Complexity Results in Optimistic/Pessimistic Preference Reasoning

Christian Bessiere*, Remi Coletta*†, Gaelle Hisler*† and Anastasia Paparrizou*
* *CNRS, University of Montpellier, France*
† *Tellmeplus, Montpellier, France*
{*bessiere,coletta,hisler,paparrizou*}@*lirmm.fr*

***Abstract*—Preference reasoning is a central problem in decision support. There exist various ways to interpret a set of qualitative preferences. Conditional preference logics allow to deal with semantics such as optimistic, pessimistic, strong or not. In this paper, we study the complexity of the main problems in optimistic/pessimistic preference logic: undominated, consistency and dominance. We show that they are all NP-hard in general, with some becoming polynomial under specific semantics. Our second contribution is to show that the dominance problem, which has an online component in its definition, is compilable to polynomial time.**

I. Introduction

Preferences appear in our everyday life each time we make a choice. When the number of alternatives becomes large, people/users unconsciously express preferences over them. Preferences help in making a faster decision rather than comparing an (potentially) exponential number of alternatives. Preference reasoning aims at supporting the user in making a choice reflecting her preferences. Reasoning with preferences is a topic of increasing interest in decision support.

Given a set of preferences defined by the user, determining the best choice(s) is one of the main problems in preference reasoning, called the *undominated* problem. Other main problems are the *consistency* of a set of preferences, or the ranking of pairs of outcomes, i.e. the *dominance* problem.

There exist two main ways to express preferences in the literature: one for quantitative preferences and one for qualitative preferences. The most popular formalism of quantitative preferences is the GAI-net (Generalized Additive Independence) [1]. GAI-net offers the advantage that dominance testing simply requires to compute the utility of each outcome. But, all quantitative preference formalisms suffer from the same drawback: how to specify the weights of the alternatives? It can be difficult for a user to state she prefers to buy a house with weight 0.7 rather than an apartment with weight 0.3.

On the other hand, qualitative preferences are easier to be expressed by the user. CP-nets [2] are a well-known formalism of qualitative preference reasoning. CP-nets use a semantics called *ceteris paribus* (i.e., "all other things being equal"), meaning that two outcomes can be compared according to a variable only if these outcomes are equal over all other variables. Both dominance and undominated problems have been widely studied for this formalism. In general, dominance testing in CP-nets has been proved PSPACE-complete [3], whereas undominated outcomes are the solutions of a set of hard constraints, thus not harder than NP [4]. Dominance has been proved polynomial for CP-nets with tree or poly-tree structures, and for acyclic binary-valued CP-nets [5]. Undominated is also polynomial for acyclic CP-nets [5].

Several semantics other than ceteris paribus have been proposed: optimistic [6], pessimistic [7], strong (called strict in [8]). The conditional logic of preferences of [9], [10] provides a unified framework to deal with all these semantics. This framework has the advantage that it allows the user to choose a pessimistic or an optimistic semantics, and to decide for each preference whether it is strong or not. Unfortunately, the complexity of reasoning in this framework is unknown. The only algorithm provided for this framework requires exponential space.

In [11], Bienvenu et al. introduced another general framework for preference reasoning, called *prototypical* preference logic (PL for short). In PL, a logic formula involves preference statements. Each preference statement involves formulas α and β and a formula F under which α is preferred to β (denoted by $\alpha \rhd \beta \parallel F$). Several well-known preference frameworks are fragments of PL. For instance, it has been shown in [11] that we can encode CP-nets by putting in F all the propositional symbols that do not appear in α and β. The expressive power of PL has a price: dominance and consistency are shown to be PSPACE-complete in the general case. In [11], Bienvenu et al. have also studied fragments with lower complexity. They isolated a part of PL in which F is empty and preferences only allow conjunctions. This fragment is called free preferences. With free preferences, consistency becomes co-NP-complete and dominance NP-complete. Free preferences are presented as "obviously related" to the strong semantics in [10].

In this paper, we study the optimistic/pessimistic preference logic defined in [9], [10]. We give a complete complexity map of the main problems in this logic: undominated, consistency and dominance. We show that their complexities depend on the semantics under which they are interpreted. All problems are NP-hard in general. When all preferences are strong, consistency and dominance are polynomial, and if in addition, the semantics is pessimistic, undominated is polynomial too. Then, we prove that the dominance problem, which has an online part in its definition, is compilable to polynomial time. After this compilation process, the dominance queries can be answered in linear time only, overcoming the practical difficulties imposed by the framework of [9], [10]. This result paves the way for using this framework in applications where we need to rank sets of outcomes on the fly.

II. Background

We first define the vocabulary on which preferences will be defined. Intuitively, a vocabulary specifies the space Ω of all possible outcomes. A vocabulary V is a pair (X, D) such that X is a set of n variables $\{x_1, \ldots, x_n\}$, and D is a set of finite domains $\{D(x_1), \ldots, D(x_n)\}$. An *instantiation* on a subset $X' \subseteq X$ of variables is an assignment of a value of $D(x_i)$ to every variable x_i in X'. Instantiations on X are called outcomes, that is, $\Omega = \Pi_{i \in 1..n} D(x_i)$. Given an outcome I and an instantiation u on $X' \subseteq X$, we say that I satisfies u (denoted by $I \models u$) if the projection of I on X' is equal to u. A preference network $N = (V, P)$ is defined by a set P of m preferences over the vocabulary $V = (X, D)$. P is any preference formulation which allows to define an irreflexive and transitive binary relation $\succ$ over Ω. For two outcomes I and I' in Ω, we say that I is strictly preferred to I' when $I \succ I'$.

A. Main problems in preference reasoning

The undominated, consistency, and dominance problems are the main problems defined in the literature for preference reasoning.

Definition 1 (Undominated): Given a preference network $N = (V, P)$, the undominated problem is to determine whether there exists an outcome $I \in \Omega$ for which there does not exist any outcome $I' \in \Omega$ such that $I' \succ I$.

Definition 2 (Consistency): Given a preference network $N = (V, P)$, the consistency problem is to determine whether there does not exist any pair of outcomes $I, I' \in \Omega$ such that $I \succ I'$ and $I' \succ I$.

Definition 3 (Dominance): Given a preference network $N = (V, P)$ and two outcomes $I, I' \in \Omega$, the dominance problem is to determine whether $I \succ I'$ and $I' \not\succ I$.

B. Optimistic/pessimistic preference logic

In this section, we present and formalize the conditional logic of preferences introduced in [9], [10]. This formalism can be used to build the relation $\succ$ over the outcomes in Ω. The relation $\succ$ is induced by a set of preferences and semantics under which the preferences are interpreted. The user can define an optimistic (Opt) (resp. pessimistic (Pes)) semantics for the whole set of preferences, to rank the outcomes from best to worst (resp. from worst to best). After setting the semantics for the whole set of preferences, each preference can be defined either as strong (e.g., strong optimistic, strong pessimistic [8][12]), or non-strong (e.g., simply optimistic [13], [6] or pessimistic [7]). Illustrations of these semantics are given later. We give the definition of a preference in conditional logic.

Definition 4 (Preference): Given a vocabulary $V = (X, D)$, a *conditional logic preference* p is defined by a triplet (h, b, w), written $p = (h : b > w)$, of instantiations in V. h defines an hypothesis under which outcomes satisfying b (better) are preferred to those satisfying w (worse). If $h = \emptyset$ then b is unconditionally preferred to w.

In the following we consider that in all preferences $p = (h : b > w)$, h, b, and w are instantiations of bounded size. Despite this restriction, all our hardness results will hold.

Having defined what a preference in conditional logic is, we define the network of conditional logic preferences.

Definition 5 (CLP-net): A *network of conditional logic preferences (CLP-net)* is a pair $(V, P[S, Str])$, where $V = (X, D)$ is the vocabulary and P is a set of conditional logic preferences defined on V. S is a semantics for P (i.e., Opt or Pes), Str is the set of the preferences of P that are strong.

Hereafter, when we refer to a preference, we always mean a conditional logic preference. We define two types of preference satisfaction, i.e. the *satisfaction* and the *full satisfaction*.

Definition 6 (Preference satisfaction): Given a preference $p = (h : b > w)$ on a vocabulary V, an outcome I in Ω *satisfies* p in *Opt*, written $I \models_{Opt} p$, if and only if $I \not\models h$ or $I \models b$ or $I \not\models w$. Respectively, I *satisfies* p in Pes, written $I \models_{Pes} p$, if and only if $I \not\models h$ or $I \not\models b$ or $I \models w$. If I does not satisfy p in S, we say that it *violates* p, written $I \not\models_S p$.

Given a CLP-net $(V, P[S, Str])$ and an outcome I, $violated(P, I)$ denotes the set of preferences from P that are violated by I in S. If $violated(P, I) = \emptyset$, we write $I \models_S P$.

An optimistic semantics ($S = Opt$) can be viewed as a first best choice (namely the best outcome is given first), whereas a pessimistic semantics ($S = Pes$) is a first worst choice (namely a choice by elimination).

We now give the definition for the preference *full satisfaction*, which requires to satisfy the hypothesis.

Definition 7 (Preference full satisfaction): Given a preference $p = (h : b > w)$ on a vocabulary V, an outcome I in Ω *fully satisfies* p in Opt, denoted by $I \models^f_{Opt} p$, if and only if $I \models h$, $I \models b$, and $I \not\models w$. Respectively, I *fully satisfies* p in Pes, denoted by $I \models^f_{Pes} p$, if and only if $I \models h$, $I \models w$, and $I \not\models b$.

We give an example to illustrate the preference *(full) satisfaction*.

Example 1 ((Full) satisfaction): When organizing a trip, there is an option between a day (d) or a night ($\bar{d}$) flight and between a stop flight (s) or a non-stop one ($\overline{s}$). The user says: "If the flight is non-stop then I prefer a night flight to a day one", written as: $p_1 = (\overline{s} : \overline{d} > d)$. An outcome I satisfies p_1 in Opt if and only if I violates $\overline{s}$ or satisfies $\overline{d}$. In other words, any stop flight or (non-stop) night flight satisfies user's preference in Opt. Similarly, any stop flight or (non-stop) day flight satisfies p_1 in Pes. An outcome I fully satisfies p_1 in Opt if and only if I satisfies $\overline{s}$ and satisfies $\overline{d}$ (i.e., a non-stop night flight). A non-stop day flight fully satisfies p_1 in Pes.

Before defining the main notion that will allow us to compare outcomes, we need to define the notion of *preference deactivation*.

Definition 8 (Preference deactivation): Given a CLP-net $(V, P[S, Str])$, we say that P *deactivates* a preference p, denoted by $P \rightsquigarrow p$, if and only if

- $p \notin Str$ and there exists an outcome $I \in \Omega$ such that I fully satisfies p in S and I satisfies P in S, that is, $P \rightsquigarrow p \leftrightarrow \exists I \in \Omega \mid I \models^f_S p$ and $I \models_S P$.
- $p \in Str$ and for any outcome $I \in \Omega$, if I fully satisfies p in S then I satisfies P in S, that is, $P \rightsquigarrow p \leftrightarrow \forall I \in \Omega$, $I \models^f_S p \rightarrow I \models_S P$.

Example 2 (Deactivation): In the problem of Example 1 where $p_1 = (\overline{s} : \overline{d} > d)$, the user adds the following preference: "I prefer a non-stop flight to a stop one", written $p_2 = (\emptyset : \overline{s} > s)$. Then, $P = \{p_1, p_2\}$. If the user sets S to Opt and p_1, p_2 are not strong, P deactivates p_1 because the outcome $I = (\overline{s}, \overline{d})$ (i.e., "non-stop night flight") fully satisfies p_1 and satisfies P. P also deactivates p_2.

Preference deactivation allows us to define *layers*, which is the central notion to compare outcomes in this conditional logic. A layer is a set of outcomes. Each layer includes the outcomes that do not belong to previous layers and that satisfy all preferences that have not been deactivated in previous layers. We give the inductive definition of layers.

Definition 9 (Layer): Given a CLP-net $(V, P[S, Str])$, $P_0 = P$, and P_i is the set of the preferences in P_{i-1} that are not deactivated by P_{i-1}, that is, $P_i = P_{i-1} \backslash \{p \mid P_{i-1} \rightsquigarrow p\}$. The *layer* E_i is the set of outcomes that do not belong to previous layers and that satisfy all preferences in P_i, that is, $E_i = \{I \in \Omega \mid I \models_S P_i\} \setminus \bigcup_{j \in 0..i-1} E_j$. The index $last$ is the smallest index such that all outcomes satisfying P_{last} belong to previous layers $E_0, \ldots, E_{last-1}$. We define E_{last} to be the set of remaining outcomes $\Omega \setminus \bigcup_{j \in 0..last-1} E_j$.

Given an outcome $I \in \Omega$, $layer(I)$ is the index of the layer to which I belongs, that is, $I \in E_{layer(I)}$.

It is important to observe that E_{last} can be empty or not. If E_{last} is non-empty, none of the outcomes in E_{last} can satisfy P_{last} and the construction of layers cannot proceed.

We define now the preference relation $\succ$ according to the semantics used in CLP-nets.

Table I
THE RANKINGS OF EXAMPLE 3

	$p_1, p_2 \notin Str$		$p_1 \notin Str, p_2 \in Str$	
i	P_i	E_i	P_i	E_i
0	p_1, p_2	$(\overline{s}, \overline{d})$	p_1, p_2	$(\overline{s}, \overline{d})$
1	$\emptyset$	$(s, \overline{d}), (s, d), (\overline{s}, d)$	p_2	$(\overline{s}, d)$
2	$\emptyset$	$\emptyset$	$\emptyset$	$(s, d), (s, \overline{d})$
3	—	—	$\emptyset$	$\emptyset$

Definition 10 (Layer based order): Given two outcomes I and I' in Ω, we have,

- I is strictly preferred to I' in Opt, denoted by $I \succ_{Opt} I'$, if and only if $layer(I') = last$ or $layer(I) < layer(I')$.
- I is strictly preferred to I' in Pes, denoted by $I \succ_{Pes} I'$, if and only if $layer(I) = last$ or $layer(I) > layer(I')$.

When $S = Opt$ (resp. $S = Pes$), the best (resp. worst) outcome is ranked first. Namely, the best outcome appears in E_0 for Opt (resp. in E_{last-1} for Pes). If an outcome I belongs to a layer which precedes (resp. follows) the layer where an outcome I' belongs to, I is strictly preferred to I'.

Example 3 (Layers): Recall the preferences of the previous examples: $p_1 = (\overline{s} : \overline{d} > d)$ and $p_2 = (\emptyset : \overline{s} > s)$. Table I illustrates the different rankings of the outcomes depending on p_2 being strong or not. Suppose that $p_1 \notin Str$ and $p_2 \notin Str$. According to Definition 6, the best outcome for $S = Opt$ is the same regardless of the set Str, i.e. a non-stop night flight $(\overline{s}, \overline{d})$. Hence, $E_0 = \{(\overline{s}, \overline{d})\}$. From Definitions 7, 8 and 9, if $p_1 \notin Str$ and $p_2 \notin Str$ then $P_0 = P = \{p_1, p_2\}$ deactivates both p_1 and p_2. Indeed, $(\overline{s}, \overline{d})$ satisfies P_0 and fully satisfies p_1 and p_2. Therefore, the set of outcomes in E_1 which satisfy $P_1 = P_0 \setminus \{p_1, p_2\} = \emptyset$ is equal to all remaining outcomes, namely $E_1 = \{(s, \overline{d}), (s, d), (\overline{s}, d)\}$. $E_2 = \emptyset$ and $last = 2$.

Suppose now that $p_1 \notin Str$ and $p_2 \in Str$. P_0 deactivates p_1 due to $(\overline{s}, \overline{d})$, but does not deactivate p_2 because $p_2 \in Str$ and there exists $I' = (\overline{s}, d)$ such that I' fully satisfies p_2 but violates p_1. Thus, $E_0 = \{(\overline{s}, \overline{d})\}$ and $P_1 = P_0 \backslash \{p_1\} = \{p_2\}$. $(\overline{s}, \overline{d})$ and $(\overline{s}, d)$ satisfy p_2, resulting in $E_1 = \{(\overline{s}, \overline{d}), (\overline{s}, d)\} \setminus E_0 = \{(\overline{s}, d)\}$. The outcomes that fully satisfy p_2 (i.e., $(\overline{s}, \overline{d})$ and $(\overline{s}, d)$) also satisfy P_1. Thus, P_1 deactivates p_2. $P_2 = \emptyset$, $E_2 = \{(s, d), (s, \overline{d})\}$, and $last = 3$.

We have shown how a preference is interpreted under its semantics. In the following section, we show the complexity of the three problems undominated/consistency/dominance depending on the semantics.

III. Complexity Map

In this section we draw the complexity map for the three problems undominated/consistency/dominance. We first focus on the optimistic semantics and then adapt the results to the pessimistic case.

A. Optimistic semantics

Theorem 1: The undominated problem is NP-complete on CLP-nets with $S = Opt$, even if $Str = P$.

Proof: Membership. Given an outcome $I \in \Omega$, checking whether I is an undominated outcome in Opt is equivalent to checking if I belongs to E_0 and $last \neq 0$. (Outcomes in E_{last} are all preferred one to each other (Definition 10). Thus, no outcome in E_{last} can be an undominated outcome.) Following Definition 9, I belongs to E_0 and $last \neq 0$ if and only if I satisfies all $p \in P$ in Opt. I satisfies a preference $p = (h : b > w)$ in Opt if and only if I violates the hypothesis h or w or satisfies b. Checking if I violates h or w or if I satisfies b is polynomial and this verification is performed exactly $|P|$ times.
Completeness. We reduce an instance $\phi = (cl_1, \dots cl_m)$ of 3-SAT to the undominated problem: the set of variables X is defined by $variables(\phi)$ and D is $\{0,1\}^X$. P is built as follows. For each clause $cl_i = (l_{i,1}, l_{i,2}, l_{i,3})$ of ϕ, we introduce a preference $p_i = (h_i : b_i > w_i)$ with $h_i = \neg l_{i,1}$, $b_i = l_{i,2}$ and $w_i = \neg l_{i,3}$. An outcome I satisfies a preference p_i if and only if I violates h_i or w_i or satisfies b_i. Hence, I satisfies p_i if and only if it satisfies $l_{i,1}$ or $l_{i,2}$ or $l_{i,3}$, that is, it satisfies the clause cl_i. Now, there exists an undominated outcome if and only if there exists an outcome I that belongs to $E_0, last \neq 0$. I belongs to E_0 with $last \neq 0$ if and only if I satisfies P, and I satisfies P if and only if it satisfies all clauses in ϕ.

The proof still holds if $Str = P$ because the satisfaction of a preference p is the same whatever p is in Str or not. ■

To prove the complexity of the consistency and dominance problems we will use the following lemmas.

Lemma 1: Given a CLP-net N with $S = Opt$, these three propositions are equivalent:
(1) N is consistent,
(2) E_{last} is empty,
(3) P_{last} is empty.

Proof: ((1) $\Rightarrow$ (3)) We show that if P_{last} is not empty, then N is inconsistent. Let p be a preference in P_{last}. There necessarily exists an outcome I in Ω violating p. I cannot be in $\bigcup_{i \in 0..last-1} E_i$ as p is still active in P_{last}. So, $I \in E_{last}$. There also necessarily exists an outcome I' in Ω that fully satisfies p. Again, I' cannot be in $\bigcup_{i \in 0..last-1} E_i$, otherwise p would have been deactivated and would not be in P_{last}. Thus, both I and I' belong to E_{last}. I' cannot be equal to I because I' satisfies p whereas I does not, and by Definition 10, we know that for any I, I' in E_{last}, $I \succ I'$ and $I' \succ I$. Therefore, N is inconsistent.
((3) $\Rightarrow$ (2)) Assume E_{last} is not empty. By Definition 9, E_{last} contains those outcomes that do not satisfy P_{last}. Therefore, P_{last} is not empty because any outcome satisfies an empty set of preferences.
((2) $\Rightarrow$ (1)) Assume E_{last} is empty. From Definition 10, there exists a pair of outcomes I, I' such that $I \succ I'$ and $I' \succ I$ if and only if both I and I' belong to E_{last}. Since E_{last} is empty, there does not exist such a pair of outcomes and then N is consistent. ■

Lemma 2: Given a CLP-net $N = (V, P[Opt, Str])$, deciding whether a preference p is deactivated by a set $P_i \subseteq P$ is polynomial if $p \in Str$.

Proof: By definition of deactivation (Definition 8), P_i deactivates a strong preference $p = (h : b > w)$ if and only if there does not exist any outcome I that fully satisfies p and violates a preference $p' = (h' : b' > w')$ in P_i. Testing whether there exists I such that I fully satisfies p and violates p' is equivalent to testing whether there exists I satisfying $h \wedge b \wedge \neg w \wedge h' \wedge \neg b' \wedge w'$. Such a test is linear in the number of variables n as all terms of the conjunction are instantiations. We do this process for each p' in P_i, $p' \neq p$. P_i deactivates p if and only if all the $|P_i| - 1$ tests return false. Therefore, deciding deactivation of a strong preference is polynomial in Opt. ■

Theorem 2: The consistency problem is NP-complete on CLP-nets with $S = Opt$.

Proof: By Lemma 1, a CLP-net with $S = Opt$ is consistent if and only if P_{last} is empty.
Membership. Let us first observe that if a set Q of preferences deactivates a preference p, then any subset of Q also deactivates p. A sequence of preferences $\langle p_{i_1}, \dots, p_{i_k} \rangle$ such that $\forall j \in 1..k$, $P \setminus \{p_{i_l} \mid l < j\}$ deactivates p_{i_j} is called *sequence of deactivations*. Given a sequence of deactivations $\langle p_{i_1}, \dots, p_{i_j}, \dots, p_{i_k} \rangle$, if $P \setminus \{p_{i_1}, \dots, p_{i_j}\}$ deactivates p_{i_k}, then $\langle p_{i_1}, \dots, p_{i_j}, p_{i_k}, \dots, p_{i_{k-1}} \rangle$ is also a sequence of deactivations because of our first observation. Hence, a polynomial certificate for $P_{last} = \emptyset$ is any sequence of deactivations that wipes out P. Let Seq be the sequence $\langle \langle p_{i_1}, I_{i_1} \rangle, \dots, \langle p_{i_m}, I_{i_m} \rangle \rangle$ such that $P = \{p_{i_1}, \dots, p_{i_m}\}$, p_{i_k} is deactivated by $P \setminus \{p_{i_j} \mid j < k\}$ for all $k \in 1..m$, and for each $p_{i_k} \notin Str$, I_{i_k} is an outcome that fully satisfies p_{i_k} and that satisfies $P \setminus \{p_{i_j} \mid j < k\}$. If $p_{i_k} \in Str$, checking if p_{i_k} is deactivated by $P \setminus \{p_{i_j} \mid j < k\}$ is polynomial by Lemma 2. If $p_{i_k} \notin Str$, checking if p_{i_k} is deactivated by $P \setminus \{p_{i_j} \mid j < k\}$ is equivalent to checking if I_{i_k} satisfies $P \setminus \{p_{i_j} \mid j < k\}$ and fully satisfies p_{i_k}, which is also polynomial. Thus, Seq is a polynomial certificate for the consistency problem.
Completeness. We reduce an instance $\phi = (cl_1, \dots cl_{m-1})$ of 3-SAT to the problem $E_{last} = \emptyset$ in the CLP-net $(V, P[Opt, Str])$. The set of variables X is defined by $variables(\phi) \cup \{l_b\}$ and $D = \{0,1\}^X$. For each clause $cl_i = (l_{i,1}, l_{i,2}, l_{i,3})$ of ϕ, we add to P a preference $p_i = (\emptyset : l_b > \neg l_{i,1} \wedge \neg l_{i,2} \wedge \neg l_{i,3})$, and $p_i \notin Str$. We add to P an extra preference $p_{extra} = (\emptyset : \neg l_b > l_b)$ with

$p_{extra} \notin Str$. We prove that ϕ is satisfiable if and only if $E_{last} = \emptyset$.

($\Rightarrow$) Let I_ϕ be a model of ϕ. By construction, $I_\phi \cup \{l_b = 0\}$ satisfies all preferences in P and fully satisfies p_{extra}. No outcome can satisfy p_{extra} and fully satisfy another preference in P. Hence, $P_1 = P \setminus \{p_{extra}\}$. By construction again, $I_\phi \cup \{l_b = 1\}$ fully satisfies all preferences in P_1, which leads to $P_2 = \emptyset$ and $E_3 = E_{last} = \emptyset$.

($\Leftarrow$) Assume E_{last} is empty. Then, $last$ cannot be equal to 0 because by definition, E_0 cannot be empty. Thus, E_0 contains at least an outcome I, and by construction I satisfies P, as $last \neq 0$. To satisfy p_{extra} I must violate l_b. Thus, I satisfies P if and only if for all p_i in P, I violates $\neg l_{i,1} \wedge \neg l_{i,2} \wedge \neg l_{i,3}$, and thus satisfies cl_i. Therefore, I satisfies ϕ. ■

Theorem 3: The dominance problem is DP-complete on CLP-nets with $S = Opt$.

Proof: Membership. Let $N = (V, P[Opt, Str])$ be a CLP-net and I, I' two outcomes. A certificate for the dominance problem is a certificate for the problem of the existence of an integer j such that $I \in \bigcup_{i \in 0..j} E_i$ and $I' \notin \bigcup_{i \in 0..j} E_i$. Let $S_j = \langle\langle p_1^1, I_1^1\rangle, \ldots, \langle p_{i_1}^1, I_{i_1}^1\rangle, \ldots, \langle p_1^j, I_1^j\rangle, \ldots, \langle p_{i_j}^j, I_{i_j}^j\rangle\rangle$ be a sequence of deactivations such that each $\langle p_y^x, I_y^x\rangle \in S_j$ has the following properties: (1) $P \setminus \{p_b^a \mid \langle p_b^a, I_b^a\rangle \in S_j \wedge a < x\} \rightsquigarrow p_y^x$, (2) if $p_y^x \notin Str$, I_y^x is a witness of the deactivation in (1), and (3) $violated(P, I) \subseteq \{p_y^x \mid \langle p_y^x, I_y^x\rangle \in S_j\}$. Let $S'_j = \langle\langle p_1'^1, I_1'^1\rangle, \ldots, \langle p_{i'_1}'^1, I_{i'_1}'^1\rangle, \ldots, \langle p_1'^j, I_1'^j\rangle, \ldots, \langle p_{i'_j}'^j, I_{i'_j}'^j\rangle\rangle$ be a sequence of deactivations built in a way similar to S_j, that is, for each $\langle p_y'^x, I_y'^x\rangle \in S'_j$, (1) $P \setminus \{p_b'^a \mid \langle p_b'^a, I_b'^a\rangle \in S'_j \wedge a < x\} \rightsquigarrow p_y'^x$, (2) if $p_y'^x \notin Str$, $I_y'^x$ is a witness of the deactivation of $p_y'^x$, and (3) $violated(P, I') \subseteq \{p_y'^x \mid \langle p_y'^x, I_y'^x\rangle \in S'_j\}$.

S_j is a certificate of $I \in \bigcup_{i \in 0..j} E_i$ and S'_j is a certificate of $I' \in \bigcup_{i \in 0..j} E_i$. They are both polynomial to check: for each pair $\langle p, I\rangle$ in S_j and S'_j, if $p \in Str$ we know by Lemma 2 that deactivation is polynomial, and if $p \notin Str$, we check whether I fully satisfies p and satisfies the set of preferences specified in (1). Finally, we check that all preferences in $violated(P, I)$ (resp. $violated(P, I')$) are covered. As dominance says 'yes' if and only if there exists such an S_j and there does not exist such an S'_j, we conclude that dominance is in DP.

Completeness. We reduce an instance of SAT-UNSAT to the dominance problem. Let $F = (cl_1, \ldots, cl_{|F|})$ and $F' = (cl'_1, \ldots, cl'_{|F'|})$ be the two 3-CNF formulas of the SAT-UNSAT instance. We build the CLP-net $((X, D), P[Opt, Str])$, and two outcomes I and I', as an instance of the dominance problem. X is the union of $variables(F)$ and $variables(F')$ plus four additional variables $\{l_h, l_{h'}, l_w, l_{w'}\}$, and $D = \{0,1\}^X$. For each clause $cl_i = (l_{i,1}, l_{i,2}, l_{i,3})$ of F, we define a preference $p_i = (l_h : l_{i,1} > \neg l_{i,2} \wedge \neg l_{i,3})$. For each clause $cl'_i = (l'_{i,1}, l'_{i,2}, l'_{i,3})$ of F', we define a preference $p'_i = (l_{h'} : l'_{i,1} > \neg l'_{i,2} \wedge \neg l'_{i,3})$. We introduce two additional preferences $p = (\emptyset : l_h > l_w)$ and $p' = (\emptyset : l_{h'} > l_{w'})$. We then have, $P = \{p_1, \ldots, p_{|F|}, p'_1, \ldots, p'_{|F'|}, p, p'\}$, and we set $Str = \emptyset$. We define I as an arbitrary outcome over X satisfying $\neg l_h \wedge l_w \wedge \neg l_{h'} \wedge \neg l_{w'}$. Similarly, we define I' as an arbitrary outcome over X satisfying $\neg l_h \wedge \neg l_w \wedge \neg l_{h'} \wedge l_{w'}$.

We now prove that I dominates I' (i.e., $I \succ I'$ and $I' \not\succ I$) if and only if F is SAT and F' is UNSAT. We first prove that F is SAT if and only if I belongs to E_1. I violates p because $I \models \neg l_h \wedge l_w$. Hence, I cannot belong to E_0. By construction, I satisfies all other preferences in P. Thus, I belongs to E_1 if and only if P deactivates p, which means that there exists an outcome I^* such that I^* satisfies P and fully satisfies p. If we set $l_{w'}$ and $l_{h'}$ to false in I^*, I^* satisfies p' and all p'_i. To fully satisfy p, I^* needs to satisfy $l_h \wedge \neg l_w$. As $I^* \models l_h$, to satisfy any p_i (obtained from a cl_i of F), I^* needs to satisfy $l_{i,1} \vee \neg(\neg l_{i,2} \wedge \neg l_{i,3})$, namely $(l_{i,1} \vee l_{i,2} \vee l_{i,3})$. Since I^* satisfies all these p_i, I^* is a model of F. Then I belongs to E_1 if and only if F is SAT.

Second, we prove that I' does not belong to E_1 if and only if F' is UNSAT. I' violates p' because $I' \models \neg l_{h'} \wedge l_{w'}$. Hence, I' cannot belong to E_0. I' does not belong to E_1 if and only if P does not deactivate p', which means that there does not exist an outcome I'^* such that I'^* satisfies P and fully satisfies p'. By using the same reasoning as previously, there does not exist such an I'^* if and only if F' is UNSAT. Thus I' does not belong to E_1 if and only if F' is UNSAT.

Finally, we observe that if F and F' are both UNSAT, neither p nor p' can be deactivated, so I and I' both belong to E_2, which is equal to E_{last}, and I does not dominate I'. ■

Theorem 1 tells us that the undominated problem is NP-complete for $S = Opt$ even if $Str = P$. Interestingly, this is not the case for the consistency and dominance problems, which become polynomial when $Str = P$.

Theorem 4: The consistency problem is polynomial on CLP-nets with $S = Opt$ and $Str = P$.

Proof: A CLP-net $(V, P[Opt, Str])$ is consistent if and only if P_{last} is empty (Lemma 1). This means that there exists a sequence of deactivations that wipes out P, that is, a sequence of preferences $\langle p_{i_1}, \ldots, p_{i_m}\rangle$ such that $P = \{p_{i_j} \mid j \in 1..m\}$ and for all $k \in 1..m$, $P \setminus \{p_{i_j} \mid j < k\} \rightsquigarrow p_{i_k}$. By Lemma 2, deciding the deactivation of a strong preference by a set of preferences is polynomial. Thus, given a set Q of strong preferences, finding whether there exists a preference in Q deactivated by Q is polynomial too. It is sufficient to iteratively check every preference in Q. Next, as in the proof of Theorem 2, we observe that if Q deactivates a preference p, then any subset of Q also deactivates p. Thus, there exits a sequence of deactivations that wipes out P if and only if a greedy algorithm iteratively finding deactivated preferences and removing them from P terminates by wiping out P. As a result, if P can be wiped out by this process, the CLP-net is consistent, otherwise it is inconsistent. In the worst case, finding a preference that is deactivated by P or by a subset of P requires $|P|$ calls to the deactivation problem. This search for deactivated preference is performed $|P|$ times. Therefore, the consistency problem is polynomial. ■

Theorem 5: The dominance problem is polynomial on CLP-nets with $S = Opt$ and $Str = P$.

Proof: (Sketch.) Given a CLP-net $(V, P[Opt, Str])$ and two outcomes I and I', dominance says 'yes' if and only if there exists an integer j such that $I \in \bigcup_{i\in 0..j} E_i$ and $I' \notin \bigcup_{i\in 0..j} E_i$. We can design an algorithm that iteratively builds the sets of preferences $P_1, \ldots, P_j$, where P_i is the set of preferences not deactivated by P_{i-1} and j is the smallest integer such that $I \models_{Opt} P_j$. P_j is such that $I \succ I'$ if and only if $I' \not\models_{Opt} P_j$.

Building $P_1, \ldots, P_j$ is polynomial because deciding the deactivation of a strong preference by a set of preferences is polynomial (Lemma 2) and the number of deactivation tests is bounded above by $|P|^2$. Finally, checking whether I' satisfies P_j is again polynomial to check. ■

B. Pessimistic semantics

As seen in Definition 6, the pessimistic semantics ranks first layers of the worst outcomes whereas optimistic semantics ranks first layers of the best outcomes. As a result, most of the proofs given in this section are slight adaptations of previous ones.

Theorem 6: The consistency problem is NP-complete on CLP-nets with $S = Pes$.

Proof: (Sketch.) Membership is essentially the same as in the proof of Theorem 2. Completeness is obtained by changing the way we encode clauses in the proof of Theorem 2: For each clause $cl_i = (l_{i,1}, l_{i,2}, l_{i,3})$ of ϕ, we introduce a preference $p_i = (\emptyset : \neg l_{i,1} \wedge \neg l_{i,2} \wedge \neg l_{i,3} > l_b)$ instead of $p_i = (\emptyset : l_b > \neg l_{i,1} \wedge \neg l_{i,2} \wedge \neg l_{i,3})$. The extra preference becomes $p_{extra} = (\emptyset : l_b > \neg l_b)$. ■

Theorem 7: The dominance problem is DP-complete on CLP-nets with $S = Pes$.

Proof: Direct adaptation of the proof of Theorem 3. ■

The undominated problem in pessimistic semantics is substantially different from the optimistic case as we have to build the whole sequence of layers from worst to best before proving the existence of an undominated outcome. It follows a strong connection between the consistency and undominated problems.

Lemma 3: A CLP-net with $S = Pes$ is consistent if and only if it has an undominated outcome.

Proof: ($\Rightarrow$) Let N be a consistent CLP-net with $S = Pes$. From a slightly adapted Lemma 1, N is consistent if and only if E_{last} is empty. As E_{last} is empty, there exists an outcome I belonging to E_{last-1}. By Definition 10, there does not exist any outcome I' such that $I' \succ I$. Thus, I is undominated.
($\Leftarrow$) Assume N is inconsistent. Then, there exist two outcomes I and I' such that $I \succ I'$ and $I' \succ I$. According to Definition 10, it can only happen if I and I' both belong to E_{last}. Suppose there exists an undominated outcome I^*. I^* necessarily belongs to E_{last}, but it is impossible as I would dominate it. ■

Corollary 1: The undominated problem is NP-complete on CLP-nets with $S = Pes$.

Proof: From Lemma 3 and Theorem 6. ■

Theorem 8: The undominated/consistency/dominance problems are polynomial on CLP-nets with $S = Pes$ and $Str = P$.

Proof: (Sketch.) Consistency and dominance are similar to the optimistic case (Theorems 4 and 5). Undominated comes from Lemma 3. ■

Table II summarizes the complexity results of this section.

Table II
SUMMARY OF OUR COMPLEXITY RESULTS

Problem	Str	$S = Opt$	$S = Pes$
Undominated	$\subsetneq P$	NP-complete (Th. 1)	NP-complete (Co. 1)
	$= P$	NP-complete (Th. 1)	polynomial (Th. 8)
Consistency	$\subsetneq P$	NP-complete (Th. 2)	NP-complete (Th. 6)
	$= P$	polynomial (Th. 4)	polynomial (Th. 8)
Dominance	$\subsetneq P$	DP-complete (Th. 3)	DP-complete (Th. 7)
	$= P$	polynomial (Th. 5)	polynomial (Th. 8)

IV. SOLVING AND COMPILING CLP-NETS

The three problems undominated/consistency/dominance introduced in Section II-A are NP-hard to solve in general. The undominated and consistency problems are static problems belonging to NP. They can be solved by a simple call to a SAT or CSP solver. The dominance problem, on the contrary, is more difficult for two reasons. First, it is beyond NP. But more importantly, dominance queries can be repeatedly asked on the same CLP-net for comparing different pairs of outcomes. Such queries arise for instance in recommendation applications, where the system/provider wants to propose to the customer the most appropriate outcome among two (or more) outcomes, according to her preferences. As these queries can be involved in an interactive process, it is crucial to have fast responses.

Compilation theory aims at improving the efficiency of on-line computation of difficult problems through pre-processing [14], [15]. The intuition is to remove sources of complexity to obtain cheaper queries (belonging to a lower complexity class) using a compiled form of the initial data Σ. The pre-processing, or off-line compilation phase, consists in transforming parts of this fixed part Σ into a compiled form of size polynomial in $|\Sigma|$. This transformation can take exponential time. The varying part of the problem appears at execution time, called on-line phase. If the output can be produced in polynomial time, then the problem is said to be compilable to polynomial time. We show that dominance queries become polynomial after a compilation phase that produces an array $indexP[\cdot]$ of indices, where $indexP[j]$ is the index of the first layer where p_j is not active, that is, $indexP[j] = i$ if and only if $p_j \in P_{i-1} \setminus P_i$.

Algorithm 1 CLP-net compilation for dominance

Require: N=$(V, P[S, Str])$
Ensure: $indexP[\cdot]$
1: $i \leftarrow 0$;
2: $nextP \leftarrow P$;
3: $indexP[j] \leftarrow +\infty, \forall j \in 1..m$;
4: **repeat**
5:   $i \leftarrow i + 1$;
6:   $oldP \leftarrow nextP$;
7:   **for all** $p_j \in oldP$ **do**
8:     **if** $p_j \in Str$ **then**
9:       $deactivate \leftarrow true$;
10:       **for all** $p \in oldP \setminus \{p_j\}$ **do**
11:         **if** $\exists I \in \Omega, I \models^f_S p_j$ **and** $I \not\models_S p$ **then**
12:           $deactivate \leftarrow false$;
13:           break;
14:         **end if**
15:       **end for**
16:       **if** $deactivate$ **then**
17:         $nextP \leftarrow nextP \setminus \{p_j\}$
18:         $indexP[j] \leftarrow i$
19:       **end if**
20:     **else if** $\exists I \in \Omega, I \models^f_S p_j$ **and** $I \models_S oldP$ **then**
21:       $nextP \leftarrow nextP \setminus \{p_j\}$
22:       $indexP[j] \leftarrow i$
23:     **end if**
24:   **end for**
25: **until** $(nextP = oldP)$
26: **return** $indexP[\cdot]$

Algorithm 1 takes as input a CLP-net $(V, P[S, Str])$ and returns the array $indexP[\cdot]$. In line 3 $indexP[\cdot]$ is initialized to a value greater than any possible layer. Then, the loop of line 4 iteratively builds $indexP[\cdot]$. The ith execution of the loop sets $oldP$ to the set of preferences still active at the previous layer $i - 1$ (line 6) and will compute the set $nextP$ of preferences still active at layer i. The loop of line 7 iterates over all preferences p_j that belong to $oldP$ to check if p_j can be deactivated. The deactivation depends on whether the semantics of p_j is strong or not. If $p_j \in Str$ (lines 8–19), we need to iterate over all other preferences p in $oldP$. If there exists an I fully satisfying p_j and violating p (line 11), flag $deactivate$ becomes false (line 12) and we proceed to the next preference in $oldP$. If such an I is not found, then p_j is deactivated (that is, removed from $nextP$), and $indexP[j]$ is set to i (lines 16–19). When $p_j \notin Str$ (lines 20–22), we need to check if there exists an outcome I that fully satisfies p_j and satisfies all preferences in $oldP$. This line is NP-complete and can be solved by a call to a SAT/CSP solver. If such an outcome is found, again p_j is deactivated (removed from $nextP$) and $indexP[j]$ is set to i (lines 21–22). Finally the test of the main loop (in line 4) detects whether at least one preference has been deactivated from one layer to another. If not, we exit the loop and return $indexP[\cdot]$.

Theorem 9: The Dominance problem in CLP-nets is compilable to linear time.

Proof: A problem is compilable to a complexity class C if it is in C once the fixed part Σ of any instance has been pre-processed, i.e. turned off-line into a data structure of size polynomial in $|\Sigma|$. Given $(V, P[S, Str])$, Algorithm 1 computes off-line the array $indexP[\cdot]$, where $indexP[j] = i$ if and only if $p_j \in P_{i-1} \setminus P_i$. This structure $indexP[\cdot]$ is linear in $|P|$, the initial data.
Correctness. We iterate over $violated(P, I)$ to compute $i = max_{p_j \in violated(P,I)} indexP[j]$ and over $violated(P, I')$ to compute $i' = max_{p_j \in violated(P,I')} indexP[j]$. By construction of $indexP[\cdot]$, $i = layer(I)$, or $i = +\infty$ and $I \in E_{last}$. Similarly, $i' = layer(I')$, or $i' = +\infty$ and $I' \in E_{last}$. As a result, I dominates I' with $S = Opt$ (resp. $S = Pes$) if and only if $i < i'$ (resp. $i > i'$).
Complexity. Checking if an outcome satisfies a preference is constant time. Thus, building $violated(P, I)$ and $violated(P, I')$ is linear in $|P|$. Computing the maximum values i and i' is done by a traversal of the structure $indexP[\cdot]$, which is linear in $|P|$ as well. ■

V. Conclusion

We have studied the theoretical complexity of the main problems in optimistic/pessimistic preference logic: undominated, consistency, and dominance. We have shown that these problems are NP-hard in general, though they become polynomial for some specific semantics. We also show that, interestingly, the dominance problem, which is the only problem beyond NP and which in addition contains an online part, can be compiled to polynomial time. This opens the door to the use of CLP-nets in applications where we need to rank sets of outcomes on the fly.

References

[1] C. Gonzales and P. Perny, "GAI networks for utility elicitation," in *KR*, D. Dubois, C. A. Welty, and M.-A. Williams, Eds. AAAI Press, 2004, pp. 224–234.

[2] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole, "Reasoning with conditional ceteris paribus preference statements," in *Uncertainty in Artificial IntelligenceI*, K. B. Laskey and H. Prade, Eds. Morgan Kaufmann, 1999, pp. 71–80.

[3] J. Goldsmith, J. Lang, M. Truszczynski, and N. Wilson, "The computational complexity of dominance and consistency in cp-nets," *J. Artif. Intell. Res. (JAIR)*, vol. 33, pp. 403–432, 2008. [Online]. Available: http://dx.doi.org/10.1613/jair.2627

[4] S. D. Prestwich, F. Rossi, K. B. Venable, and T. Walsh, "Constraint-based preferential optimization," in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, M. M. Veloso and S. Kambhampati, Eds. AAAI Press / The MIT Press, 2005, pp. 461–466. [Online]. Available: http://www.aaai.org/Library/AAAI/2005/aaai05-073.php

[5] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole, "CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements," *Journal of Artificial Intelligence Research*, vol. 21, pp. 135–191, 2004.

[6] J. Pearl, "System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning," in *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, March 1990*, R. Parikh, Ed. Morgan Kaufmann, 1990, pp. 121–135.

[7] S. Benferhat, D. Dubois, S. Kaci, and H. Prade, "Bipolar representation and fusion of preferences on the possibilistic logic framework," in *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*, D. Fensel, F. Giunchiglia, D. L. McGuinness, and M. Williams, Eds. Morgan Kaufmann, 2002, pp. 421–448.

[8] C. Boutilier, "Toward a logic for qualitative decision theory," in *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn, Germany, May 24-27, 1994.*, J. Doyle, E. Sandewall, and P. Torasso, Eds. Morgan Kaufmann, 1994, pp. 75–86.

[9] S. Kaci and L. van der Torre, "Algorithms for a nonmonotonic logic of preferences," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, ECSQARU 2005, Barcelona, Spain, July 6-8, 2005, Proceedings*, ser. Lecture Notes in Computer Science, L. Godo, Ed., vol. 3571. Springer, 2005, pp. 281–292.

[10] S. Kaci and L. van der Torre, "Reasoning with various kinds of preferences: logic, non-monotonicity, and algorithms," *Annals of Operation Research*, vol. 163, no. 1, pp. 89–114, 2008.

[11] M. Bienvenu, J. Lang, and N. Wilson, "From preference logics to preference languages, and back," in *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, F. Lin, U. Sattler, and M. Truszczynski, Eds. AAAI Press, 2010. [Online]. Available: http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1360

[12] S. Benferhat and K. Souhila, "A possibilistic logic handling of strong preferences," in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 2, July 2001, pp. 962–967 vol.2.

[13] S. Benferhat, D. Dubois, and H. Prade, "Representing default rules in possibilistic logic," in *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, October 25-29, 1992.*, B. Nebel, C. Rich, and W. R. Swartout, Eds. Morgan Kaufmann, 1992, pp. 673–684.

[14] M. Cadoli and F. M. Donini, "A survey on knowledge compilation," *AI Commun.*, vol. 10, no. 3-4, pp. 137–150, 1997. [Online]. Available: http://content.iospress.com/articles/ai-communications/aic133

[15] P. Liberatore, "Compilation of intractable problems and its application to artificial intelligence," Ph.D. dissertation, Dipartimento di Informatica e Sistemistica, Universitá di Roma "La Sapienza", 1998.