

Introducing Ontological CP-Nets

Tommaso Di Noia¹ and Thomas Lukasiewicz²

¹ Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy
t.dinoia@poliba.it

² Department of Computer Science, University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Abstract. Preference representation and reasoning is a key issue in many real-world scenarios. Currently, there are many approaches allowing preferences to be assessed in a qualitative or quantitative way. The most prominent qualitative approach for representing preferences are CP-nets. Their clear graphical structure unifies an easy representation of user desires with nice computational properties when computing the best outcome. Here, we introduce ontological CP-nets, which allow the representation of preferences using a CP-net over an ontological domain, i.e., variable values are logical formulas constrained relative to a background domain ontology.

1 Motivation

During the last five years, we have seen two main phenomena emerging from the classical Web. On the one side, we have had the social revolution, where the users act as first-class citizens in the creation and delivery of contents over the Web. On the other side, we have seen an increasing interest in the so-called Web of Data as a special case of the Semantic Web vision. These two technological waves (Social Web and Semantic Web) led to what is known as Web 3.0, i.e., a Web where, on top of the classical Web of interlinked documents, we have the two layers represented by: (a) user contents, connections, interactions, reviews, tags, etc.; and (b) semantic data and tags constrained by ontologies. All this information may be exploited to create semantic user profiles containing a representation of users' preferences. Given a query, these latter play a fundamental role when a crisp yes/no answer is not enough to satisfy a user's needs, since there is a certain degree of uncertainty in possible answers [4]. We have two main ways of modeling preferences: (a) *quantitative preferences* are associated with a number representing their worth or they are represented as an ordered set of objects (e.g., "my preference for WiFi 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 WiFi over cable connection"). The two approaches can also be combined (see, e.g., [6]).

As also stated in [4], the qualitative approach seems to be a more natural way of representing preferences, since humans are not very comfortable in expressing their "wishes" in terms of a 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 [4]. Among the diverse qualitative frameworks for preference representation and reasoning, one of the most powerful are CP-nets. They are a graphical language that unifies an easy representation of user desires with nice computational properties when computing the best outcome. In this paper, we propose an enhancement of CP-nets by adding ontological information associated to preferences. The rest of this paper is structured as follows. In Section 2, we briefly introduce CP-nets and constrained CP-nets. Section 3 then shows some of their limits, and introduces the notion of ontological CP-nets.

2 CP-Nets and Constrained CP-Nets

In this section, we introduce some necessary preliminary notions and formalisms. Given a set of variables \mathcal{V} , an *outcome* is an assignment to all the variables in \mathcal{V} . A preference relation \succeq

is a total pre-order over the set of outcomes. We write $o_1 \succ o_2$ iff o_1 is strictly preferred to o_2 , and $o_1 \succeq o_2$ iff o_1 is strictly or equally preferred to o_2 ; we then also say that o_2 is *dominated* by o_1 . If there is no outcome o such that $o \succ o_1$, we say that o_1 is *undominated*.

Conditional preference networks (CP-nets) [2] are a formalism to represent and reason with qualitative preferences. This compact but powerful language allows the specification of preferences based on the notion of conditional preferential independence. Fundamental for CP-nets is the notion of *conditionally preferentially independent (CPI)*. Let $P, Q \in \mathcal{V}$ be two variables and $\mathcal{R} \subset \mathcal{V}$ be a set of variables such that P, Q , and \mathcal{R} partition \mathcal{V} , and $Dom(P)$, $Dom(Q)$, and $Dom(\mathcal{R})$ represent all possible assignments for P, Q , and all the variables in \mathcal{R} , respectively. We say that P is *conditionally preferentially independent (CPI)* of Q given an assignment $r \in \mathcal{R}$ iff, for all $p_1, p_2 \in P$ and $q_1, q_2 \in Q$, we have that $p_1 q_1 r \succeq p_2 q_1 r$ iff $p_1 q_2 r \succeq p_2 q_2 r$. Here, \succeq represents the preference order among assignments for sets of variables. CP-nets are a graphical language to model CPI statements. Formally, a CP-net N consists of a directed graph G representing preference relations among variables P_i and a set of conditional preference tables $CPT(P_i)$ (one for each variable). Given the set of variables $\mathcal{V} = \{P_i \mid i \in \{1, \dots, n\}\} \cup \{P_{n+1}\}$ representing the nodes of G , such that P_i is a parent of P_{n+1} in G , the corresponding $CPT(P_{n+1})$ contains a preference for each pair of values of P_{n+1} conditioned to all possible assignments of variables P_i . The representation of a CP-net assumes that the user explicitly specifies her preferences over the values of P_{n+1} for each complete assignment of P_i , with $i \in \{1, \dots, n\}$. Given a CP-net N , we denote by CPT_i the set of all conditional preferences represented in $CPT(P_i)$, and $CPT_N = \{CPT_i \mid i \in \{1, \dots, n\}\}$.

Given a CP-net, the two main queries one may ask are:

- *Dominance query*: Given two outcomes o_1 and o_2 , decide whether $o_1 \succeq o_2$.
- *Outcome optimization*: What is the optimal outcome given the preferences represented in the CP-net? That is, we look for one of the undominated outcomes.

Given an acyclic CP-net, one can compute the best outcome in linear time. The algorithm just follows the order among variables represented by the graph and assigns values to the variables P_i from top to bottom satisfying the preference order in the corresponding $CPT(P_i)$. Finding the optimal outcome in cyclic CP-nets is NP-hard.

In *constrained CP-nets* [5, 3], constraints among variables are added to the basic formalism of CP-nets. Adding constraints among variables may reduce the set of possible outcomes. The approach to finding the optimal outcome proposed in [5] relies on a reduction of the preferences represented in the CP-net to a set of hard constraints (which can be represented in clause form for binary variables) taking into account the variables occurring in the preferences. Given a CP-net N and a set of constraints \mathcal{C} , an outcome o is *feasible* if it satisfies all the constraints in \mathcal{C} . A feasible outcome is *Pareto optimal* [3] iff it is undominated. In [5], the authors present an algorithm to find the optimal outcome solving a constraint satisfaction problem. For binary variables, given a conditional preference $(p_{n+1} \succeq \neg p_{n+1} \mid \bigwedge_{i=1 \dots n} \tilde{p}_i)$, where $\tilde{p}_i \in \{p_i, \neg p_i\}$, the corresponding constraint is the clause $\bigwedge_{i=1 \dots n} \tilde{p}_i \rightarrow p_{n+1}$ (analogously, for $(\neg p_{n+1} \succeq p_{n+1} \mid \bigwedge_{i=1 \dots n} \tilde{p}_i)$, we have $\bigwedge_{i=1 \dots n} \tilde{p}_i \rightarrow \neg p_{n+1}$).

Given a CP-Net N and a set of constraints \mathcal{C} , a feasible Pareto optimal outcome is an assignment satisfying the corresponding set of clauses and all the constraints in \mathcal{C} (and vice versa).

3 Ontological CP-Nets

We now introduce a framework for preference representation harnessing the technologies described in the previous section. The idea is to combine CP-nets and ontologies represented in description logics (DLs) [1]. In this combination, variable values are satisfiable DL concepts. For ease of presentation, we describe our approach using variables whose domain contains only two values, i.e., two concepts; the extension to more than two concepts is straightforward.

Two conditional preferences $(\alpha \succeq \beta \mid \gamma)$ and $(\alpha' \succeq \beta' \mid \gamma')$ are *equivalent* relative to an ontology \mathcal{T} iff $\mathcal{T} \models \gamma \equiv \gamma'$, $\mathcal{T} \models \alpha \equiv \alpha'$, and $\mathcal{T} \models \beta \equiv \beta'$.

Definition 1. An *ontological CP-net* (\mathcal{T}, N) consists of an ontology \mathcal{T} and a CP-net N such that: (1) for each variable P in N , $Dom(P) = \{\alpha, \beta\}$, where α and β are DL concepts that are satisfiable relative to \mathcal{T} , and such that $\mathcal{T} \not\models \alpha \equiv \top$, $\mathcal{T} \not\models \beta \equiv \top$, and $\mathcal{T} \not\models \alpha \equiv \beta$; (2) any two conditional preferences in CPT_N are pairwise not equivalent.

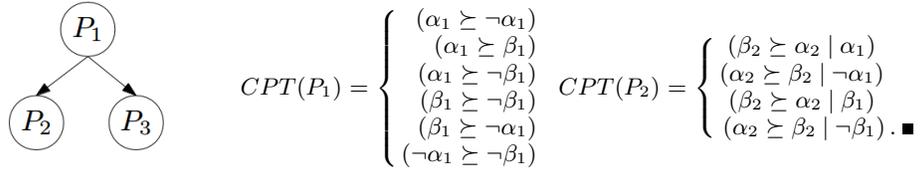
Note that each variable $P \in \mathcal{V}$ with $Dom(P) = \{\alpha, \beta\}$ may have one of the four values α , $\neg\alpha$, β , and $\neg\beta$. That is, the variables are not strictly binary.

Example 1 (Hotel). Consider a simple ontology, describing the services offered by a hotel:

$$\begin{aligned} \text{Scooter} &\sqsubseteq \text{Motorcycle} \\ \text{Motorcycle} &\sqsubseteq \neg\text{Bike} \\ \text{rent.Scooter} &\sqsubseteq \exists\text{facilities}.\text{(Parking} \sqcap \exists\text{payment} \sqcap \forall\text{payment.Free)}. \end{aligned}$$

A simple ontological CP-net is depicted in the following together with possible CPTs related to the nodes P_1 and P_2 . The domains of P_1 , P_2 and P_3 are:

$$\begin{aligned} Dom(P_1) &= \{\alpha_1 = \exists\text{location.OnTheSea}, \beta_1 = \exists\text{location.NearTheAirport}\} \\ Dom(P_2) &= \{\alpha_2 = \exists\text{rent.Bike}, \beta_2 = \exists\text{facilities}.\text{(Parking} \sqcap \exists\text{payment} \sqcap \forall\text{payment.Free})\} \\ Dom(P_3) &= \{\alpha_3 = \exists\text{rent} \sqcap \forall\text{rent.Scooter}, \beta_3 = \exists\text{facilities.Shuttle}\} \end{aligned}$$



Although we know how to reason with expressive DLs and with CP-nets, their combination leads to diverse issues both from the modeling and the reasoning perspective. We now sketch the main ideas behind our approach, using illustrative examples whenever possible.

Implicitly constrained variables. Even if we do not have any explicit hard constraint expressed among the variables of the CP-net, due to the background ontology, we have a set of implicit constraints among α and β values of the variables \mathcal{V} in the CP-net.

Example 2 (Hotel cont'd). Consider the ontology \mathcal{T} of Example 1 and the two variables P_2 and P_3 . Because of \mathcal{T} , we have the implicit constraint $\mathcal{T} \models \alpha_3 \sqsubseteq \beta_2$. ■

One way to infer all possible constraints among variable values is to adopt the ontology compilation technique presented in [7]. There, the authors propose an algorithm to elicit all possible hidden constraints (represented in clausal form) occurring among a set of DL concepts.

Preference satisfiability. Following [5], for each preference $\Phi = (\tilde{\alpha} \succeq \tilde{\beta} \mid \gamma) \in CPT_N$, we may write the clause: $\gamma \rightarrow \tilde{\alpha}$ (i.e., $\neg\gamma \sqcup \tilde{\alpha}$). In ontological CP-nets, this is not sufficient. Indeed, since α and β belong to the domain of the same variable, we have to explicitly state that they are disjoint with each other relative to \mathcal{T} . Hence, for each preference we have to add one more clause of the form $\tilde{\alpha} \rightarrow \neg\tilde{\beta}$. This may also lead to unsatisfiable clauses.

Example 3 (Hotel cont'd). Consider the preference (not allowed by the CP-net represented in Example 1) $\Phi = (\alpha_3 \succeq \beta_2 \mid \neg\alpha_1)$. If we imposed $\alpha_3 \rightarrow \neg\beta_2$ then we had $\mathcal{T} \models (\neg\alpha_1 \rightarrow \alpha_3) \sqcap (\alpha_3 \rightarrow \neg\beta_2) \sqsubseteq \perp$. In fact, we know that $\mathcal{T} \models \alpha_3 \sqsubseteq \beta_2$. That is, we are saying that Φ is never satisfied. ■

Hence, the preference Φ is *satisfiable* iff $\mathcal{T} \models (\gamma \rightarrow \tilde{\alpha}) \sqcap (\tilde{\alpha} \rightarrow \neg\tilde{\beta}) \not\sqsubseteq \perp$. The notion of satisfiability can be extended also to the whole CP-net.

Definition of outcome. In a constrained CP-net, if we had propositional true/false variables, an outcome would be a model, i.e., a true/false assignment that satisfies all the

constraints and some of the clauses built starting from the preferences represented in CPT_N . In ontological CP-nets, we deal with DL concepts, so a model satisfying the constraints cannot be explicitly represented. Actually, we have more than one equivalent outcome, i.e., all the models that satisfy the same preferences. A solution to this issue is to compute a formula whose models satisfy the same preferences. Also, in this case, such formula can be computed by adapting the techniques proposed in [7], where the preference satisfaction problem for DL concepts is solved via Integer Linear Programming encoding.

Dominance test and eligibility of CP-statements. A set of conditional preferences (CP-statements) is eligible iff it has an undominated outcome. In case of a set of eligible CP-statements, once we introduce an ontology to describe the background knowledge we can make the undominated outcome unsatisfiable. Then we have to be very careful when evaluating the dominance test among a set of possible outcomes of the CP-net. In fact, in case we have cycles in the dependency graph [2] associated to the CP-net, due to the presence of an ontology, we could be unable to find an undominated outcome.

Complexity of reasoning. As also argued in [5], having background knowledge may introduce implicit cycles in the graph representing the CP-net. This affects the computational complexity related to the computation of an outcome.

4 Conclusion

The availability of semantic information over the Web and the social revolution, pave the way to a new wave of personalized applications where ontological knowledge plays a fundamental role. User's preferences may act as a filter to the information accessed by the user in order to provide a personalized experience while interacting with the Semantic Web. Among the various formalisms proposed in the literature to represent preferences, a very promising one is that of CP-nets. They have a strong theoretical background and many results are already available in the literature both related to their computational properties and to modeling aspects. Nevertheless, to the best of our knowledge, almost nothing has been done to combine CP-nets with ontological modeling and reasoning. In this paper we introduce the notion of Ontological CP-nets and highlight some issues related to the ontological nature of the information we deal with when combined with conditional preferences arranged in a CP-net.

Acknowledgments. This work was supported by the EPSRC grant EP/J008346/1 ("PrOQAW"), the ERC under the EU's 7th Framework Programme (FP7/2007-2013/ERC) grant 246858 ("DIADEM"), a Google Research Award, and a Yahoo! Research Fellowship. Tommaso Di Noia also acknowledges support by HP IRP 2011, grant CW267313.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. 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. *JAIR*, 21:135–191, 2004.
3. C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Preference-based constrained optimization with CP-nets. *Comput. Intell.*, 20(2):137–157, 2004.
4. C. Domshlak, E. Hüllermeier, S. Kaci, and H. Prade. Preferences in AI: An overview. *Artif. Intell.*, 175(7/8):1037–1052, 2011.
5. C. Domshlak, S. Prestwich, F. Rossi, K. B. Venable, and T. Walsh. Hard and soft constraints for reasoning about qualitative conditional preferences. *J. Heuristics*, 12(4/5):263–285, 2006.
6. T. Lukasiewicz and J. Schellhase. Variable-strength conditional preferences for ranking objects in ontologies. *J. Web Sem.*, 5(3):180–194, 2007.
7. A. Ragone, T. D. Noia, F. M. Donini, E. D. Sciascio, and M. P. Wellman. Weighted description logics preference formulas for multiattribute negotiation. In *Proc. SUM-2009*, pp. 193–205, 2009.