Logical preference representation and combinatorial vote *

Jérôme Lang (lang@irit.fr)
IRIT
Université Paul Sabatier
31062 Toulouse Cedex
France

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.

* This paper is a revised and extended version of the paper entitled "From preference representation to combinatorial vote", *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002)*, pages 277-288, Morgan Kaufmann, 2002.

1. Introduction

For a few years, AI researchers have been concerned with interaction, cooperation or negotiation 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 succinct representation of the description of the problem, without having to enumerate a prohibitive number of states. Languages based on propositional logic have been proposed recently for some multi-agent problems, for instance for combinatorial auctions [8, 46, 43] and automated negotiation [52].

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 negotiation). Formulated as such, this can be identified as a *vote problem*. Vote problems have been investigated by researchers in social choice theory (see for instance [40] for an overview) who have studied extensively all properties of various families of vote rules, up to an important detail: candidates are supposed to be listed explicitly (typically, they are individuals 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 exponential in the number of variables and it is therefore not reasonable asking the voters to rank all alternatives or evaluate them on a utility scale.

Consider for example that the voters have to agree on a common menu to be composed of 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 vote 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 AI community; they are often build upon propositional logic, but not always (see for instance utility networks [1, 33] or valued constraint satisfaction [47] – however in this paper we restrict the study to logical approaches); 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 in the literature, 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 vote rule. In Section 5 we evoke other important issues in combinatorial vote and we mention some possible 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. If $\vec{x}$ is an interpretation then $for(\vec{x})$ denotes the formula – unique up to logical equivalence – such that $Mod(for(\vec{x})) = \{\vec{x}\}$.

The complexity results we give in this paper refer to some complexity classes which we now briefly recall (see [44] 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.

- $\mathsf{BH}_2 = \mathsf{(2)}$ is the class of all languages L such that $L = L_1 \cap L_2$, where L_1 is in NP and L_2 in coNP. The canonical BH_2-complete problem is SAT-UNSAT: $\langle \varphi_1, \varphi_2 \rangle$ is a positive instance of SAT-UNSAT if and only if φ_1 is satisfiable and φ_2 is unsatisfiable. $\mathsf{NP(3)}$ 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 satisfiable and (φ_2 is unsatisfiable or φ_3 is satisfiable). Theses classes are members of the *Boolean hierarchy* [13].

- $\Delta_2^p = \mathsf{P}^{\mathsf{NP}}$ 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 an NP problem 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.

- $\Theta_3^p = \Delta_3^p[\mathcal{O}(log\ n)]$ is the class of all languages that can be recognized in polynomial time by a Turing machine using a number of Σ_2^p oracles bounded by a logarithmic function of the size of the input data.

- 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:

$$\mathsf{NP}, \mathsf{coNP} \subseteq \mathsf{BH}_2 \subseteq \mathsf{NP(3)} \subseteq \Theta_2^p \subseteq \Delta_2^p \subseteq \Sigma_2^p \subseteq \Theta_3^p \subseteq \mathsf{PSPACE}$$

It is strongly believed by researchers in complexity theory that these inclusions are strict.

3. Logical representation of preference

In this Section we are concerned with the preferences of a *single voter* over a finite set of candidates $\mathcal{X}$. Candidates are denoted by $\vec{x}$, $\vec{y}$, $\vec{z}$ etc. – exactly as interpretations (the reason of this choice will be made clear soon).

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 candidates? 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 $\vec{x} > \vec{y}$ 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).

A candidate $\vec{x}$ is said to be *non-dominated* for a preference relation $\geq$ (respectively, a utility function u) if and only if there is no $\vec{y}$ such that $\vec{y} > \vec{x}$ (respectively, such that $u(\vec{y}) > u(\vec{x})$).

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 $\mathcal{O}(|\mathcal{X}|)$ for cardinal structures and $\mathcal{O}(|\mathcal{X}|^2)$ for ordinal structures, respectively, where $|\mathcal{X}| = 2^{|VAR|}$.

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

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

3.2. Principles of logical representation of preference

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 large in n. Because specifying a preference structure explicitly in such a case is unreasonable, 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 enable 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 *logical* (and propositional), i.e., each v_i is a binary variable: $D_1 = ... = D_n = \{0, 1\}$. Languages based on propositional logic are not only compact, but also particularly *expressive*, thanks to the expressive power of logic, and therefore they are 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 (especially for satisfiability).

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 restricting the set of (physically) feasible worlds. This strong assumption implies that agents have a full and common knowledge of this set of feasible alternatives[4]. 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" [50]).

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, inference is not as important, whereas the problems that are particularly important are the following:

DEFINITION 1 (comparison).
Given a logical specification GB (goal base) of the preferences of an

[4] This assumption implies that we study in this paper *pure preference representation language* only; therefore, languages for expressing logically decision making problems under uncertainty are not considered in this paper as soon as they lose their significance in uncertainty-free environments.

agent, and two candidates $\vec{x}$ and $\vec{y}$, the COMPARISON *problem consists in determining whether $\vec{x} \geq_{GB} \vec{y}$.*

DEFINITION 2 (NON-DOMINANCE).
Given a logical specification GB of the preferences of an agent and a candidate $\vec{x}$, the NON-DOMINANCE *problem consists in determining whether $\vec{x}$ is non-dominated for $\geq_{GB}$.*

DEFINITION 3 (CAND-OPT-SAT).
Given a logical specification GB of the preferences of an agent and a property represented by a formula ψ, the CAND-OPT-SAT *problem consists in determining whether there exists an non-dominated candidate satisfying ψ.*

Note that if for a given representation language COMPARISON is in P then NON-DOMINANCE is in coNP and CAND-OPT-SAT in Σ_2^p.

Other relevant problems that could be considered are the search problem associated with NON-DOMINANCE, i.e., *search for one non-dominated candidate*, and the (computationally much harder) function problem *determine all non-dominated candidates*. For the sake of simplicity, we focus here on decision problems and leave the latter two problems for further research.

3.3. A 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. Proofs are in the Appendix.

3.3.1. *Basic propositional representation*

The simplest, or prototypical, logical representation of preference, that we call "basic" and denote by R_{basic}, consists simply in giving a single goal G (i.e. a propositional formula) (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_{basic} :*

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

There are two straightforward ways of refining the basic representation, which both consist in considering that GB is a sets of propositional formulas, namely $GB = \{G_1, ..., G_n\}$. They order candidates according to the *number* (respectively the *set*) of formulas satisfied in GB.

R_{card} $u_{GB}^{card}(\vec{x}) = |\{i | \vec{x} \models G_i\}|$

$R_{\subseteq}$ $\vec{x} \geq_{GB}^{\subseteq} \vec{y}$ if and only if $sat_{GB}(\vec{x}) \supseteq sat_{GB}(\vec{y})$, where $sat_{GB}(\vec{x}) = \{i | \vec{x} \models G_i\}$.

The partial preorder $\geq_{GB}^{\subseteq}$ on $\mathcal{X}$ generated by GB according to $R_{\subseteq}$ is nothing but the Pareto ordering: indeed, a candidate $\vec{x}$ is non-dominated for $\geq_{GB}^{\subseteq}$ if and only if there is no candidate $\vec{y} \in Mod(K)$ satisfying the goals satisfied by $\vec{x}$ and at least another one.

3.3.2. *Weighted goals*

The refinement R_{card} of R_{basic} considers all-or-nothing but independent goals, and enables compensations. This representation can be generalized to R_{wg} by weighting goals with numerical valuations. The utility of a candidate is computed by first gathering the valuations of the goals it satisfies, the valuations of the goals it violated, and then by aggregating these valuations in a suitable way (see for instance [34]) :

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

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

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 can be assumed that only the violated goals count (see for instance [34, 51]), which leads to $u^F_{GB}(\vec{x}) = F(\{\alpha_i | \vec{x} \models \neg G_i\})$; F has of course to satisfy a number of desirable properties (see [34])[5]. Usual choices for F are, for instance, sum (weights are then usually called *penalties*) or maximum (which corresponds to possibilistic logic). The complexity of decision problems for weighted logics can easily be derived from the complexity of distance-based belief merging ([39],[30]). Assuming that the aggregation functions F_1, F_2 and F_3 can be computed in polynomial time, we get the following results as byproducts from known results, especially the complexity of distance-based belief merging ([39], and especially [30]): 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 Θ_2^p-complete even for the simple representation language R_{card}.

3.3.3. *Prioritized goals*

Instead of weighting formulas by numerical weights, several approaches proceed by ordering them with a *priority relation.* Consider $GB = \langle\{G_1, ..., G_n\}, \succeq\rangle$ where $\succeq$ is a weak order (called a priority relation) on $\{1, ..., n\}$. Let $\succ$ be the strict order induced by $\succeq$, and let $nonsat_{GB}(\vec{x}) = \{1, ..., n\} \setminus sat_{GB}(\vec{x})$. A common choice (see *e.g* [29] [3]), called the *discrimin* ordering when $\succeq$ is complete, is the following:

$R_{discrimin}$:

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

Note that $R_{discrimin}$ generalizes $R_{\subseteq}$ – namely, it coincided with $R_{\subseteq}$ when $\succeq$ is chosen to be the relation defined by $i \geq j$ for all i, j. An alternative common way of inducing preference on candidates from priorities is the representation $R_{leximin}$ based on *leximin* ordering [3]. It generalizes R_{card} (and assumes that the priority relation $\succeq$ is complete). We omit its definition.

It is not hard to show that as soon as $\vec{x} \geq_{GB} \vec{y}$ can be decided in polynomial time from $\succeq$, which is the case for $R_{\subseteq}$, $R_{discrimin}$ and $R_{leximin}$, then COMPARISON is polynomial, therefore NON-DOMINANCE is in coNP and CAND-OPT-SAT in Σ_2^p. Moreover, NON-DOMINANCE is coNP-complete for $R_{\subseteq}$. Lastly, from results in [41, 26], it can be easily derived that CAND-OPT-SAT is Σ_2^p-complete for $R_{\subseteq}$, and from results in [14, 27], that CAND-OPT-SAT is Δ_2^p-complete for $R_{leximin}$.

[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].

A recent approach [10] proposes a preference representation language QCL based on prioritized goals, where priority is expressed by means of a new connector $\times$: $\varphi_1 \times \varphi_2 \ldots \times \varphi_n$ has to be understood as the priority relation: φ_1 preferred to $\neg\varphi_1 \wedge \varphi_2$ preferred to $\neg\varphi_1 \wedge \neg\varphi_2 \wedge \varphi_3$... Formulas expressed in this language can be translated into a set of prioritized goals, the preference relation then being based on the *leximin* ordering (however the transformation is not polynomial). Therefore the problems associated with R_{QCL} are at least as hard as with $R_{leximin}$ and possibly above (we did not investigate them).

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:

3.3.4.1. *Standard preference relation* $R_{cond,S}$ consists in considering that a candidate is at least as good as another one if and only if this holds in *all* models of GB. Formally: $\vec{x} \geq_{GB}^{cond,S} \vec{y}$ if and only if for any $\geq$ satisfying GB we have $\vec{x} \geq \vec{y}$.

Note that $\geq_{GB}$ is only a partial preoder which is generally very weak, often much too weak (i.e., does not enable enough comparisons) to be a good candidate for preference representation, 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).

3.3.4.2. *Preference relation based on Z-ranking* While $R^{cond,S}$ considered *all* models satisfying a set of conditionals, the approach based on the Z-completion of GB, at work in System-Z [45] and similar approaches, selects *one* model and allows much more consequences to be derived. Given a set $GB = \mathcal{D}$ of conditional rules $\phi : \psi$ and a set of hard facts, System-Z proceeds by partitioning $\mathcal{D}$ into a collection $\mathcal{D}_0 \cup \ldots \cup \mathcal{D}_n$; if a conditional rule $\varphi : \psi$ is in $\mathcal{D}_i$ then its *rank* is i. We omit to explain how the ranking is computed as this is only derivative for the purposes of the paper. Note that ranks intuitively respect specificity relations between rules, i.e., more specific rules are assigned higher ranks. Now, the ranking function on conditional rules induces a ranking function κ on candidates: a candidate $\vec{x}$

being said to violate a rule $\varphi : \psi$ if and only if $\vec{x} \models \varphi \wedge \neg\psi$, then for any $\vec{x} \models K$, $\kappa_{GB}(\vec{x}) = 0$ if $\vec{x}$ does not violate any rule of D and $\kappa_{GB}(\vec{x}) = 1 + \max\{i | d \in \mathcal{D}_i$ and $\vec{x}$ violates $d\}$ otherwise. Note that *more preferred candidates have lower ranks*, henceforth, $\geq_{GB}^{cond,Z}$ is defined by $\vec{x} \geq_{GB}^{cond,Z} \vec{y}$ if and only if $\kappa_{GB}(\vec{x}) \leq \kappa_{GB}(\vec{y})$.

Intuitively speaking, $\geq_{GB}^{cond,Z}$ is the preference relation, among those satisfying GB, maximizing preference world by world ([5], page 79). The obtained relation $\geq_{GB}^{cond,Z}$ is much more discriminant (hence much better) than $\geq_{GB}^{cond,S}$.

Complexity results for $R_{cond,S}$ and $R_{cond,Z}$ can be derived as byproducts of complexity results for conditional logics [28] and for conditional knowledge bases [27]:

PROPOSITION 2.

1. *for* $R_{cond,S}$*:* COMPARISON *is* coNP*-complete* [6]*.*
2. *for* $R^{cond,Z}$*:* COMPARISON^{cond,Z}, NON-DOMINANCE^{cond,Z} *and* CAND-OPT-SAT^{cond,Z} *are* Θ_2^p*-complete.*

One drawback of $R^{cond,Z}$ is that, as for $R^{cond,S}$), 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 [36]. This principle is further generalized by introducing numerical strengths and polarities in [51, 38]. We did not investigate complexity issues for these approaches.

3.3.5. *Ceteris paribus preferences*

C, G and G' being three propositional formulas and V being a subset of VAR such that $Var(G) \cup Var(G') \subseteq V$, the *ceteris paribus desire* $C : G > G'[V]$ is interpreted by : "all irrelevant things being equal, I prefer $G \wedge \neg G'$ to $G' \wedge \neg G$", where the "irrelevant things" are the variables that are not in V. The definitions proposed in various places [23, 24, 49, 7] differ somehow. We take as a basis the definition of [23], slightly generalized, in the spirit of [49] but with less complications, by introducing the explicit set of variables V which expresses, in an explicit way, which variables are referred to when saying "all other things being equal" (namely, those not in V). For natural reasons, and

[6] As to NON-DOMINANCE and CAND-OPT-SAT, I could not manage to identify exactly their complexity; the best I can say is that they are BH_2-hard and (obviously) in Σ_2^p, but since this is not very significant (the gap between BH_2 and Σ_2^p being large), I omit the technical details.

to remain consistent with the origiginal definitions, we impose that $Var(G) \cup Var(G') \subseteq V$. This modification is simple, it does not affect significantly the computational aspects of the framework and answers (to a certain extent) a criticism addressed to *ceteris paribus* desires in [2] (pages 545-546), thus giving a little more expressivity to the framework, for instance by allowing ramifications of the goals to be taken into account (by default, V is considered to be the set of variables mentioned in G and G'[7]).

DEFINITION 4 (R_{CP}).
Let $GB = \langle K, \mathcal{D} \rangle$ with $\mathcal{D} = \{C_1 : G_1 > G'_1[V_1], \ldots, C_m : G_m > G'_m[V_m]\}$ such that for all i, C_i, G_i and G'_i are propositional formulas and $Var(G_i) \cup Var(G'_i) \subseteq V_i \subseteq VAR$. For two candidates $\vec{x}, \vec{y} \in \mathcal{C}$, $\vec{x}$ is said to dominate *$\vec{y}$ with respect to the desire $D_i = (C_i : G_i > G'_i[V_i])$, denoted by $\vec{x} >_{D_i} \vec{y}$, if and only if the following conditions hold:*

1. $\vec{x} \models K \wedge C_i \wedge G_i \wedge \neg G'_i$;
2. $\vec{y} \models K \wedge C_i \wedge \neg G_i \wedge G'_i$;
3. *$\vec{x}$ and $\vec{y}$ coincide on all variables that are not in V_i.*

Now, the strict order $>_{GB}$ is defined from the above dominance relations by transitive closure, i.e.: for two candidates $\vec{x}, \vec{y} \in \mathcal{C}$, $\vec{x} >^{cp}_{GB} \vec{y}$ if and only if there exists a finite chain $\vec{x}_0 = \vec{x}, \vec{x}_1, \ldots, \vec{x}_{q-1}, \vec{x}_q = \vec{y}$ of candidates such that for all $j \in \{0, \ldots, q-1\}$ there is a $i \in \{1, \ldots, m\}$ such that $\vec{x}_j >_{D_i} \vec{x}_{j+1}$. Lastly, $\geq_{GB}$ is defined by $\vec{x} \geq_{GB} \vec{y}$ if and only if $\vec{x} >_{GB} \vec{y}$ or $\vec{x} = \vec{y}$.

For the complexity results we consider not only the general case but also the following restriction (*simple desires*): each goal G_i is a literal l_i and $G'_i = \neg l_i$.

PROPOSITION 3. *The complexity of* COMPARISON, NON-DOMINANCE *and* CAND-OPT-SAT *for* ceteris paribus *desires is reported on the following table.*

	COMPARISON	NON-DOMINANCE	CAND-OPT-SAT
general case	PSPACE-*comp.*	coNP-*comp.*	Σ_2^p-*comp.*
simple desires	PSPACE-*comp.*	coNP-*comp.*	Σ_2^p-*comp.*

[7] This could be refined further by considering the variables on which G and G' do not *semantically* depend ([49, 37]). This is not be considered further in this paper.

The surprising point is that, for the general case, the comparison problem is much more difficult than the other ones (in contrast to other representation languages). This is due to possible exponentially long "preference paths" (see the proof in Appendix).

As we can see, imposing that goals are literals does not make the problems easier[8]. Studying further the complexity of the comparison and the non-dominance problems for *ceteris paribus* desires under various assumptions is a promising topic. Some significant results have been obtained in the resrtricted case of *CP-nets* in [21, 20], especially tractable cases (however, the general comparison problem for the restriction of *ceteris paribus* desires corresponding to CP-nets is still open).

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 [34, 35, 31], denoted by R_d, 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 [26, 42] and of distance-based belief merging [30] that COMPARISON (which amounts to deciding whether $d(\vec{x}, G) \leq d(\vec{y}, G)$), 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 goal being associated with 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. Now, the complexity of the key problems COMPARISON

[8] This would however be the case with variants of the framework, that we do not consider here for the sake of brevity.

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 is polynomial, or, at least, for which NON-DOMINANCE lays at the first level of the polynomial hierarchy. The complexity results given in the Section are summarized in the table below[9].

	COMPARISON	NON-DOMINANCE	CAND-OPT-SAT
R_{basic}	P	coNP-complete	NP(3)-complete
R_{wg}	P	coNP-complete	up to Δ_2^p-complete
$R_{discrimin}$	P	coNP-complete	Σ_2^p-complete
$R_{leximin}$	P	coNP-complete	Δ_2^p-complete
$R_{cond,S}$	coNP-complete	BH_2-hard, in Σ_2^p	BH_2-hard, in Σ_2^p
$R_{cond,Z}$	Θ_2^p-complete	Θ_2^p-complete	Θ_2^p-complete
R_{cp}	PSPACE-complete	coNP-complete	Σ_2^p-complete
R_d	up to Δ_2^p-complete	up to Δ_2^p-complete	up to Δ_2^p-complete

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. 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[10].

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

[9] For R_{wg}, we assume that aggregation functions are computable in polynomial time; for $R_{\succ}$, that preference relation between candidates is computable in polynomial time from priorities; and for R_d, that $d(\vec{x}, \vec{y})$ is computed in polynomial time.

[10] However, see [6] 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.

language R, and second, one or several optimal (i.e., non-dominated) candidate(s) is (are) determined automatically, using a fixed *vote rule*.

A *preference profile* P consists of a preference structure (cardinal or ordinal) for each of the voters. Social choice theory (see [40] for a good introduction) defines a *vote 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*[11]. The set of elected candidates, given a preference profile P and a vote rule V, will be denoted by $Select_V(P)$.

Research in social choice theory focuses on the properties of vote 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 vote rule will both depend of the rule V itself and of the representation language R chosen.

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

We will now look at several vote 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 vote rule still "mean" something when the set of candidates has a combinatorial structure?);

2. *computational complexity*. Let V be a vote rule and R a representation language; we consider the following decision problems:

DEFINITION 5 (AMONG-WINNERS$_{V,R}$).
AMONG-WINNERS$_{V,R}$: *given $B = \langle GB_1, ..., GB_p \rangle$ and a candidate $\vec{x}$, determine whether $\vec{x} \in Select_V(Induce_R(B))$*

DEFINITION 6 (ELECT-SAT$_{V,R}$).
ELECT-SAT$_{V,R}$: *given $B = \langle GB_1, ..., GB_p \rangle$ and a formula ψ, determine whether there is a candidate in $Select_V(Induce_R(B))$ satisfying ψ.*

[11] For the sake of simplicity we will not distinguish between what is usually called a vote rule, which selects a unique candidate, and a *vote correspondance*, which selects a subset of candidates; we will use the terminology "vote 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 a tie-breaking rule, either by giving up neutrality, or by randomly choosing one of the elected candidates.

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 the languages studied in Section 3, 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 [25]), namely: for each $\vec{x}$, the scores $u_i(\vec{x})$ are synthesized by an aggregation function $* : I\!R^p \rightarrow I\!R$ into a so-called social welfare function $u_B(\vec{x}) = *\{u_i(\vec{x}) | i = 1 \dots p\}$ reflecting the satisfaction of the group (see [40]). $Select_{agreg(*)}(P)$ is then the set of candidates maximizing $u_B(\vec{x})$. Classically, the function $*$ is commutative (for guaranteeing the anonymity 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-interpreted 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 [30] it can be shown that, provided that $u_i(\vec{x})$ is polynomially computable from 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 (note that this ordinal structure 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 only of its position in the relation $\geq_i$. 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 $\geq_i$ (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* (which goes back to the eighteenth century mathematician Borda). 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 monotonicity and some forms of strategyproofness that are not satisfied by other rules (see [40]).

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 at 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 very high, since candidate counting is required. One may thus advance the following general principle that *a vote 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 of voters who 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 4. *Let R be a language for which* NON-DOMINANCE *is in* NP*or in* coNP*(this includes a fortiori languages for which* COMPARISON *is polynomial).*

1. AMONG-WINNERS$_{plurality,R}$ *is in* Θ_3^p*;*

2. ELECT-SAT$_{plurality,R}$ *is in* Θ_3^p*;*

3. for $R = R_{basic}$, AMONG-WINNERS$_{plurality,R_{basic}}$ *is* coNP*-complete and* ELECT-SAT$_{plurality,R_{basic}}$ *is Θ_2^p-complete.*

(3) gives a lower complexity bound since R_{basic} is the most elementary logical representation language. 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 total order, this comes down to count the number of voters who do not rank $\geq_i$ last. Complexity results for the veto rule are similar to those for the plurality rule.

The following example illustrates how scoring rules are applied when preference is expressed compactly.

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 B be the following triple of goal bases:*

$$B = \langle GB_1, GB_2, GB_3 \rangle$$

where

- $GB_1 = \{a, a \vee b\}$;
- $GB_2 = \{\neg a, b \wedge c, a \rightarrow (b \vee c)\}$;
- $GB_3 = \{\neg b, a \leftrightarrow \neg c\}$;

The preference orderings $\geq_{GB_1}$, $\geq_{GB_2}$ and $\geq_{GB_3}$ are:

$$\begin{array}{ll}
 & (a,b,c),(a,b,\neg c),(a,\neg b,c),(a,\neg b,\neg c)\\
>_{GB_1} & (\neg a,b,c),(\neg a,b,\neg c)\\
>_{GB_1} & (\neg a,\neg b,c),(\neg a,\neg b,\neg c)\\
 & (\neg a,b,c)\\
>_{GB_2} & (a,b,c),(\neg a,b,\neg c),(\neg a,\neg b,c),(\neg a,\neg b,\neg c)\\
>_{GB_2} & (a,b,\neg c),(a,\neg b,c)\\
>_{GB_2} & (a,\neg b,\neg c)\\
 & (a,\neg b,\neg c),(\neg a,\neg b,c)\\
>_{GB_3} & (a,b,\neg c),(\neg a,b,c),(a,\neg b,c),(\neg a,\neg b,\neg c)\\
>_{GB_3} & (a,b,c),(\neg a,b,\neg c)
\end{array}$$

Candidates written on a same line are equally preferred: for instance, for GB_1, we have $(a,b,c) \sim_{GB_1} (a,b,\neg c) \sim_{GB_1} (a,\neg b,c) \sim_{GB_1}$

$(a, \neg b, \neg c)$, any of these four candidates being strictly preferred to both $(\neg a, b, c)$ and $(\neg a, b, \neg c)$ and so on.

Note the role of redundancy in goal bases. Take for instance $GB_1 = \{a, a \vee b\}$: the redundancy between goal a and goal $a \vee b$ expresses that voter 1 prefers to see a true, and that in addition to this he has a preference to see $a \vee b$ satisfied (in particular in the case where a is false). Therefore, the preference profile of voter 1 can be expressed as I desire a, and if not possible then b, *reflected by the preference relation $>_{GB_1}$ above.*

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.*, $\forall \vec{y} \in Mod(K), |\{i | \vec{x} >_i \vec{y}\}| > |\{j | \vec{y} >_i \vec{x}\}|$ ([12]). This notion goes back to 1785 [19] 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 $R = R_{basic}$ (and holds as well for more sophisticated representation languages). We easily get that $\vec{x}$ is the Condorcet winner for $B = \langle G_1, ..., G_n \rangle$ if and only if the number of i such that $\vec{x} \models G_i$ is maximal *and no other interpretation* than $\vec{x}$ maximizes this number of voters whose goal is satisfied. This has the consequence that there almost

[12] 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.

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, $R = R_{basic}$, 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 *near Condorcet winners*. Another way (much more usual) of weakening the notion of Condorcet winner consists in replacing $>$ by $\geq$ in the above inequality (candidates satisfying the resulting condition are called *weak Condorcet winners*).

DEFINITION 7. *Let* $B = \langle GB_1, \ldots, GB_n\rangle$, *and for any two candidates* $\vec{x}, \vec{y} \in \mathcal{C}$, $C(\vec{x}, \vec{y}, B) = |\{i \mid \vec{x} >_{GB_i} \vec{y}\}|$.

- $\vec{x}$ *is a* Condorcet winner *(CW) for* B *if and only if for any candidate* $\vec{y} \in \mathcal{C}$ *s.t.* $\vec{y} \neq \vec{x}$, *we have* $C(\vec{x}, \vec{y}, B) > C(\vec{y}, \vec{x}, B)$;

- $\vec{x}$ *is a* near Condorcet winner *(NCW) for* B *if and only if* $\forall \vec{y} \in \mathcal{C}$ *s.t.* $\vec{y} \neq \vec{x}$, *we have either* $C(\vec{x}, \vec{y}, B) > C(\vec{y}, \vec{x}, B)$ *or* $\forall i = 1, \ldots, n$, $\vec{x} \sim_{GB_i} \vec{y}$;

- $\vec{x}$ *is a* weak Condorcet winner *(WCW) for* B *if and only if* $\forall \vec{y} \in \mathcal{C}$ *s.t.* $\vec{y} \neq \vec{x}$, *we have* $C(\vec{x}, \vec{y}, B) \geq C(\vec{y}, \vec{x}, B)$.[13]

Note that we have the straightforward properties:

- $\vec{x}$ is a CW for $B \Rightarrow \vec{x}$ is a NCW for $B \Rightarrow \vec{x}$ is a WCW for B ;

- the existence of a WCW (and *a fortiori* of a NCW nor a CW) is not guaranteed; there is at most a CW (but there may be more than one NCW/WCW);

- let $\sim_B$ be the equivalence relation with respect to the goal base B, defined by $\vec{x} \sim_B \vec{y}$ if and only if for every $i = 1, \ldots, n$, $\vec{x} \sim_{GB_i} \vec{y}$; let us denote by $[\vec{x}]$ the equivalence class of $\vec{x}$ for $\sim_B$, let $X_{/\sim_B}$ be the quotient of X by $\sim_B$, and let $[\vec{x}] \geq_i [\vec{y}]$ if and only if $\vec{x} \geq_{GB_i} \vec{y}$. Then $\vec{x}$ is a NCW if and only if $[\vec{x}]$ is a CW for the problem defined by the set of candidates $\mathcal{X}_{/\sim_B}$ and the profile $\{\geq_1, \ldots, \geq_p\}$.

[13] Both definitions of near and weak Condorcet winners could be "mixed", namely, it is possible to define an even weaker notion; this is not considered here.

Let R be a representation language for which COMPARISON is polynomial. We consider the problem CONDORCET WINNER VERIFICATION (and similarly WEAK and NEAR CONDORCET WINNER VERIFICATION) the problem of determining whether a given candidate $\vec{x}$ is a CW (and similarly a NCW, a WCW) for a given B ; and the problem CONDORCET WINNER EXISTENCE (and similarly WEAK and NEAR CONDORCET WINNER EXISTENCE) the problem of determining whether there exists a CW (and similarly a NCW, a WCW) for a given B .

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 harder when preference are represented in a succinct language:

PROPOSITION 5. *Let R be a language for which* COMPARISON *is polynomial.*

1. CW VERIFICATION, NCW VERIFICATION *and* WCW VERIFICATION *are in* coNP*; when $R = R_{basic}$,* CW VERIFICATION *and* NCW VERIFICATION *are* coNP*-complete*[14].

2. CW EXISTENCE, NCW EXISTENCE *and* WCW EXISTENCE *are all in* Σ_2^p*; when $R = R_{basic}$,* CW VERIFICATION *is in* Θ_2^p *and* coNP*-hard;* NCW VERIFICATION *is in* Θ_2^p *and* BH_2*-hard*[15].

Since the existence of a CW is not guaranteed, several vote 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, it is reasonable to ask for a Condorcet-consistent rule to elect 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 therefore is not well-suited to combinatorial vote; the Simpson rule is more interesting.

[14] When $R = R_{basic}$, there always exists a WCW, therefore WCW VERIFICATION is trivial.

[15] Note that I could not manage to prove Θ_2^p-hardness. Moreover, 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.

4.3.2. *The Simpson rule*
Let P be a profile, $\vec{x}$ and $\vec{y}$ two candidates. Let $C(\vec{x}, \vec{y}, P)$ be the number of voters who strictly prefer $\vec{x}$ to $\vec{y}$ w.r.t. P. The *Simpson score* $S(\vec{x}, P)$ is defined by $S(\vec{x}, P) = \min_{\vec{y} \neq \vec{x}} C(\vec{x}, \vec{y}, P) - C(\vec{y}, \vec{x}, P)$, and $Select_{Simpson}(P)$ is the set of candidates maximizing $S(\vec{x}, P)$ (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 6. *Let R be a language for which comparison is polynomial.*

1. AMONG-WINNERS$_{Simpson,R}$ *is in* Θ_3^p.
2. ELECT-SAT$_{Simpson,R}$ *in* Θ_3^p.
3. *For* $R = R_{basic}$*:* AMONG-WINNERS$_{Simpson,R_{basic}}$ *is* coNP*-complete and* ELECT-SAT$_{Simpson,R_{basic}}$ *is* Θ_2^p*-complete.*

Example 1 (continued).
On the above table we give the values of $C(\vec{x}, \vec{y}, B)$ ($\vec{x}$ 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	$\times$	0	0	0	0	-1	0	0
110	0	$\times$	0	0	0	-1	1	0
101	0	0	$\times$	0	0	-1	1	0
100	0	0	0	$\times$	-1	-1	0	-1
011	0	0	0	1	$\times$	-2	-1	-2
010	1	1	1	1	2	$\times$	0	0
001	0	-1	-1	0	1	0	$\times$	-1
000	0	0	0	1	2	0	1	$\times$
$S(\vec{x}, B) \rightarrow$	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 near Condorcet winner.

4.4. Discussion

The conclusions are not simple, because several parameters have to be taken into account. From a strict computational point of view, aggregative rules may be easier to implement than scoring rules or Condorcet-consistent rules, but they require a numerical input which is often hard to obtain. Note that our complexity results for the Simpson rule and for the plurality rule almost coincide[16].

5. Further related work and perspectives

This paper contains both a survey of several logical languages for preference representation proposed so far in the literature, and a very preliminary work about combinatorial vote. We have merely shown that it is often hard (from the points of view of relevance and computational complexity) to import in a straightforward way vote rules from the area of social choice and to have them work on problems having a heavy combinatorial structure. Lots of things remain to be studied. We list here several topics related to this paper and point to possible directions for further research.

5.1. Compact languages for preference representation: going further

The role of the first half of the paper was to give a survey of logical representation of preference, especially from a computational point of view. I do not claim that it is exhaustive, but I tried to refer to most significant approaches. At least three kinds of languages have not been considered, and I'll try to explain here why:

- so-called "preferential logics", developed either in a nonmonotonic reasoning framework ([48, 32] and subsequent works), or more recently in a logic programming framework, are not considered because they do not really deal with (decision-theoretic) preference representation. The terminology "preference" in the latter approaches is rather technical ("preference" is used as a synonym of "[weak/strict] order") and not especially connected to decision-theoretic issues.

- compact languages for preference representation that are not based on logic, such as utility networks (e.g., [1]) or weighted CSPs (e.g., [47]). I did not study them because for the sake of brevity I focused on pure logical approaches, but these approaches *are* relevant to combinatorial vote. Some of them have a structure close enough to propositional logic to make me think that the complexity results

[16] Actually, they differ in the condition required: for the Simpson rule we required that COMPARISON is polynomial whereas for plurality we required the weaker condition that NON-DOMINANCE is in NPor in coNP.

would not be so far from those obtained, but again, this is left for further research.

– logical languages for qualitative decision theory (see next paragraph).

As to complexity results, some are really new (especially those related to *ceteris paribus* desires) and some others are byproducts of already existing results. There are several problems whose complexity was not entirely identified in this paper, such as finding the upper bound for CW existence problems for simple languages such as R_{card} (Proposition 5). Another issue which is lacking here is a study of the *representational complexity* [12] of these languages, which would assess precisely their concision power: see [18].

5.2. Qualitative decision theory

The survey I gave in Section 2 differs from Doyle and Thomason's review [22] on *qualitative decision theory*. Qualitative approaches to decision theory and compact, logical approaches for preference representation are two distinct issues, even if some papers are concerned with both of them (especially [5]): qualitative decision theory aims at studying criteria for decision making under uncertainty that refer as little as possible to numbers (in contrast with the standard expected utility criterion) and it is not surprising that several approaches are based on non-classical logics whose semantics is defined by means of orderings (as [5]), or on nonmonotonic logics [50, 9]; however the goal in these approaches is not to describe a pure preference relation on states, and most of these approaches degenerate when the assumption of perfect knowledge is made.

5.3. Preference elicitation

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.

Preference elicitation traditionally focus on *one agent*; now, the recent paper [17] considers the elicitation issue is a vote context; given some partial data about the votes of a number of agents, it studies the complexity of determining which piece of preference and from which voter in order to be able to determine the winner of the election. We did not consider this issue in the context of combinatorial vote, and no doubt that it is extremely relevant to it.

5.4. Manipulation

It has been known for long in social choice theory that there is no vote rule being both non-dictatorial and strategyproof (this is the well-known Gibbard-Satterthwhite theorem); not being strategyproof means that it is not always the interest of the voters to report their preference truthfully. The vote rules considered in this paper in the context of combinatorial vote do not escape this; however, manipulation is probably computationally hard (this is the best we can expect) when the set of candidates has a strong combinatorial structure[17]; demonstrating this formally would be a positive aspect of combinatorial vote.

5.5. More vote rules; partial vote and negotiation

Fopr the sake of brevity, there are a number of vote rules that we did not consider in this paper, such as *single transferable vote* or sequential (several-rounds) rules.

Moreover, the combinatorial structure of the set of candidates offers new perspectives: in some situations, it is not compulsory to assign *all* variables at once; one may then 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 negotiation phase which is out of our concern), and the process can then be iterated until all variables have been assigned. This process requires that vote rules for combinatorial vote be adapted; this is an issue for further work.

5.6. 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., choosing a person to be recruited), or when the variables are independent (or almost) from each other, preferences can

[17] Manipulation is already hard when the number of candidates is small, see [16, 15].

be aggregated manually, but this is not the case as soon as the former has a strong combinatorial structure[18].

Acknowledgements

I would like to thank Sébastien Konieczny for a careful reading of a preliminary version of the paper, Thomas Eiter and Pierre Marquis for discussions about Θ_2^p-hardness, and Carmel Domshlak for discussions about *ceteris paribus* desires and CP-nets (and for pointing to me a problem in a preliminary version of the paper).

Appendix: proofs

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

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

Proof:

1. from the definition of $\geq_G$ we get immediately that for any $\vec{x}, \vec{y} \in$C, we have $\vec{x} \geq_G \vec{y}$ if and only if $\vec{x} \models G$ or $\vec{y} \models \neg G$, which is obviously checked in polynomial time;
2. using the above equivalence, we get that $\vec{x} \in$X is non-dominated for $\geq_G$ if and only if $\vec{x} \models G$ or $G \wedge K$ is unsatisfiable, which proves membership of NON-DOMINANCE to coNP. Hardness comes from the following polynomial reduction from UNSAT: let φ be a propositional

[18] 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.

formula and let $K = \top$, $G = \varphi \wedge p$, where p is a new propositional symbol (i.e., $p \notin Var(\varphi)$). Let $\vec{x}$ be any candidate such that $\vec{x} \models \neg p$. $\vec{x}$ is undominated for G if and only if $\vec{x} \models \varphi \wedge p$ or $\varphi \wedge p$ is unsatisfiable, i.e., if and only if φ is unsatisfiable.

3. there exists an optimal candidate satisfying ψ iff ψ is satisfiable and ($K \wedge G$ is unsatisfiable or $K \wedge G \wedge \psi$ is satisfiable), 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 if and only if φ_1 is satisfiable and (φ_2 is unsatisfiable or φ_3 is satisfiable). Assuming without loss of generality that φ_1, φ_2 and φ_3 do not share any variables, and p being a new variable, we let $K = \top$, $G = (\varphi_2 \wedge \neg p) \vee (\varphi_2 \wedge \varphi_3)$, $\psi = \varphi_1 \wedge p$. Let us distinguish four cases:

   - φ_1 is satisfiable and φ_2 unsatisfiable. In this case, $G \equiv \bot$, therefore any candidate is non-dominated: hence, since φ_1 is satisfiable, so is ψ , and $\langle K, G, \psi \rangle$ is a positive instance of CAND-OPT-SAT.
   - φ_1 and φ_3 are both satisfiable. If φ_2 is unsatisfiable then we are in the above case and $\langle K, G, \psi \rangle$ is a positive instance of CAND-OPT-SAT, so assume that φ_2 is consistent as well; since φ_1, φ_2 and φ_3 do not share variables, their conjunction $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$ is satisfiable, and for the same reason, so is $\varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge p$. Let $\vec{x} \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge p$. We have $\vec{x} \models G$, therefore $\vec{x}$ is undominated, and moreover we have $\vec{x} \models \psi$, hence, $\langle K, G, \psi \rangle$ is a positive instance of CAND-OPT-SAT.
   - φ_1 is unsatisfiable. In this case, so is ψ, and $\langle K, G, \psi \rangle$ is a negative instance of CAND-OPT-SAT.
   - φ_2 is satisfiable and φ_3 is unsatisfiable. In this case, $K \wedge G \equiv (\varphi_2 \wedge \neg p)$ is satisfiable, and $K \wedge G \wedge \psi \equiv ((\varphi_2 \wedge \neg p) \vee (\varphi_2 \wedge \varphi_3)) \wedge \varphi_1 \wedge p \equiv \bot \vee (\varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \wedge p)$, therefore $K \wedge G \wedge \psi$ is unsatisfiable, and we get that $\langle K, G, \psi \rangle$ is a negative instance of CAND-OPT-SAT.

   Noticing that these four cases cover all possible situations, we get the result that $\langle K, G, \psi \rangle$ is a positive instance of CAND-OPT-SAT if and only if $\langle \varphi_1, \varphi_2, \varphi_3 \rangle$ is a positive instance of SAT-SAT-UNSAT.

■

Proposition 2

1. *for $R_{cond,S}$:* COMPARISON *is* coNP*-complete.*

2. *for* $R_{cond,Z}$*:* COMPARISON*,* NON-DOMINANCE *and* CAND-OPT-SAT *are* Θ_2^p*-complete.*

Proof:

1.a For $R_{cond,S}$*,* COMPARISON *is in* coNP
Let us first remark that $\vec{x} \geq_{\mathcal{D}}^{cond,S} \vec{y}$ holds iff $\mathcal{D} \wedge D(for(\vec{y})|for(\vec{x}) \vee for(\vec{y}))$ is unsatisfiable in the conditional logic CO^*[5], whose semantics in namely in terms of complete preorders on worlds (without loss of generality, we identify here a CO^*-model M with its complete preorder $\geq_M$): indeed, let M be a CO^*-model; M satisfies the conditional desire $D(for(\vec{y})|for(\vec{x}) \vee for(\vec{y}))$ if and only if $Max(\geq_M, \{\vec{x}, \vec{y}\}) \subseteq \{\vec{y}\}$, i.e., if and only if $\vec{y} >_M \vec{x}$. Therefore, if $\mathcal{D} \wedge D(for(\vec{y})|for(\vec{x}) \vee for(\vec{y}))$ is satisfiable then there is a model M satisfing $\mathcal{D}$ such that $\vec{y} >_M \vec{x}$, and thus $\vec{x} \geq_{\mathcal{D}}^{cond,S} \vec{y}$ does not hold. Reciprocally, if $\mathcal{D} \wedge D(for(\vec{y})|for(\vec{x}) \vee for(\vec{y}))$ is unsatisfiable then any model M satisfying $\mathcal{D}$ is such that $\vec{y} \leq_M \vec{x}$, therefore $\vec{x} \geq_{\mathcal{D}}^{cond,S} \vec{y}$ holds. Therefore, the comparison problem reduces to an unsatisfiability test in CO^*. Now, the class of CO^*-models is the class of models satisfying absoluteness and connectedness in [28]; now, point (a) of Theorem 1 of [28] states that deciding satisfiability for conditional formulas without nested conditional in the latter logic is NP-complete, hence our comparison problem is in coNP.

1b. For $R_{cond,S}$*,* COMPARISON *is* coNP*-hard*
Let us consider the following polynomial reduction from UNSAT: to any propositional formula φ over an alphabet VAR, let $F(\varphi) = \langle \mathcal{D}, \vec{x}, \vec{y} \rangle$ where $\vec{x}$ and $\vec{y}$ are any two distinct candidates (therefore with distinct valuations), and $\mathcal{D} = \{D(\neg for(\vec{y})|for(\vec{x}) \vee for(\vec{y}) \vee \varphi)\}$. A CO^*-model M satisfies $\mathcal{D}$ if and only if $\vec{y} \notin Max(\geq_M, \{\vec{x}, \vec{y}\} \cup Mod(\varphi))$. Now, if φ is consistent then let $\vec{z}$ such that $\vec{z} \models \varphi$ and let then M be a model such that $\vec{z} >_M \vec{y} >_M \vec{w}$ for all worlds $\vec{w} \neq \vec{z}, \vec{y}$; M satisfies $\mathcal{D}$ and $\vec{x} \geq_M \vec{y}$ does not hold in M, therefore $\vec{x} \geq_{\mathcal{D}}^{cond,S} \vec{y}$ does not hold. Conversely, if φ is inconsistent then $M \models_{\mathsf{CO}^*} \mathcal{D}$ if and only if $\vec{x} \geq_M \vec{y}$, and hence $\vec{x} \geq_{\mathcal{D}}^{cond,S} \vec{y}$ holds. We have shown that $\vec{x} \geq_M \vec{y}$ holds if and only if φ is inconsistent, hence the coNP-hardness of COMPARISON.

2a. For $R_{cond,Z}$*,* COMPARISON*,* NON-DOMINANCE *and* CAND-OPT-SAT *are in* Θ_2^p*.*
Considering $GB = \langle K, \mathrm{D} \rangle$ as a conditional belief base (in the sense of [27]), we have that $\langle K, \mathcal{D}, \psi \rangle$ is a positive instance of CAND-OPT-SAT if and only if $GB \not\models D(\neg\varphi|\top)$. Now, determining whether a conditional belief base Z-entails a conditional is Θ_2^p-complete [27], and Θ_2^p is closed under complement, therefore CAND-OPT-SAT^{cond,Z} is in Θ_2^p. Moreover,

$\vec{x}$ is undominated if and only if $\langle K, \mathcal{D}, for(x)\rangle$ is a positive instance of CAND-OPT-SAT^{cond,Z}, and $\vec{x} \geq_{GB}^{cond,Z} \vec{y}$ if and only if $\vec{x}$ is undominated with respect to $\langle K \wedge (for(\vec{x}) \vee for(\vec{y})), \mathcal{D}\rangle$ (the latter holds because $\geq_Z$ is a connected), therefore COMPARISON^{cond,Z} and NON-DOMINANCE^{cond,Z} are in Θ_2^p as well.

2b.For $R_{cond,Z}$, COMPARISON, NON-DOMINANCE and CAND-OPT-SAT are Θ_2^p-hard.
We have just seen above that COMPARISON can be polynomially reduced to NON-DOMINANCE which in turn can be polynomially reduced to CAND-OPT-SAT which in turn can be polynomially reduced to CAND-OPT-SAT, therefore it is sufficient to prove that COMPARISON is Θ_2^p-hard. In [27], it is shown that Z-entailment is Θ_2^p-complete even once the default rankings have been computed (and therefore are part of the input); we give here a polynomial reduction from the complement of this problem (deciding non-entailment given the default rankings – Θ_2^p-complete as well, since Θ_2^p is closed under complement) to deciding comparison *given the default rankings* as well. Let $KB = \langle L, D_1, \ldots, D_p\rangle$ be a "pre-ranked" conditional belief base, where each D_i is a nonempty set of conditional rules and any let γ, δ be two propositional formulas. Let p and q be two new variables and let $K = \varphi \wedge (\delta \rightarrow q)$; let $\mathcal{D} = \langle \Delta_1, \ldots, \Delta_p\rangle$ be the ordered list of sets of conditional rules, defined as follows: $\Delta_i = \{D(\psi|\varphi \wedge p) \mid D(\psi|\varphi) \in D_i\}$; lastly, let $\vec{x}$ and $\vec{y}$ be any two candidates such that $\vec{x} \models \neg p \wedge \neg q$ and $\vec{y} \models \neg p \wedge q$. We have $\kappa_{K,\mathcal{D}}(\vec{x}) = \kappa_{KB}(\varphi \wedge \neg\psi)$ and $\kappa_{K,\mathcal{D}}(\vec{y}) = \kappa_{KB}(\varphi)$, therefore $\vec{x} \geq_{K,\mathcal{D}}^{cond,Z} \vec{y}$ if and only if $\kappa_{KB}(\varphi \wedge \neg\psi) \leq \kappa_{KB}(\varphi)$, i.e., if and only if KB does not entail $D(\psi|\varphi)$. ■

Proposition 3 *The complexity of* COMPARISON, NON-DOMINANCE *and* CAND-OPT-SAT *for* ceteris paribus *desires is reported on the following table.*

	COMPARISON	NON-DOMINANCE	CAND-OPT-SAT
general case	PSPACE-*comp.*	coNP-*comp.*	Σ_2^p-*comp.*
simple desires	PSPACE-*comp.*	coNP-*comp.*	Σ_2^p-*comp.*

We start by proving the most difficult result:

1. COMPARISON *for* ceteris paribus *preferences is* PSPACE-*complete.*

Proof: The membership proof is the same as the PSPACE-membership proof for plan existence with deterministic actions represented in STRIPS [11]: the key points are that (i) the length of a dominance path from $\vec{x}$ to $\vec{y}$, if such a dominance path, is bounded by $2^{|VAR|-1}$([19]) and (ii) for any two candidates $\vec{x}_1$, $\vec{x}_2$, checking whether $\vec{x}_1 >_{C:G>G'[V]} \vec{x}_2$ can be done in polynomial time (it is sufficient to check that $\vec{x}_1 \models C \wedge G \wedge \neg G'$, that $\vec{x}_2 \models C \wedge \neg G \wedge G'$, and that $\vec{x}_1$ and $\vec{x}_2$ coincide in all variables not in V).

The hardness proof is much more complex. It comes from a polynomial reduction from plan existence with deterministic actions represented in STRIPS[20]; the latter problem has been shown PSPACE-complete in [11]. We actually exhibit a polynomial reduction from a restriction (still PSPACE-hard) of propositional STRIPS PLAN EXISTENCE to COMPARISON : this restriction, called COMPLETE-GOAL-STATE (CGS) STRIPS PLAN EXISTENCE, is obtained by requiring that the goal is a maximal conjunction of literals, or in other terms, that there is a single goal state (in the general case, the goal is expressed by a conjunction G of literals which is not necessarily complete). ■

LEMMA 1. CGS STRIPS PLAN EXISTENCE *is* PSPACE-*complete.*

Proof of Lemma 1: PSPACE-hardness is proven by a polynomial reduction from (general) STRIPS PLAN EXISTENCE. Let us define an instance I of CGS STRIPS PLAN EXISTENCE as a 5-uple $J = \langle VAR = \{v_1, \ldots, v_n\}, ACT, E, \vec{x}_0, \vec{x}_G\rangle$ where VAR is a finite set of propositional variables, $ACT = \{\alpha_1, \ldots, \alpha_m\}$ is a set of (deterministic) actions, $\vec{x}_0$ and $\vec{x}_G$ are two complete states (i.e., full assignments over VAR) and E is a STRIPS description of the actions, containing for each $\alpha_i \in ACT$ a pair $\langle pre_i, post_i\rangle$ where pre_i and $post_i$ are consistent conjunctions of literals[21]. Consider the following reduction from general STRIPS PLAN EXISTENCE to CGS STRIPS PLAN EXISTENCE where to each instance $J = \langle VAR = \{v_1, v_2, \ldots, v_n\}, ACT, E, \vec{x}_0, G\rangle$ of general STRIPS PLAN EXISTENCE we associate the following instance $F(J) =$

[19] As in [11], the reason for this is that there are $|Mod(K)| \leq 2^{|VAR|}$ candidates and that if a dominance path has a length $> 2^{|VAR|-1}$ then it has at least a candidate appearing twice in it, and the path between its two occurrences can be safely deleted; repeating this process until no candidate appears twice in the dominance path gives a dominance path from $\vec{x}$ to $\vec{y}$ of length $\leq 2^{|VAR|-1}$.

[20] Note that this connection between CP-desires and STRIPS planning is already mentioned in [7], which by the way gives an immediate proof of membership to PSPACE of the restriction of CP-desires they consider.

[21] The assumption that each action is described by only one STRIPS effect rule can be done without loss of generality.

$\langle VAR', ACT', E', \vec{x}'_0, \vec{x}'_G\rangle$ of CGS STRIPS PLAN EXISTENCE where $VAR' = VAR \cup \{new\}$, where $new \notin VAR$; $ACT' = ACT \cup \{\alpha^*\}$; E' consists of adding to E the STRIPS description of α^*, namely: $\langle pre = G, post = v_1 \wedge \ldots v_n \wedge new\rangle$; $\vec{x}'_0 = \vec{x}_0 \wedge \neg new$; $\vec{x}_G = v_1 \wedge \ldots v_n \wedge new$. Clearly, π is a valid plan for J if and only if $(\pi; \alpha^*)$ is a valid plan for $F(J)$, from which PSPACE-hardness of COMPLETE-GOAL-STATE STRIPS PLAN EXISTENCE, and hence PSPACE-completeness, follows.
End of proof of Lemma 1

Coming back to the proof of Proposition 3, we now give a polynomial reduction from CGS STRIPS PLAN EXISTENCE to COMPARISON. Let us consider the following reduction H from in CGS STRIPS PLAN EXISTENCE to COMPARISON which, to any instance $I = \langle VAR = \{v_1, \ldots, v_n\}, ACT = \{\alpha_1, \ldots, \alpha_m\}, E, \vec{x}_0, \vec{x}_G\rangle$ of CGS STRIPS PLAN EXISTENCE) associates the following instance $H(I)$ of COMPARISON: $H(I) = \langle VAR', K = \top, \mathcal{D}, \vec{x}, \vec{y}\rangle$ where

– $VAR' = \{v_1, \ldots, v_n, t_1, \ldots, t_n\} = VAR \cup T$;

– $\mathcal{D} = \{C_{i,j}, i = 0, 1, \ldots m, j = \ldots n\}$, where the CP-desires $C_{i,j}$'s are defined as follows: for each $i \geq 1$, let $pre_i \setminus NEG(post_i)$ be the conjunction of all literals of pre_i whose negation is not in $post_i$ and let $pre_i \cap NEG(post_i)$ be the conjunction of all literals of pre_i whose negation is in $post_i$. Lastly, for $k = 1, \ldots, n$, let $\rho_k = \neg t_k \wedge t_{k-1} \wedge \ldots \wedge t_1$ (in particular, $\rho_1 = \neg t_1$) and $\xi_k = t_k \wedge \neg t_{k-1} \wedge \ldots \wedge \neg t_1$ (in particular, $\xi_1 = t_1$). Then, for each $i \geq 1$, $C_{i,j}$ is the CP-desire

$$(pre_i \setminus NEG(post_i)) : (pre_i \cap NEG(post_i)) \wedge \rho_j > post_i \wedge \xi_j$$

and for $i = 0$, $C_{0,j}$ is the *ceteris paribus* desire

$$\top : \rho_j > \xi_j$$

– $\vec{x} = \vec{x}_0 \wedge \neg t_1 \ldots \wedge \neg t_n$;

– $\vec{y} = \vec{x}_G \wedge t_1 \ldots \wedge t_n$.

It may help the reader seeing how this reduction works on an example. Let $VAR = \{v_1, v_2\}$, ACT is the set of 4 actions $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ described by their respective effects $E = \{\langle v_1, \neg v_1\rangle, \langle \neg v_1, v_1\rangle, \langle \neg v_1 \wedge v_2, \neg v_2\rangle, \langle v_1, v_2\rangle\}$, $\vec{s}_0 = (v_1, v_2)$, $\vec{s}_G = (\neg v_1, \neg v_2)\rangle$. Then
$H(I) = \langle\{v_1, v_2, t_1, t_2\}, \mathcal{D}, \vec{x}_1 = (v_1, v_2, \neg t_2, \neg t_1), \vec{v}_2 = (\neg v_1, \neg v_2, t_2, t_1)\rangle$, with $\{\mathcal{D} = C_{0,1}, C_{0,2}, C_{1,1}, C_{1,2}, C_{2,1}, C_{2,2}, C_{3,1}, C_{3,2}, C_{4,1}, C_{4,2}\}$, where

the *ceteris paribus* desires are $C_{i,j}$ are in the following table:

	$j=1$	$j=2$
$i=0$	$\top : \neg t_1 > t_1$	$\top : \neg t_2 \wedge t_1 > t_2 \wedge \neg t_1$
$i=1$	$\top : v_1 \wedge \neg t_1 > \neg v_1 \wedge t_1$	$\top : v_1 \wedge \neg t_2 \wedge t_1 > \neg v_1 \wedge t_2 \wedge \neg t_1$
$i=2$	$\top : \neg v_1 \wedge \neg t_1 > v_1 \wedge t_1$	$\top : \neg v_1 \wedge \neg t_2 \wedge t_1 > v_1 \wedge t_2 \wedge \neg t_1$
$i=3$	$\neg v_1 : v_2 \wedge \neg t_1 > \neg v_2 \wedge t_1$	$\neg v_1 : v_2 \wedge \neg t_2 \wedge t_1 > \neg v_2 \wedge t_2 \wedge \neg t_1$
$i=4$	$v_1 : \neg t_1 > v_2 \wedge t_1$	$v_1 : \neg t_2 \wedge t_1 > v_2 \wedge t_2 \wedge \neg t_1$

The role of the variables t_i is to ensure the consistency of $\mathcal{D}$. Without them, inconsistent sets of ceteris paribus constraints would be introduced when translating reversible actions such as the two actions α_1 and α_2: translating the action set above without the t_i's would give $\{\top : v_1 > \neg v_1, \top : \neg v_1 > v_1, \neg v_1 : v_2 > \neg v_2, v_1 : \top > v_2\rangle$, which contains a cycle and therefore is inconsistent[22].

We now have to show that I is a positive instance of COMPLETE-GOAL-STATE STRIPS PLAN EXISTENCE if and only if $H(I)$ is a positive instance of COMPARISON. The proof is based on the following lemma. First we need the following notations to be introduced:

- candidates, i.e. full assignments over $VAR \cup T$, are denoted as usually by $\vec{x}, \vec{x}'$ etc. Let $\vec{x} \in 2^{VAR \cup T}$, then we write $\vec{x} = (\vec{v}, \vec{t})$ where $\vec{v} \in 2^{VAR}$ and $\vec{t} \in 2^{T}$ are the projections of $\vec{x}$ on VAR and T respectively;

- for $0 \leq k \leq 2^n - 1$, let γ_k be the conjunction of literals over the variables $\{t_1, ..., t_n\}$ corresponding to the expression of the k in base 2: for instance, if $n = 4$, $\gamma_5 = (\neg t_4 \wedge t_3 \wedge \neg t_2 \wedge t_3)$; conversely, to each $\vec{t} \in 2^T$ we let $Int(\vec{t})$ be the integer such that $\vec{t} \equiv \gamma_{Int(t)}$; for instance, $Int(\neg t_4 \wedge t_3 \wedge \neg t_2 \wedge t_3) = 5$.

- for $\vec{v} \in 2^{VAR}$ and $\alpha \in ACT$, $next(\vec{v}, \alpha)$ denotes the state obtained after performing α in state $\vec{v}$ according to the STRIPS action description of α in E.

[22] This problem was pointed to me by Carmel Domshlak (personal communication). Actually, we could cope with such inconsistent *ceteris paribus* desires, since the definition of $>_{\mathcal{D}}$ is not model-theoretic, so that the set of desires does not trivialize : for instance, from the inconsistent set of desires $\mathcal{D} = \{\top : a > \neg a, \top : \neg a > a, a : b > \neg b\}$, using Definition 4 we get for example the cycle $(a, b) > (\neg a, b) > (a, b)$ but nevertheless there is no way to induce, by transitive closure, the preference $(a, \neg b) > (a, b)$. This treatment of local inconsistencies in *ceteris paribus* sets of preferences (see [20] for a discussion in the more specific framework of CP-nets) has a paraconsistent flavour; this issue is not considered further in this paper, and we go on assuming that sets of constraints are consistent, i.e., cycle-free.

LEMMA 2. *For any* $\vec{x} = (\vec{v}, \vec{t}), \vec{x}' = (\vec{v}\,', \vec{t}\,') \in 2^{VAR \cup T}$ *and for any* $i \geq 1$, $\vec{x} > \vec{x}'$ *is a direct (i.e., within a single application) consequence of one of the CP-desires* $C_{i,j}$ *(for some* j*) if and only if the following two conditions hold:*

1. $\vec{v}\,' = next(\vec{v}, \alpha_i)$;
2. $K(\vec{t}\,') = K(\vec{t}) + 1$.

Proof of Lemma 2: let $i \in 1 \dots p$ and $j \in 1 \dots n$ and let us note $\vec{x} >_{C_{i,j}} \vec{x}'$ for "$\vec{x} > \vec{x}'$ is a direct consequence of $C_{i,j}$". Then

$$\vec{x} >_{C_{i,j}} \vec{x}'$$

$$\Leftrightarrow$$

$$\begin{cases} \vec{x} \models (pre_i \setminus NEG(post_i)) \wedge (pre_i \cap NEG(post_i)) \wedge \rho_j \wedge \neg(post_i \wedge \xi_j) \\ \vec{x}' \models (pre_i \setminus NEG(post_i)) \wedge post_i \wedge \xi_j \wedge \neg((pre_i \cap NEG(post_i)) \wedge \rho_j) \\ \vec{x} \text{ and } \vec{x}' \text{ coincide on all variables not in } post_i, \xi_j, \rho_j \end{cases}$$

$$\Leftrightarrow$$

$$\begin{cases} \vec{x} \models pre_i \wedge \rho_j \wedge (\neg post_i \vee \neg \xi_j) \\ \vec{x}' \models (pre_i \setminus NEG(post_i)) \wedge post_i \wedge \xi_j \wedge (\neg(pre_i \cap NEG(post_i)) \vee \neg \rho_j) \\ \vec{x} \text{ and } \vec{x}' \text{ coincide on all variables not in } post_i, \xi_j, \rho_j \end{cases}$$

Using the fact that $\rho_j \wedge \xi_j$ is inconsistent, this is equivalent to (1) and (2) where

$$(1) : \begin{cases} \vec{v} \models pre_i \\ \vec{v}\,' \models (pre_i \setminus NEG(post_i)) \wedge post_i \\ \vec{v} \text{ and } \vec{v}\,' \text{ coincide on all variables not in } post_i \end{cases}$$

and

$$(2) : \begin{cases} \vec{t} \models \rho_j \\ \vec{t}\,' \models \xi_j \\ \vec{t} \text{ and } \vec{t}\,' \text{ coincide on all variables not in } \rho_j, \xi_j \end{cases}$$

Now, (1) is equivalent to $\vec{v}\,' = next(\vec{v}, \alpha_i)$ and (2) is equivalent to $Int(\vec{t}) = 2^j.k + 2^{j-1} - 1$ for some integer k and $Int(\vec{t}\,') = Int(\vec{t}) + 1$. From this we conclude that $\vec{x} >_{C_{i,j}} \vec{x}'$ for some j if and only if $\vec{v}\,' = next(\vec{v}, \alpha_j)$ and $K(\vec{t}\,') = K(\vec{t}) + 1$, which proves Lemma 2. *End of proof of Lemma 2.*

Now, assume there is a valid plan from $\vec{x}_0$ to $\vec{x}_G$ w.r.t. ACT; then it is known (see [11]) that there exists then a valid plan $\vec{x}_0$ to $\vec{x}_G$ w.r.t. ACT, whose length is bounded by $2^n - 1$. Then, Lemma 2 tells us that there is a dominance chain w.r.t. $\mathcal{D}$ from $\vec{x}_0, \neg t_0, \ldots, \neg t_n)$ to $(\vec{x}_G, \vec{t})$ for some $\vec{t}$. Now, applying the constraints $C_{0,j}$ exactly $2^n - 1 - K(\vec{t})$ times (note that for each $\vec{t}$ there is one and only one constraint $C_{0,j}$ applicable), we get then a dominance chain from $\vec{x}_0$ to $(\vec{v}_G, t_1, \ldots, t_n)$. Conversely, if there is a dominance chain

$$\vec{x}_0 > (\vec{v}_1, \vec{t}_1) > \ldots > (\vec{v}_G, (t_1, \ldots, t_n))$$

then for each i, either $\vec{v}_{i+1} = \vec{v}_i$ or $\vec{v}_{i+1} = next(\vec{v}_i, \alpha_j)$ for some $\alpha_j \in ACT$, and the obtained sequence of actions is a valid plan from $\vec{v}_0$ to $\vec{v}_G$ w.r.t. ACT. Henceforth, H is a polynomial reduction from COMPLETE-GOAL-STATE STRIPS PLAN EXISTENCE to COMPARISON, which completes the proof of Proposition 3. ■

2. COMPARISON *is still* PSPACE*-hard under the restriction that desires are simple.*

Proof: This is shown by the following polynomial reduction from the COMPARISON problem in the general case: let $I = \langle VAR, K, \mathcal{D}, \vec{x}, \vec{y}\rangle$ be an instance of COMPARISON, where $\mathcal{D} = \{D_1, \ldots, D_m\}$ where $D_i = (C_i : G_i > G'_i[V_i])$. Let $J = G(I) = \langle VAR', K', \mathcal{D}', \vec{x}', \vec{y}'\rangle$ where

- $VAR' = VAR \cup \{p_1, \ldots, p_m\}$, where the p_i are new variables;
- $K' = K \cup \bigwedge\{p_i \leftrightarrow (G_i \wedge \neg G'_i) | i = 1, \ldots, m\}$;
- $\mathcal{D}' = \{D'_1, \ldots, d'_m\}$ where D'_i is $C_i \wedge (G_i \leftrightarrow \neg G'_i) : p_i > \neg p_i[V_i]$;
- $\vec{x}'$ and $\vec{y}'$ are the respective extensions of $\vec{x}$ and $\vec{y}$ to the alphabet VAR' (this is unambiguous since the values of the new variables p_i are completely determined from $\vec{x}$ – and similarly from $\vec{y}$).

Suppose that there is a dominance path $\vec{x}_0 = \vec{x} > \ldots \vec{x}_q = \vec{y}$. For any $i \in \{0, \ldots, q-1\}$, let j be the CP-desire such that $\vec{x}_i >_{D_j} \vec{x}i + 1$. We have $\vec{x}_i \models C_j \wedge (G_j \wedge \neg G'_j)$ and $\vec{x}_i \models C_j \wedge (\neg G_j \wedge G'_j)$, therefore, if $\vec{x}'_i$ and $\vec{x}'_{i+1}$ denote the extensions of $\vec{x}'_i$ and $\vec{x}'_{i+1}$ to VAR', we have: $\vec{x}'_i \models C_j \wedge (G_j \leftrightarrow \neg G'_j) \wedge p_j$; $\vec{x}_{i+1} \models C_j \wedge (G_j \leftrightarrow \neg G'_j) \wedge \neg p_j$; lastly, $\vec{x}'_i$ and $\vec{x}'_{i+1}$ coincide on all variables not in V_j. Therefore, we have $\vec{x}_i >_{D'_j} \vec{x}_{i+1}$ and $\vec{x}'_0 = \vec{x}' > \ldots \vec{x}'_q = \vec{y}'$ is a dominance path from $\vec{x}'$ to $\vec{y}'$. The converse (any dominance path from $\vec{x}'$ to $\vec{y}'$ induces a dominance path from $\vec{x}'$ to $\vec{y}'$) works exactly in the same way. This proves the equivalence of the two instances I and $G(I)$ of COMPARISON – and the latter involves only simple goals. ■

3. in the general case, NON-DOMINANCE *is in* coNP.

Proof: Membership to coNP comes from the following property: $\vec{x}$ is dominated if and only if there is a CP-desire $D_i = (C_i : G_i > G'_i[V_i])$ in $\mathcal{D}$ and a $\vec{y}$ such that $\vec{y} >_{D_i} \vec{x}$. Given $\vec{y}$, checking that $\vec{y} >_{D_i} \vec{x}$, i.e., that (1) $\vec{x} \models K \wedge C_i \wedge \neg G_i \wedge G'_i$, (2) $\vec{y} \models K \wedge C_i \wedge G_i \wedge \neg G'_i$ and (3) $\vec{x}$ and $\vec{y}$ coincide on V_i, can be done in polynomial time; hence the membership of NON-DOMINANCE to coNP. ∎

4. under the restriction that desires are simple, NON-DOMINANCE *is* coNP*-hard.*

Proof: coNP-hardness comes from the fact that the preference representation language based on simple desires generalizes the basic representation (and recall that NON-DOMINANCE is coNP-complete for R_{basic}). Indeed, it is straightforward to verify that any instance of NON-DOMINANCE for R_{basic} defined by a set of variables VAR, two formulas K, G and a candidate $\vec{x}$ is equivalent to the instance of NON-DOMINANCE for R_{cp} defined by VAR, K, the single CP-desire $G > \neg G[VAR]$ and $\vec{x}$. ∎

5. in the general case, CAND-OPT-SAT *is in* Σ_2^p*.*

This comes immediately from the fact that NON-DOMINANCE is in coNP.

6. under the restriction that desires are simple, CAND-OPT-SAT *is* Σ_2^p*-hard.*

Proof: Σ_2^p-hardness is shown by the following polynomial reduction from standard base revision (inclusion-based and without priorities): let $B = \{\phi, \ldots, \phi_n\}$ be a set of propositional formulas (a *belief base*) then $\beta \in K \star \alpha$ if and only if for any subset B' of B such that $B' \wedge \alpha$ is consistent and there is no strict subset B'' of B' such that $B' \wedge \alpha$, we have $B' \wedge \alpha \models \beta$. Now, we map $\langle B, \alpha, \beta \rangle$ to the following instance of CAND-OPT-SAT for R_{cp}: let $p_1, \ldots, p_n$ be n new propositional symbols, let $V = Var(K) \cup Var(\alpha) \cup Var(\beta)$; $K = \alpha \wedge \bigwedge \{p_i \leftrightarrow \phi_i | i = 1, \ldots, n\}$; $\mathcal{D} = \{\top : p_1 > \neg p_1[V], \ldots, \top : p_n > \neg p_n[V]\}$ and $\psi = \beta$. $\vec{x}$ is undominated for $\mathcal{D}$ if and only if $\vec{x}$ satisfies a maximal consistent subset of $\{\phi_1, \ldots, \phi_n\}$, which enables us to conclude. ∎

Proposition 4 *Let R be a language for which* NON-DOMINANCE *is in* NP*or in* coNP*.*

1. AMONG-WINNERS$_{plurality,R}$ *is in* Θ_3^p*;*
2. ELECT-SAT$_{plurality,R}$ *is in* Θ_3^p*;*

3. for $R = R_{basic}$, AMONG-WINNERS$_{plurality,R_{basic}}$ *is* coNP*-complete and* ELECT-SAT$_{plurality,R_{basic}}$ *is* Θ_2^p*-complete.*

Proof:

AMONG-WINNERS$_{plurality,R}$ *is in* Θ_3^p.
Let $Score_B(\vec{x}) = \sum_{i=1...n} s_i(\vec{x}) = |\{i, \vec{x} \text{ undominated for } GB_i\}|$. Since $s_i(\vec{x}) = 1$ if and only if $\vec{x}$ is undominated for GB_i (and $s_i(\vec{x}) = 0$ otherwise), and since NON-DOMINANCE is in NP or in coNP, computing $s_i(\vec{x})$ amounts to an NP oracle. Therefore, $Score_B(\vec{x})$ can be computed within n NP oracles. Given an integer k, the problem (P) of determining whether there exists a candidate $\vec{x}$ such that $Score_B(\vec{x}) \geq k$ is in Σ_2^p. Now, let $S^* = \max_{\vec{y} \in \mathcal{C}} Score_B(\vec{y})$; since $0 \leq S^* \leq n$, S^* can be computed by dichotomy by solving $\lceil \log_2 n + 1 \rceil$ problems (P). Now, $\vec{x}$ is a plurality winner for B if and only if $Score_B(\vec{x}) = S^*$. The following algorithm then determines whether $\vec{x}$ is a plurality winner for B : 1. compute S^*; 2. check that $Score_B(\vec{x}) = S^*$. Step 1 requires a logarithmic numbers of calls to Σ_2^p oracles, and step 2 is in Δ_2^p, therefore in Σ_2^p. This shows membership of AMONG-WINNERS$_{\text{plurality},R}$ to Θ_3^p.

ELECT-SAT$_{plurality,R}$ *is in* Θ_3^p.
The following algorithm determines whether there exists a plurality winner $\vec{x}$ for B satisfying ψ : 1. compute $S^* = \max_{\vec{y} \in \mathcal{C}} Score_B(\vec{y})$; 2. guess $\vec{x}$; 3. check that $Score_B(\vec{x}) = S^*$; 4. check that $\vec{x} \models \psi$. Steps 3-4 require a polynomial number of NP oracles, hence, steps 2-4 amount to an oracle Σ_2^p; step 1 requires a logarithmic numbers of calls to Σ_2^p oracles; therefore 1-4 requires a logarithmic numbers of calls to oracles Σ_2^p, which shows membership to Θ_3^p.

AMONG-WINNERS$_{plurality,R_{basic}}$ *is* coNP*-complete.*
Let $B = \langle G_1, ..., G_n \rangle$. Let $N_B(\vec{x}) = |\{i | \vec{x} \models G_i\}|$. We have $Score_B(\vec{x}) = N_B(\vec{x}) + |\{i | G_i \text{ unsatisfiable}\}|$; therefore, $\vec{x}$ is a plurality winner for B if and only if $\forall \vec{y} \in \mathcal{C}, N_B(\vec{x}) \geq N_B(\vec{y})$; since $N_B(\vec{x})$ is computed in polynomial time, determining whether $\vec{x}$ is a plurality winner for B is in coNP. coNP-hardness comes from the following polynomial reduction from UNSAT: for any propositional formula φ , let $B = \{\varphi \wedge p\}$ where p is a new propositional symbol, and let $\vec{x}$ be any candidate such that $\vec{x} \models \neg p$. We have $N_B(\vec{x}) = 0$. If φ is satisfiable then for any $\vec{y}$ such that $\vec{y} \models \varphi \wedge p$ (and there exist some, since φ is satisfiable and so is $\varphi \wedge p$) we have $N_B(\vec{y}) = 1$, therefore $\vec{x}$ is not a plurality winner. If φ is unsatisfiable then $\forall \vec{y} \in \mathcal{C}, N_B(\vec{y}) = 0$ and thus $\vec{x}$ is a plurality winner. This shows that $\vec{x}$ is a plurality winner for B if and only if φ is unsatisfiable.

ELECT-SAT$_{plurality,R_{basic}}$ *is* Θ_2^p*-complete.*
Both membership and hardness come easily from the Θ_2^p-complete problem CARDINALITY-MAXIMIZING BASE REVISION [42], abbreviated in CMBR. There is a plurality-winner satisfying ψ if and only if $\psi \in B \star_c \top$, where B is considered as a belief base (without priorities) in the sense of [42], and $\star_c$ denotes cardinality-maximizing base revision; this shows membership to Θ_2^p. As to hardness, to any triple $\langle B, \varphi, \psi \rangle$ we associate $F(B, \varphi, \psi) = \langle K, B, \psi' \rangle$ where $K = \varphi$, $(B = B)$, and $\psi' = \neg\psi$. (B, φ, ψ) is a positive instance of ELECT-SAT if and only if there exists $\vec{x}$ such that $N_B(\vec{x})$ is maximal in $Mod(K)$ and $\vec{x} \models \psi$, i.e., if and only if there is a $B' \subseteq B$, such that (i) $B' \wedge \varphi$ is consistent, (ii) there is no $B'' \subseteq B$ such that $B'' \wedge \varphi$ is consistent and $|B''| \geq |B'|$, and (iii) $B' \wedge \neg\psi$ is consistent. This is equivalent to $\psi \notin B \star_c \varphi$. Now, CMBR being Θ_2^p-complete and Θ_2^p being closed under complement, $\overline{\text{CMBR}}$ is also Θ_2^p-complete. Now, since $\langle B, \varphi, \psi \rangle$ is a negative instance of CMBR if and only if $F(B, \varphi, \psi)$ is a positive instance of ELECT-SAT, the latter problem is Θ_2^p-hard. ■

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

1. CW VERIFICATION, NCW VERIFICATION *and* WCW VERIFICATION *are in* coNP*; when $R = R_{basic}$,* CW VERIFICATION *and* NCW VERIFICATION *are* coNP*-complete.*

2. CW EXISTENCE, NCW EXISTENCE *and* WCW EXISTENCE *are all in* Σ_2^p*; when $R = R_{basic}$,* CW VERIFICATION *is in* Θ_2^p *and* coNP*-hard;* NCW VERIFICATION *is in* Θ_2^p *and* BH_2*-hard.*

Proof:

Verification problems, general case.
Since COMPARISON in R is polynomial, $C(\vec{x}, \vec{y}, B)$ can be computed in polynomial time; therefore, the verification that $\vec{x}$ is a CW, which amounts to checking that there is no $\vec{y}$ such that $C(\vec{x}, \vec{y}, B) \geq C(\vec{y}, \vec{x}, B)$, is in coNP. This is very similar with WCW (replacing $\geq$ by $>$ in the latter inequality). As to NCW, checking that $\vec{x} \sim_{GB_i} \vec{y}$ can also be done in polynomial time because COMPARISON is polynomial, and by similar considerations as above, the verification that $\vec{x}$ is a NCW for B is in coNP.

Verification problems, $R = R_{basic}$.
coNP-hardness for *cw verification* is shown by exhibiting a polynomial reduction from UNSAT. For any propositional formula φ we let $F(\varphi) = \langle K, B, \vec{x} \rangle$ where p_1, p_2 are two propositional variables not appearing

in φ ; $K = \top$; $\vec{x}$ is any candidate satisfying $p_2 \wedge \neg p_1$; $B = \langle \mathrm{G}_1, \mathrm{G}_2 \rangle$ with $\mathrm{G}_1 = \varphi \wedge p_1$ and $\mathrm{G}_2 = for(\vec{x})$. Let $\vec{y} \neq \vec{x}$. We have $C(\vec{x}, \vec{y}, B) - C(\vec{x}, \vec{y}, B) = \begin{cases} 0 & \text{if } \vec{y} \models \varphi \wedge p_1 \\ +1 & \text{if } \vec{y} \models \neg(\varphi \wedge p_1) \end{cases}$. Therefore, $\vec{x}$ is a CW for B if and only if there is no $\vec{y}$ such that $\vec{y} \models \varphi \wedge p_1$, i.e., if and only if φ is inconsistent. This proves that CW VERIFICATION for $R = R_{basic}$ is coNP-hard (and therefore coNP-complete). The very same reduction also shows coNP-hardness for *ncw verification* : $\vec{x}$ is a CW for B if and only if there is no $\vec{y}$ such that (1) $C(\vec{x}, \vec{y}, B) - C(\vec{x}, \vec{y}, B) \geq 0$ or (2) $\vec{y} \sim_{GB_i} \vec{x}$ holds for $i = 1, 2$; now, (2) cannot be verified by any $\vec{y} \neq \vec{x}$, since $\vec{x}$ is the unique model of G_2; therefore, $\vec{x}$ is a NCW for B if and only if it is a CW for B , i.e., if and only if φ is inconsistent.

Existence problems, general case.
The Σ_2^p-membership result comes directly from the membership of the verification problem to coNP: it suffices to guess a candidate and show that it is a CW (resp. NCW, WCW) for B .

CW existence problems, $R = R_{basic}$
Let $B = \{G_1, \ldots, G_n\}$ and $N(\vec{x}, B) = |\{i | \vec{x} \models G_i\}|$. $\vec{x}$ is a CW for B if and only if there is no $\vec{y} \neq \vec{x}$ such that $N(\vec{y}, B) \geq N(\vec{x}, B)$ – in other words, if $\vec{x}$ is the *unique* candidate maximizing the number of formulas of B satisfied. Recall now that the maximal number of formulas from $\{G_1, \ldots, G_n\}$ that are simultaneously satisfiable is in Θ_2^p can be computed using $\mathcal{O}(\log n)$ NP oracles[23]. Now, verifying that $\vec{x}$ is a CW for B can be done by the following algorithm:

1. compute the maximal number k of simultaneously satisfiable formulas in $\{G_1, \ldots, G_n\}$;
2. guess $\vec{x}$;
3. check that $N(\vec{x}, B) = k$.
4. check that there are no two distinct candidates $\vec{y}$, $\vec{z}$ such that $N(\vec{y}, B) = N(\vec{z}, B) = k$.

Steps 1 requires a logarithmic NP oracles; step 2 requires a single NP oracle; step 3 can be done in polynomial time; step 4 is a problem in coNPand need thus a single NP oracle. Hence the membership to Θ_2^p.

[23] It is actually well-known result that the considered problem is $\mathsf{F}\Theta_2^p$-complete – see for instance [44] – this is due to the fact that the problem of determining whether there is a candidate $\vec{x}$ such that $\vec{x}$ satisfies at least k formulas in $\{G_1, \ldots, G_n\}$ is in NP: therefore, the maximum number of simultaneously satisfiable formulas can be computed by dichotomy on $\{0, \ldots, N\}$.

coNP-hardness comes from this polynomial reduction from UNSAT: for any given φ we consider $K = \top$ and $B = \langle G_1 \rangle$ with $G_1 = (\varphi \wedge \neg p) \vee for(\vec{x})$ where p is a new propositional symbol and $\vec{x}$ any candidate satisfying p. Since $\vec{x} \models G_1$, if there exists a CW for B, this is $\vec{x}$; now, $\vec{x}$ is a CW for B if and only if G_1 has no other model than $\vec{x}$, which is the case if and only if φ is unsatisfiable.

NCW existence problems, $R = R_{basic}$ $\vec{x}$ is a NCW for B if and only if $\vec{x}$ maximizes $N(\vec{x}, B)$ and for any candidate $\vec{y} \neq \vec{x}$ such that $N(\vec{y}, B) = N(\vec{x}, B)$, we have $\{i, \vec{y} \models G_i\} = \{i, \vec{x} \models G_i\}$. Therefore, there is a NCW for B if and only if there is a unique consistent subset of B of maximum cardinality (let us call such a subset a *maxcard consistent subset* of B). Now, the membership proof is similar to the one for CW existence, the last step being replaced by

4' check that there are no two distinct candidates $\vec{y}$, $\vec{z}$ such that $N(\vec{y}, B) = N(\vec{z}, B) = k$ and $\vec{y} \not\sim_B \vec{z}$.

BH_2-hardness is shown by the following polynomial reduction from SAT-UNSAT: for any pair $\langle \varphi, \psi \rangle$ we let $G(\langle \varphi, \psi \rangle) = B = \langle \varphi \vee p, \varphi \vee \neg p, \psi \wedge q, \psi \wedge \neg q \rangle$. If both φ and ψ are satisfiable then B has two maxcard consistent subsets, namely $\{\varphi \vee p, \varphi \vee \neg p, \psi \wedge q\}$ and $\{\varphi \vee p, \varphi \vee \neg p, \psi \wedge \neg q\}$. If φ is satisfiable and ψ is unsatisfiable then B has a unique maxcard consistent subset, namely $\{\varphi \vee p, \varphi \vee \neg p\}$. If φ is unsatisfiable and ψ is satisfiable then B has four maxcard consistent subsets, namely $\{\varphi \vee p, \psi \wedge q\}$, $\{\varphi \vee p, \psi \wedge \neg q\}$, $\{\varphi \vee \neg p, \psi \wedge q\}$,$\{\varphi \vee \neg p, \psi \wedge \neg q\}$. If φ and ψ are both unsatisfiable then B has two maxcard consistent subsets, namely $\{\varphi \vee p\}$ and $\{\varphi \vee \neg p\}$. Therefore, B has a unique maxcard consistent subset if and only if φ is satisfiable and ψ is unsatisfiable, i.e., if and only if $\langle \varphi, \psi \rangle$ is an instance of SAT-UNSAT. ■

Proposition 6 *Let R be a language for which* COMPARISON *is polynomial.*

1. AMONG-WINNERS$_{Simpson,R}$ *is in* Θ_3^p.
2. ELECT-SAT$_{Simpson,R}$ *in* Θ_3^p.
3. *For $R = R_{basic}$:* AMONG-WINNERS$_{Simpson,R_{basic}}$ *is* coNP*-complete and* ELECT-SAT$_{Simpson,R_{basic}}$ *is* Θ_2^p*-complete.*

Proof: Let $S(\vec{x}, B) = \min_{\vec{y} \neq \vec{x}} C(\vec{x}, \vec{y}, B) - C(\vec{y}, \vec{x}, B)$.

AMONG-WINNERS$_{Simpson,R}$ *is in* Θ_3^p.
$C(\vec{x}, \vec{y}, B)$ is computed in polynomial time, therefore the problem of

determining whether $S(\vec{x}, B) \geq k$, which is equivalent to the non-existence of a $\vec{y}$ such that $C(\vec{x}, \vec{y}, B) \geq k$, is in coNP. Since $-n \leq S(\vec{x}, B) \leq n$, $S(\vec{x}, B)$ can be computed using $\lceil log_2(2n) \rceil$ oracles NP. The following algorithm then determines whether $\vec{x}$ is a Simpson winner for B : 1. compute $S^* = \max_{\vec{z} \in \mathcal{C}} S(\vec{z}, B)$; 2. check that $S(\vec{x}, B) = S^*$. Then, by similar considerations as those used for plurality winners, we conclude that AMONG-WINNERS$_{\text{Simpson},R}$ is in Θ_3^p.

ELECT-SAT$_{Simpson,R}$ *is in* Θ_3^p.
Similar to the proof of Θ_3^p-membership of ELECT-SAT$_{\text{plurality},R}$.

AMONG-WINNERS$_{Simpson,R_{basic}}$ *is* coNP*-complete and* ELECT-SAT$_{Simpson,R_{basic}}$ *is* Θ_2^p*-complete.*
Let $B = \langle \{G_1\}, ..., \{G_n\} \rangle$. For any two candidates $\vec{x}$ and $\vec{y}$, we have $C(\vec{x}, \vec{y}, , B) = N(\vec{x}, , B) - N(\vec{y}, B)$, where $N(\vec{z}, B) = |\{i | \vec{z} \models G_i\}|$, and therefore $S(\vec{x}, B) = \min_{\vec{y} \in \mathcal{C}} (N(\vec{x}, B) - N(\vec{y}, B)) = N(\vec{x}, B) - \min_{\vec{y} \in \mathcal{C}} N(\vec{y}, B))$. Therefore, $\vec{x}$ is a Simpson winner for B if and only if it maximizes $N(\vec{x}, B)$, i.e., if and only if it is a plurality winner for B. The results are therefore corollaries of point 2 of Proposition 6. ■

References

1. Bacchus, F. and A. Grove: 1995, 'Graphical models for preference and utility'. In: *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*. pp. 3–10.
2. Bacchus, F. and A. Grove: 1996, 'Utility independence in a qualitative decision theory'. In: *Proceedings of the* 5^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*. pp. 542–552.
3. Benferhat, S., C. Cayrol, D. Dubois, J. Lang, and H. Prade: 1993, 'Inconsistency management and prioritized syntax- based entailment'. In: *Proceedings of the* 13^{th} *International Joint Conference on Artificial Intelligence (IJCAI'93)*. pp. 640–645.
4. Benferhat, S., D. Dubois, S. Kaci, and H. Prade: 2002, 'Bipolar representation and fusion of preference in the possibilistic logic framework'. In: *Proceedings of the* 11^{th} *European Conference on Artificial Intelligence (ECAI2002)*. pp. 421–429.
5. Boutilier, C.: 1994, 'Toward a logic for qualitative decision theory'. In: *Proceedings of the* 4^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*. pp. 75–86.
6. Boutilier, C., F. Bacchus, and R. Brafman: 2001, 'UCP-networks: a directed graphical representation of conditional utilities'. In: *Proceedings of the 17th Conf. on Uncertainty in Artificial Intelligence (UAI-2001)*. pp. 56–64.

7. Boutilier, C., R. Brafman, H. Hoos, and D. Poole: 1999, 'Reasoning with conditional *ceteris paribus* statements.'. In: *Proceedings of the 15th Conf. on Uncertainty in Artificial Intelligence (UAI'99)*. pp. 71–80.
8. Boutilier, C. and H. Hoos: 2001, 'Bidding languages for combinatorial auctions'. In: *Proceedings of the* 17^{th} *International Joint Conference on Artificial Intelligence (IJCAI'01)*. pp. 1211–1217.
9. Brafman, R. and N. Friedman: 1995, 'On decision-theoretic foundations for defaults'. In: *Proceedings of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'95)*. pp. 1458–1465.
10. Brewka, G., S. Benferhat, and D. Le Berre: 2002, 'Qualitative choice logic.'. In: *Proceedings of the* 8^{th} *International Conference on Knowledge Representation and Reasoning (KR2002)*. pp. 158–169.
11. Bylander, T.: 1994, 'The computational complexity of propositional STRIPSplanning.'. *Artificial Intelligence* (69), 161–204.
12. Cadoli, M., F. Donini, P. Liberatore, and M. Schaerf: 1996, 'Comparing space efficiency of propositional knowledge representation formalisms'. In: *Proceedings of the* 5^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*. pp. 364–373.
13. Cai, J.-Y., T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. W. Wagner, and G. Wechsung.: 1988, 'The Boolean hierarchy I: structural properties.'. *SIAM Journ. on Computing* **17**(6), 1232–1252.
14. Cayrol, C., M. Lagasquie-Schiex, and T. Schiex: 1998, 'Nonmonotonic reasoning: from complexity to algorithms'. *Annals of Mathematics and Artificial Intelligence* **22**(3-4), 207–236.
15. Conitzer, V., J. Lang, and T. Sandholm: 2003, 'How many candidates are required to make an election hard to manipulate?'. In: *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-03)*. pp. 201–214.
16. Conitzer, V. and T. Sandholm: 2002a, 'Complexity of manipulating elections with few candidates'. In: *Proceedings of the* 18^{th} *National Conference on Artificial Intelligence (AAAI-02)*. pp. 314–319.
17. Conitzer, V. and T. Sandholm: 2002b, 'Vote elicitation: complexity and strategy-proofness'. In: *Proceedings of the* 18^{th} *National Conference on Artificial Intelligence (AAAI-02)*. pp. 392–397.
18. Coste-Marquis, S., J. Lang, P. Liberatore, and P. Marquis: 2003, 'Expressivity and succinctness power of preference representation language'. Manuscript.
19. de Condorcet, M.: 1785, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Paris.
20. Domshlak, C.: 2002, 'Modelling and reasoning about preferences with CP-nets'. Ph.D. thesis, Ben-Gurion University.
21. Domshlak, C. and R. Brafman: 2002, 'CP-nets: reasoning and consistency testing'. In: *Proceedings of the* 8^{th} *International Conference on Knowledge Representation and Reasoning (KR2002)*. pp. 121–132.
22. Doyle, J. and R. Thomason: 1999, 'Background to qualitative decision theory'. *AI Magazine* **20**, 55–68.
23. Doyle, J. and M. P. Wellman: 1991, 'Preferential semantics for goals'. In: *Proceedings of AAAI-91*. pp. 698–703.
24. Doyle, J., M. P. Wellman, and Y. Shoham: 1991, 'A logic of relative desire (preliminary report)'. In: *Proceedings of the 6th Intern. Symposium on Methodologies for Intelligent Systems*. pp. 16–31.

25. Dubois, D., H. Fargier, and P. Perny: 2002, 'On the limits of ordinality in decision making'. In: *Proceedings of the* 8^{th} *International Conference on Knowledge Representation and Reasoning (KR2002)*. pp. 133–144.
26. Eiter, T. and G. Gottlob: 1992, 'On the complexity of propositional knowledge base revision, updates and counterfactuals'. *Artificial Intelligence* **57**, 227–270.
27. Eiter, T. and T. Lukasiewicz: 2000, 'Complexity results for default reasoning from conditional knowledge bases'. In: *Proceedings of KR-2000*. pp. 62–73.
28. Friedman, N. and J. Halpern: 1994, 'On the complexity of conditional logics'. In: *Proceedings of the* 4^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*. pp. 202–213.
29. Geffner, H.: 1992, 'Default reasoning : causal and conditional theories'. *MIT Press*.
30. Konieczny, S., J. Lang, and P. Marquis: 2002, 'Distance-based merging: a general framework and some complexity results'. In: *Proceedings of the* 8^{th} *International Conference on Knowledge Representation and Reasoning (KR2002)*. pp. 97–108.
31. Konieczny, S. and R. Pino-Pérez: 1998, 'On the logic of merging'. In: *Proc. of KR'98*. pp. 488–498.
32. Kraus, S., D. Lehmann, and M. Magidor: 1990, 'Nonmonotonic reasoning, Preferential Models and Cumulative Logics'. *Artificial Intelligence* **44**, 167–207.
33. La Mura, P. and Y. Shoham: 1999, 'Expected utility networks'. In: *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)*. pp. 366–373.
34. Lafage, C. and J. Lang: 2000, 'Logical representation of preferences for group decision making'. In: *Proceedings of the* 7^{th} *International Conference on Knowledge Representation and Reasoning (KR2000)*. pp. 457–468.
35. Lafage, C. and J. Lang: 2001, 'Propositional distances and preference representation'. In: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2001)*, Vol. 2143 of *Lecture Notes in Computer Science*. Springer Verlag, pp. 48–59.
36. Lang, J.: 1996, 'Conditional desires and utilities - an alternative logical approach to qualitative decision theory'. In: *Proceedings of the* 11^{th} *European Conference on Artificial Intelligence (ECAI'96)*. pp. 318–322.
37. Lang, J. and P. Marquis: 1998, 'Complexity results for independence and definability in propositional logic'. In: *Proceedings of the* 6^{th} *International Conference on Knowledge Representation and Reasoning (KR'98)*. 356–367.
38. Lang, J., L. van der Torre, and E. Weydert: 2002, 'Utilitarian desires'. *International Journal on Autonomous Agents and Multi-Agent Systems* **5**, 329–363.
39. Liberatore, P. and M. Schaerf: 2002, 'BReLS: a system for the integration of knowledge bases.'. In: *Proceedings of the* 7^{th} *International Conference on Knowledge Representation and Reasoning (KR2000)*. pp. 145–152.
40. Moulin, H.: 1988, *Axioms of Cooperative Decision Making*. Cambridge University Press.
41. Nebel, B.: 1991, 'Belief Revision and Default Reasoning: Syntax-Based approaches'. In: *Proceedings of KR'91*. pp. 417–428.
42. Nebel, B.: 1998, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, Chapt. How hard is it to revise a knowledge base? Kluwer Academics.
43. Nisan, N.: 2000, 'Bidding and allocations in combinatorial auctions'. In: *ACM Conference on Electronic Commerce (EC-2000)*.

44. Papadimitriou, C. H.: 1994, *Computational complexity*. Addison–Wesley.
45. Pearl, J.: 1990, 'System Z : a natural ordering of defaults with tractable applications for default reasoning.'. In: M. Vardi (ed.): *Proceedings of theoretical aspects of reasoning about knowledge*. San Mateo, pp. 121–135.
46. Sandholm, T.: 1999, 'An Algorithm for Optimal Winner Determination in Combinatorial Auctions,'. In: *Proceedings of the* 16^{th} *International Joint Conference on Artificial Intelligence (IJCAI'99)*. pp. 542–547.
47. Schiex, T., H. Fargier, and G. Verfaillie: 1995, 'Valued constraints satisfaction problems : hard and easy problems'. In: *Proceedings of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'95)*. pp. 631–637.
48. Shoham, Y.: 1988, *Reasoning about change – Time and Causation from the Standpoint of Artificial Intelligence*. MIT Press.
49. Tan, S. and J. Pearl: 1994, 'Specification and evaluation of preferences for planning under uncertainty'. In: *Proceedings of KR-94*.
50. Thomason, R.: 2000, 'Desires and defaults: a framework for planning with inferred goals'. In: *Proceedings of the* 7^{th} *International Conference on Knowledge Representation and Reasoning (KR2000)*. pp. 702–713.
51. van der Torre, L. and E. Weydert: 2001, 'Parameters for utilitarian desires in a qualitative decision theory'. *Applied Intelligence* **14**(3), 285–302.
52. Wooldridge, M. and S. Parsons: 2000, 'Languages for negociation'. In: *Proceedings of the* 14^{th} *European Conference on Artificial Intelligence (ECAI2000)*. pp. 393–397.