From preference representation to combinatorial vote

Jérôme Lang
IRIT
Université Paul Sabatier, Toulouse (France)
lang@irit.fr

Abstract

We introduce the notion of combinatorial vote, where a group of agents (or voters) is supposed to express preferences and come to a common decision concerning a set of non-independent variables to assign. We study two key issues pertaining to combinatorial vote, namely preference representation and the automated choice of an optimal decision. For each of these issues, we briefly review the state of the art, we try to define the main problems to be solved and identify their computational complexity.

1 Introduction

For a few years, AI researchers have been concerned with interaction, cooperation or negociation within agent societies. For these problems, it often occurs that the set of all feasible states has a very large size, due to its combinatorial nature. For this reason, research has been done so as to develop representation languages aiming at enabling a succint representation of the description of the problem, without having to enumerate a prohibitive number of states. Recently, languages based on propositional logic have been proposed for some multi-agent problems, namely, combinatorial auctions [8] and automated negociation [39].

In this paper we focus on *combinatorial vote*. Combinatorial vote is located within the larger class of *group decision making* problems. Each one of a set of agents (called *voters*) initially expresses her preferences on a set of alternatives (called *candidates*), these preferences are then aggregated so as to identify (or *elect*) an acceptable common alternative in an automated way (without negociation). Formulated as such, this can be identified as a *vote problem*. Vote problems have been investigated by resarchers in social choice theory (see [29] for an overview) who have studied extensively all properties of various families of voting rules, up to an important detail: candidates are supposed to be listed explicitly (typically, they are individual or lists of individuals, as in political elections), which assumes that they should not be too numerous.

In this paper, we focus on the case where the set of candidates has a *combinatorial structure*, i.e., is a Cartesian product of finite value domains for each one of a set of variables: this problem will be referred to as *combinatorial vote*. In this case, the space of possible alternatives has a size being exponential in the number of variables and it is therefore not reasonable asking the voters to rank or evaluate on a utility scale all alternatives.

Consider for example that the voters have to agree on a common menu to be composed if a first course dish, a main course dish, a dessert and a wine, with a choice of 6 items for each. This makes 6^4 candidates. This would not be a problem if the four items to choose were independent from the other ones: in this case, this vote problem over a set of 6^4 candidates would come down to four independent problems over sets of 6 candidates each, and any standard voting rule could be applied without difficulty. Things become more complicated if voters express dependencies between items, such as "I would like to have risotto ai funghi as first course, except if the main course is a vegetable curry, in which case I would prefer smoked salmon as first course", "I prefer white wine if one of the courses is fish and none is meat, red wine if one of the courses is meat and none is fish, and in the remaining cases I would like equally red or white wine", etc.

Since the preference structure of each voter cannot be expressed explicitly by listing all candidates, what is needed is a compact *preference representation language*. Such preference representation languages have been developed within the KR community; they are often build up on propositional logic, but not always (see for instance utility networks [1] [23] or valued constraint satisfaction [35]); they enable a much more concise representation of the preference structure, while preserving a good readability (and hence a proximity with the way agents express their preferences in natural language).

Therefore, the first parameter to be fixed, for a combinatorial vote problem, is the language for representing the preferences of the voters. In Section 3, we recall different logical representation languages proposed so far, we discuss their relevance to combinatorial vote, and discuss their computational complexity.

In Section 4, we investigate the problem of electing a candidate; by "candidate"; we mean here an assignment of a value to each of the variables involved in the problem. We study the computational complexity of the different problems obtained from the choice of a given representation language and a given voting rule. In Section 5 we evoke other important issues in combinatorial vote and we mention some of its potential applications.

2 Preliminaries

In this paper, $\mathcal{L}$ is a propositional language built upon a finite set of propositional variables VAR, the usual connectives, and the symbols $\top$ (tautology) and $\bot$ (contradiction). A *literal* is a propositional variable or its negation. $W = 2^{VAR}$ is the set of all interpretations for VAR. Elements of W are denoted $\vec{x}$, $\vec{y}$ etc. If $\varphi \in \mathcal{L}$ then $Mod(\varphi)$ denotes the set of the models of φ and $Var(\varphi)$ is the set of propositional variables mentioned in formula φ.

The complexity results we give in this paper refer to some complexity classes which we now briefly recall (see [32] for more details). Given a problem A, we denote by $\overline{\text{A}}$ its complement. We assume the reader familiar with the classes P, NP and coNP and we now introduce the following classes.

- $\mathsf{NP(3)}$ is the class of all languages L such that $L = L_1 \cap (L_2 \cup L_3)$, where L_1 and L_2 are in NP and L_3 in coNP. The canonical $\mathsf{NP(3)}$-complete problem is SAT-SAT-UNSAT: $\langle \varphi_1, \varphi_2, \varphi_3 \rangle$ is a positive instance of SAT-SAT-UNSAT if and only if φ_1 is consistent and (φ_2 is inconsistent or φ_3 is consistent). Theses classes are members of the *Boolean hierarchy* [10].

- $\Delta_2^p = \mathsf{P}^{\mathsf{NP}}$ is the class of all languages that can be recognized in polynomial time by a Turing machine equipped with an NP oracle, where an NP oracle solves whatever instance of a problem NP in unit time. $\Theta_2^p = \Delta_2^p[\mathcal{O}(log\ n)]$) is the class of all languages that can be recognized in polynomial time by a Turing machine using a number of NP oracles bounded by a logarithmic function of the size of the input data.

- $\Sigma_2^p = \mathsf{NP}^{\mathsf{NP}}$ is the class of all languages recognizable in polynomial time by a nondeterministic Turing machine equipped with an NP oracle telling in unit time whether a given propositional formula is satisfiable or not.

- PSPACE is the class of all languages recognizable by a (deterministic or not) Turing machine working with polynomial space.

Note that the following inclusions hold:
$\mathsf{NP}, \mathsf{coNP} \subseteq \mathsf{NP(3)} \subseteq \Theta_2^p \subseteq \Delta_2^p \subseteq \Sigma_2^p \subseteq \mathsf{PSPACE}$

3 Logical representation of preference

In this Section we are concerned with the preferences of a *single voter* over a finite set of candidates $\mathcal{X}$.

3.1 Preference structures

What is a preference structure? In other terms, what is the mathematical model underlying the preferences that an agent has concerning a set of alternatives? There is not a unique answer to this important question that has been discussed for long by researchers in decision theory. One may, roughly speaking, distinguish two main families (not disjoint, however) of preference structures:

- a *cardinal preference structure* consists of an evaluation function (generally called *utility*) $u : \mathcal{X} \to Val$, where Val is a set of numerical valuations (typically, $\mathbb{N}$, $\mathbb{R}$, $[0, 1]$, $\mathbb{R}^+$ etc.).

- a *relational preference structure* consists of a binary[1] relation R on $\mathcal{X}$. R is generally supposed to be a weak order (i.e., a reflexive and transitive relation); in this case the preference structure R is *ordinal*[2], and we then note $\geq$ instead of R. $\geq$ is not necessarily complete (some candidates may not be comparable by a given agent). We note

[1] More generally, this relation may be fuzzy so as to enable representing intensities of preference. We omit this eventuality in this paper.

[2] See [13] for an extensive discussion on the limits of ordinality in decision making under uncertainty, multicriteria decision making and social choice.

$\vec{x} > \vec{y}$ for $\vec{x} \geq \vec{y}$ and not $(\vec{y} \geq \vec{x})$ (strict preference of $\vec{y}$ over $\vec{x}$), and $\vec{x} \sim \vec{y}$ for $\vec{x} \geq \vec{y}$ and $\vec{y} \geq \vec{x}$ (indifference).

Note that any cardinal preference induces an ordinal preference, namely from a utility function u we may define the complete weak order $\geq_u$ defined by $\vec{x} \geq_u \vec{y}$ iff $u(\vec{x}) \geq u(\vec{y})$.

The *explicit representation* of a preference structure consists of the data of all candidates with their utilities (for cardinal preferences) or the whole relation[3] R (for ordinal preferences). These representations have a spatial complexity in $\mathcal{O}(|\mathcal{X}|)$ and $\mathcal{O}(|\mathcal{X}|^2)$, respectively, i.e., in $\mathcal{O}(2^{|VAR|})$ and $\mathcal{O}(2^{|VAR|^2})$.

3.2 Principles of logical representation of preferences

We now assume that $\mathcal{X}$ has a combinatorial nature, namely, $\mathcal{X}$ is a set of possible assignments of each of a certain number of variables to a value of its domain: $\mathcal{X} = D_1 \times ... \times D_n$, where D_i is the set of possible values for variable v_i; the size of $\mathcal{X}$ is exponentially larger than the size of the input. Because explicit representations are unreasonable for such problems. the AI community has developed several preference representation languages that escape this combinatorial blow up. Such languages are said to be *factorized*, or *succinct*, because they enables a much more concise representation of preference structures than explicit representations. In the rest of the paper, for simplicity reasons we make the following two important hypotheses:

1. the representation languages considered are *propositional*, i.e., each v_i is a binary variable: $D_1 = ... = D_n = \{0,1\}$. Propositional languages are not only compact, but also particularly *expressive*, thanks to the expressive power of logic, and therefore close to intuition (ideally, a preference representation language should be easily obtained from its specification in natural language by the agent); moreover, propositional logic benefits from many well-worked algorithms.

2. the set of possible decisions (candidates) $\mathcal{X}$ is identical to the set of physically realizable worlds (by the agents), $Mod(K)$, where K is a propositional formula expressing a set of integrity constraints[4]. This strong assumption assumes that there is *no uncertainty* on the initial state of the world nor on the effects of actions (and this is known by all agents). It is important to keep in mind the strong distinction between K, which represents *knowledge*, and the formulas that represent *goals* (confusing both notions leads to "wishful thinking" [37]).

Now we briefly survey different logical preference representation languages. For each of these languages we discuss its computational complexity. Since the problem at hand is not to *reason about knowledge* but to *decide from preferences*, the important problems are not quite the same as for logical languages for knowledge representation. In particular, the role of inference is smaller, whereas the problems that are particularly important are:

Given a logical specification GB (goal base) of the preferences of an agent ...

- ... *and two candidates $\vec{x}$ and $\vec{y}$, determine if $\vec{x} \geq_{GB} \vec{y}$* : COMPARISON;
- ... *and a candidate $\vec{x}$, determine if $\vec{x}$ is optimal for* $\geq_{GB}$: NON-DOMINANCE;
- ... *and a property represented by a formula ψ, determine if there exists an optimal candidate satisfying ψ* : CAND-OPT-SAT.

3.3 A synthetic survey and some complexity results

An intuitive way for an agent to express his preferences consists in enumerating a set of *goals* (or *desires*), each of which is represented by a propositional formula, possibly with extra data such as weights, priorities, contexts or distances. From now on, G_i and C_i denote propositional formulas and α_i is a positive number. GB is a "goal base" and u_{GB} (resp. $\geq_{GB}$) denotes the utility function (resp. the preference relation) induced by GB.

Some of the complexity results we give are not quite new: indeed, although the complexity of preference representation languages has not been studied in a specific way, several of the problems below are very close to similar to problems of *knowledge* representation, and although the relevant problems are not quite the same whether it is a matter of knowledge or preference, some complexity results established in the former context can induce easily similar results for the latter one.

For the results that can be easily derived from other results, we will give the appropriate reference(s); in

[3] or possibly a subset of R from which R is drawn by transitive closure – this is a detail since this does not enable escaping the combinatorial blow up.

[4] This hypothesis that all models of K are deterministically reachable from the initial situation comes down to say that there exists an individual action $a_{\vec{x}}$ for each $\vec{x} \in Mod(K)$, whose unique possible consequence is $\vec{x}$.

the absence of such references, they are new.

3.3.1 Brute propositional representation

The simplest, or prototypical, logical representation of preference, that we call "brute" and denote by R_{brute}, consists simply in giving a single goal (i.e. a propositional formula) G (or equivalently, a finite set of propositional formulas, interpreted conjunctively). The utility function u_G generated by G is extremely basic: for each possible world $\vec{x} \in Mod(K)$, $u_G(\vec{x}) = 1$ if $\vec{x} \models G$, $u_G(\vec{x}) = 0$ if $\vec{x} \models \neg G$. This representation, very rough since it does not enable more than a distinction between goal states and non-goal states ("binary" utility), is of little interest in practice but we will often refer to it because it provides lower bounds for complexity results.

Proposition 1 *For $R = R_{brute}$:*

1. COMPARISON *is in* P *;*
2. NON-DOMINANCE *is* coNP*-complete ;*
3. CAND-OPT-SAT *is* NP(3)*-complete.*

Sketch of proof : (1) is obvious. The membership part of (2) comes from the fact that candidate $\vec{x} \in Mod(K)$ is optimal for $\geq_G$ iff $\vec{x} \models G$ or $G \wedge K$ is unsatisfiable. Lastly, there exists an optimal candidate satisfying ψ iff ψ is consistent and ($K \wedge G$ is inconsistent or $K \wedge G \wedge \psi$ is consistent), which gives the membership part of (3). Hardness comes from the following reduction from SAT-SAT-UNSAT: $\langle \varphi_1, \varphi_2, \varphi_3 \rangle$ is a positive instance of SAT-SAT-UNSAT iff φ_1 is consistent and (φ_2 is inconsistent or φ_3 is consistent). Assuming w.l.o.g. that φ_1, φ_2 and φ_3 do not share any variables, and p being a new variable, we let $K = \top$, $G = (\varphi_2 \wedge \neg p) \vee (\varphi_2 \wedge \varphi_3)$, $\psi = \varphi_1 \wedge p$.

3.3.2 Weighted goals

Let us start by defining the representation R_{card}, whose input is a multiset of goals $GB = \{G_1, ..., G_n\}$, and $u_{GB}(\vec{x}) = |\{i | \vec{x} \models G_i\}|$. This refinement of R_{brute} considers all-or-nothing but independent goals, and enables compensations. This representation can be easily generalized to R_{wg} by weighting goals with numerical valuations, to be then aggregated (see for instance [21]) :

$GB = \{\langle \alpha_1, G_1 \rangle, ..., \langle \alpha_n, G_n \rangle\}$ and

$$u_{GB}(\vec{x}) = F_1(F_2(\{\alpha_i | \vec{x} \models G_i\}, F_3(\{\alpha_j | \vec{x} \models \neg G_j\}))$$

i.e.$u_{GB}(\vec{x})$ is a function of the weights of goals satisfied by $\vec{x}$ and of the weights of the goals violated by $\vec{x}$

When utility can be considered as a relative notion rather than an absolute one (which means that only differences of utilities between candidates are relevant), it is often assumed that only the violated goals count (see for instance [21, 38]), which leads to $u_{GB}(\vec{x}) = F(\{\alpha_i | \vec{x} \models \neg G_i\})$; F has of course to satisfy a number of desirable properties (see [21])[5]. Usual choices for F are, for instance, sum (weights are then usually called *penalties*) or maximum (which leads to possibilistic logic). The complexity of decision problems for weighted logics can easily be derived from the complexity of distance-based belief merging ([28],[19]). Assuming that the aggregation functions can be computed in polynomial time, we get the following results as byproducts from known results, especially the complexity of distance-based belief merging ([28], and especially [19]): COMPARISON is polynomial; NON-DOMINANCE is coNP-complete; CAND-OPT-SAT is in Δ_2^{P}, and in Θ_2^p when the set $\{u_{GB}(\vec{x}) \mid \vec{x} \in Mod(K)\}$ can be computed in polynomial time (and therefore has a polynomial size); CAND-OPT-SAT is already Θ_2^p-complete for R_{card}.

3.3.3 Prioritized goals

Let us now define $R_{\subseteq}$: given $GB = \{G_1, ..., G_n\}$, let $sat_{GB}(\vec{x}) = \{i | \vec{x} \models G_i\}$ and $nonsat_{GB}(\vec{x}) = \{1, ..., n\} \setminus sat_{GB}(\vec{x})$. The preference relation induced by GB is: $\vec{x} \geq_{GB} \vec{y}$ if and only if $sat_{GB}(\vec{x}) \supseteq sat_{GB}(\vec{y})$

This partial preorder on $\mathcal{X}$ is nothing but the Pareto ordering: indeed, a candidate $\vec{x}$ is optimal for $R_{\subseteq}$ iff there is no candidate $\vec{y} \in Mod(K)$ satisfying all goals that $\vec{x}$ satisfies and at least another one.

This representation can be generalized, considering $GB = \langle \{G_1, ..., G_n\}, > \rangle$ where $>$ is a complete weak order (priority relation) on $\{1, ..., n\}$. The most common choice (see *e.g* [18] [3]), called the *discrimin* ordering when $>$ is ranked (i.e., the complement of $>$ is complete), is the following:
$R_{discrimin}$:

$$\vec{x} \geq_{GB,>}^{discrimin} \vec{y} \text{ if and only if}$$
$$\forall i \in nonsat_{GB}(\vec{x}) \setminus nonsat_{GB}(\vec{y})$$
$$\exists j \in nonsat_{GB}(\vec{y}) \setminus nonsat_{GB}(\vec{x}) \text{ such that } j \geq i.$$

It is not hard to show that, as soon as $\vec{x} \geq_{GB,>} \vec{y}$ can be decided in polynomial time from $>$ – which is obviously the case for $R_{\subseteq}$ and $R_{discrimin}$, then COMPARISON is polynomial and NON-DOMINANCE is in coNPand CAND-OPT-SAT is in Σ_2^p. Moreover, NON-DOMINANCE

[5] However, this assumption is sometimes not made; in this case, positive preference items (goals) have to be formally distinguished from negative ones (constraints); see [4].

is coNP-complete for $R_{\subseteq}$. Lastly, from results in [30], [15] it can be easily derived that CAND-OPT-SAT is already Σ_2^p-complete for $R_{\subseteq}$.

3.3.4 Conditional logics

Each goal G_i is now attached to a *context* C_i : $GB = \{C_1 : G_1, ..., C_n : G_n\}$, and $C : G$ is interpreted as $C \Rightarrow G$ in the simplest conditional logics: $\geq$ being a complete weak order on candidates, we say that $C : G$ is satisfied by $\geq$ if and only if $Max(Mod(C), \geq) \subseteq Mod(G)$; this can be interpreted as *"ideally G if C"* [5]. This constraint does not fully determine the preference relation induced from $\mathcal{D}$. Several possibilities exist:

(1) R_{CL}: $\vec{x} \geq_{GB} \vec{y}$ iff for any $\geq$ satisfying GB we have $\vec{x} \geq \vec{y}$. $\geq_{GB}$ is a partial preoder which is much too weak (i.e., does not enable enough comparisons) to be a good candidate for preference representation[6], as it can be seen on the following example:
Let us consider the propositional language generated by two propositional variables a and b and let $GB = \{\top : a\}$. Then, for any $\vec{x}, \vec{y} \in \{(a, b), (a, \neg b), (\neg a, b), (\neg a, \neg b)\}$, $\vec{x} \geq_{GB} \vec{y}$ holds if and only if $\vec{x} = \vec{y}$ (and therefore $\vec{x} >_{GB} \vec{y}$ never holds).

(2) R_{CLZ}: $\geq^Z_{GB}$ is the preference relation, among those satisfying GB, maximizing preference world by world ([5], page 79), which is equivalent to the Z-completion of GB [33]. The obtained relation $\geq^Z_{GB}$ is much more discriminant (hence much better) than $\geq_{GB}$. Details are omitted. Complexity results for R_{CLZ} can be derived as byproducts from complexity results for conditional knowledge bases ([16], Section 3) : COMPARISON, NON-DOMINANCE and CAND-OPT-SAT are in Δ_2^p.

One drawback of R_{CLZ} is that, as for $\geq_{GB}$), a so-called "drowning effect" occurs (some goals are ignored while they should not); this can be remedied for instance by adding extra constraints expressing that violating a conditional desire induces an explicit utility loss [24]. This principle is further generalized by introducing numerical strengths and polarities in [38] [26]. We did not investigate complexity issues for these approaches.

3.3.5 Ceteris paribus preferences

In this second interpretation of conditional desires, $C : G > G'$ is interpreted by : *all irrelevant things being equal, I prefer G to G', or in more formal terms, "all irrelevant things being equal, I prefer a $C \wedge G \wedge \neg G'$-world to a $C \wedge \neg G \wedge G'$-world.* The definitions proposed in various places [14, 36, 6] differ somehow. We take as a basis the definition of [14], slightly generalized so as to take account for integrity constraints (K).

R_{CP} : the strict order $>_{GB}$ is defined as the transitive closure of the following set of strict inequalities:

$$\left\{ \begin{array}{c|l} & \text{there exists } C : G > G' \text{ in } GB \text{ such that} \\ & 1.\ \vec{x} \models K \wedge C \wedge G \wedge \neg G', \\ \vec{x} > \vec{y} & 2.\ \vec{y} \models K \wedge C \wedge \neg G \wedge G' \\ & 3.\ \vec{x} \text{ and } \vec{y} \text{ coincide on all variables} \\ & \quad \text{not appearing in } G, G' \end{array} \right\}$$

Note that the the definition of the variables included in the "ceteris paribus" scope of $C : G > G'$ could be refined further by considering, instead of the variables not appearing G, G', the variables on which G and G' do not semantically depend ([36, 25]). This is not be considered further in this paper.

For the complexity results we consider not only the general case but also the following two restrictions:

- *GL*: each goal G_i is a conjunction of literals and $G'_i = \neg G_i$ (which corresponds to the definition in [6], restricted to propositional domains);
- *CIG*: the contexts C_i are independent from the goals, *i.e.*, for any goal base $GB = \{C_1 : G_1 > G'_1, \ldots, C_n : G_n > G'_n\}$ we have $Var(C_i) \cap Var(G_j) = Var(C_i) \cap Var(G'_j) = \emptyset$ for any $i, j = 1 \ldots n$. This condition is realized in particular when the contexts C are empty ($C = \top$).

Proposition 2 *The complexity of* COMPARISON, NON-DOMINANCE *and* CAND-OPT-SAT *for* ceteris paribus *preferences is reported on the following table (where $-c$ means "-complete").*

	general case	*GL*	*CIG*	*GL+ CIG*
COMPAR.	PSPACE-c	PSPACE-c	NP-c	P
NON-DOM.	coNP-c	P	coNP-c	P
C.-O.-SAT	Σ_2^p-c	NP-c	Σ_2^p-c	NP-c

Sketch of proof:
- the somewhat surprising *PSPACE*-completeness of COMPARISON comes from reductions to and from plan existence with deterministic actions represented in STRIPS (STRIPS-PLAN-EXISTENCE); the latter problem has been shown PSPACE-complete in [9] (PSPACE-hardness is due to the fact that plans can be exponentially long). We give a reduction from STRIPS-PLAN-EXISTENCE to the *GL* restriction of COMPARISON (the reduction in the converse direction is in [6]): from any action α with the consistent set of literals *pre* as precondition and the consistent set of literals *post* as postcondition, we draw the *ceteris paribus* preference item

[6] Just to mention it, note that complexity for R_{CL} can be derived from complexity results for conditional logics [17] and is surprisingly low : CAND-OPT-SAT is NP-complete and NON-DOMINANCE coNP-complete.

$$pre \setminus NEG(post) : post > (pre \cap NEG(post))$$

where $pre \setminus NEG(post)$ the conjunction of all literals that are in pre and whose negation is not in $post$, and $pre \cap NEG(post)$ the conjunction of all literals that are in pre and whose negation is in $post$. The existence of a plan leading from an initial state s to a final state s' is equivalent to a strict preference of s' to s with respect to the set of c.p. preference items corresponding to the actions. Now, under the *CIG* assumption, preference items need to be "fired" only once, therefore, if $x > y$ then there is a "path" of polynomial length between x and y, hence the membership to NP.

- for NON-DOMINANCE, membership to coNPcomes from the following property: $\vec{x}$ is non-dominated if and only if for each $C_i : G_i > G'_i$ in GB such that $\vec{x} \models K \wedge C_i \wedge \neg G_i \wedge G'_i$, there is no $\vec{y}$ such that $\vec{y} \models K \wedge C_i \wedge G_i \wedge \neg G'_i$ and such that $\vec{x}$ and $\vec{y}$ coincide on all variables that do not appear in G_i and G'_i. coNP-hardness for the *CIG* restriction comes from the following reduction from UNSAT: let $K = \top$ and $GB = \{\top : \varphi \wedge new > \neg(\varphi \wedge new)\}$, where $new \notin Var(\varphi)$, and let $\vec{x}$ such that $\vec{x} \models \neg new$. Then $\vec{x}$ is undominated if and only if φ is unsatisfiable. Under *GL*, NON-DOMINANCE is equivalent to: for all $C_i : G_i > \neg G_i$ of GB we have $\vec{x} \models C_i \Rightarrow G_i$, hence the membership to P.

- for CAND-OPT-SAT, membership to, respectively, Σ_2^p and NPcomes easily from the membership of NON-DOMINANCE to, respectively, coNPand p. Σ_2^p-hardness under the *CIG* restriction comes from a reduction from base revision (or inference from maximal consistent subsets), using the fact that $\vec{x}$ is undominated for $K = \top$ and $GB = \{\top : \phi_1 > \neg\phi_1, \ldots, \top : \phi_n > \neg\phi_n\}$ if and only if $\vec{x}$ satisfies a maximal consistent subset of $\{\phi_1, \ldots, \phi_n\}$. Under *GL*+*CIG*, NP-hardness comes from the fact that when $K = \top$ and $GB = \emptyset$, there is an undominated candidate satisfying 'φ if and only if φ is satisfiable.

The surprising point is that, for the general case, the comparison problem is much more difficult than the other ones (in contrast with other representation languages). This is due to possible exponential "preference paths". Note that the connection between the comparison problem in ceteris paribus nets and STRIPS planning was first remarked in [6].

3.3.6 Distances

Let d be a (pseudo-)distance on $\mathcal{X}$, *i.e.*, a function from $Mod(K) \times Mod(K)$ to $\mathbb{N}$ such that (i) $d(\vec{x}, \vec{y}) = 0$ if and only if $\vec{x} = \vec{y}$ and (ii) $d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x})$. Distance-based logical representations of preference [21, 22, 20] are based on the intuitive idea that, when expressing a goal G, then ideally, $\vec{x}$ must satisfy G, and when it is no longer the case, then, the "further" $\vec{x}$ is from G, the less preferred $\vec{x}$.

Formally, a pair $\langle\{G\}, d\rangle$, where G is a propositional formula and d a pseudo-distance, induces the utility function

$$u_{GB}(\vec{x}) = -d(\vec{x}, G) = -\min_{\vec{y} \models G} d(\vec{x}, \vec{y})$$

When d is computable in polynomial time, one easily derives from the literature on the complexity of belief revision [15] [31] and of distance-based belief merging [19] that COMPARISON, NON-DOMINANCE and CAND-OPT-SAT are in Δ_2^p (and, in particular, Θ_2^p-complete when d is the Hamming distance).

This principle can be generalized by considering a set of goals, each of which is associated to a pseudo-distance: $GB = \{\langle G_1, d_1\rangle, ..., \langle G_n, d_n\rangle\}$, and $u_{GB}(\vec{x}) = F(d_1(\vec{x}, G_1), ..., d_n(\vec{x}, G_n))$, where F is an aggregation function. This has no strong impact on complexity.

3.3.7 Discussion

These results have a value of their own, since they enable a first comparison of propositional preference representation from a computational point of view[7]. Now, the complexity of the key problems COMPARISON and NON-DOMINANCE also have a strong impact on the rest of the paper, i.e., to the question whether these languages are suitable for combinatorial vote. Interesting languages are then (in principle) those for which either COMPARISON or NON-DOMINANCE is polynomial, namely, in addition to the brute representation: weighted or prioritized goals, and specific restrictions of ceteris paribus desires.

The problem with weighted goals is the well-known difficulty of eliciting numerical preferences from agents; as to prioritized goals, their lack of expressivity (no compensation allowed between goals) somewhat limits their range of use. "Ceteris paribus" preferences are interesting from a cognitive point of view, being rather close to human intuition and rather easy to elicit [6] [12]. However, they have a high computational complexity in the general case, and furthermore, from the point of view of expressivity, they are not very discriminant[8].

[7] Note however that what is lacking here is a study of their *representational complexity* [11] which would assess precisely their concision power (this is an interesting topic for further research).

[8] However, see [7] for an introduction of numerical utilities in ceteris paribus networks that remedies this problem, but on the other hand reintroduces the cognitive problem inherent to numerical utilities ...

4 Combinatorial vote

We now assume that a group of p agents (or voters) have to come up with a common assignment of values to dependent variables. A combinatorial vote problem consists of two steps: first, the agents express their preferences within a fixed (and common) representation language R, and second, one or several optimal candidate(s) is (are) determined automatically, using a fixed *voting rule*.

A *preference profile* consists of a preference structure (cardinal or ordinal) for each of the voters. Social choice theory (see [29] for a good introduction) defines a voting rule V as a function mapping every preference profile P to an elected candidate or a subset of $\mathcal{X}$, called the set of *elected candidates*[9]. The set of elected candidates, given a preference profile P and a voting rule V, will be denoted by $Select_V(P)$.

Research in social choice theory focuses on the properties of voting rules (and establishes for instance impossibility results) but does not care about the representation language and computational complexity, since the preferences are supposed to be expressed explicitly. However, in combinatorial vote, the representation language is an important parameter, and the computational properties of a voting rule will both depend of the rule V itself and of the representation language R chosen.

For any representation language R, one defines a *p-voters R-profile* as a collection $\mathcal{B} = \langle GB_1, ..., GB_p\rangle$ of goal bases (one for each voter), expressed in the language R, generating a profile $P = Induce_R(\mathcal{B})$, *i.e.*, depending whether preferences are cardinal or ordinal: $Induce_R(\mathcal{B}) = \{u_{GB_1}, ..., u_{GB_p}\}$ or $Induce_R(\mathcal{B}) = \{\geq_{GB_1}, ..., \geq_{GB_p}\}$.

We will now look at several voting rules that are well-known in the social choice community, and discuss these with respect to two criteria:

1. *relevance* for combinatorial vote (i.e., does the voting rule still mean something when the set of candidates has a combinatorial structure?);
2. *computational complexity.* Let V be a voting rule and R a representation language; we consider the following decision problems:

- AMONG-WINNERS$_{V,R}$: given $\mathcal{B} = \langle GB_1, ..., GB_p\rangle$ and a candidate $\vec{x}$, determine whether $\vec{x} \in Select_V(Induce_R(\mathcal{B}))$;
- ELECT-SAT$_{V,R}$: given $\mathcal{B} = \langle GB_1, ..., GB_p\rangle$ and a formula ψ, determine whether there is a candidate in $Select_V(Induce_R(\mathcal{B}))$ satisfying ψ.

The complexity of these problems depends a lot on the complexity of COMPARISON and NON-DOMINANCE for R. It has been seen that for some of these languages, comparison and/or non-dominance is polynomial. For the rest of the paper, we focus on these.

4.1 Aggregative rules for cardinal preferences

When preference is cardinal, i.e., $P = \langle u_1, ..., u_n\rangle$, the simplest way consists in "aggregating before comparing" (see [13]), namely: for each $\vec{x}$, the scores $u_i(\vec{x})$ are synthesized by an aggregation function $* : \mathbb{R}^p \to \mathbb{R}$ into a so-called social welfare function $u_{\mathcal{B}}(\vec{x}) = *\{u_i(\vec{x}) | i = 1 \ldots p\}$ reflecting the satisfaction of the group (see [29]). $Select_{agreg(*)}(P)$ is then the set of candidates maximizing $u_{\mathcal{B}}(\vec{x})$. Classically, the function $*$ is commutative (for guaranteeing the anonymousness property) and non-decreasing.

In Section 3 it has been seen that the languages that lead to cardinal preference structures are weighted goals and distance-equipped goals (note that for the latter the polynomiality of comparison does not hold except if some restrictions are made).

As a consequence of Section 3 and [19] it can be shown that, provided that $u_i(\vec{x})$ is polynomially computable from $\mathcal{B}$, and $*$ is polynomially computable, AMONG-WINNERS$_{agreg(*),R}$ is in coNP and ELECT-SAT$_{agreg(*),R}$ is in Δ_2^{P}.

The problem with these methods are, again, the difficulty to elicit a numerical utility function. From now on we focus on rules whose input is an ordinal structure (which may possibly have been induced from a cardinal structure).

4.2 Scoring rules

4.2.1 General principle

Scoring rules consist in translating the preference relation $\geq_i$ of voters into score functions $s_i(\vec{x})$, such that the score $s_i(\vec{x})$ of a candidate $\vec{x}$ with respect to voter i is a function of its position in the relation $\geq_i$, *only.*

[9] For the sake of simplicity we will not distinguish between what is usually called a voting rule, which selects a unique candidate, and a voting correspondance, which selects a subset of candidates; we will use the terminology "voting rule" in all cases. The rules that we will consider in the rest of the paper are strictly speaking correspondances, from which a standard rule can be defined by tie-breaking ex-aequos either by giving up neutrality or by randomly choosing one of the elected candidates.

The global score $s(\vec{x})$ of $\vec{x}$ is then computed by summing up all scores $s_i(\vec{x})$ for $i = 1$ to n. This is particularly clear in the case where $\geq$ is a *complete order* (i.e., no ex-aequos, no incomparabilities), $r_i(\vec{x})$ is the rank of $\vec{x}$ in $\leq_i$ (1 if it is the favorite candidate for i, 2 if it is the second favorite etc.), then $s_i(\vec{x})$ is a non-increasing function of $r_i(\vec{x})$.

The most common choice is the *Borda rule.* In the case of a complete order, the Borda score is defined by $s_i(\vec{x}) = |\mathcal{X}| - r_i(\vec{x})$, which can be generalized by: let $s_i(\vec{x})$ be defined from $>_i$ as the number of candidates that do not dominate $\vec{x}$, i.e.,

$$s_i(\vec{x}) = |Mod(K)| - |\{\vec{x}' \in Mod(K) | \vec{x}' >_i \vec{x}\}|$$

One may think that scoring rules translate purely ordinal preference into a numerical utility function in a rather arbitrary way, which is true; but on the other hand, these rules satisfy several desirable properties such as monotony and some forms of strategyproofnesses that are not satisfied by other rules (see [29]).

However, in the context of combinatorial vote, scoring rules present a major drawback: being based on a counting of candidates, they are extremely syntax-sensitive, *i.e.*, to the choice of propositional symbols used for representing the problem in hand, and moreover it really makes little sense to use a rule such as Borda on a set of candidates of cardinality 2^n as soon as n does not have a reasonable size. Therefore, most scoring rules are much more arbitrary in the context of combinatorial vote than in classical vote contexts. As if it were not enough, their computational complexity, in most cases, is really high, since candidate counting is required. One may thus advance the following general principle that *a voting rule that requires a counting of candidates is not suitable to combinatorial vote.* This rules out almost all scoring rules; however, a few of them (very specific) avoid this counting, such as the plurality and the veto rules, that we study now.

4.2.2 Plurality and veto

The *plurality* rule is the scoring rule obtained by taking $s_i(\vec{x}) = 1$ if and only if $\vec{x}$ is non-dominated for $\geq_i$, *i.e.*, iff there is no $\vec{y} \models K$ such that $\vec{y} >_i \vec{x}$. $Select_{\text{plurality}}(P)$ is therefore the set of candidates maximizing the number of voters for whom $\vec{x}$ is non-dominated (or, in the simple case of a complete order, the number agents that rank $\vec{x}$ in first position). It is known to be less satisfactory than many other scoring rules, but on the other hand it is syntax-insensitive and does not require any counting of candidates , therefore it is relevant for combinatorial vote. We now look at its complexity, when the preference representation language R varies.

Proposition 3 *Let R be a language for which* NON-DOMINANCE *is in* NP*or in* coNP*.*

1. AMONG-WINNERS$_{plurality,R}$ *is in* Θ_2^p*;*
2. ELECT-SAT$_{plurality,R}$ *is in* Σ_2^p*;*
3. *for $R = R_{brute}$,* AMONG-WINNERS$_{plurality,R_{brute}}$ *is* coNP*-complete and* ELECT-SAT$_{plurality,R_{brute}}$ *is* Θ_2^p*-complete.*

Proof sketch: (1) is easy, once noticed that $s(\vec{x})$ is bounded by n. (2) comes easily from (1). As to (3), let $\mathcal{B} = \{GB_1, ..., GB_n\}$ with $GB_i = \{G_i\}$. $s(\vec{x})$ is maximal if and only if the number of i such that $\vec{x} \models G_i$ is maximal, hence the membership of AMONG-WINNERS$_{\text{plurality},R_{brute}}$ to coNP. coNP-hardness is a consequence of coNP-hardness of NON-DOMINANCE. As to ELECT-SAT$_{\text{plurality},R_{brute}}$, both membership and hardness come straightforwardly from cardinality-maximizing base revision [31].

(3) gives a lower complexity bound since all representation languages considered in this paper generalize R_{brute}. The upper bound for ELECT-SAT is reached for instance for $R = R_{\subseteq}$: ELECT-SAT$_{\text{plurality},R_{\subseteq}}$ is Σ_2^p-complete. For R_{wg}, ELECT-SAT is in Δ_2^p.

The *veto* rule is obtained by letting $s_i(\vec{x}) = 1$ if and only if there is at least a candidate $\vec{y}$ such that $\vec{x} >_{G_i} \vec{y}$. When $>_{G_i}$ is a complete order, this comes down to count the number of voters who do not for $\geq_i$, *i.e.*, iff there is no $\vec{y} \models K$ such that $\vec{y} >_i \vec{x}$. $Select_{\text{plurality}}(P)$ is therefore the set of candidates maximizing the number of voters for whom $\vec{x}$ is non-dominated (or, in the simple case of a complete order, the number of voters who rank $\vec{x}$ in first position). Complexity results for the veto rule are similar to those for the plurality rule.

Example 1 *Let us fix $R = R_{card}$, $\mathcal{L}$ the language generated by the propositional variables $\{a, b, c\}$ and let the number of voters be 3. Let $\mathcal{B}$ be the following 4-uple of goal bases:*

$$\mathcal{B} = \langle GB_1, GB_2, GB_3, GB_4 \rangle$$

where

- $GB_1 = \{a, a \vee b\}$*;*
- $GB_2 = \{\neg a, b \wedge c, b \vee c\}$*;*
- $GB_3 = \{\neg b, a \Leftrightarrow \neg c\}$*;*

The preference orderings $\geq_{G_1}$, $\geq_{G_2}$ and $\geq_{G_1}$ are:

$$\begin{array}{ll} & (a,b,c),(a,b,\neg c),(a,\neg b,c),(a,\neg b,\neg c) \\ >_{G_1} & (\neg a,b,c),(\neg a,b,\neg c) \\ >_{G_1} & (\neg a,\neg b,c),(\neg a,\neg b,\neg c) \end{array}$$

$$\begin{array}{ll} & (\neg a,b,c) \\ >_{G_2} & (a,b,c),(\neg a,b,\neg c),(\neg a,\neg b,c),(\neg a,\neg b,\neg c) \\ >_{G_2} & (a,b,\neg c),(a,\neg b,c) \\ >_{G_2} & (a,\neg b,\neg c) \end{array}$$

$$\begin{array}{ll} & (a,\neg b,\neg c),(\neg a,\neg b,c) \\ >_{G_3} & (a,b,\neg c),(\neg a,b,c),(a,\neg b,c),(\neg a,\neg b,\neg c) \\ >_{G_3} & (a,b,c),(\neg a,b,\neg c) \end{array}$$

Candidates written on a same line are indifferent: for instance, for G_1, we have $(a,b,c) \sim_{G_1} (a,b,\neg c) \sim_{G_1} (a,\neg b,c) \sim_{G_1} (a,\neg b,\neg c)$, any of these four candidates being strictly preferred to both $(\neg a,b,c)$ and $(\neg a,b,\neg c)$ and so on.

The above table shows the Borda, plurality and veto scores of all candidates.

	s_1^B	s_2^B	s_3^B	s^B	s^p	s^v
(a,b,c)	8	7	2	17	1	2
$(a,b,\neg c)$	8	3	6	17	1	$\boxed{3}$
$(a,\neg b,c)$	8	3	6	17	1	$\boxed{3}$
$(a,\neg b,\neg c)$	8	1	8	17	$\boxed{2}$	2
$(\neg a,b,c)$	4	8	6	$\boxed{18}$	1	$\boxed{3}$
$(\neg a,b,\neg c)$	4	7	2	13	0	2
$(\neg a,\neg b,c)$	2	7	8	17	1	2
$(\neg a,\neg b,\neg c)$	2	7	6	15	0	2

Framed scores correspond to elected candidates.

4.3 Condorcet-consistent rules

4.3.1 Condorcet winner

A *Condorcet winner* (CW) for a profile P is a candidate $\vec{x}$ such that for any candidate $\vec{y} \neq \vec{x}$, there are strictly more agents preferring $\vec{x}$ to $\vec{y}$ than agents preferring $\vec{y}$ to $\vec{x}$, *i.e.*, $\forall \vec{y} \in Mod(K), |\{i | \vec{x} >_i \vec{y}\}| > |\{j | \vec{y} >_i \vec{x}\}|$.[10] This notion comes back to 1785 and it is known since then that there are profiles for which there is no CW. Importantly, when there exists a CW, it is unique.

A first problem is that the usual definition of a CW is not well-suited to combinatorial vote. This can be seen easily in the case where $R = R_{brute}$ (and is easily be generalized to more sophisticated representation languages). We easily get that $\vec{x}$ is the Condorcet winner for $\mathcal{B} = \langle G_1, ..., G_n \rangle$ if and only if the number of i such that $\vec{x} \models G_i$ is maximal *and there is no other interpretation* than $\vec{x}$ maximizing this number of voters whose goals is satisfied. This has the consequence that there almost *never* exists a Condorcet winner in practical cases, because the number of candidates being much larger than the number of voters, there are generally pairs of worlds upon which all voters are indifferent. For instance, let $\mathcal{L}$ be the propositional language generated by the variables $\{a,b,c\}$; let the numbers of voters be 4, $R = R_{brute}$, and $GB = \langle a, a, \neg a, b \rangle$. We get that both (a,b,c) and $(a,b,\neg c)$ beat any of the other six candidates by a majority of voters, however neither of both beats the other one by a majority of voters, therefore there is no Condorcet winner. Intuitively speaking, we would like not to distinguish $\vec{x}$ and $\vec{y}$ and to have them both as winners. For this purpose, we now define *weak Condorcet winners.*

Let us first define the equivalence relation $\sim$ on $\mathcal{X}$ by $\vec{x} \sim \vec{y}$ if and only if $\vec{x} \sim_i \vec{y}$ for every voter i (unanimous indifference between $\vec{x}$ and $\vec{y}$). A *weak Condorcet winner* (WCW) for a profile P is a candidate $\vec{x}$ such that for any other candidate $\vec{y}$, either there are strictly more agents preferring $\vec{x}$ to $\vec{y}$, or there is an unanimous indifference between $\vec{x}$ and $\vec{y}$. An equivalent definition consists in first quotienting $\mathcal{X}$ by $\sim$, and then looking for Condorcet winners in $\mathcal{X}_{/\sim}$; a candidate $\vec{x} \in \mathcal{X}$ is a WCW if and only if its equivalence class for $\sim$ is *the* CW in $\mathcal{X}_{/\sim}$.

Obviously, a CW is a WCW. Noticeably, there are profiles for which there are several WCWs (in which case there is no CW), profiles for which there is a single WCW (which is then the CW) and profiles for which there is no WCW.

When P is represented explicitly, the existence of a CW or of a WCW can be determined in quadratic time. This is much more complex when preference are represented in a succint language:

Proposition 4 *Let R be a language for which* COMPARISON *is polynomial.*

1. *determining whether $\vec{x}$ is a Condorcet winner (resp. a weak Condorcet winner) for $\mathcal{B}$ w.r.t. R is in* coNP. *It is* coNP*-complete for $R = R_{brute}$ (and a fortiori for more refined languages).*

2. *determining whether there exists a Condorcet winner (resp. a weak Condorcet winner) for $\mathcal{B}$ w.r.t. R is in Σ_2^p. For $R = R_{brute}$, the existence of a Condorcet winner is in Θ_2^p,* NP*-hard and* coNP-

[10] In particular, when $\geq$ is a complete order, $\vec{x}$ is a Condorcet winner if and only if it is preferred to any other candidate by a majority of voters.

hard)[11].

Sketch of proof: all membership proofs are easy, once noticed that for any two candidates, $\vec{x}$, $\vec{y}$, it can be decided in polynomial time whether $\vec{x} >_i \vec{y}$ for every voter i, and therefore it can be decided in polynomial time whether $|\{i|\vec{x} >_i \vec{y}\}| > |\{i|\vec{y} >_i \vec{x}\}|$, and similarly, $\vec{x} \sim \vec{y}$ can also be decided in polynomial time. For $R = R_{brute}$, we get that $\vec{x}$ is a WCW for $\mathcal{B} = \{G_1, ..., G_n\}$ if and only if $\vec{x}$ satisfies a maximal number of G_i's and $\vec{x}$ is a CW if furthermore $\vec{x}$ is the *unique* candidate maximizing the number of G_i's satisfied. This leads to three conclusions: (a) when $R = R_{brute}$, there is always a WCW, (b) checking that $\vec{x}$ is the CW is in coNP, and (c) the existence of a CW is in Θ_2^p[12].

Since the existence of a CW is not guaranteed, several voting rules have been proposed in social choice theory, which guarantee that each time there exists a CW, it is elected. These rules are said *Condorcet-consistent.* Considering now weak Condorcet winners, a Condorcet-consistent rule elects the set of WCWs when the latter is not empty.

Each Condorcet-consistent rule is thus characterized by its results in the cases where there are no WCWs.

Two simple Condorcet-consistent rules have received a particular interest, namely the *Copeland* and the *Simpson* rules. The *Copeland rule* requires to count candidates and, as said above, it is not well-suited to combinatorial vote; the Simpson rule is more interesting.

4.3.2 The Simpson rule

Let P be a profile, $\vec{x}$ and $\vec{y}$ two candidates. $N(\vec{x}, \vec{y})$ is defined as the number of voters who prefer $\vec{x}$ to $\vec{y}$, *i.e.*, $N(\vec{x}, \vec{y}) = |\{i|\vec{x} >_i \vec{y}\}|$. The *Simpson score* $S(x)$ is defined by $S(\vec{x}) = \min_{\vec{y} \neq \vec{x}} N(\vec{x}, \vec{y}) - N(\vec{y}, \vec{x})$, and $Select_{Simpson}(P)$ is the set of candidates maximizing $S(\vec{x})$ (the Simpson winners). It is easy to check that when there exists at most one $\vec{x}$ with a strictly positive score, and when such a candidate exists, it is the CW: the Simpson rule is Condorcet-consistent.

Proposition 5 *Let R be a language for which comparison is polynomial.*

1. AMONG-WINNERS$_{Simpson,R}$ *is in* Θ_2^p.
2. ELECT-SAT$_{Simpson,R}$ *in* Σ_2^p.
3. *For* $R = R_{brute}$*:* AMONG-WINNERS$_{Simpson,R_{brute}}$ *is* coNP*-complete and* ELECT-SAT$_{Simpson,R_{brute}}$ *is* Θ_2^p*-complete.*

Sketch of proof: the first point comes easily from the fact that $-n \leq S(\vec{x}) \leq n$ (n being the number of voters), making possible a binary search over a set of $2.n$ possible values. The second point is then derived easily from the first one. For the last point, ELECT-SAT-hardness is obtained by a straightforward reduction from cardinality-maximizing base revision [31].

Example 1 (continued).

On the above table we give the values of $N(\vec{x}, \vec{y})$ ($\vec{x}$ in column and $\vec{y}$ in row) and then the Simpson scores. We represent candidates by vectors of digits: (a, b, c) by 111, $(a, b, \neg c)$ by 110 and so on.

$\downarrow \vec{y}$ $\vec{x} \rightarrow$	111	110	101	100	011	010	001	000
111	/	0	0	0	0	−1	0	0
110	0	/	0	0	0	−1	1	0
101	0	0	/	0	0	−1	1	0
100	0	0	0	/	−1	−1	0	−1
011	0	0	0	1	/	−2	−1	−2
010	1	1	1	1	2	/	0	0
001	0	−1	−1	0	1	0	/	−1
000	0	0	0	1	2	0	1	/
$S(\vec{x}) \rightarrow$	0	−1	−1	0	−1	−2	−1	−2

There are two Simpson winners, namely (a, b, c) and $(a, \neg b, \neg c)$. None of them is a weak Condorcet winner.

4.4 Discussion

The conclusions are not simple, because several parameters have to be taken into account. From a strictly computational point of view, aggregative rules are probably easier to implement than scoring rules or Condorcet-consistent rules, but they require a numerical input which is often hard to obtain, and they are thus considered little efficient from a cognitive point of view. Note that our complexity results for the Simpson rule and for the plurality rule coincide[13], namely, its complexity is not above the second level of the polynomial hierarchy. Therefore it seems that the Simpson rule, considered more acceptable than the plurality rule [29], could be considered as an acceptable tradeoff.

[11] and the existence of a weak Condorcet winner is trivial (see the sketch of proof).

[12] Note that I could not manage to prove Θ_2^p-hardness, which nevertheless I strongly believe to hold. Similarly, I strongly believe that when R varies, then the upper bound, i.e., Σ_2^p-completeness, is reached for simple languages such as R_{card}, but again I could not find a proof.

[13] This does not mean, however, that the complexity of both rules would coincide for all R.

5 Future directions

5.1 Electronic vote

A first possible application of combinatorial vote is *electronic vote* concerning decisions about several dependent variables, that have to be taken in small organizations (small companies, laboratories, recruiting committees etc.). When the set of candidates has little or no combinatorial structure (e.g., choose a person to be recruited on a position), or when the variables are independent (or almost) from each other, preferences can be aggregated manually, but this is not the case as soon as the former has a strong combinatorial structure[14].

In this paper we only briefly mentioned the key issue of automated *preference elicitation* (i.e., how to interact with a voter so as to obtain her preference relation). This issue, which has received some attention in the last years, is a necessary upstream task for combinatorial vote and relationships between both problems should be studied further. As well, identifying preferential independences between some variables for a given voter (see [2]) is extremely relevant in this context.

5.2 Partial vote and negociation

In some situations, it is not compulsory to assign *all* variables at once: one may try to localize conflicts and look for a partial decision (an assignment of a subset of the variables of interest) that is both as consensual as possible and as complete as possible; this decision can then be communicated to the voters, who can then update their preferences about the remaining variables (possibly after a negociation phase which is out of our concern), and the process can then be iterated until all variables have been assigned. This process requires that voting rules for combinatorial vote be adapted; this is an issue for further work.

Other less obvious possible applications of combinatorial vote are fair division problems (see [27]) or combinatorial auctions. This is left for further research.

[14] This can be seen for instance on the following real-world problem, concerning a decision to be taken by a recruiting committee: when not a single, but k individuals (out of n) can be recruited, the space of possible decisions has a combinatorial structure: a "candidate" is no longer an individual but a set of k individuals. The problem can be solved manually only if the dependencies between individuals are ignored, which means that the voters cannot express *correlations* between individuals, as for instance: "In the first place I prefer recruiting A, then B, then C, but since A and B work on similar subjects whereas C works on a complementary subject, the joint recruiting of A and C, or even of B and C, suits me better than the joint recruting of A and B." Allowing for the expression of such preference needs a sophisticated language for preference representation such as those studied in this article, and the aggregation of such preference is clearly a combinatorial vote problem.

Acknowledgements: I would like to thank Sébastien Konieczny for a careful reading of a preliminary version of the paper, as well as Thomas Eiter and Pierre Marquis for discussions about Θ_2^p-hardness.

References

[1] F. Bacchus and A. Grove. Graphical models for preference and utility. *Proceedings of the 11th Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, 3-10.

[2] F. Bacchus and A. Grove. Utility independence in a qualitative decision theory. *Proc. of KR'96*, 542-552.

[3] S. Benferhat, C. Cayrol, D. Dubois, J. Lang and H. Prade. Inconsistency management and prioritized syntax-based entailment. *Proceedings of IJCAI'93*, 640-647.

[4] S. Benferhat, D. Dubois, S. Kaci and H. Prade. Bipolar representation and fusion of preferences in the possibilistic logic framework. *Proceedings of KR-2002* (these Proceedings).

[5] C. Boutilier. Towards a logic of qualitative decision theory. *Proceedings of KR'94*, p. 75-86, 1994.

[6] C. Boutilier, R. Brafman, H. Hoos and D. Poole. Reasoning with conditional *ceteris paribus* statements. *Proceedings of the 15th Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, 71-80.

[7] C. Boutilier, F. Bacchus and R. Brafman. UCP-networks: a directed graphical representation of conditional utilities. *Proceedings of UAI-2001*, 56-64.

[8] C. Boutilier and H. Hoos. Bidding languages for combinatorial auctions. *Proceedings if IJCAI'01*, 1211-1217.

[9] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69, 161-204, 1994.

[10] J.-Y. Cai, T. Gundermann, J. Hartmanis, L.A. Hemachandra, V. Sewelson, K. W. Wagner and

G. Wechsung. The Boolean hierarchy I: structural properties. *SIAM Journ. on Computing*, 17 (6), 1232-1252, 1988.

[11] M. Cadoli, F. Donini, P. Liberatore and M. Schaerf. Comparing space efficiency of propositional knowledge representation formalisms. *Proc. of KR'96*, 364-373.

[12] C. Domshlak, R.I. Brafman and S. Eyal Shimony. Preference-Based Configuration of Web Page Content. *Proc. of IJCAI'2001.*

[13] D. Dubois, H. Fargier and P. Perny. On the limits of ordinality in decision making. In *Proceedings of KR-02* (these Proceedings).

[14] J. Doyle and M. P. Wellman. Preferential semantics for goals. Proceedings of *AAAI'91*, p. 698-703, 1991.

[15] T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence* 57, 227-270, 1992.

[16] T. Eiter and T. Lukasiewicz. Complexity results for default reasoning from conditional knowledge bases. *Proceedings of KR-2000*, 62-73.

[17] N. Friedman and J. Halpern. On the complexity of conditional logics. *Proceedings of KR'94*, 202-213.

[18] H. Geffner. *Default Reasoning: Causal and Conditional Theories*. MIT Press, 1992.

[19] S. Konieczny, J. Lang et P. Marquis. Distance-based merging: a general framework and some complexity results. In *Proceedings of KR-02* (these Proceedings).

[20] S. Konieczny and R. Pino-Pérez, On the logic of merging. *Proceedings of KR'98*, 488-498, 1998.

[21] C. Lafage, J. Lang. Logical representation of preferences for group decision making. *Proceedings of KR'2000*, 457-468, 2000.

[22] C. Lafage and J. Lang. Propositional distances and preference representation. *Proceedings of ECSQARU-2001, LNAI 2143*, Springer-Verlag, 48-59, 2001.

[23] P. La Mura and Y. Shoham. Expected utility networks. *Proceedings of UAI'99*, 366-373, 1999.

[24] J. Lang. Conditional desires and utilities - an alternative approach to qualitative decision theory. *Proceedings of ECAI'96*, 318-322, 1996.

[25] J. Lang and P. Marquis. Complexity results for independence and definability in propositional logic, *Proceedings of KR'98*, 356-367.

[26] J. Lang, L. van der Torre and E. Weydert. Utilitarian desires. To appear in *International Journal on Autonomous Agents and Multi-Agent Systems*, 2002.

[27] M. Lemaitre, G. Verfaillie, and N. Bataille. Exploiting a common property resource under a fairness constraint: a case study. *Proceedings of IJCAI'99*, 206-211.

[28] P. Liberatore and M. Schaerf. BReLS: a system for the integration of knowledge bases. *Proceedings of KR-2000*, 145-152.

[29] H. Moulin, *Axioms of Cooperative Decision Making*, Cambridge University Press, 1988.

[30] B. Nebel. Belief revision and default reasoning: syntax-based approaches. *Proceedings of KR-91*, 417-428.

[31] B. Nebel, How hard is it to revise a knowledge base? In D.M. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 3, 77-145, Kluwer Academics.

[32] C. H. Papadimitriou, *Computational complexity*, Addison-Wesley Publishing Company, 1994.

[33] J. Pearl. System Z: a natural ordering of defaults with tractable applications to default reasoning. *Proceedings of TARK'90*, 121-135.

[34] T. Sandholm. An Algorithm for Optimal Winner Determination in Combinatorial Auctions, *Proc. of IJCAI'99*, 542-547.

[35] T. Schiex, H. Fargier and G. Verfaillie. Valued constraint satisfaction problems: hard and easy problems, *Proceedings of IJCAI'95*, 631-637, 1995.

[36] S.-W. Tan and J. Pearl, Specification and evaluation of preferences for planning under uncertainty, *Proc. KR'94.*

[37] R. H. Thomason. Desires and defaults: a framework for planning with inferred goals. *Proceedings of KR-2000*, 702-713.

[38] L. van der Torre and E. Weydert. Parameters for utilitarian desires in a qualitative decision theory, *Applied Intelligence*, 14 (3), 285-302, 2001.

[39] M. Wooldridge and S. Parsons. Languages for negociation. *Proceedings of ECAI-2000*, 393-397.