

Ontological CP-Nets

Tommaso Di Noia¹, Thomas Lukasiewicz²(✉), Maria Vanina Martinez²,
Gerardo I. Simari², and Oana Tifrea-Marcuska²

¹ Department of Electrical and Information Engineering,
Polytechnic University of Bari, Bari, Italy
tommaso.dinoia@poliba.it

² Department of Computer Science, University of Oxford, Oxford, UK
{thomas.lukasiewicz, vanina.martinez,
gerardo.simari, oana.tifrea}@cs.ox.ac.uk

Abstract. Representing and reasoning about preferences is a key issue in many real-world scenarios in which personalized access to information is required. Many approaches have been proposed and studied in the literature that allow a system to work with qualitative or quantitative preferences; among the qualitative models, one of the most prominent are CP-nets. Their clear graphical structure unifies an easy representation of user preferences with good computational properties when computing the best outcome. In this paper, we show how to reason with CP-nets when the attributes modeling the knowledge domain are structured via an underlying domain ontology. We show how the computation of all undominated feasible outcomes of an ontological CP-net can be reduced to the solution of a constraint satisfaction problem, and study the computational complexity of the basic reasoning problems in ontological CP-nets.

1 Introduction

During the recent years, several revolutionary changes have been taking place in the classical Web. First, the so-called Web of Data is increasingly being realized as a special case of the Semantic Web. Second, as part of the Social Web, users are acting more and more as first-class citizens in the creation and delivery of contents on the Web. The combination of these two technological waves is called the *Social Semantic Web* (or also *Web 3.0*), where the classical Web of interlinked documents is gradually turning into (i) semantic data and tags constrained by ontologies, and (ii) social data, such as connections, interactions, reviews, and tags. The Web is thus shifting away from data on linked Web pages and towards more semantic and social data. This requires new technologies for search and query answering, where the ranking of search results is no longer solely based on the link structure between Web pages, but rather on the information available in the Social Semantic Web—in particular, the underlying ontological knowledge and the preferences of the users. Given a query, the latter play a fundamental role when a crisp yes/no answer is not enough to satisfy a user's needs, as there might be a certain degree of uncertainty in the possible answers [14].

There exist two main ways of modeling preferences: (a) *quantitative preferences* are associated with a number (or a quantity) representing their worth (e.g., “my preference for wi-fi connection is 0.8” and “my preference for cable connection is 0.4”), while (b) *qualitative preferences* are related to each other via pairwise comparisons (e.g., “I prefer wi-fi over cable connection”). The two approaches can also be combined (see, e.g., [24]). In many applications in practice, it is more natural to use a qualitative approach, as humans are not always comfortable or capable of expressing their preferences via a meaningful numerical value. To have a quantitative representation of her preferences, the user needs to explicitly determine a value for a large number of alternatives, usually described by more than one attribute. It is generally much easier to provide information about preferences as pairwise qualitative comparisons [14]. One of the most powerful qualitative preference formalisms are perhaps CP-nets [5], which are described in a graphical way and unifies an easy representation of user desires with nice computational properties when computing the best outcome.

Previous work on CP-nets, and more generally on preference representation approaches, mainly deals with a propositional representation of preferences. In this paper, we propose an enhancement of CP-nets by adding ontological information associated with preferences. We especially aim at using the resulting ontology-based partial strict orders in personalized semantic search on the Social Web. This is a first step towards a new type of ranking technologies, which are based on ontological and personalized information, and which go beyond PageRank and similar rankings. They will exploit ontological background knowledge as well as social information (e.g., from social networks and other platforms) and model them as semantic-enabled user preferences.

The main contributions of this paper can be briefly summarized as follows.

- We introduce ontological CP-nets, which combine CP-nets with description logics (DLs) so that variable values (which can be non-Boolean) correspond to DL concepts relative to an underlying domain TBox. We define the notions of feasible outcomes, dominance between such outcomes, and consistency in this context.
- We define the semantics of ontological CP-nets by a reduction to the notion of constrained CP-nets, which allows the problem of finding optimal outcomes in ontological CP-nets to be reduced to a constraint satisfaction problem.
- We study the complexity of the main reasoning problems for ontological CP-nets, namely consistency checking, whether a given outcome is undominated, and dominance testing, in relation to both the complexity of checking satisfiability of the underlying ontological language and the structure of the CP-net:
 - For tractable ontology languages, we show that the complexity is determined by that of CP-nets; i.e., the problems are complete for PSPACE.
 - For EXP- (resp., NEXP-) complete ontology languages, the complexity of these problems is dominated by the complexity of the ontology language.
 - Finally, if the CP-net is a polytree, and the ontology language is tractable, we show that dominance can be decided in polynomial time.

The rest of this paper is organized as follows. In Sect. 2, we briefly recall the background on description logics (DLs) and on CP-nets. Section 3 introduces ontological CP-nets, i.e., CP-nets enriched with ontological descriptions, while Sect. 4 describes how to compute optimal outcomes of ontological CP-nets. In Sects. 5 and 6, we provide complexity results and discuss related work, respectively. Finally, we give a summary of the results presented in this paper and an outlook on future work.

2 Preliminaries

In this section, we briefly recall from the literature the basics of description logics (DLs) and of CP-nets (a graphical representation for conditional preferences), which both form the main components of the formalism that we present in this paper.

2.1 Description Logics

Intuitively, description logics (DLs) [1] model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations on classes of individuals, respectively. A DL knowledge base (or ontology) encodes in particular (i) subsumption relationships between concepts, (ii) subsumption relationships between roles, (iii) instance relationships between individuals and concepts, and (iv) instance relationships between pairs of individuals and roles, which represent (i) subset relationships between classes of individuals, (ii) subset relationships between binary relations on classes of individuals, (iii) the membership of individuals to classes, and (iv) the membership of pairs of individuals to binary relations on classes, respectively. There are many different DLs of different expressiveness [1]. In this section, we recall the DLs $\mathit{SHIF}(\mathbf{D})$ and $\mathit{SHOIN}(\mathbf{D})$, which stand behind the web ontology languages OWL Lite and OWL DL [18], respectively. Note, however, that the approach in this paper does not depend on a specific DL and may, e.g., also be applied to the very expressive $\mathit{SROIQ}(D)$, which is the logic language behind OWL 2 [30].

Syntax. We now recall the syntax of $\mathit{SHIF}(\mathbf{D})$ and $\mathit{SHOIN}(\mathbf{D})$. We first describe the syntax of the latter, which has the following datatypes and elementary components. We assume a set of *elementary datatypes* and a set of *data values*. A *datatype* is an elementary datatype or a set of data values (called *datatype oneOf*). A *datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a *datatype domain* $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each elementary datatype a subset of $\Delta^{\mathbf{D}}$ and to each data value an element of $\Delta^{\mathbf{D}}$. We extend $\cdot^{\mathbf{D}}$ to all datatypes by $\{v_1, \dots\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \dots\}$. Let \mathbf{A} , \mathbf{R}_A , \mathbf{R}_D , and \mathbf{I} be pairwise disjoint sets of *atomic concepts*, *abstract roles*, *datatype roles*, and *individuals*, respectively. We denote by \mathbf{R}_A^- the set of inverses R^- of all $R \in \mathbf{R}_A$.

Roles and concepts are defined as follows. A *role* is any element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$. *Concepts* are inductively defined as follows. Each $\phi \in \mathbf{A}$ is a concept, and if $o_1, \dots, o_n \in \mathbf{I}$, then $\{o_1, \dots, o_n\}$ is a concept (called *oneOf*). If ϕ, ϕ_1 , and ϕ_2 are concepts and if $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, then also $\neg\phi$, $(\phi_1 \sqcap \phi_2)$, and $(\phi_1 \sqcup \phi_2)$ are concepts (called *negation*, *conjunction*, and *disjunction*, respectively), as well as $\exists R.\phi$, $\forall R.\phi$, $\geq nR$, and $\leq nR$ (called *existential*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. Note that for decidability reasons, number restrictions will be restricted to simple abstract roles (see below). If D is a datatype and $U \in \mathbf{R}_D$, then $\exists U.D$, $\forall U.D$, $\geq nU$, and $\leq nU$ are concepts (called *datatype existential*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. We use \top (resp., \perp) to abbreviate the *top* (resp., *bottom*) *concept* $\phi \sqcup \neg\phi$ (resp., $\phi \sqcap \neg\phi$). Furthermore, we write $\exists R$ to abbreviate $\exists R.\top$, and we eliminate parentheses as usual.

We next define axioms and knowledge bases. An *axiom* is an expression of one of the following forms: (1) $\phi \sqsubseteq \psi$ (called *concept inclusion axiom*), where ϕ and ψ are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A \cup \mathbf{R}_A^-$ or $R, S \in \mathbf{R}_D$; (3) $\text{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $\phi(a)$ (called *concept membership axiom*), where ϕ is a concept and $a \in \mathbf{I}$; (5) $R(a, b)$ (resp., $U(a, v)$) (called *role membership axiom*), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value); and (6) $a = b$ (resp., $a \neq b$) (*equality* (resp., *inequality*) *axiom*), where $a, b \in \mathbf{I}$. Two axioms $\phi \sqsubseteq \psi$ and $\psi \sqsubseteq \phi$ of the form (1) are also abbreviated as $\phi \equiv \psi$ (called *definition axiom*). Note that such axioms can be used to define a new atomic concept ϕ used as synonym of a concept ψ . A *TBox* \mathcal{T} is a finite set of axioms of the form (1), (2), (3), and (6), while an *ABox* \mathcal{A} is a finite set of axioms of the form (4) and (5). A *knowledge base* (or *ontology*) KB is a finite set of axioms (1)–(6).

We next define simple abstract roles. For abstract roles $R \in \mathbf{R}_A$, we define $\text{Inv}(R) = R^-$ and $\text{Inv}(R^-) = R$. Let \sqsubseteq_{KB}^* denote the reflexive and transitive closure of \sqsubseteq on $\bigcup \{ \{ R \sqsubseteq S, \text{Inv}(R) \sqsubseteq \text{Inv}(S) \} \mid R \sqsubseteq S \in KB, R, S \in \mathbf{R}_A \cup \mathbf{R}_A^- \}$. An abstract role S is *simple* relative to KB iff for each abstract role R such that $R \sqsubseteq_{KB}^* S$, it holds that (i) $\text{Trans}(R) \notin KB$ and (ii) $\text{Trans}(\text{Inv}(R)) \notin KB$. Informally, an abstract role S is simple iff it is neither transitive nor has transitive subroles. For decidability, number restrictions in KB are restricted to simple abstract roles [19].

In $\text{SHOIN}(\mathbf{D})$, concept and role membership axioms can also be expressed in terms of concept inclusion axioms, since $\phi(a)$ can be expressed by $\{a\} \sqsubseteq \phi$, while $R(a, b)$ (resp., $U(a, v)$) can be expressed by $\{a\} \sqsubseteq \exists R.\{b\}$ (resp., $\{a\} \sqsubseteq \exists U.\{v\}$).

The syntax of $\text{SHIF}(\mathbf{D})$ is as the above syntax of $\text{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the atleast and atmost constructors limited to 0 and 1.

Example 1 (Conference Organization). A simple TBox \mathcal{T}_{conf} may describe a conference organization website and consist of the following axioms:

$Hotel \sqsubseteq \exists hasRoom;$	$\exists hasRoom^- \sqsubseteq Room;$
$Hotel \sqsubseteq \exists hasBuilding;$	$\exists hasBuilding^- \sqsubseteq Building;$
$Room \sqsubseteq \exists hasFeature;$	$\exists hasFeature^- \sqsubseteq Feature;$
$Room \sqsubseteq \exists hasRoomPack;$	$\exists hasRoomPack^- \sqsubseteq Package;$
$Old \sqsubseteq Building;$	$Small \sqsubseteq \neg Medium;$
$New \sqsubseteq Building;$	$Small \sqsubseteq \neg Large;$
$Old \sqsubseteq \neg New;$	$Medium \sqsubseteq \neg Large;$
$Wifi \sqsubseteq Feature;$	$LuxuryPackage \sqsubseteq Package;$
$VideoConference \sqsubseteq Feature;$	$StandardPackage \sqsubseteq Package;$
$\exists hasFeature. Wifi \sqsubseteq \exists hasFeature. VideoConference;$	$StandardPackage \sqsubseteq \neg LuxuryPackage;$
$\exists hasRoomPack. StandardPackage \sqsubseteq \neg \exists hasRoomPack. LuxuryPackage.$	

Roughly speaking, the above axioms describe that hotels have either new or old buildings, and they have either small, medium, or large rooms. In turn, rooms have associated features such as wi-fi, video conference, etc.; if a room has wi-fi, then one can use the wi-fi connection for video conferences. Finally, rooms have associated stay packages; there are two types of packages; luxury packages in general (though not always) provide features that standard packages do not, for instance wi-fi connection. ■

Semantics. We now define the semantics of $SHIF(\mathbf{D})$ and $SHOIN(\mathbf{D})$ in terms of general first-order interpretations, as usual. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a nonempty (*abstract*) domain $\Delta^{\mathcal{I}}$ disjoint from $\Delta^{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $\phi \in \mathbf{A}$ a subset of $\Delta^{\mathcal{I}}$, to each individual $o \in \mathbf{I}$ an element of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{R}_A$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each datatype role $U \in \mathbf{R}_D$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$. We extend $\cdot^{\mathcal{I}}$ to all roles and concepts as usual (where $\#S$ denotes the cardinality of a set S):

- $(R^-)^{\mathcal{I}} = \{(y, x) \mid (x, y) \in R^{\mathcal{I}}\};$
- $\{o_1, \dots, o_n\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \dots, o_n^{\mathcal{I}}\}; (\neg\phi)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \phi^{\mathcal{I}};$
- $(\phi_1 \sqcap \phi_2)^{\mathcal{I}} = \phi_1^{\mathcal{I}} \cap \phi_2^{\mathcal{I}}; (\phi_1 \sqcup \phi_2)^{\mathcal{I}} = \phi_1^{\mathcal{I}} \cup \phi_2^{\mathcal{I}};$
- $(\exists R.\phi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y: (x, y) \in R^{\mathcal{I}} \wedge y \in \phi^{\mathcal{I}}\};$
- $(\forall R.\phi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y: (x, y) \in R^{\mathcal{I}} \rightarrow y \in \phi^{\mathcal{I}}\};$
- $(\geq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(y \mid (x, y) \in R^{\mathcal{I}})\} \geq n\};$
- $(\leq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(y \mid (x, y) \in R^{\mathcal{I}})\} \leq n\};$
- $(\exists U.D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y: (x, y) \in U^{\mathcal{I}} \wedge y \in D^{\mathbf{D}}\};$
- $(\forall U.D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y: (x, y) \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\};$
- $(\geq nU)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(y \mid (x, y) \in U^{\mathcal{I}})\} \geq n\};$
- $(\leq nU)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(y \mid (x, y) \in U^{\mathcal{I}})\} \leq n\}.$

The *satisfaction* of an axiom F in an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$, denoted $\mathcal{I} \models F$, is defined as follows: (1) $\mathcal{I} \models \phi \sqsubseteq \psi$ iff $\phi^{\mathcal{I}} \subseteq \psi^{\mathcal{I}}$; (2) $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$; (3) $\mathcal{I} \models \text{Trans}(R)$ iff $R^{\mathcal{I}}$ is transitive; (4) $\mathcal{I} \models \phi(a)$ iff $a^{\mathcal{I}} \in \phi^{\mathcal{I}}$; (5) $\mathcal{I} \models R(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$; (6) $\mathcal{I} \models U(a, v)$ iff $(a^{\mathcal{I}}, v^{\mathbf{D}}) \in U^{\mathcal{I}}$; (7) $\mathcal{I} \models a = b$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$; and (8) $\mathcal{I} \models a \neq b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

The interpretation \mathcal{I} satisfies the axiom F , or \mathcal{I} is a *model* of F , iff $\mathcal{I} \models F$. We say that \mathcal{I} satisfies a knowledge base KB , or \mathcal{I} is a *model* of KB , denoted $\mathcal{I} \models KB$, iff $\mathcal{I} \models F$ for all $F \in KB$. We say that KB is *satisfiable* (resp., *unsatisfiable*) iff KB has a (resp., no) model. An axiom F is a *logical consequence* of KB , denoted $KB \models F$, iff each model of KB satisfies F . We say that ϕ is *subsumed* (resp., *not subsumed*) by ψ relative to KB , denoted $\phi \sqsubseteq_{KB} \psi$ (resp., $\phi \not\sqsubseteq_{KB} \psi$), iff $KB \models \phi \sqsubseteq \psi$ (resp., $KB \not\models \phi \sqsubseteq \psi$). We say that ϕ is *unsatisfiable* (resp., *satisfiable*) relative to KB , denoted $\phi \sqsubseteq_{KB} \perp$ (resp., $\phi \not\sqsubseteq_{KB} \perp$), iff $KB \models \phi \sqsubseteq \perp$ (resp., $KB \not\models \phi \sqsubseteq \perp$).

2.2 CP-Nets

CP-nets [5] are a widespread formalism to represent and reason with qualitative preferences. More specifically, they are a graphical representation for conditional *ceteris paribus* (all else being equal) preference statements, which allows the specification of preferences based on the notion of conditional preferential independence (CPI) [20].

We assume a finite set of variables \mathbf{V} , where each variable $X^i \in \mathbf{V}$ has a finite *domain of values*, denoted $Dom(X^i)$. A *value* for a set of variables $\mathbf{X} = \{X^1, \dots, X^n\} \subseteq \mathbf{V}$ is a mapping $\mathbf{x}: \mathbf{X} \rightarrow \bigcup_{i=1}^n Dom(X^i)$ such that $\mathbf{x}(X^i) \in Dom(X^i)$ for all $i \in \{1, \dots, n\}$; the *domain* of \mathbf{X} , denoted $Dom(\mathbf{X})$, is the set of all values for \mathbf{X} . If \mathbf{x} and \mathbf{y} are values for disjoint sets of variable $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$, then \mathbf{xy} denotes the combination of \mathbf{x} and \mathbf{y} . A *preference relation* \succ is a strict partial order (an irreflexive and transitive binary relation). We write $o_1 \succeq o_2$ iff either $o_1 \succ o_2$ or $o_1 = o_2$. We say that o_1 is *strictly preferred* (resp., *strictly or equally preferred*) to o_2 iff $o_1 \succ o_2$ (resp., $o_1 \succeq o_2$). We say that o_2 is *dominated* by o_1 iff $o_1 \succ o_2$, and that o_2 is *directly dominated* by o_1 , denoted $o_1 \succ_d o_2$ iff (i) o_2 is dominated by o_1 , and (ii) no o exists such that $o_1 \succ o$ and $o \succ o_2$. We say that o_1 is *undominated* iff no o exists with $o \succ o_1$.

A CP-net \mathcal{N} over \mathbf{V} consists of a directed graph $G = (\mathbf{V}, E)$, where the nodes are variables in \mathbf{V} and if there is a directed edge $(X^j, X^i) \in E$, this shows that the preferences over values of X^i is influenced by the values of X^j . Each node $X^i \in \mathbf{V}$ has an annotated conditional preference table (CPT), denoted $CPT(X^i)$, which associates a total (or partial) order $\succ_{\mathbf{u}}^i$ over the values of X^i with each value \mathbf{u} of X^i 's parents in G , denoted $Pa(X^i)$. Intuitively, given a particular value assignment to $Pa(X^i)$, one is able to determine a preference order for the values of X^i , all other things being equal.

An *outcome* of \mathcal{N} is a value $\mathbf{o} \in Dom(\mathbf{V})$. A preference relation \succ on the set of all outcomes of \mathcal{N} is defined via the notion of worsening flip, which informally combines the preference relations on the values of the variables of \mathcal{N} (as encoded by their CPTs) to a preference relation on the outcomes of \mathcal{N} . More specifically, given two outcomes \mathbf{o} and \mathbf{o}' of \mathcal{N} , it holds $\mathbf{o} \succ_{wf} \mathbf{o}'$ in \mathcal{N} iff (i) there exist a variable $X^i \in \mathbf{V}$, values $x, x' \in Dom(X^i)$, and a value $\mathbf{u} \in Dom(Pa(X^i))$ such that (i) $\mathbf{o}(X^i) = x$, (ii) $\mathbf{o}'(X^i) = x'$, (iii) $\mathbf{o}(X^j) = \mathbf{o}'(X^j)$ for all $X^j \in \mathbf{V} \setminus \{X^i\}$, (iv) $\mathbf{o}(Pa(X^i)) = \mathbf{u}$, and (v) $x \succ_{\mathbf{u}}^i x'$; here, the single change from x to x' , moving from \mathbf{o} to \mathbf{o}' , is called a *worsening flip*. The preference relation \succ is

then defined as the transitive closure of \succ_{wf} . Intuitively, $\mathbf{o} \succ \mathbf{o}'$ iff there exists a sequence of worsening flips from \mathbf{o} to \mathbf{o}' .

CP-net actually assumes that its directed graph encodes conditional preferential independences (CPIs). In detail, let \mathbf{X} , \mathbf{Y} , and \mathbf{Z} partition \mathbf{V} , and let \succ be a preference relation over $Dom(\mathbf{V})$. Then, \mathbf{X} is *conditionally preferentially independent* of \mathbf{Y} given \mathbf{Z} iff for all $\mathbf{x}, \mathbf{x}' \in Dom(\mathbf{X})$, $\mathbf{y}, \mathbf{y}' \in Dom(\mathbf{Y})$, $\mathbf{z} \in Dom(\mathbf{Z})$, we have that $\mathbf{x}\mathbf{y}\mathbf{z} \succ \mathbf{x}'\mathbf{y}\mathbf{z}$ iff $\mathbf{x}\mathbf{y}'\mathbf{z} \succ \mathbf{x}'\mathbf{y}'\mathbf{z}$. Hence, every variable in a CP-net, given its parents, is conditionally preferentially independent of all the other variables in the CP-net.

Example 2. A CP-net with five variables R , W , B , C , and P that expresses the preferences over different features of hotels is shown in Fig. 1. The variables have the domains $\{r_s, r_m, r_l\}$, $\{w_y, w_n\}$, $\{b_o, b_n\}$, $\{c_y, c_n\}$, and $\{p_s, p_l\}$, respectively. This CP-net tells us that large rooms (r_l) are preferred to medium rooms (r_m), which in turn are preferred to small rooms (r_s). If only small rooms are available then a hotel in an old building is preferred to one in a new building (perhaps for historical reasons). A hotel that has a wi-fi connection (w_y) is preferred to one that does not (w_n). Furthermore, a hotel that facilitates video conferences (c_y) is preferred to one that does not (c_n). If the hotel is a new building, and the rooms facilitate video conferences, then the luxury package (p_l) is preferred to the standard package, otherwise the standard package is preferred (p_s). Here, we have a worsening flip moving from $\mathbf{o}_1 = r_l w_y b_n c_y p_l$ to $\mathbf{o}_2 = r_l w_y b_o c_y p_l$, since b_n is preferred to b_o , given r_l . Similarly, we have a worsening flip from \mathbf{o}_2 to $\mathbf{o}_3 = r_l w_n b_o c_y p_l$. Hence, $\mathbf{o}_1 \succ \mathbf{o}_2$, $\mathbf{o}_2 \succ \mathbf{o}_3$, and $\mathbf{o}_1 \succ \mathbf{o}_3$. Given the room size, the building type is conditionally preferentially independent of all the other variables. ■

The following are the two main computational tasks for CP-nets:

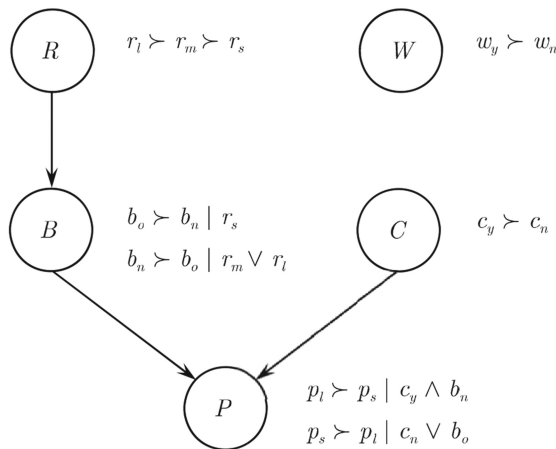


Fig. 1. CP-net for Example 2

- *Dominance query*: given a CP-net \mathcal{N} and two outcomes o_1 and o_2 of \mathcal{N} , decide whether $o_1 \succ o_2$ holds in \mathcal{N} .
- *Outcome optimization*: given a CP-net \mathcal{N} , compute an undominated outcome of \mathcal{N} .

Acyclic CP-nets (i.e., the associated directed graph does not have any directed cycles) with total orders in their conditional preference tables have only one undominated outcome, which can be computed in linear time [4]. The algorithm just follows the order among the variables that is represented by the directed graph and assigns values to the variables A_i from top to bottom, satisfying the preference relations in the CPTs corresponding to the variables. For arbitrary CP-nets, deciding dominance queries are PSPACE-complete [16], and computing an optimal outcome is NP-hard [12].

2.3 Constrained CP-Nets

In constrained CP-nets [6, 26], constraints among variables are added to the basic formalism of CP-nets, which may reduce the set of possible outcomes. The approach in [26] to finding all optimal outcomes of a CP-net (if some exist) relies on a reduction of the preferences represented in the CP-net to a set of hard constraints taking into account the variables occurring in the preferences. Given a CP-net \mathcal{N} and a set of constraints \mathcal{C} , an outcome o is *feasible* iff it satisfies all the constraints in \mathcal{C} . A feasible outcome is *Pareto-optimal* iff it is undominated [6]. In [13], the authors present an approach to finding the Pareto-optimal outcomes by solving a constraint satisfaction problem in the presence of soft and hard constraints. Here, we focus only on the latter. For every variable A and every instantiation γ of its parents in \mathcal{N} , the conditional preferences for $Dom(A)$ encoded in \mathcal{N} yield an optimality constraint. The undominated outcomes of $(\mathcal{N}, \mathcal{C})$ are then exactly the solutions of the conjunction of all optimality constraints. For example, consider a variable A in \mathcal{N} with domain $Dom(A) = \{a_1, \dots, a_m\}$ and an instantiation γ of its parents in \mathcal{N} . If the conditional preferences encoded in \mathcal{N} are a total order over the values in $Dom(A)$, i.e., $a_1 \succ a_2 \succ \dots \succ a_m$, which means that a_1 is undominated relative to all other a_i , then the optimality constraint is given by $\gamma \rightarrow a_1$. In the most general case, where we have more than one undominated value, the optimality constraint is given by:

$$\gamma \rightarrow \bigvee_{a_i \in undom(A|\gamma)} a_i, \quad (1)$$

where $undom(A|\gamma)$ denotes the set of all undominated values of A under γ . In the following, we consider only the case with one undominated value, i.e., $|undom(A|\gamma)| = 1$. The approach can be extended to the general case in a straightforward way.

3 Ontological CP-Nets

We now introduce an approach to ontological CP-net-based preference representation, which combines CP-nets and DLs, harnessing the technologies described

in the previous section. Intuitively, the main idea behind this approach is to use certain satisfiable concepts relative to an underlying TBox as values of the variables of a CP-net. More precisely, the values are taken from a finite nonempty set \mathcal{C} of *basic classification concepts* (or *basic c-concepts* for short), which are (not necessarily atomic) concepts C in $\mathit{SHIF}(\mathbf{D})$ (resp., $\mathit{SHOIN}(\mathbf{D})$) that are free of individuals from \mathbf{I} .

Definition 1 (ontological CP-net). Let \mathbf{V} be a finite set of variables. An *ontological CP-net* $(\mathcal{N}, \mathcal{T})$ over \mathbf{V} consists of a CP-net \mathcal{N} over \mathbf{V} and a TBox \mathcal{T} such that the domain of each variable $A \in \mathbf{V}$ is of the form $\text{Dom}(A) = \{\alpha_1, \dots, \alpha_m\}$, where:

1. every $\alpha_i, i \in \{1, \dots, m\}$, is a concept from \mathcal{C} that is satisfiable relative to \mathcal{T} ,
2. $\mathcal{T} \models \alpha_i \sqcap \alpha_j \sqsubseteq \perp$ for all $i, j \in \{1, \dots, m\}$ with $i < j$, and
3. $\mathcal{T} \models \top \sqsubseteq \alpha_i \sqcup \dots \sqcup \alpha_m$.

The following example illustrates the above notion of ontological CP-net.

Example 3 (Conference Organization cont'd). An ontological CP-net $(\mathcal{N}, \mathcal{T})$ over \mathbf{V} is given by the TBox \mathcal{T} from Example 1 and the CP-net \mathcal{N} over \mathbf{V} from Example 2, where the values of the variables are now defined as the following DL concepts:

$$\begin{aligned}
 r_s &= \exists \text{hasRoom.Small}; & r_m &= \exists \text{hasRoom.Medium}; \\
 r_l &= \exists \text{hasRoom.Large}; \\
 w_y &= \exists \text{hasRoom} . (\exists \text{hasFeature.Wifi}); & w_n &= \neg w_y; \\
 b_o &= \exists \text{hasBuilding.Old}; & b_n &= \exists \text{hasBuilding.New}; \\
 c_y &= \exists \text{hasRoom} . (\exists \text{hasFeature.VideoConference}); & c_n &= \neg c_y; \\
 p_l &= \exists \text{hasRoom} . (\exists \text{hasRoomPack.LuxuryPackage}); \\
 p_s &= \exists \text{hasRoom} . (\exists \text{hasRoomPack.StandardPackage}).
 \end{aligned}$$

Observe here that even if we do not have any explicit hard constraint expressed among the values of the variables of the CP-net, due to their logical structure and the underlying TBox, we have a set of implicit constraints among these values. We will show in Sect. 4.2 below how to explicitly encode these constraints. Hence, due to these constraints among the values of the variables of the CP-net, some outcomes are infeasible, where outcomes are values o of the set \mathbf{V} of all variables of the CP-net. The following definition formally introduces feasible outcomes and undominated feasible outcomes (which are not dominated by any other feasible outcome) as well as the consistency of CP-nets as the existence of at least one undominated feasible outcome.

Definition 2 (feasible outcome, dominance, and consistency). Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over \mathbf{V} , an outcome $o \in \text{Dom}(\mathbf{V})$ is *feasible* iff $\prod o(\mathbf{V})$ ($= \prod_{A \in \mathbf{V}} o(A)$) is satisfiable relative to \mathcal{T} . A feasible outcome o is *undominated* iff no feasible outcome o' exists such that $o' \succ o$. We say that $(\mathcal{N}, \mathcal{T})$ is *consistent* iff it has a feasible outcome. ■

The following example briefly illustrates the above notions of feasible outcomes, undominated feasible outcomes, and consistent CP-nets.

Example 4 (Conference Organization cont'd). Reconsider the ontological CP-net of the running example. The outcome $r_l b_n w_y c_y p_l$ is feasible and undominated, while $r_l b_n w_y c_n p_s$ is not feasible (as the availability of wi-fi implies the availability of video conferences). Thus, the ontological CP-net of the running example is consistent. ■

4 Computing Optimal Outcomes

The main computational task around ontological CP-nets that we want to solve in this paper is how to determine all undominated feasible outcomes of a consistent ontological CP-net. In this section, we show how to compute them, given an ontological CP-net. The approach mainly relies on the HARD-PARETO algorithm of [26] (see Algorithm 1).

Generalizing the results of Sect. 2.3 to ontologies, we express the undominated feasible outcomes of an ontological CP-net $(\mathcal{N}, \mathcal{T})$, if some exist, in an ontological way as follows. For every variable A and every value γ of its parents in $(\mathcal{N}, \mathcal{T})$, the conditional preferences for $Dom(A)$ encoded in $(\mathcal{N}, \mathcal{T})$ yield the optimality constraint

$$\prod \gamma(Pa(A)) \sqsubseteq \bigsqcup undom(A|\gamma).$$

Let $DL-opt(\mathcal{N})$ denote the set of all these optimality constraints. The undominated feasible outcomes of $(\mathcal{N}, \mathcal{T})$ are then exactly the set of all outcomes $o \in Dom(\mathbf{V})$ such that $\prod o(\mathbf{V})$ is satisfiable relative to $\mathcal{T} \cup DL-opt(\mathcal{N})$.

4.1 Propositional Compilation of DL Formulas

A TBox can be seen as a set of logical constraints that reduces the set of models for a formula. Given a set of concepts \mathcal{F} , we now show how to compute a compact representation of a TBox \mathcal{T} as a set of clauses whose variables have a one-to-one mapping to the concepts in \mathcal{F} . Hereafter, we write $\tilde{\phi}$ to denote $\tilde{\phi} \in \{\phi, \neg\phi\}$.

Definition 3 (ontological constraint). Given a TBox \mathcal{T} and a set of satisfiable concepts $\mathcal{F} = \{\phi_i \mid i \in \{1, \dots, n\}\}$ relative to \mathcal{T} , we say that \mathcal{F} is *minimally constrained* relative to \mathcal{T} iff

1. there exists a concept $\tilde{\phi}_1 \sqcup \dots \sqcup \tilde{\phi}_n$ such that $\mathcal{T} \models \top \sqsubseteq \tilde{\phi}_1 \sqcup \dots \sqcup \tilde{\phi}_n$, and
2. there is no proper subset $\mathcal{E} \subset \mathcal{F}$ such that the previous condition holds.

The ontological axiom $\top \sqsubseteq \tilde{\phi}_1 \sqcup \dots \sqcup \tilde{\phi}_n$ is an *ontological constraint*.

An ontological constraint is an explicit representation of the constraints existing among a set of concepts, due to the information encoded in the TBox \mathcal{T} .

Definition 4 (ontological closure). Given a TBox \mathcal{T} and a set of satisfiable concepts $\mathcal{F} = \{\phi_1, \dots, \phi_n\}$ relative to \mathcal{T} , the *ontological closure* of \mathcal{F} and \mathcal{T} , denoted $\mathcal{OCC}(\mathcal{F}, \mathcal{T})$, is the set of all ontological constraints, if any, for each subset of \mathcal{F} .

The ontological closure of \mathcal{F} and \mathcal{T} is an explicit representation of all the logical constraints among a set of concepts \mathcal{F} , considering also an underlying TBox \mathcal{T} .

Proposition 1. *Given a TBox \mathcal{T} and a set of satisfiable concepts $\mathcal{F} = \{\phi_1, \dots, \phi_n\}$ relative to \mathcal{T} , if $\mathcal{T} \models \prod_{i=1}^n \tilde{\phi}_i \sqsubseteq \perp$, then $\mathcal{OCC}(\mathcal{F}, \mathcal{T}) \models \prod_{i=1}^n \tilde{\phi}_i \sqsubseteq \perp$.*

Proof. Since $\mathcal{T} \models \prod_{i=1}^n \tilde{\phi}_i \sqsubseteq \perp$, this means that we have the corresponding clause $\psi = \bigsqcup_{i=1}^n \neg \tilde{\phi}_i$ such that $\mathcal{T} \models \top \sqsubseteq \psi$. If $\mathcal{F} = \{\phi_1, \dots, \phi_n\}$ is minimally constrained, then $\psi \in \mathcal{OCC}(\mathcal{F}, \mathcal{T})$, otherwise, by definition of $\mathcal{OCC}(\mathcal{F}, \mathcal{T})$, there will be a clause $\psi' \in \mathcal{OCC}(\mathcal{F}, \mathcal{T})$ such that $\psi' \subset \psi$. \square

Hence, if we are interested only in the relationships between predefined concepts (due to their logical structure and \mathcal{T}), then the corresponding ontological closure is a compact and complete representation. Note that we are only looking for minimal clauses, as we are interested in computing the actual constraints between the formulas representing the domain of a variable in the CP-net. Hence, to compute the final outcomes, it is preferable to deal with a compact representation of all possible constraints.

Example 5 (Conference Organization cont'd). Given the set of concepts $\mathcal{F} = \{w_y, c_y, p_l, p_s\}$, due to the axioms in the TBox, we have the minimally constrained sets $\mathcal{F}' = \{w_y, c_y\}$ and $\mathcal{F}'' = \{p_l, p_s\}$ and the two corresponding ontological constraints $\top \sqsubseteq \neg w_y \sqcup c_y$ (indeed $w_y \sqsubseteq_{\mathcal{T}} c_y$) and $\top \sqsubseteq \neg p_l \sqcup \neg p_s$ (as $p_l \sqcap p_s \sqsubseteq_{\mathcal{T}} \perp$). The corresponding ontological closure is then $\mathcal{OCC}(\mathcal{F}, \mathcal{T}) = \{\top \sqsubseteq \neg w_y \sqcup c_y, \top \sqsubseteq \neg p_l \sqcup \neg p_s\}$. \blacksquare

The set $\tilde{\mathcal{F}} = \{\tilde{\phi}_1, \dots, \tilde{\phi}_n\}$ is a *feasible* assignment for \mathcal{F} and \mathcal{T} iff

$$\mathcal{OCC}(\mathcal{F}, \mathcal{T}) \not\models \prod_{i=1}^n \tilde{\phi}_i \sqsubseteq \perp.$$

We are interested in feasible assignments, since (as we will show in the following), they represent feasible outcomes for an ontological CP-net. Note that by Proposition 1, if $\tilde{\mathcal{F}} = \{\tilde{\phi}_1, \dots, \tilde{\phi}_n\}$ is a feasible assignment for \mathcal{F} and \mathcal{T} , then $\mathcal{T} \not\models \prod_{i=1}^n \tilde{\phi}_i \sqsubseteq \perp$, i.e., $\prod_{i=1}^n \tilde{\phi}_i$ is satisfiable relative to \mathcal{T} . The next proposition shows that there always exist feasible assignments for sets of satisfiable concepts relative to an underlying TBox.

Proposition 2. *Let \mathcal{T} be a TBox, and $\mathcal{F} = \{\phi_1, \dots, \phi_n\}$ be a set of satisfiable concepts relative to \mathcal{T} . Then, there always exists a feasible assignment for \mathcal{F} and \mathcal{T} .*

Proof. If $\mathcal{OCC}(\mathcal{F}, \mathcal{T}) = \emptyset$, then every $\tilde{\mathcal{F}} = \{\tilde{\phi}_1, \dots, \tilde{\phi}_n\}$ is a feasible assignment for \mathcal{F} and \mathcal{T} . Otherwise, since $\mathcal{T} \models \mathcal{OCC}(\mathcal{F}, \mathcal{T})$, every interpretation \mathcal{I} that satisfies \mathcal{T} also satisfies $\mathcal{OCC}(\mathcal{F}, \mathcal{T})$. That is, \mathcal{I} satisfies every $\top \sqsubseteq \tilde{\phi}_{i_1} \sqcup \dots \sqcup \tilde{\phi}_{i_m} \in \mathcal{OCC}(\mathcal{F}, \mathcal{T})$, which means that $(\tilde{\phi}_{i_j})^{\mathcal{I}} \neq \emptyset$ for some $j \in \{1, \dots, m\}$ (as every interpretation \mathcal{I} has a nonempty domain $\Delta^{\mathcal{I}}$). Thus, there exists a feasible assignment for \mathcal{F} and \mathcal{T} . \square

4.2 Computing Optimal Outcomes

If we have an ontological CP-net $(\mathcal{N}, \mathcal{T})$, the variable values (concepts) in a set \mathcal{F} may constrain each other, and the corresponding constraints are encoded in $\mathcal{OCC}(\mathcal{F}, \mathcal{T})$. The ontological closure of a set of concepts explicitly represents all the logical constraints among them with respect to an underlying ontology. The computation of all undominated feasible outcomes for an ontological CP-net goes through the Boolean encoding of both the ontology \mathcal{T} and of the clauses corresponding to the preferences represented in the CPTs of \mathcal{N} for each variable $A \in \mathbf{V}$. To use HARD-PARETO, we need a few pre-processing steps. Given the ontological CP-net $(\mathcal{N}, \mathcal{T})$:

1. for every $A^i \in \mathbf{V}$ and every $\alpha_j^i \in \text{Dom}(A^i) = \{\alpha_1^i, \dots, \alpha_{m_i}^i\}$, choose a fresh atomic concept V_j^i ;
2. define the ontology $\mathcal{T}' = \mathcal{T} \cup \{V_j^i \equiv \alpha_j^i \mid A^i \in \mathbf{V}, j \in \{1, \dots, m_i\}\}$;
3. define the ontological CP-net $(\mathcal{N}', \mathcal{T}')$, where \mathcal{N}' is obtained from \mathcal{N} by isomorphically replacing every α_j^i by V_j^i , for all $A^i \in \mathbf{V}$ and $j \in \{1, \dots, m_i\}$;
4. define $\mathcal{F} = \{V_j^i \mid A^i \in \mathbf{V}, j \in \{1, \dots, m_i\}\}$;
5. compute $\mathcal{OCC}(\mathcal{F}, \mathcal{T}')$;
6. introduce a Boolean variable v_j^i for each $V_j^i \in \mathcal{F}$;
7. transform $\mathcal{OCC}(\mathcal{F}, \mathcal{T}')$ into the corresponding set of Boolean clauses \mathcal{C} by replacing V_j^i by the corresponding binary variable v_j^i (and the ontological constant “ \top ” and the ontological connectives “ \sqcup ” and “ \sqsubseteq ” by the Boolean constant “*true*” and the Boolean connectives “ \vee ” and “ \rightarrow ”, respectively);
8. transform $DL\text{-}opt(\mathcal{N}')$ into the set of Boolean clauses $opt(\mathcal{N}')$ by replacing every V_j^i by the corresponding variable v_j^i (and the ontological connectives “ \sqcap ”, “ \sqcup ”, and “ \sqsubseteq ” by the Boolean connectives “ \wedge ”, “ \vee ”, and “ \rightarrow ”, respectively).

Note that \mathcal{T} is logically equivalent to \mathcal{T}' , as we only use equivalence axioms to define new concepts V_j^i as synonyms of complex concepts α_j^i . The same holds for $(\mathcal{N}, \mathcal{T})$ and $(\mathcal{N}', \mathcal{T}')$, as we just replace concepts in $\text{Dom}(A^i)$ by equivalent concepts.

Once we have \mathcal{C} and $opt(\mathcal{N}')$, we can compute the optimal outcome of $(\mathcal{N}, \mathcal{T})$ by using the slightly modified version of HARD-PARETO, shown in Algorithm 1. The function $sol(\cdot)$ used in Algorithm 1 computes all the solutions for the Boolean constraint satisfaction problem represented by \mathcal{C} , $opt(\mathcal{N}')$ and $\mathcal{C} \cup opt(\mathcal{N}')$. Differently from the original HARD-PARETO, by Proposition 2, we know that \mathcal{C} is always consistent, and so we do not need to check its consistency

at the beginning of the algorithm. Moreover, note that the algorithm works with propositional variables although we are computing undominated feasible solutions for an ontological CP-net. That is, the dominance test in line 11 can be computed using well-known techniques for Boolean problems.

Input: $opt(\mathcal{N}')$ and \mathcal{C}

```

1  $S_{opt} \leftarrow sol(\mathcal{C} \cup opt(\mathcal{N}'))$ ;
2 if  $S_{opt} = sol(\mathcal{C})$  then
3   | return  $S_{opt}$ ;
4 end
5 if  $sol(opt(\mathcal{N}')) \neq \emptyset$  and  $S_{opt} = sol(opt(\mathcal{N}'))$  then
6   | return  $S_{opt}$ ;
7 end
8  $S \leftarrow sol(\mathcal{C}) - S_{opt}$ ;
9 repeat
10  | choose  $o \in S$ ;
11  | if  $\forall o' \in sol(\mathcal{C}) - o, o' \not\prec o$  then
12  |   |  $S_{opt} \leftarrow S_{opt} \cup \{o\}$ ;
13  |   end
14  |  $S \leftarrow S - \{o\}$ ;
15 until  $S = \emptyset$ ;
16 return  $S_{opt}$ .

```

Algorithm 1. Algorithm HARD-PARETO adapted to ontological CP-nets

The outcomes returned by Algorithm 1 in S_{opt} are *true/false* assignments to the Boolean variables v_j^i . To compute undominated outcomes for the original ontological CP-net $(\mathcal{N}, \mathcal{T})$, we need to revert to a DL setting. Hence, we build the set $DL-S_{opt}$, where for each outcome $o \in S_{opt}$, we add to $DL-S_{opt}$ the following value o' :

$$o'(A^i) = V_j^i \text{ iff } o(v_j^i) = \textit{true}, \text{ for all } A_i \in \mathbf{V} \text{ and } j \in \{1, \dots, m_i\}.$$

The following example shows a trace of Algorithm 1 for our running example.

Example 6 (Conference Organization cont'd). For the CP-net in Fig. 1, we obtain:

$$\begin{aligned}
- \mathcal{T}' = \mathcal{T} \cup \{ & V_1^1 \equiv \exists hasRoom.Small, \quad V_2^1 \equiv \exists hasRoom.Medium, \\
& V_3^1 \equiv \exists hasRoom.Large, \quad V_1^2 \equiv \exists hasRoom.(\exists hasFeature.Wifi), \\
& V_2^2 \equiv \neg \exists hasRoom.(\exists hasFeature.Wifi), \\
& V_1^3 \equiv \exists hasBuilding.Old, \quad V_2^3 \equiv \exists hasBuilding.New, \\
& V_1^4 \equiv \exists hasRoom.(\exists hasFeature.VideoConference), \\
& V_2^4 \equiv \neg \exists hasRoom.(\exists hasFeature.VideoConference), \\
& V_1^5 \equiv \exists hasRoom.(\exists hasRoomPack.LuxuryPackage), \\
& V_2^5 \equiv \exists hasRoom.(\exists hasRoomPack.StandardPackage) \};
\end{aligned}$$

- $\mathcal{F} = \{V_1^1, V_2^1, V_3^1, V_1^2, V_2^2, V_1^3, V_2^3, V_1^4, V_2^4, V_1^5, V_2^5\}$;
- $\mathcal{OCC}(\mathcal{F}, \mathcal{T}') = \{\top \sqsubseteq \neg V_1^2 \sqcup V_1^4, \top \sqsubseteq \neg V_1^1 \sqcup \neg V_2^1, \top \sqsubseteq \neg V_2^1 \sqcup \neg V_3^1, \top \sqsubseteq \neg V_1^1 \sqcup \neg V_3^1, \top \sqsubseteq V_1^1 \sqcup V_2^1 \sqcup V_3^1, \top \sqsubseteq \neg V_1^2 \sqcup \neg V_2^2, \top \sqsubseteq V_1^2 \sqcup V_2^2, \top \sqsubseteq \neg V_1^3 \sqcup \neg V_2^3, \top \sqsubseteq V_1^3 \sqcup V_2^3, \top \sqsubseteq \neg V_1^4 \sqcup \neg V_2^4, \top \sqsubseteq V_1^4 \sqcup V_2^4, \top \sqsubseteq \neg V_1^5 \sqcup \neg V_2^5, \top \sqsubseteq V_1^5 \sqcup V_2^5\}$;
- $\mathcal{C} = \{\neg v_1^2 \vee v_1^4, \neg v_1^1 \vee \neg v_2^1, \neg v_2^1 \vee \neg v_3^1, \neg v_1^1 \vee \neg v_3^1, v_1^1 \vee v_2^1 \vee v_3^1, \neg v_1^2 \vee \neg v_2^2, v_1^2 \vee v_2^2, \neg v_1^3 \vee \neg v_2^3, v_1^3 \vee v_2^3, \neg v_1^4 \vee \neg v_2^4, v_1^4 \vee v_2^4, \neg v_1^5 \vee \neg v_2^5, v_1^5 \vee v_2^5\}$;
- $DL-opt(N') = \{V_3^1, V_1^2, V_1^1 \sqsubseteq V_1^3, V_2^1 \sqsubseteq V_2^3, V_1^4, V_3^1 \sqsubseteq V_2^3, V_2^3 \sqcap V_1^4 \sqsubseteq V_1^5, V_1^3 \sqcap V_1^4 \sqsubseteq V_2^5, V_2^3 \sqcap V_2^4 \sqsubseteq V_2^5, V_1^3 \sqcap V_2^4 \sqsubseteq V_2^5\}$;
- $opt(N') = \{v_3^1, v_2^1, v_1^1 \rightarrow v_3^1, v_2^1 \rightarrow v_2^3, v_1^4, v_3^1 \rightarrow v_2^3, v_2^3 \wedge v_1^4 \rightarrow v_5^1, v_1^3 \wedge v_1^4 \rightarrow v_5^2, v_2^3 \wedge v_2^4 \rightarrow v_5^2, v_1^3 \wedge v_2^4 \rightarrow v_5^2\}$.

Then, $S_{opt} = \{v_3^1 v_2^1 v_2^3 v_1^4 v_1^5\}$, and $r_l w_y b_n c_y p_l$ is the only optimal outcome. \blacksquare

The following theorem shows the correctness of the algorithm.

Theorem 1. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over \mathbf{V} , the values $o' \in DL-S_{opt}$ are all the undominated feasible outcomes for $(\mathcal{N}, \mathcal{T})$.*

Proof. We start by showing that o' is a feasible outcome. If we consider the final assignment $o = \bigwedge_{i=1}^{|\mathbf{V}|} \bigwedge_{j=1}^{m_i} \tilde{v}_j^i$, the corresponding formula $o' = \prod_{i=1}^{|\mathbf{V}|} \prod_{j=1}^{m_i} \tilde{V}_j^i$ is a feasible assignment. In fact, if we had $\mathcal{OCC}(\mathcal{F}, \mathcal{T}') \models \prod_{i=1}^{|\mathbf{V}|} \prod_{j=1}^{m_i} \tilde{V}_j^i \sqsubseteq \perp$, then we should have the corresponding constraint (or one that implies) $\bigvee_{i=1}^{|\mathbf{V}|} \bigvee_{j=1}^{m_i} \neg \tilde{v}_j^i$ in \mathcal{C} , thus not allowing o to be a solution. By Definition 1, for each variable A^i , both $\mathcal{T} \models a_j^i \sqcap a_{j'}^i \sqsubseteq \perp$, for all $j, j' \in \{1, \dots, m_i\}$ with $j < j'$, and $\mathcal{T} \models \top \sqsubseteq a_1^i \sqcup \dots \sqcup a_{m_i}^i$. These axioms are encoded in the corresponding binary constraints $\neg v_j^i \vee \neg v_{j'}^i$ and $v_1^i \vee \dots \vee v_{m_i}^i$ saying that, given A^i , in o , we have all v_j^i negated but one. As a consequence, in o' , we have only one V_j^i for each A^i , i.e., o' is an outcome. Overall, o' is a feasible outcome. Finally, as o satisfies all the optimality constraints, o' is an undominated outcome. \square

5 Computational Complexity

We now explore the complexity of the main computational problems in ontological CP-nets for underlying ontological languages with typical complexity of deciding knowledge base satisfiability, namely, tractability and completeness for EXP and NEXP. We also provide some special tractable cases of dominance testing in ontological CP-nets.

5.1 General Results

For tractable ontology languages (i.e., those for which deciding knowledge base satisfiability is tractable), the complexity of ontological CP-nets is dominated

by the complexity of CP-nets. That is, deciding (a) consistency, (b) whether a given outcome is undominated, and (c) dominance of two given outcomes are all PSPACE-complete. Note also that the same complexity results hold for ontology languages with PSPACE-complete knowledge base satisfiability checks and that even computing the set of all undominated outcomes (generalizing (b)) is PSPACE-complete under the condition that there are only polynomially many of them.

Theorem 2. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over a tractable ontology language,*

- (a) *deciding whether $(\mathcal{N}, \mathcal{T})$ is consistent,*
- (b) *deciding whether a given outcome o is undominated,*
- (c) *deciding whether $o \succ o'$ for two given outcomes o and o'*

are all PSPACE-complete.

Proof (sketch). The lower bounds follow immediately from the fact that ontological CP-nets generalize CP-nets, for which these problems are all PSPACE-complete [16].

As for the upper bounds, compared to standard CP-nets, these problems additionally involve knowledge base satisfiability checks, which can all be done in polynomial time and thus also in polynomial space. Note that in (a) (resp., (b)), one has to go through all outcomes o' and check that it is not the case that $o \succ o'$ (resp., $o' \succ o$), which can each and thus overall be done in polynomial space. \square

In particular, if the ontological CP-net is defined over a DL of the *DL-Lite* family [9] (which all allow for deciding knowledge base satisfiability in polynomial time, such as *DL-Lite_R*, which stands behind the important OWL 2 QL profile [31]), deciding (a) consistency, (b) whether a given outcome is undominated, and (c) dominance of two given outcomes are all PSPACE-complete.

Corollary 1. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over a DL from the *DL-Lite* family,*

- (a) *deciding whether $(\mathcal{N}, \mathcal{T})$ is consistent,*
- (b) *deciding whether a given outcome o is undominated,*
- (c) *deciding whether $o \succ o'$ for two given outcomes o and o'*

are all PSPACE-complete.

For EXP (resp., NEXP) complete ontology languages (i.e., those for which knowledge base satisfiability is complete for EXP (resp., NEXP)), the complexity of ontological CP-nets is dominated by the complexity of the ontology languages. That is, deciding (a) inconsistency, (b) whether a given outcome is dominated, and (c) dominance of two given outcomes are all complete for EXP (resp., NEXP). Note that computing the set of all undominated outcomes (generalizing (b)) is also EXP-complete for EXP-complete ontology languages.

Theorem 3. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over an EXP (resp., NEXP) complete ontology language,*

- (a) *deciding whether $(\mathcal{N}, \mathcal{T})$ is inconsistent,*
- (b) *deciding whether a given outcome o is dominated,*
- (c) *deciding whether $o \succ o'$ for two given outcomes o and o'*

are all complete for EXP (resp., NEXP).

Proof (sketch). The lower bounds follow from the fact that all three problems in ontological CP-nets can be used to decide knowledge base satisfiability in the underlying ontology language.

As for the upper bounds, in (a) and (b), we have to go through all outcomes, which is in EXP (resp., NEXP). Then, we have to perform knowledge base satisfiability checks, which are also in EXP (resp., NEXP), and dominance checks in standard CP-nets, which are in PSPACE, and thus also in EXP (resp., NEXP). Overall, (a) to (c) are thus in EXP (resp., NEXP). \square

In particular, if the ontological CP-net is defined over the expressive DL $\mathit{SHIF}(\mathbf{D})$ (resp., $\mathit{SHOIN}(\mathbf{D})$) [18] (which stands behind OWL Lite (resp., OWL DL) [30, 32], and allows for deciding knowledge base satisfiability in EXP [18, 29] (resp., NEXP, for both unary and binary number encoding; see [25, 29] and the NEXP-hardness proof for ACCQIO in [29], which implies the NEXP-hardness of $\mathit{SHOIN}(\mathbf{D})$), deciding (a) inconsistency, (b) whether a given outcome is dominated, and (c) dominance of two given outcomes are all complete for EXP (resp., NEXP).

Corollary 2. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over the DL $\mathit{SHIF}(\mathbf{D})$ (resp., $\mathit{SHOIN}(\mathbf{D})$),*

- (a) *deciding whether $(\mathcal{N}, \mathcal{T})$ is inconsistent,*
- (b) *deciding whether a given outcome o is dominated,*
- (c) *deciding whether $o \succ o'$ for two given outcomes o and o'*

are all complete for EXP (resp., NEXP).

5.2 Tractability Results

If the ontological CP-net is a polytree (i.e., the underlying undirected graph has no cycles) and defined over a tractable ontology language, deciding dominance of two outcomes can be done in polynomial time. Note that polytree ontological CP-nets are always consistent.

Theorem 4. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over a tractable ontology language, where \mathcal{N} is a polytree, deciding whether $o \succ o'$ for two given outcomes o and o' can be done in polynomial time.*

Proof (sketch). We have to decide whether (i) $o \succ o'$ holds in \mathcal{N} and (ii) o and o' are feasible outcomes of $(\mathcal{N}, \mathcal{T})$. The former can be done in polynomial time, as for standard polytree CP-nets, dominance can be decided in polynomial time [5], while the latter can also be done in polynomial time in tractable ontology languages. \square

In particular, if the ontological CP-net is a polytree and defined over a DL of the *DL-Lite* family, deciding dominance of two outcomes can be done in polynomial time.

Corollary 3. *Given an ontological CP-net $(\mathcal{N}, \mathcal{T})$ over a DL from the DL-Lite family, where \mathcal{N} is a polytree, deciding whether $o \succ o'$ for two given outcomes o and o' can be done in polynomial time.*

6 Related Work

Modeling and dealing with preferences has traditionally been studied in several areas within computer science. In the databases community, the work of [21] stands out as the seminal work in the area; see [28] for a survey of notable works in this line. Much work has also been carried out in the intersection of databases and knowledge representation and reasoning, such as in preference logic programs [17], incorporation of preferences into formalisms such as answer set programs [8], and answering k -rank queries in ontological languages [22].

On the other hand, in the philosophical tradition, preferences are usually expressed over mutually exclusive “worlds”, such as truth assignments to formulas. The work of [2] is framed in this interpretation of preferences, aiming at bridging the gap between several formalisms from the AI community such as CP-nets and those studied traditionally in philosophy. In this regard, CP-nets [5] is one of the most widely known formalisms. More recently, the work of Wang et al. [33] proposes an efficient algorithm and indexing scheme for top- k retrieval in CP-nets.

Constrained CP-nets were originally proposed in [6], along with algorithm SEARCH-CP, which uses branch and bound to compute undominated outcomes. The algorithm has an anytime behavior: it can be stopped at any time, and the set of computed solutions are a subset of the set containing all the undominated outcomes. This means that in case one is interested in any undominated outcome, one can use the first one returned by SEARCH-CP. In [26], HARD-PARETO is presented; the most notable difference is that HARD-PARETO does not rely on topological information like SEARCH-CP, but it exploits only the CP-statements, thus allowing to work also with cyclic CP-nets. Differently from the previous two papers, in our work, we allow the variable domains to contain DL formulas constrained via ontological axioms.

Recently, there has been some interest regarding the combination of Semantic Web technologies with preference representation and reasoning. A combination of conditional preferences (very different from CP-nets) with DL reasoning for ranking objects is introduced in [24]. A ranking function is

described that exploits conditional preferences to perform a semantic personalized search and ranking over a set of resources annotated via an ontological description. In [22], Datalog+/- is extended with preference management formalisms closely related to those previously studied for relational databases; the authors further extend this formalism to the case where ontological inferences involve probabilistic uncertainty [23]. Another interesting approach to mixing qualitative preferences with Semantic Web technology is presented in [27], where an extension of SPARQL that can encode user preferences in the query is proposed.

There is, however, very little work on the particular combination of Semantic Web technologies and CP-nets. To our knowledge, the most notable work is that of [3], developed within an information retrieval context; in this work, Wordnet is used to add semantics to CP-net variables.

7 Summary and Outlook

In classical decision theory and analysis, the preferences of decision makers are modeled by utility functions. Unfortunately, the effort needed to obtain a good utility function requires a significant involvement of the user [15]. This is one of the main reasons behind the success obtained by CP-nets since they were originally proposed [4]: they are compact, easily understandable, and well-suited for combinatorial domains, such as multi-attribute ones. In this paper, we have described how to reason with CP-nets that are augmented by assigning description logic axioms to its variable values—such axioms refer to a common underlying ontology and constrain the possible outcomes in the CP-net. Furthermore, we studied the complexity of the problems of consistency checking, whether a given outcome is undominated, and dominance testing for ontological CP-nets, showing how the complexity of checking satisfiability of the underlying ontological language and the structure of the CP-net affects the complexity of solving these problems. The proposed framework is very useful in many semantic retrieval scenarios, among which we distinguish semantic search.

Other formalisms related to the original CP-nets have been subsequently proposed in the literature, such as TCP-nets (Trade-off CP-nets) [7] or CP-theories [34]. TCP-nets extend CP-nets by allowing to express also statements of relative importance between variables. With TCP-nets, the user is allowed to express her preferences over compromises that sometimes may be required. CP-theories generalize (T)CP-nets allowing conditional preference statements on the values of a variable, along with a set of variables that are allowed to vary when interpreting the preference statement. In future work, we plan to enrich these frameworks by introducing ontological descriptions and reasoning, thus allowing the development of more powerful semantic-enabled preference-based retrieval systems.

Acknowledgements. This work was supported by the UK EPSRC grant EP/J0083-46/1 “ProQAW: Probabilistic Ontological Query Answering on the Web”, by a Google

European Doctoral Fellowship, by the ERC (FP7/2007-2013) grant 246858 (“DIADEM”), by a Yahoo! Research Fellowship, and by PON02.00563.3470993 (“VINCENTE”). This paper is a significantly extended and revised version of papers that appeared in *Proceedings URSW-2013* [10] and *Proceedings SUM-2013* [11].

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook*. Cambridge University Press, Cambridge (2002)
2. Biennu, M., Lang, J., Wilson, N.: From preference logics to preference languages, and back. In: *Proceedings of KR*, pp. 214–224. AAAI Press (2010)
3. Boubekur, F., Boughanem, M., Tamine-Lechani, L.: Semantic information retrieval based on CP-nets. In: *Proceedings of FUZZ-IEEE*, pp. 1–7. IEEE Computer Society (2007)
4. Boutilier, C., Brafman, R.I., Hoos, H.H., Poole, D.: Reasoning with conditional ceteris paribus preference statements. In: *Proceedings of UAI*, pp. 71–80. Morgan Kaufmann (1999)
5. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.* **21**, 135–191 (2004)
6. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: Preference-based constrained optimization with CP-nets. *Comput. Intell.* **20**(2), 137–157 (2004)
7. Brafman, R.I., Domshlak, C., Shimony, S.E.: On graphical modeling of preference and importance. *J. Artif. Intell. Res.* **25**(1), 389–424 (2006)
8. Brewka, G.: Preferences, contexts and answer sets. In: Dahl, V., Niemelä, I. (eds.) *ICLP 2007*. LNCS, vol. 4670, pp. 22–22. Springer, Heidelberg (2007)
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. *J. Autom. Reasoning* **39**(3), 385–429 (2007)
10. Di Noia, T., Lukasiewicz, T.: Introducing ontological cp-nets. In: *Proceedings of URSW, CEUR Workshop Proceedings*, vol. 900, pp. 90–93. CEUR-WS.org (2012)
11. Di Noia, T., Lukasiewicz, T., Simari, G.I.: Reasoning with semantic-enabled qualitative preferences. In: Liu, W., Subrahmanian, V.S., Wijsen, J. (eds.) *SUM 2013*. LNCS, vol. 8078, pp. 374–386. Springer, Heidelberg (2013)
12. Domshlak, C., Brafman, R.I.: CP-nets-reasoning and consistency testing. In: *Proceedings of KR*, pp. 121–132. Morgan Kaufmann (2002)
13. Domshlak, C., Prestwich, S.D., Rossi, F., Venable, K.B., Walsh, T.: Hard and soft constraints for reasoning about qualitative conditional preferences. *J. Heuristics* **12**(4/5), 263–285 (2006)
14. Domshlak, C., Hüllermeier, E., Kaci, S., Prade, H.: Preferences in AI: an overview. *Artif. Intell.* **175**(7/8), 1037–1052 (2011)
15. French, S.: *Decision Theory: An Introduction to the Mathematics of Rationality*. Prentice Hall, Englewood Cliffs (1988)
16. Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in CP-nets. *J. Artif. Intell. Res.* **33**, 403–432 (2008)
17. Govindarajan, K., Jayaraman, B., Mantha, S.: Preference logic programming. In: *Proceedings of ICLP*, pp. 731–745. MIT Press (1995)

18. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 17–29. Springer, Heidelberg (2003)
19. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. In: Ganzinger, H., McAllester, D., Voronkov, A. (eds.) LPAR 1999. LNCS, vol. 1705, pp. 161–180. Springer, Heidelberg (1999)
20. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Trade-Offs. Cambridge University Press, Cambridge (1993)
21. Lacroix, M., Lavency, P.: Preferences: putting more knowledge into queries. In: Proceedings of VLDB, pp. 1–4. Morgan Kaufmann (1987)
22. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Preference-based query answering in Datalog+/- ontologies. In: Proceedings of IJCAI, pp. 1017–1023 (2013)
23. Lukasiewicz, T., Martinez, M.V., Simari, G.I., Tifrea-Marcuska, O.: Preference-based query answering in probabilistic Datalog+/- ontologies. *J. Data Semant.* 1–21 (2014). <http://dx.doi.org/10.1007/s13740-014-0040-x>
24. Lukasiewicz, T., Schellhase, J.: Variable-strength conditional preferences for ranking objects in ontologies. *J. Web Sem.* **5**(3), 180–194 (2007)
25. Pratt-Hartmann, I.: Complexity of the two-variable fragment with counting quantifiers. *J. Log. Lang. Inf.* **14**(3), 369–395 (2005)
26. Prestwich, S.D., Rossi, F., Venable, K.B., Walsh, T.: Constraint-based preferential optimization. In: Proceedings of AAI, pp. 461–466. AAAI Press (2005)
27. Siberski, W., Pan, J.Z., Thaden, U.: Querying the semantic web with preferences. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 612–624. Springer, Heidelberg (2006)
28. Stefanidis, K., Koutrika, G., Pitoura, E.: A survey on representation, composition and application of preferences in database systems. *ACM Trans. Database Syst.* **36**(3), 19 (2011)
29. Tobies, S.: Complexity results and practical algorithms for logics in knowledge representation. Ph.D. thesis, RWTH Aachen, Germany (2001)
30. W3C: OWL 2 Web Ontology Language Primer (2nd Edition). W3C Recommendation, 11 December 2012. <http://www.w3.org/TR/owl2-primer/>
31. W3C: OWL 2 Web Ontology Language Profiles. W3C Recommendation, 27 October 2009. <http://www.w3.org/TR/owl2-profiles/>
32. W3C: OWL Web Ontology Language Overview. W3C Recommendation, 10 February 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
33. Wang, H., Zhou, X., Chen, W., Ma, P.: Top-k retrieval using conditional preference networks. In: Proceedings of CIKM, pp. 2075–2079. ACM Press (2012)
34. Wilson, N.: Extending CP-nets with stronger conditional preference statements. In: Proceedings of AAI, pp. 735–741. AAAI Press (2004)