice and Ranking with the Borda Count

Nawal Benabbou, Serena Di Sabatino Di Diodoro, Patrice Perny, Paolo Viappiani

► **To cite this version:**

Nawal Benabbou, Serena Di Sabatino Di Diodoro, Patrice Perny, Paolo Viappiani. Incremental Preference Elicitation in Multi-Attribute Domains for Choice and Ranking with the Borda Count. The tenth International Conference on Scalable Uncertainty Management, Sep 2016, Nice, France. pp.81-95, 10.1007/978-3-319-45856-4_6 . hal-01342500

HAL Id: hal-01342500

<https://hal.sorbonne-universite.fr/hal-01342500>

Submitted on 6 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.

Copyright

Incremental Preference Elicitation in Multi-Attribute Domains for Choice and Ranking with the Borda Count

Nawal Benabbou¹, Serena Di Sabatino Di Diodoro^{1,2},
Patrice Perny¹, and Paolo Viappiani¹

¹ Sorbonne Universités, UPMC Univ Paris 06 and CNRS
LIP6, UMR 7606, Paris, France
name.surname@lip6.fr

² Politecnico di Milano
Dept. of Electronics and Information (DEIB), Milano, Italy

Abstract. In this paper, we propose an interactive version of the Borda method for collective decision-making (social choice) when the alternatives are described with respect to multiple attributes and the individual preferences are unknown. More precisely, assuming that individual preferences are representable by linear multi-attribute utility functions, we propose an incremental elicitation method aiming to determine the Borda winner while minimizing the communication effort with the agents. This approach follows the recent work of Lu and Boutilier [8] relying on the minimax regret as a criterion for dealing with uncertainty in the preferences. We show that, when preferences are expressed on a multi-attribute domain and are additively separable over attributes, regret-based incremental elicitation methods can be made more efficient to determine or approximate the Borda winner. Our approach relies on the representation of incomplete preferences using convex polyhedra of possible utilities and is based on linear programming both for minimizing regrets and selecting informative preference queries. It enables to incrementally collect preference judgements from the agents until the Borda winner can be identified. Moreover, we provide an incremental technique for eliciting a collective ranking instead of a single winner.

1 Introduction

Voting is an effective method for collective decision-making, used in political elections, technical committees, academic institutions. Recently, interest in voting has increased in computer science, given the possibility offered by online web systems to support voting protocols, or protocols inspired by voting, for group decision-making (for example, for scheduling a meeting). In many real situations, however, it may be necessary to reason with partial preferences, as some preferences are not available and too expensive to obtain (with respect to a cognitive or economic cost). This observation has motivated a number of recent works on social choice with partial preferences, e.g., [8, 9, 3–6, 2, 12]. In this research stream, typical questions concern the determination of possible and necessary winners, the selection of preference queries to ask to the agents for further eliciting preferences, the approximation of optimal solutions or the determination of robust recommendations based on the available preference information.

Acquiring agents' preferences is expensive (with respect to time and cognitive cost). It is therefore essential to provide techniques that allow to reason with partial preference information, and that can effectively elicit the most relevant part of preferences to make a decision. Adaptive utility elicitation [11, 10, 1] tackles the challenges posed by preference elicitation by representing the system knowledge about the agents' preferences in the form of a set of admissible utility functions. This set includes all functions compatible with the preferences collected so far, and is updated following agents' responses. In this way, one can often make good (or even optimal) recommendations with sparse knowledge of the users' utility functions.

The aim of this paper is to introduce an adaptive utility elicitation procedure in the context of voting, for the fast determination of a Borda winner or a social ranking based on the Borda score, and to test the practical efficiency of this procedure. In particular, we extend the work of [8] to the multi-attribute case. Multiple attributes may appear in well-known collective decision problems such as committee elections or voting in multi-issue domains [7]. In these cases, attributes are boolean and represent elementary decisions on candidates or issues. More generally, the multi-attribute case occurs when the alternatives of a collective decision problem are described by different features, non-necessarily boolean. Individual preferences are assumed here to be representable by a linear function of the attribute values. Since utilities are decomposable over attributes, a set of preference statements formulated by an agent on some pairs of alternatives will possibly allow to infer other preference statements with respect to other pairs, without asking them explicitly. We show in the paper how this type of inference mechanism can be implemented using mathematical programming to reduce the number of queries and speed-up the determination of a necessary Borda winner.

The paper is organized as follows: in Section 2, we introduce the basic framework for voting on multi-attribute domains. Then, we present the minimax regret decision criterion as a useful tool for decision under uncertainty and preference elicitation. In Section 3, we introduce a new method based on mathematical programming to minimize regrets based on the Borda count. Section 4 deals with preference elicitation for the Borda count; we introduce different strategies for generating preference queries and compare them experimentally. Finally, in Section 5, we extend the approach to ranking problems based on the Borda score and provide additional numerical tests to evaluate the efficiency of our approach in ranking.

2 Social Choice in Multi-attribute Domains with Incomplete Preferences

We consider a set of n voters or agents and a set X of m alternatives (candidates, options, items), characterized by a finite set of q attributes or criteria; an alternative is associated to a vector $x = (x_1, \dots, x_q)$ where each x_k represents the value of an attribute k or a performance with respect to a given point of view.

Individual preferences are assumed here to be represented by linear utilities of the form $u^i(x) = \sum_{k=1}^q \omega_k^i x_k$, where $\omega^i = (\omega_1^i, \dots, \omega_q^i)$ is a vector of weights characterizing the preferences of agent i . Hence, an alternative x is as least as good as y for agent i whenever $\sum_{k=1}^q \omega_k^i x_k \geq \sum_{k=1}^q \omega_k^i y_k$. Our framework can be used to address

two different cases: a multi-criteria decision setting or a multi-attribute utility where the utility is defined as the weighted sum of attribute values. Formally, these preferences are defined by the following relation \succsim_i :

$$x \succsim_i y \quad \text{iff} \quad \sum_{k=1}^q \omega_k^i (x_k - y_k) \geq 0$$

A preference profile $\langle \succsim_1, \dots, \succsim_n \rangle$ of an election is therefore completely characterized by the weight vectors $\omega^1, \dots, \omega^n$ (each associated with an agent). We can now define the Borda score in our multi-attribute settings, where preferences are defined by the utility weights. Given $\omega = \langle \omega^1, \dots, \omega^n \rangle$, the Borda score $s(x, \omega)$ of an alternative x is

$$s(x, \omega) = \sum_{i=1}^n s^i(x, \omega^i)$$

where $s^i(x, \omega^i) = |\{y \in X \mid x \succ_i y\}|$ counts the number of alternatives that are strictly beaten by x according to the preference relation induced from ω^i , where \succ_i is the asymmetric part of \succsim_i : $x \succ_i y$ iff \succeq_i and $\neg(y \succeq_i x)$. Our definition allows for ties in each ranking. When using only linear orders (i.e. the ω^i 's are such that there are no ties) we get the usual Borda count.

When the weights of the agents are not known to the system with certainty, we need to reason about partially specified preferences. This is done by assuming a vector $\Omega = \langle \Omega^1, \dots, \Omega^n \rangle$ where each Ω^i is the set of feasible ω^i that are consistent with the available preference information on agent i . Later, we will use Ω (that represents our uncertainty about the weights associated with the agents) in order to provide a recommendation based on minimax regret. At the level of a single agent i , we can check whether pairs of alternatives are in a necessary preference relation given Ω^i .

Definition 1. *Alternative x is necessarily weakly preferred to y for agent i , written $x \succsim_i^N y$, iff $\forall \omega^i \in \Omega^i, \sum_{k=1}^q \omega_k^i (x_k - y_k) \geq 0$. Similarly, x is necessarily strictly preferred to y for agent i , written $x \succ_i^N y$, iff $\forall \omega^i \in \Omega^i, \sum_{k=1}^q \omega_k^i (x_k - y_k) > 0$.*

The necessarily strictly preferred relation \succ_i^N should not be confused with the asymmetric part³ of the necessarily weakly preferred relation \succsim_i^N .

At the level of the community of the agents, a possible Borda winner is an alternative such that there exists a feasible instantiation of the weights that makes it a Borda winner; a necessary Borda winner is a Borda winner for all feasible instantiations of the weights.

In general the sets $\Omega^1, \dots, \Omega^n$ are not given directly but are inferred by available preference statements. Any preference statement of type $x \succsim_i y$ for agent i is indeed interpreted as a linear constraint $\omega^i \cdot (x - y) \geq 0$. Therefore, after collecting several preferences of this type, Ω^i is a convex polyhedron in the space of weights.

When the utility weights are known and characterized by $\omega = \langle \omega^1, \dots, \omega^n \rangle$, the actual loss or *real regret* of an alternative x is the shortfall in Borda score that occurs by choosing x instead of the optimal choice x_ω^* ; more formally:

$$\text{Regret}(x, \omega) = \max_{y \in X} \{s(y, \omega)\} - s(x, \omega) = s(x_\omega^*, \omega) - s(x, \omega).$$

³ The asymmetric part \triangleright_i^N of \succsim_i^N is defined as $x \triangleright_i^N y$ iff $(x \succsim_i^N y) \wedge \neg(y \succsim_i^N x)$.

Instead, when the actual weights $\omega = \langle \omega^1, \dots, \omega^n \rangle$ are not known, but some preferences are available, we are interested in quantifying how “bad” a choice can be with respect to the current uncertainty about the weights, encoded by $\Omega = \langle \Omega^1, \dots, \Omega^n \rangle$. To this end, we first define pairwise max regret, then max regret and finally minimax regret as proposed in [10, 8]. The *pairwise max regret* $\text{PMR}(x, y, \Omega)$ of alternative x relative to y under Ω is the worst-case loss, in terms of Borda score, of selecting the alternative x instead of y . The *max regret* $\text{MR}(x, \Omega)$ is the worst-case loss of choosing x : this can be viewed as an adversarial selection of the instantiation of the weights ω to maximize the loss between x and the true winner under ω . We want to choose the alternative x minimizing max regret: the *minimax regret* $\text{MMR}(\Omega)$ represents the smallest max regret under Ω . These concepts are formalized below:

$$\begin{aligned} \text{PMR}(x, y, \Omega) &= \max_{\omega \in \Omega} [s(y, \omega) - s(x, \omega)], \\ \text{MR}(x, \Omega) &= \max_{y \in X} \text{PMR}(x, y, \Omega), \end{aligned} \tag{1}$$

$$\text{MMR}(\Omega) = \min_{x \in X} \text{MR}(x, \Omega). \tag{2}$$

Finally the *minimax optimal alternative* x_Ω^* is any alternative x minimizing regret MR over Ω (i.e. $x_\Omega^* \in \arg \min_{x \in X} \text{MR}(x, \Omega)$). Solution x_Ω^* is an approximate winner of the current election according to the minimax regret criterion; it gives us the safest choice with respect to the uncertainty on the preference weights attached to the agents; this will be suggested as a recommendation for the social choice problem given the available preference information. We recall from [8] the observation that the regret-minimizing alternative may not be a possible winner. Another important property is that, if $\text{MMR}(\Omega) = 0$, then x_Ω^* is a necessary winner.

3 Minimax Regret Computation for Borda

We are now interested in the computation of minimax regret, given the uncertainty sets $\langle \Omega^1, \dots, \Omega^n \rangle$, when using Borda count as voting rule on a multi-attribute domain. Note that the computation of the pairwise max-regret values PMR is the cornerstone of the problem: once we have computed $\text{PMR}(x, y, \Omega)$ for all $x, y \in X$, max regret $\text{MR}(x, \Omega)$ for all x and then minimax regret $\text{MMR}(\Omega)$ can be computed directly from the definitions (Equations 1 and 2).

The main intuition for computing minimax regrets comes from [8]; however, in our multi-attribute settings, computing PMR is more involved as we need to deal with the multi-attribute structure of the domain. The key idea is to exploit the decomposition of PMR with respect to the different agents:

$$\text{PMR}(x, y, \Omega) = \sum_{i=1}^n \max_{\omega^i \in \Omega^i} [s^i(y, \omega^i) - s^i(x, \omega^i)]$$

This decomposition allows to decompose the PMR maximization problem into a series of simpler maximization problems. For each agent i , we maximise the contribution to

PMR separately, which is defined as follows:

$$\text{PMR}_i(x, y, \Omega^i) = \max_{\omega^i \in \Omega^i} [s^i(y, \omega^i) - s^i(x, \omega^i)]$$

This optimization problem gives the maximal difference between the number of alternatives strictly less preferred than y and the number of alternatives strictly less preferred than x (according to the i^{th} -agent's preferences); note that, if there is no tie, this corresponds to maximizing the difference between their rank. Let ω^i be the weighted vector maximizing this value and \succsim_i be the preference relation induced by ω^i . From the definition of the scores, we have:

$$s^i(y, \omega^i) - s^i(x, \omega^i) = \begin{cases} -|\{z \in X, x \succ_i z \succsim_i y\}| & \text{if } x \succsim_i y \\ |\{z \in X, y \succ_i z \succsim_i x\}| & \text{otherwise} \end{cases}$$

However, since we do not know in which case we are (ω^i is not known), we make use of the necessarily preferred relation \succsim_i^N in order to check whether some conclusions can be drawn from the available information about the preference between x and y . More precisely, we distinguish whether it is known that x is necessarily weakly preferred to y or not. Then, we deduce the weighting vector that maximizes the contribution to regret of agent i . Note that checking whether $x \succsim_i^N y$ can be simply performed using a linear program, by testing the condition $\min_{\omega^i \in \Omega^i} \{(x - y) \cdot \omega^i\} \geq 0$. We now express two mutually exclusive cases using the necessary preference relation.

1) case $x \succsim_i^N y$: in that case, we have $s^i(y, \omega^i) - s^i(x, \omega^i) \leq 0$ for all $\omega^i \in \Omega^i$ by definition of \succsim_i^N . This induces that the contribution to $\text{PMR}(x, y, \Omega)$ is non-positive and more precisely, we have $\text{PMR}_i(x, y, \Omega^i) = -\min_{\omega^i \in \Omega^i} |\{z \in X, x \succ_i z \succsim_i y\}|$. Hence, to maximize the pairwise max regret $\text{PMR}(x, y, \Omega)$, we need to minimize over Ω^i the cardinality of the set $\{z \in X, x \succ_i z \succsim_i y\}$ as much as possible.

2) case $\neg(x \succsim_i^N y)$: there exists $\omega^i \in \Omega^i$ such that $s^i(y, \omega^i) - s^i(x, \omega^i) \geq 0$ by definition of \succsim_i^N . Therefore, we know that the contribution to $\text{PMR}(x, y, \Omega)$ is non-negative here. More precisely, we have $\text{PMR}_i(x, y, \Omega^i) = \max_{\omega^i \in \Omega^i} |\{z \in X, y \succ_i z \succsim_i x\}|$. Hence, we need to maximize the cardinality of the set $\{z \in X, y \succ_i z \succsim_i x\}$ to maximize the pairwise max regret $\text{PMR}(x, y, \Omega)$.

In the following, we consider the problem of computing $\text{PMR}_i(x, y, \Omega^i)$ for any x, y and i . First of all, we need to define the following sets for any $a \in \{x, y\}$:

$$U^a = \{z \in X \setminus \{a\}, z \succsim_i^N a\}, L^a = \{z \in X, a \succ_i^N z\}, V^a = X \setminus (\{a\} \cup U^a \cup L^a)$$

and for any pair of alternatives $(a, b) \in \{(x, y), (y, x)\}$:

$$M^{a,b} = L^a \cap U^b, Z_1^{a,b} = L^a \cap V^b, Z_2^{b,a} = U^b \cap V^a, Z_3^{a,b} = V^a \cap V^b$$

These sets are computed for each user i using linear programming (repeatedly testing \succsim_i^N or \succ_i^N on pairs of alternatives) and allow us to partition the set X for the computation of $\text{PMR}_i(x, y, \Omega^i)$. We refer the reader to Figure 1 where the different cases are visualized; for simplicity, we only show the transitive reduction of the preference relation and we distinguish whether it is known that y is necessarily weakly preferred to x or not (if not, set $M^{y,x}$ is empty). Note that, in the following, we may write Z_1, Z_2 and Z_3 (dropping the superscripts) when the case considered is clear from the context.

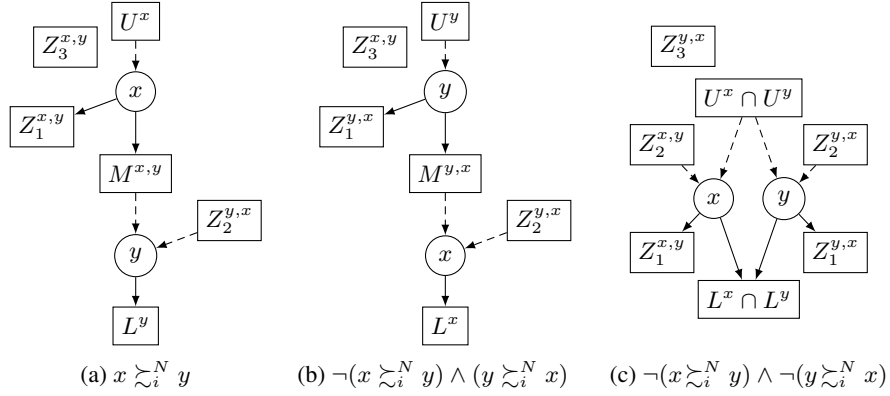


Fig. 1: Partition of set X with respect to the value of \succsim_i^N with x and y for agent i . The solid (resp. dashed) arcs represent necessary strict (resp. weak) preferences.

1) case $x \succsim_i^N y$ (Figure 1a): We want to compute $\text{PMR}_i(x, y, \Omega^i)$. Recall that, in this case, $\text{PMR}_i(x, y, \Omega^i) = -\min_{\omega^i \in \Omega^i} |\{z \in X, x \succ_i z \succsim_i y\}|$. Hence, we want to find a feasible $\omega^i \in \Omega^i$ such that as few of the alternatives $z \in X$ are such that $x \succ_i z \succsim_i y$. First, let us note that none of the alternatives z in $U^x \cup L^y$ verify $x \succ_i z \succsim_i y$ for some $\omega^i \in \Omega^i$ (by definition of U^x and L^y). Moreover, $x \succ_i z \succsim_i y$ for all alternatives $z \in M^{x,y}$ and all $\omega^i \in \Omega^i$ (by definition of $M^{x,y}$). Therefore, we have:

$$\text{PMR}_i(x, y, \Omega^i) = -|M^{x,y}| - \min_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{y\}, x \succ_i z \succsim_i y\}|$$

Thus, we need to compute $\min_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{y\}, x \succ_i z \succsim_i y\}|$ to determine $\text{PMR}_i(x, y, \Omega^i)$. We propose now a mixed-integer programming formulation (named $\text{MIP}_{x,y}$) to solve the latter optimization problem:

$$\begin{aligned}
 (\text{MIP}_{x,y}): \quad & \min \quad b_0 + \sum_{z \in Z_1} b_1^z + \sum_{z \in Z_2} b_2^z + \sum_{z \in Z_3} b_3^z \\
 \text{s.t.} \quad & \sum_{j=1}^q \omega_j^i = 1 \tag{3} \\
 & \omega^i \cdot (a - b) \geq 0, \quad \forall (a, b) \in \mathcal{P}_{\geq}^i \tag{4} \\
 & \omega^i \cdot (a - b) \geq \varepsilon, \quad \forall (a, b) \in \mathcal{P}_{>}^i \tag{5} \\
 & \omega^i \cdot (y - x) + Cb_0 \geq 0 \tag{6} \\
 & \omega^i \cdot (y - z) + Cb_1^z \geq \varepsilon, \quad \forall z \in Z_1 \cup Z_3 \tag{7} \\
 & \omega^i \cdot (z - x) + Cb_2^z \geq 0, \quad \forall z \in Z_2 \cup Z_3 \tag{8} \\
 & b_3^z \geq b_1^z + b_2^z - 1, \quad \forall z \in Z_3 \tag{9} \\
 & \omega_j^i \geq 0, \quad \forall j \in \{1, \dots, q\}; \quad b_0 \in \{0, 1\}; \quad b_3^z \in \{0, 1\}, \quad \forall z \in Z_3 \\
 & b_1^z \in \{0, 1\}, \quad \forall z \in Z_1 \cup Z_3; \quad b_2^z \in \{0, 1\}, \quad \forall z \in Z_2 \cup Z_3
 \end{aligned}$$

In this program, the variables are $\omega^i = (\omega_1^i, \dots, \omega_q^i)$, a vector of q positive real numbers, binary variable b_0 and binary variables b_1^z for each $z \in Z_1 \cup Z_3$, b_2^z for each $z \in Z_2 \cup Z_3$, and b_3^z for each $z \in Z_3$ (we therefore have $q + |Z_1| + |Z_2| + 3|Z_3| + 1$ variables). C is an arbitrary large constant value and ε is an arbitrary small and positive constant modelling strict inequalities. Constraint 3 simply states that the weights should be normalized to add up to 1. Constraints 4 and 5 model the fact that weight ω^i should satisfy both the weak preference statements in \mathcal{P}_{\geq}^i and the strict preference statements in $\mathcal{P}_{>}^i$ obtained from agent i ; indeed, set Ω^i is defined by these preference statements.

Proposition 1 *If $x \succ_i^N y$, then $\text{PMR}_i(x, y, \Omega^i) = -|M^{x,y}| - \text{OPT}$, where OPT is the optimum of mixed-integer program $\text{MIP}_{x,y}$.*

Proof. We want to prove that $\min_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{y\}, x \succ_i z \succsim_i y\}|$ is the optimum of $\text{MIP}_{x,y}$, i.e. we want to show that the objective function counts the cardinality of $\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{y\}, x \succ_i z \succsim_i y\}$. In this program, we use a set of binary variables b_1^z , b_2^z and b_3^z to represent the condition $x \succ_i z \succsim_i y$ for alternatives z in Z_1 , Z_2 and Z_3 respectively. Binary variable b_0 represents whether x is strictly preferred to y (otherwise the contribution to PMR is null). The objective function sums up over all variables b_0 , b_1^z , b_2^z and b_3^z , so that we count the cardinality of $\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{y\}, x \succ_i z \succsim_i y\}$. We now prove that each binary variable is equal to one iff the corresponding constraint is satisfied. Since the objective is a minimization, the values of the binary variables b_0 , b_1^z , b_2^z and b_3^z (that appear in the objective function), will be 0 unless forced to 1.

The binary variable b_1^z , for $z \in Z_1$, represents whether alternative z verifies $x \succ_i z \succsim_i y$. Equation 7 indeed enforces $b_1^z = 1$ when $\omega^i \cdot (z - y) \geq 0$, i.e. when $z \succsim_i y$; otherwise, variable b_1^z is set to zero since we are minimizing the objective function. Then, since $x \succ_i z$ (by definition of Z_1), we have that $b_1^z = 1$ iff $x \succ_i z \succsim_i y$.

For all alternatives $z \in Z_2$, we know that $z \succsim_i y$ by definition. Therefore, z will be such that $x \succ_i z \succsim_i y$ iff $x \succ_i z$. The binary variable b_2^z will take value 1 in this case. This is indeed guaranteed by Constraint 8 enforcing $b_2^z = 1$ when $\omega^i \cdot (z - x) < 0$, i.e. if x is strictly preferred to z . If instead z is preferred to x , then the value $\omega^i \cdot (z - x)$ is positive and Constraint 8 is vacuous; in this case, b_2^z will take value 0, as desired, because we are minimizing.

For all alternatives $z \in Z_3$, the two previous conditions need to be satisfied in order for z to contribute to the score difference. Constraint 9 implements an *and* between these two conditions ($b_3^z = 1$ iff $x \succ_i z$ and $z \succsim_i y$).

Finally, while we know that y cannot be strictly preferred to x (since $x \succ_i^N y$), it might be the case that they are equally preferred. The binary variable b_0 represents whether x is strictly preferred to y ; more precisely, Constraint 6 enforces that $b_0 = 1$ whenever $\omega^i \cdot (y - x) < 0$. \square

2) case $\neg(x \succ_i^N y)$ (Figures 1b and 1c): Recall that, in this case, $\text{PMR}_i(x, y, \Omega^i) = \max_{\omega^i \in \Omega^i} |\{z \in X, y \succ_i z \succsim_i x\}|$. Therefore, we aim to find a feasible $\omega^i \in \Omega^i$ so that as many of the alternatives $z \in X$ are such that $y \succ_i z \succsim_i x$. Since we are maximizing, the optimal $\omega^i \in \Omega^i$ will be such that $y \succsim_i x$; thus, the case represented in Figure 1c reduces to the one depicted in Figure 1b. We now focus on the optimization

of $\text{PMR}_i(x, y, \Omega^i)$ for Figure 1b. Similarly to the first case, note that none of the alternatives z in $U^y \cup L^x$ verifies $y \succ_i z \succsim_i x$ for some $\omega^i \in \Omega^i$. Moreover, all alternatives $z \in M^{y,x}$ are such that $y \succ_i z \succsim_i x$ for all $\omega^i \in \Omega^i$. Therefore:

$$\text{PMR}_i(x, y, \Omega^i) = |M^{y,x}| + \max_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{x\}, y \succ_i z \succsim_i x\}|$$

Thus, we need to compute $\max_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{x\}, y \succ_i z \succsim_i x\}|$. This can be performed by solving the following program (named $\text{MIP}_{y,x}$ hereafter):

$$\begin{aligned} (\text{MIP}_{y,x}): \quad & \max \quad b_0 + \sum_{z \in Z_1} b_1^z + \sum_{z \in Z_2} b_2^z + \sum_{z \in Z_3} b_3^z \\ \text{s.t.} \quad & \sum_{j=1}^q \omega_j^i = 1 \\ & \omega^i \cdot (a - b) \geq 0, \quad \forall (a, b) \in \mathcal{P}_{\geq}^i \\ & \omega^i \cdot (a - b) \geq \varepsilon, \quad \forall (a, b) \in \mathcal{P}_{>}^i \\ & \omega^i \cdot (y - x) + (1 - b_0)C \geq \varepsilon \tag{10} \\ & \omega^i \cdot (z - x) + (1 - b_1^z)C \geq 0, \quad \forall z \in Z_1 \cup Z_3 \tag{11} \\ & \omega^i \cdot (y - z) + (1 - b_2^z)C \geq \varepsilon, \quad \forall z \in Z_2 \cup Z_3 \tag{12} \\ & b_3^z \leq b_1^z, \quad \forall z \in Z_3 \tag{13} \\ & b_3^z \leq b_2^z, \quad \forall z \in Z_3 \tag{14} \\ & \omega_j^i \geq 0, \quad \forall j \in \{1, \dots, q\}; \quad b_0 \in \{0, 1\}; \quad b_3^z \in \{0, 1\}, \quad \forall z \in Z_3 \\ & b_1^z \in \{0, 1\}, \quad \forall z \in Z_1 \cup Z_3; \quad b_2^z \in \{0, 1\}, \quad \forall z \in Z_2 \cup Z_3 \end{aligned}$$

Proposition 2 *If $\neg(x \succsim_i^N y)$, then $\text{PMR}_i(x, y, \Omega^i) = |M^{y,x}| + \text{OPT}$, where OPT is the optimum of mixed-integer program $\text{MIP}_{y,x}$.*

The proof is similar to that of the previous condition, however since the objective is a maximization, the values of the binary variables b_0, b_1^z (for $z \in Z_1$), b_2^z (for $z \in Z_2$) and b_3^z (for $z \in Z_3$) will be 1 unless forced to be 0. Constraints 10-14 formalize the required behaviour: the value of each binary variable, relative to a specific z , will be set to 1 unless ω^i is chosen in a way such that $y \succ_i z \succsim_i x$.

Note that the MIP formulations might be too computationally demanding for problems involving a large number of alternatives (since there are one or more integer variables per alternative). For this reason, we will consider the linear programming relaxation of these programs, i.e., the linear programs obtained by replacing boolean variables b_0, b_1^z, b_2^z, b_3^z by continuous variables belonging to the unit interval. The resulting optimization problems are solvable in polynomial time using linear programming; however the solution gives an upper bound on pairwise max regret values (instead of the exact value). The relaxed values for PMR are then aggregated giving a relaxed MMR value. Note that, since optimizing the relaxed problem gives an upper bound, the result

can still be used in order to provide a robust recommendation with worst-case guarantees; the guarantee is less strong than if pairwise max regret values were computed exactly, but computation times are significantly improved as shown in Subsection 4.2.

4 Incremental Elicitation

Given the available preference information, the worst-case loss ensured by the minimax regret might be at unacceptable level. In order to approximate the Borda winner with the desired guarantee (expressed by the minimax regret value), we may ask additional preference information to the agents. By incorporating the responses to additional questions, we can indeed refine the uncertainty sets and therefore reduce this loss.

4.1 Elicitation Strategies

We adopt an incremental setting where preference queries are selected incrementally according to the current available information until the minimax regret is zero; at that point, we know that alternative x_Ω^* is a necessary Borda winner. We allow asking queries that may induce either weak or strict preference statements. In order to limit the cognitive effort of the agents, it is important to ask queries that are informative (roughly, a query is informative if it significantly reduces regrets whatever the answer); in particular, the computation of minimax regret can suggest queries that may be able to impose a significant reduction of regrets. One common technique, also known as the *Current Solution Strategy* (CSS), is to consider one of the current “best challenger” y_Ω^* of the approximate winner: $y_\Omega^* \in \arg \max_{y \in X} \text{PMR}(x_\Omega^*, y, \Omega)$. New preference information involving the pair (x_Ω^*, y_Ω^*) is indeed often useful to reduce the minimax regret efficiently, which is equal to $\text{PMR}(x_\Omega^*, y_\Omega^*, \Omega)$. We propose now two elicitation strategies of different complexity, that are aimed to reduce $\text{PMR}(x_\Omega^*, y_\Omega^*, \Omega)$.

Multi-Attribute-CSS0 (MA-CSS0). This strategy selects a pair (agent, query) such that the answer may reduce the agent’s contribution to $\text{PMR}(x_\Omega^*, y_\Omega^*, \Omega)$. More precisely, an agent i is selected at random and the strategy proceeds as follows:

1) *case $x_\Omega^* \succsim_i^N y_\Omega^*$:* recall that, in this case, $\text{PMR}_i(x_\Omega^*, y_\Omega^*, \Omega^i) = -|M^{x_\Omega^*, y_\Omega^*}| - \min_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{y\}, x_\Omega^* \succ_i z \succsim_i y_\Omega^*\}|$. We distinguish two cases:

- *case $Z_1 \cup Z_2 \cup Z_3 = \emptyset$:* if $\neg(x_\Omega^* \succ_i^N y_\Omega^*)$, then we ask the agent whether x_Ω^* is strictly preferred to y_Ω^* . If, instead, $x_\Omega^* \succ_i^N y_\Omega^*$, we know precisely the difference of scores between x_Ω^* and y_Ω^* for agent i , that is $-|M^{x_\Omega^*, y_\Omega^*}| - 1$. In this case, asking a query to agent i is useless (since his/her contribution to $\text{PMR}(x_\Omega^*, y_\Omega^*, \Omega)$ cannot be decreased) and so the strategy selects another agent at random.
- *case $\neg(Z_1 \cup Z_2 \cup Z_3 = \emptyset)$:* an alternative z in $Z_1 \cup Z_2 \cup Z_3$ is selected at random. For each $z \in Z_1 \cup Z_2 \cup Z_3$, our current knowledge about the agent’s preferences is not sufficient to conclude on whether $x_\Omega^* \succ_i z \succsim_i y_\Omega^*$ is satisfied or not. More precisely, if $z \in Z_1$, then we know that x_Ω^* is strictly preferred to z by definition, but there exists $\omega^i \in \Omega^i$ such that $\neg(z \succsim_i y_\Omega^*)$. Therefore, we ask the agent whether

z is (weakly) preferred to y_Ω^* so as to obtain the missing information. Similarly, if $z \in Z_2$, then we know that z is preferred to y_Ω^* , and so the agent is asked whether x_Ω^* is strictly preferred to z . Finally, if $z \in Z_3$, then we ask one of the two previous questions, the choice between the two questions being randomly made.

2) case $\neg(x_\Omega^* \succsim_i^N y_\Omega^*)$: recall that, in this case, $\text{PMR}_i(x_\Omega^*, y_\Omega^*, \Omega^i) = |M^{y_\Omega^*, x_\Omega^*}| + \max_{\omega^i \in \Omega^i} |\{z \in Z_1 \cup Z_2 \cup Z_3 \cup \{x\}, y_\Omega^* \succ_i z \succsim_i x_\Omega^*\}|$. We distinguish three cases:

- case $\neg(y_\Omega^* \succsim_i^N x_\Omega^*)$: in this case, x_Ω^* and y_Ω^* are incomparable for the system, and so we ask the agent to compare them directly.
- case $(y_\Omega^* \succsim_i^N x_\Omega^*) \wedge (Z_1 \cup Z_2 \cup Z_3 = \emptyset)$: if $\neg(y_\Omega^* \succ_i^N x_\Omega^*)$, then the agent is asked whether y_Ω^* is strictly preferred to x_Ω^* . If, instead, $y_\Omega^* \succ_i^N x_\Omega^*$, then the difference of scores between x_Ω^* and y_Ω^* for this agent is equal to $|M^{y_\Omega^*, x_\Omega^*}| + 1$. In this case, asking a query to agent i is useless and another agent is selected at random.
- case $(y_\Omega^* \succsim_i^N x_\Omega^*) \wedge \neg(Z_1 \cup Z_2 \cup Z_3 = \emptyset)$: an alternative z in $Z_1 \cup Z_2 \cup Z_3$ is selected at random and we want to know whether $y_\Omega^* \succ_i z \succsim_i x_\Omega^*$. More precisely, if $z \in Z_1$, then we ask the agent whether z is (weakly) preferred to x_Ω^* . Instead, if $z \in Z_2$, then we ask the agent if y_Ω^* is strictly preferred to z . Finally, if $z \in Z_3$, then we ask one of the two previous questions.

Multi-Attribute-CSS1 (MA-CSS1). This strategy is based on the heuristics proposed by Lu and Boutilier [8] but adapted to our multi-attribute setting. The aim is to choose the query with the highest potential of reducing $\text{PMR}(x^*, y^*, \Omega)$. More precisely, instead of choosing the agent and the alternative $z \in Z_1 \cup Z_2 \cup Z_3$ at random (as in MA-CSS0), strategy MA-CSS1 selects the pair (agent, query) that maximizes the minimax regret reduction in the most optimistic scenario; it therefore requires the computation of the resulting minimax regret for each pair (agent, query).

4.2 Numerical Tests

We performed a number of numerical experiments in order to evaluate the proposed elicitation procedures for determining the Borda winner in an incremental process. In these experiments, the attribute values for each alternative are randomly sampled in $[0, 1]^q$. Starting from an empty set of preference statements, we repeatedly compute minimax regret and we ask a new question to one of the agent according to an elicitation strategy. We simulate answers to queries according to randomly generated vectors $\omega^1, \dots, \omega^n$ (one vector per agent). Optimizations are performed using the Gurobi solver; the simulation environment is implemented in Java.

In the first experiment, we evaluate the impact of exploiting the fact that the domain is multi-attribute. We implemented the elicitation procedure proposed in [8] (named CSS1 hereafter) where no assumption is made about the “structure” of the agents preferences, and compare it with our strategies MA-CSS0 and MA-CSS1.⁴ In Figure 2a,

⁴ Note that CSS1 and MA-CSS1 adopt the same heuristics for choosing the pair (agent, query); the difference is that MA-CSS1 makes use of the multi-attribute structure (using linear programming) for identifying the sets Z_1, Z_2 , etc., and computing regrets, while CSS1 does not.

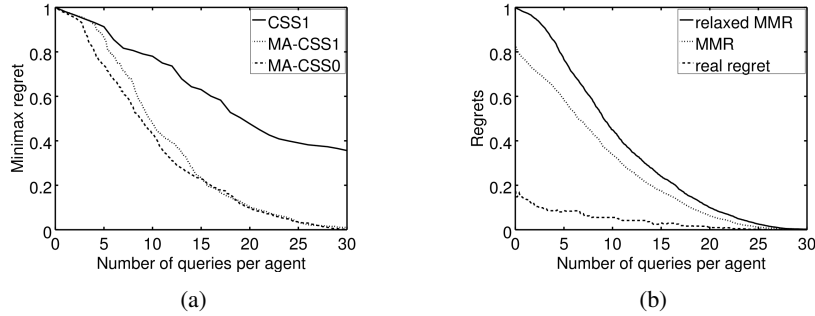


Fig. 2: Evaluation of the elicitation strategies; regret reduction is plotted as a function of the average number of queries per agent (30 alternatives, 5 criteria and 10 agents; results averaged over 30 runs). In (a) we plot the reduction of minimax regret obtained by different elicitation strategies; in (b) we compare the upper bound of MMR obtained with the relaxed optimization, the exact computation of MMR and the real regret.

we report the minimax regret, computed at each step of the incremental elicitation procedure. Regret values are expressed on a normalized scale, with 1 corresponding to the initial MMR (computed before acquiring any preference information). Note that a value of 0 for MMR implies identification of a Borda winner. We observe that the MMR reduces much more slowly with CSS1 than with its multi-attribute version MA-CSS1; after 20 queries, the MMR is still above 40% of the initial value with CSS1, while it is under 10% with MA-CSS1. Moreover, after 30 queries per agent on average, the MMR is still around 40% of the initial regret with CSS1 while MA-CSS1 has identified the Borda winner. Then, we observe (somewhat surprisingly) that the heuristics used by MA-CSS1 is less effective than MA-CSS0. Since MA-CSS1 is much more computational demanding than MA-CSS0, in the following experiments, we use MA-CSS0.

The second experiment evaluates the quality of the upper bound obtained when using the linear programming relaxation of the MMR optimization. Figure 2b shows the minimax regret, the upper bound obtained by linear programming relaxation and the real regret (the actual loss in terms of Borda score) at each iteration step of the elicitation procedure. We can see that the linear programming relaxation gives us a relatively tight upper bound on the minimax regret and its quality improves with the number of preference statements. Recall that the relaxed version is significantly faster than the exact version, as the former solves linear programming problems instead of mixed integer linear problems. For instance, when no preferences are given, the relaxed optimization takes about 1s on average while the exact method needs 30s to compute the value of initial minimax regret. The determination of the next query is also faster when using the relaxed optimization (2s against 12s). Even if, by optimizing the relaxed problem, we are potentially ignoring some valuable information, the experiment shows that the elicitation performs well. The recommended choice is the alternative whose “relaxed” MMR is lowest; the real regret associated to this choice is small and quickly

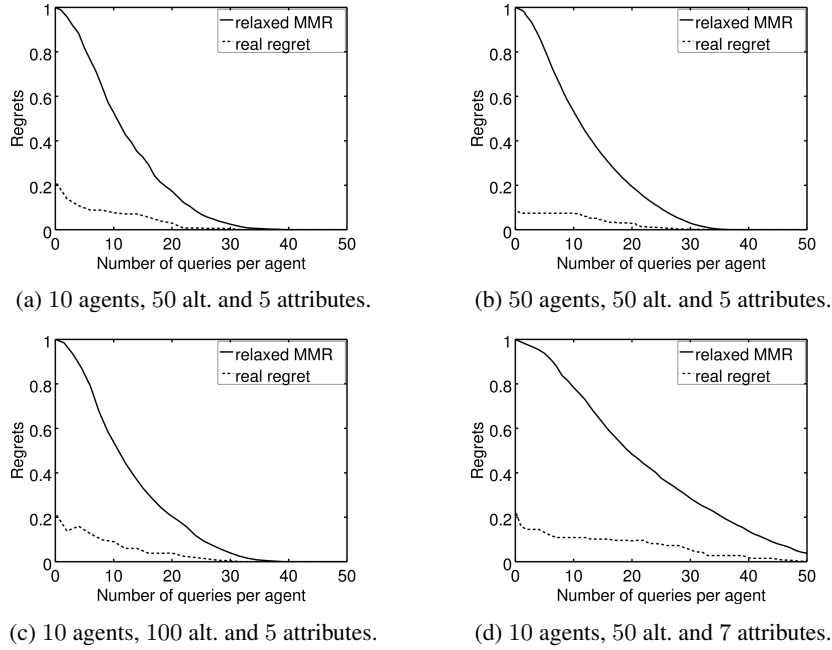


Fig. 3: Performance of MA-CSS0 with the relaxed version of minimax regret (30 runs).

decreases to zero. Note that the fact that real regret is much smaller than minimax regret in practice has already been observed [10].

The third experiment aims to evaluate the performance of MA-CSS0, using the relaxed optimization of regrets, when increasing the size of the problem (number of agents, number of alternatives and number of criteria). Figure 3 shows that, with 5 attributes, our incremental elicitation procedure determines a necessary Borda winner in about 30-35 queries asked to each agent; however, with 7 attributes, slightly more than 50 queries are needed. In all cases, the real regret is low even after a few queries.

5 Determination of the Social Ranking Induced by Borda Scores

There are many decision situations where knowing the top- k alternatives is the desirable output. When the preference profile is fully known, ranking alternatives with a scoring rule is straightforward. However, when preferences are incomplete, incremental elicitation methods need to be adapted to efficiently focus the elicitation effort on the determination of the top- k alternatives. We address here the problem of ranking as one of repeated choices, assuming that we want to incrementally rank alternatives from best to worst; we can generate preference queries until the minimax regret drops to 0, meaning that the Borda winner has been identified. Then, this alternative is put aside⁵

⁵ It may still be associated with a binary variable b^z in the optimization problems for computing regrets (as it can impact the Borda score of other alternatives).

and the selection process is iterated on the remaining set of alternatives. The alternative selected in the second stage will be the second best alternative in the ranking induced by Borda scores and so forth.

Numerical Tests. We perform an experiment that evaluates the performance of our incremental assessment of ranking (when used with MA-CSS0) in comparison to approaches that are more systematic. We consider the following two elicitation procedures: strategy *S1* determines the preference order of each agent by adapting a standard sorting algorithm (it requires $O(m \log_2(m))$ comparison queries per agent); the ranking is then obtained by straightforward computation of the Borda scores. Instead, strategy *S2* iteratively applies a regret-based incremental elicitation procedure for the determination of the best alternative in terms of a linear utility model *for a single agent*. The procedure is repeated in order to find the second item, the third, and so on; this is done for all agents and finally Borda scores are computed. In Table 1, we report the average number of comparison queries per agent required to identify the top-10 alternatives, varying n the number of agents, m the number of alternatives and q the number of criteria. Our incremental ranking procedure based on Borda scores is referred to as Incremental Ranking Elicitation (IRE); overall, IRE outperforms both *S1* and *S2*.

n	m	q	IRE	S1	S2
10	30	5	43.3	147.2	58.7
10	50	5	43.7	282.2	67.4
100	30	5	51.1	147.2	87.2
10	30	10	93.3	147.2	178.2

Table 1: Average number of queries per agent for determining the top-10 (30 runs).

We now present some experimental results about our incremental ranking method (when used with MA-CSS0). Figure 4 shows the average number of queries needed to determine the top- k alternatives in domains with 20 agents and 5 criteria. We observe that the marginal amount of queries needed to determine the next best alternative decreases as the rank of the alternatives increases. Actually, most of the elicitation “cost” in terms of queries occurs when determining the top alternative.

6 Conclusions

This paper dealt with social choice in a context where preferences are dictated by a latent (linear) utility function. We provided algorithms for the computation of an approximate winner and elicitation strategies based on minimax regret, extending previous work [8] to multi-attribute domains. We also provided an iterative procedure for top- k ranking and compared our results with full elicitation procedures. Possible directions for future research include: dealing with other voting rules in multi-attribute domains, considering different kinds of queries, and addressing combinatorial domains.

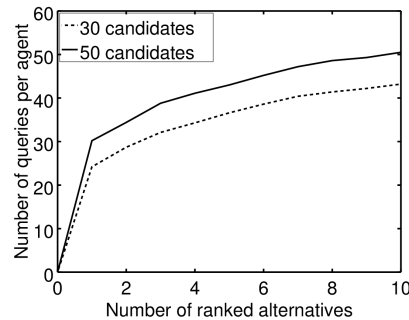


Fig. 4: Performance of top- k elicitation with MA-CSS0 (30 runs).

Acknowledgements

This work is supported by the ANR project 14-CE24-0007-01-Cocorico-CoDec.

References

1. Darius Braziunas and Craig Boutilier. Elicitation of factored utilities. *AI Magazine*, 29(4):79–92, 2008.
2. Lihi Naamani Dery, Meir Kalech, Lior Rokach, and Bracha Shapira. Reaching a joint decision with minimal elicitation of voter preferences. *Information Sciences*, 278:466–487, 2014.
3. Ning Ding and Fangzhen Lin. Voting with partial information: what questions to ask? In *Proceedings of AAMAS-13*, pages 1237–1238, 2013.
4. Meir Kalech, Sarit Kraus, and Gal A. Kaminka. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, 2010.
5. Kathrin Konczak and Jerome Lang. Voting procedures with incomplete preferences. In *Proceedings of IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20, 2005.
6. Jérôme Lang, Maria Silvia Pini, Francesca Rossi, Domenico Salvagnin, Kristen Brent Venable, and Toby Walsh. Winner determination in voting trees with incomplete preferences and weighted votes. *Autonomous Agents and Multi-Agent Systems*, 25(1):130–157, 2012.
7. Jérôme Lang and Lirong Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical social sciences*, 57(3):304–324, 2009.
8. Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proceedings of IJCAI 2011*, pages 287–293, 2011.
9. Tyler Lu and Craig Boutilier. Vote elicitation with probabilistic preference models: Empirical estimation and cost tradeoffs. In *Proceedings of ADT 2011*, pages 135–149, 2011.
10. Tianhan Wang and Craig Boutilier. Incremental utility elicitation with the minimax regret decision criterion. In *Proceedings of IJCAI 2003*, pages 309–316, Acapulco, 2003.
11. 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.
12. Lirong Xia and Vincent Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research (JAIR)*, 41:25–67, 2011.