J Heuristics (2006) 12: 263–285
DOI 10.1007/s10732-006-7071-x

Hard and soft constraints for reasoning about qualitative conditional preferences

C. Domshlak · S. Prestwich · F. Rossi · K.B. Venable · T. Walsh

Abstract Many real life optimization problems are defined in terms of both hard and soft constraints, and qualitative conditional preferences. However, there is as yet no single framework for combined reasoning about these three kinds of information. In this paper we study how to exploit classical and soft constraint solvers for handling qualitative preference statements such as those captured by the CP-nets model. In particular, we show how hard constraints are sufficient to model the optimal outcomes of a possibly cyclic CP-net, and how soft constraints can faithfully approximate the semantics of acyclic conditional preference statements whilst improving the computational efficiency of reasoning about these statements.

Keywords Preferences · Hard and Soft Constraints · CP-nets

*This material is based in part upon works supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075

C. Domshlak
William Davidson Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Technion City, Haifa, Israel
e-mail: dcarmel@ie.technion.ac.il

S. Prestwich
Department of Computer Science, University College Cork, Cork, Ireland
e-mail: s.prestwich@cs.ucc.ie

F. Rossi · K.B. Venable (✉)
Department of Mathematics, University of Padova, Padova, Italy
e-mail: {frossi, kvenable}@math.unipd.it

T. Walsh
National ICT Australia and School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
e-mail: tw@cse.unsw.edu.au

1. Introduction and motivation

Representing and reasoning about preferences is an area of increasing interest in theoretical and applied AI. In many real-life optimization problems we have both hard and soft constraints, as well as qualitative conditional preferences. For example, in a product configuration problem (Sabin and Weigel, 1998), the producer may pose some hard constraints on the problem (e.g., component compatibility) and soft constraints (e.g., supply time), while the user may provide the system with her subjective preferences over alternative products expressed in some natural language of preference statements. While soft constraint solvers (Bistarelli et al., 1997; Schiex et al., 1995) and various models for reasoning about qualitative preferences (Doyle and Thomason, 1999) have been proposed in AI research, there is as yet no single framework for efficient and effective combined reasoning about these different kinds of information. Although this is the long-term goal of our research, in this paper we focus on the connections between constraints and qualitative preferences representable by the CP-nets model (Boutilier et al., 2004), and exploit these connections to provide a purely constraint-based framework for combined reasoning about these two formalisms.

Soft constraints (Bistarelli et al., 1997; Schiex et al., 1995) are one of the main methods for dealing with preferences in constraint optimization. Each assignment to the variables of a constraint is annotated with the level of its desirability, and the desirability of a complete assignment is computed by a combination operator applied to the "local" preference values. Whilst soft constraints are very expressive and have a powerful computational machinery for reasoning about them, the fact that they are based on *quantitative* measures of preference and a global preference combination operator make the preference elicitation process somewhat difficult for a naive user.

To make preference elicitation process accessible to the masses, one must support reasoning about qualitative statements of preference that all of us express in our everyday activities (Doyle and Thomason, 1999; Hansson, 2001). Several models for representation and reasoning about such statements have been proposed in AI (Lang, 2002), of which the most studied to date is the CP-nets model (Boutilier et al., 2004), where CP stands for Conditional Preference. CP-nets are devoted to reasoning about qualitative (and possibly conditional) statements of preference where each statement expresses preference over the values of a single property of the outcomes, such as "I prefer a red dress to a yellow dress", or "If the car is convertible, I prefer a soft top to a hard top". The core notions exploited by the CP-nets model are the *ceteris paribus* (all else being equal) interpretation of the statements, and conditional preferential independence. However, while elicitation of CP-nets from lay users is very intuitive, the Achilles' heel of CP-nets and other sophisticated representation models of qualitative preferences is the complexity of reasoning with them (Domshlak and Brafman, 2002; Boutilier et al., 2004; Lang, 2002; Goldsmith et al., 2005).

In this paper we make a step towards bridging the gap between the attractiveness of preference elicitation with CP-nets, and the expressiveness and efficiency of reasoning with soft constraints. First, we consider the complexity of reasoning about qualitative preference statements, and in particular of verifying consistency of preference specification. We show how a set of hard constraints can model the set of optimal outcomes of a possibly cyclic CP-net, making off-the-shelf hard constraint solvers sufficient for finding optimal assignments with respect to such preference models. To tackle the complexity of reasoning about the whole preference orderings induced by CP-nets, we consider compiling CP-nets into soft constraint satisfaction problems. First, we show that the expressiveness of these two formalisms is incomparable, and hence the desired compilation must approximate the information captured by CP-nets. We then focus on acyclic CP-nets, and introduce two sound approximation schemes for such orderings that are based on the soft constraints formalism. Finally, we compare our two approximations in terms of both expressivity and complexity, and show how they can be combined into another approximation that is closer to the original information.

Returning to the configuration example described above, the preferences of the agents can be modelled via a CP-net, which can then be approximated by soft constraints. Such soft constraints can then be added to the hard constraints of the problem and solved using soft constraint satisfaction machinery.

To the best of our knowledge, this work provides the first formally sound framework for using hard and soft constraint solvers for reasoning about qualitative preference statements. In addition, our framework provides a platform for a combined (exact or approximate) reasoning about hard constraints, quantitative soft constraints, and certain sets of qualitative statements of preference.

The paper is organized as follows. In Section 2 we give the main notions about soft constraints and CP-nets. Then in Section 3 we show how hard constraints are sufficient to find optimal outcomes of a possibly cyclic set of statements. In Section 4 we compare the expressive power of soft constraints and CP-nets, while in Section 5 we show how to approximate an acyclic CP-net via two classes of soft constraints. Finally, Section 6 discusses related work, and Section 7 summarizes the main results and points at possible future directions for further work.

This paper is partially based on results contained in Domshlak et al. (2003). In addition, it includes the study of eligibility via optimality constraints and the comparison between the expressive power of soft constraints and CP-nets.

2. Formalisms for describing preferences

In this section we provide an essential background hard and soft constraints, as well as CP-nets.

2.1. Soft constraints

While several formalisms for describing *soft constraints* have been proposed in the literature, here we adopt the *c-semi-ring* formalism (Bistarelli et al., 1997), which is equivalent to the valued-CSP with total orders (Bistarelli et al., 1996), and generalizes numerous soft constraint settings (Schiex, 1992; Dubois et al., 1993; Freuder and Wallace, 1992). In brief, each soft constraint is defined over a certain set of variables, and it associates each instantiation of its variables with a value from a partially ordered set. In addition, we are given a pair of operations devoted for combining ($\times$) and comparing ($+$) values provided by soft constraints. A semi-ring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ such that: A is a set and $\mathbf{0}, \mathbf{1} \in A$; $+$ is commutative, associative and $\mathbf{0}$ is its unit element; $\times$ is associative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element. A *c-semi-ring* is a semi-ring $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ in which $+$ is idempotent, $\mathbf{1}$ is its absorbing element and $\times$ is commutative.

Let us consider the relation $\leq$ over A such that $a \leq b$ iff $a + b = b$. Then $\leq$ is a partial order, $+$ and $\times$ are monotone on $\leq$, $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum, $\langle A, \leq\rangle$ is a complete lattice and, for all $a, b \in A$, $a + b = lub(a, b)$. Moreover, if $\times$ is idempotent: $+$ distributes over $\times$; $\langle A, \leq\rangle$ is a complete distributive lattice and $\times$ its glb. Informally, the relation $\leq$ compares semi-ring values and constraints. When $a \leq b$, we say that *b is better than* (or *preferred to*) *a*. Given a semi-ring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$, a finite set D (variable domains) and an ordered set of variables V, a *constraint* is a pair $\langle def, con\rangle$ where $con \subseteq V$ and $def : D^{|con|} \rightarrow A$. A constraint specifies a set of variables, and assigns to each tuple of values of these variables an element of the semi-ring.

A (SCSP) is given by a set of soft constraints. For example, a classical CSP is an SCSP with the c-semi-ring $S_{CSP} = \langle\{false, true\}, \vee, \wedge, false, true\rangle$, a fuzzy CSP (Schiex, 1992) is an SCSP with the c-semi-ring $S_{FCSP} = \langle[0, 1], max, min, 0, 1\rangle$, and probabilistic and weighted CSPs are SCSPs with the c-semi-rings $S_{prob} = \langle[0, 1], max, \times, 0, 1\rangle$ and $S_{weight} = \langle\mathcal{R}, min, +, 0, +\infty\rangle$, respectively. A solution to an SCSP is a complete assignment to its variables. The preference value associated with a solution is obtained by multiplying the preference values of the projections of the solution to each constraint. A solution is considered to be better than some another solution if the preference value of the former is higher in the order than this of the latter.

Finding an optimal solution for an SCSP is an NP-hard problem, since SCSPs include classical CSPs which are NP-hard. On the other hand, given two solutions, checking whether one is preferable to another is easy: simply compute the semi-ring values of the two solutions and compare the resulting values. This takes time linear in the number of constraints if the number of variables involved in each constraint is bounded.

2.2. CP-nets

Soft constraints are the main tool for representing and reasoning about preferences in constraint satisfaction problems. However, they require specifying a numeric semi-ring value for each variable assignment in each constraint. In many applications, it is more natural for users to express preferences via generic qualitative (usually partial) preference relations over variable assignments. For example, it is often more intuitive for the user to state “I prefer red wine to white wine”, rather than “Red wine has preference 0.7 and white wine has preference 0.4” (with the assumption that a higher preference value expresses higher desirability). Although the former statement provides us with less information, it does not require *careful* selection of preference values for (possibly partial) variable assignments as required in soft constraints.

CP-nets (Boutilier et al., 1999, 2004) (that is, Conditional Preference nets) are a graphical model for representation and reasoning about certain sets of qualitative preference statements, interpreted under the *ceteris paribus (cp)* assumption. For instance, under the ceteris paribus interpretation, the statement *“I prefer red wine to white wine if meat is served”* asserts that, given two meals that differ *only* in the kind of wine served *and* both containing meat, the meal with a red wine is preferred to the meal with a white wine. Observe that this interpretation corresponds to a “least committing” interpretation of the information provided by the user, and many philosophers (see (Hansson, 2001) for an overview) and AI researchers (Doyle and Wellman, 1994) have argued on behalf of adopting this interpretation scheme. To emphasize the ceteris paribus interpretation, such statements are usually called cp-statements.

Informally, each CP-net compactly captures the preference relation induced by a set of such (possibly conditional) cp-statements. Structurally, CP-nets bear some similarity to Bayesian networks, as both utilize directed graphs where each node stands for a domain variable, and assume a set of features $\mathbf{F} = \{X_1, \ldots, X_n\}$ with finite, discrete domains $\mathcal{D}(X_1), \ldots, \mathcal{D}(X_n)$ (these play the same role as variables in soft constraints). Yet another similarity between CP-nets and Bayesian networks is that graphical structure in both models relies on a certain notion of independence between the variables: Bayesian networks utilize the notion of probabilistic independence, while CP-nets utilize the notion of preferential independence.

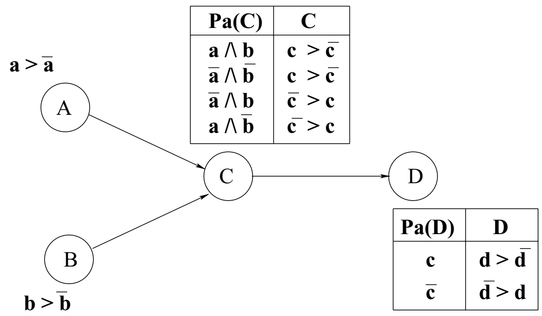


Fig. 1 A CP-net

During preference elicitation, for each feature X_i the user is asked to specify a set of *parent* features $Pa(X_i)$, the values of which affect her preferences over the values of X_i. This information is used to create the directed *dependency graph* of the CP-net in which each node X_i has $Pa(X_i)$ as its immediate predecessors. Given this structural information, the user is asked to explicitly specify her preference over the values of X_i for *each complete assignment* on $Pa(X_i)$, and this preference is assumed to take the form of a total (Boutilier et al., 1999) or partial (Boutilier et al., 2004) order over $\mathcal{D}(X)$. These conditional preferences over the values of X_i are captured by a *conditional preference table* $CPT(X_i)$ which is annotated with the node X_i in the CP-net. To illustrate the CP-nets model, consider a CP-net in Figure 1. Here, statement $a \succ \overline{a}$ represents the unconditional preference of the user for $A = a$ over $A = \overline{a}$, while statement $c : d \succ \overline{d}$ represents that the user prefers $D = d$ to $D = \overline{d}$, given that $C = c$.

The semantics of CP-nets depends on the notion of a *worsening flip*. A worsening flip is a change in the value of a single feature to a value which is less preferred according to a cp-statement for that feature. For example, in the CP-net in Figure 1, "moving" from $abcd$ to $ab\overline{c}d$ is a legitimate worsening flip since, according to $CPT(C)$, c is preferred to $\overline{c}$ given a and b. We say that an outcome, that is, a complete assignment of the features in their domains, α is better than (or preferred to) an outcome β, written $\alpha \succ \beta$, if and only if there is a chain of worsening flips from α to β. This definition induces a strict preorder over the outcomes, which defines the so-called *induced graph*, where nodes represent outcomes and directed arcs represent worsening flips.[1] An outcome is optimal if it is undominated in this preorder.

From the point of view of reasoning about preferences, several types of query can be asked about CP-nets. First, given a CP-net N one might be interested in finding an optimal assignment to the features of N. For acyclic CP-nets, such a query is answerable in linear time (Boutilier et al., 1999), while the complexity of this query for cyclic CP-nets has been left as an open problem. Second, given a CP-net N and a pair of complete assignments α and β, one might be interested in determining whether N implies $\alpha \succ \beta$, i.e. α is preferred to β. Though some tractable special cases of this query do exist (Boutilier et al., 2004), the general problem is known to be NP-hard even for acyclic CP-nets (Domshlak and Brafman, 2002), and PSPACE-complete for cyclic CP-nets (Goldsmith et al., 2005).

[1] For the formally precise semantics of CP-nets we refer the reader to (Boutilier et al., 2004).

3. Eligibility of cp-statements

Given a set of preference cp-statements Ω extracted from a user, we might be interested in testing the consistency of the preference relation induced by these statements. In this section we provide some complexity results for this reasoning task, and show how it can be accomplished by solving a set of hard constraints.

In general, there is no single notion of preferential consistency (Hansson, 2001). In (Boutilier et al., 2004), a CP-net N is considered to be consistent if and only if the preorder $\succ$ induced by N is *asymmetric*, that is, there exists at least one total ordering of the outcomes consistent with $\succ$. In many situations, however, one can ignore cycles in the preference relation, as long as these do not prevent the user making a rational choice, that is, there exist an outcome that is not dominated by any other outcome with respect to $\succ$ (Hansson, 2001). Here we consider the second approach and say that a CP-net is *eligible* if it induces at least one undominated outcome. Likewise, in what follows we will use the same notions of asymmetry and eligibility for sets of cp-statements that are not representable by CP-nets. For instance, such a situation occurs when there exists a feature X, such that the assignments to its parents in the preference statements over the values of X are not mutually exclusive.

When a set of cp-statements Ω defines an acyclic CP-net, the preorder induced by Ω is guaranteed to be asymmetric (Boutilier et al., 2004). For cyclic CP-nets, however, asymmetry is no longer guaranteed. In the more general case, we are given a set Ω of conditional preference statements without any guarantee that they define a CP-net. Formally, we will consider cp-statements of the form $X_1 = v_1 \wedge \ldots \wedge X_k = v_k : Y = w_1 \succ \ldots \succ Y = v_l$, which will sometimes be written without the names of the variables if they are not significant.

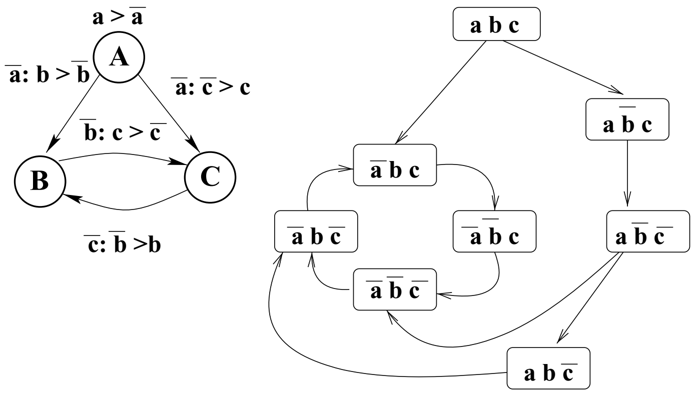
Let the *dependency graph* of such a set Ω of cp-statements be defined similarly to the graph of a CP-net: the nodes stand for problem features, and a directed arc goes from X_i to X_j iff Ω contains a statement expressing preference on the values of X_j conditioned on the value of X_i.

For example, the set $\Omega = \{a : b \succ \overline{b}, a \wedge c : \overline{b} \succ b\}$ does not induce a CP-net (the two conditionals are not mutually exclusive), and the preference relation induced by Ω is not asymmetric, despite the fact that the dependency graph of Ω is acyclic. Another example of a set of cp-statements which does not represent a CP-net, but is eligible, can be seen in Figure 2. As can be seen in the figure, the induced ordering is not asymmetric since there is a cycle in the induced preference graph. Nevertheless, this order contains an undominated outcome abc.

Given such a set of cp-statements, and assuming that the *true* preferences of the user are asymmetric, one can try to continue questioning the user, attempting to eliminate the detected cyclic preferences. However, even ignoring the fact that detecting asymmetry can be hard in general, long interactions with the user (for example in online configuration tasks) should be avoided as far as possible. In fact, given such an eligible set of statements, it is often sufficient to prompt the user with one of the undominated assignments without further refining the available preference information. Hence, testing preferences for eligibility and identifying undominated outcomes is a practically important reasoning task.

3.1. Testing eligibility is NP-complete

We begin with showing in Theorem 1 that determining eligibility of a set of cp-statements is in general NP-complete. In contrast, note that determining asymmetry of the preference relation even for CP-nets has been recently shown to be PSPACE-complete (Goldsmith et al., 2005).



(a) Dependence Graph **(b) Induced Graph**

Fig. 2 The dependency graph of a set of cp statements and the induced graph

Theorem 1. ELIGIBILITY *of a set of conditional preference statements* Ω *is* NP-*complete.*

Proof: Membership in NP is straightforward, as an assignment is a polynomial-size witness that can be checked for non-dominance in time linear in the size of Ω. To show hardness, we reduce 3-SAT to our problem: given a 3-cnf formula F, for each clause $(x \vee y \vee z) \in F$ we construct the conditional preference statement: $\overline{x} \wedge \overline{y} : z \succ \overline{z}$. This set of conditional preferences is eligible iff the original original formula F is satisfiable. In fact, in any satisfying assignment of the original 3-cnf formula, at least one of x, y, and z is true. If x or y are true, then the condition of the conditional preference statement is not satisfied and thus the statement is true. If x and y are both false, then z is true, which satisfies the conditional preference statement as this is the most preferred value. Hence, any model of the original 3-cnf formula is an optimal assignment of the set of statements. The argument reverses: any optimal assignment is also a model. $\square$

3.2. Hard constraints to test for eligibility

Next we show that the problem of testing a set of cp-statements for eligibility can be translated into a hard constraint satisfaction problem. In fact, the latter formulation of the problem using hard constraints also allows us to find the actual undominated outcomes.

Definition 1 (Optimality constraints). Given a set of cp-statements Ω, the optimality constraint corresponding to the statement

$$\langle \varphi : (X = x_1) \succ (X = x_2) \succ \ldots \succ (X = x_i) \rangle \in \Omega$$

is

$$\varphi \rightarrow (X = x_1).$$

In general, suppose that φ induces a strict partial preference ordering over $\mathcal{D}(X)$ with $\mathcal{D}_\varphi(X) \subseteq \mathcal{D}(X)$ being the set of all most preferred (i.e., undominated) values of X given φ. Then the optimality constraint corresponding to this statement is:

$$\varphi \rightarrow \bigvee_{x_j \in \mathcal{D}_\varphi(X)} (X = x_j)$$

The optimality constraints opt (Ω) corresponding to the entire set Ω is the union of the optimality constraints corresponding to all the cp-statements in Ω.

For example, the cp-statements $a \succ \overline{a}$ and $(a \wedge b) : c \succ \overline{c}$ are translated to the optimality constraints a and $(a \wedge b) \rightarrow c$, respectively. Since in this example the features are Boolean, for each cp-statement there is just one corresponding optimality constraint.

Theorem 2. *An outcome is optimal in the ordering induced by a set of cp-statements Ω iff it is a satisfying assignment for $opt(\Omega)$.*

Proof: Assume first that there is at least one optimal outcome α with respect to Ω, and suppose that this outcome does not satisfy *opt*(Ω). If so, then there exists at least one optimality constraint

$$\varphi \rightarrow \bigvee_{x_j \in \mathcal{D}_\varphi(X)} X = x_j$$

that is not satisfied by α. An implication is unsatisfied only when the hypothesis is true and the conclusion is false. That is, φ holds, yet for all $x_{j \in \mathcal{D}_\varphi(X)}$, $X = x_j$ does not hold. Hence X must be assigned by α to a value from $\mathcal{D}(X) \setminus \mathcal{D}_\varphi(X)$, that is, to one of the dominated values of X given φ. Flipping the value of X in α to any $x_j \in \mathcal{D}_\varphi(X)$ will then be an improving flip for this conditional preference, contradicting the fact that α is optimal. Hence all the optimality constraints in *opt*(Ω) must be satisfied by all the outcomes being optimal with respect to Ω. The proof in the other direction is similar as it is based on exactly the same arguments. □

Corollary 1. *A CP-net N is eligible iff opt*(Ω) *is consistent.*

Proof: Follows directly from Theorem 2. □

In particular, Theorem 2 provides us with the first general method and complexity bound for finding optimal outcomes with respect to cyclic CP-nets. We note that a similar technique restricted to Boolean features has been independently proposed in (Brafman and Dimopoulos, 2004), and it can be seen as a special case of our proposal for general, multi-valued CP-nets.

Notice that the optimality constraints cannot simply be added to other hard constraints, since this would preserve only feasible solutions that are undominated in the CP-net. However, if there are no solutions with these features, then then there could still be solutions which are feasible and undominated by other feasible solutions in CP-net, which would not be found. Thus the optimality constraints can also be used in unconstrained CP-nets. On the contrary, the approximation technique we will describe in Section 5 can handle also such situations. Exact techniques to deal with both CP-nets and hard constraints can be found in (Prestwich et al., 2005; Boutilier et al., 2004).

3.3. Tractable cases

While testing eligibility is hard in general, Theorem 3 presents a wide class of statement sets that can be tested for eligibility in polynomial time.

Theorem 3. *A set of conditional preference statements Ω can be tested for eligibility in polynomial time if the dependency graph of Ω is acyclic and its node in-degree bounded by a constant.*

Proof: The proof is constructive, and the algorithm is as follows: First, for each feature $X \in \mathbf{F}$, we construct a table T_X with an entry for each assignment $\pi \in \mathcal{D}(Pa(X))$, where each entry $T_X[\pi]$ contains all the values of X that are not dominated given Ω and π. Subsequently, we remove all the empty entries. For example, let A, B and C be a set of Boolean problem features, and let $\Omega = \{c \succ \overline{c}, a : b \succ \overline{b}, a \wedge c : \overline{b} \succ b\}$. The corresponding table will be as follows:

Feature	π	Values
T_A	$\emptyset$	$\{a, \overline{a}\}$
T_C	$\emptyset$	$\{c\}$
T_B	$a \wedge \overline{c}$	$\{b\}$
	$\overline{a} \wedge \overline{c}$	$\{b, \overline{b}\}$
	$\overline{a} \wedge c$	$\{b, \overline{b}\}$

Observe that the entry $T_B[a \wedge c]$ has been removed, since, given $a \wedge c$, b and $\overline{b}$ are dominated according to the statements $a \wedge c : \overline{b} > b$ and $a : b > \overline{b}$, respectively. Since the in-degree of each node X in the dependency graph of Ω is bounded by a constant k (i.e. $|Pa(X)| \leq k$), these tables take space and can be constructed in time $O(n2^k)$. Given such tables for all the features in $\mathbf{F}$, we traverse the dependency graph of Ω in a topological order of its nodes, and for each node X being processed we remove all the entries in T_X that are not "supported" by (already processed) $Pa(X)$: an entry $T_X[\pi]$ is not supported by $Pa(X)$ if there exists a feature $Y \in Pa(X)$ such that the value provided by π to Y appears in no entry of T_Y. For instance, in our example, the rows corresponding to $a \wedge \overline{c}$ and $\overline{a} \wedge \overline{c}$ will be removed, since $\overline{c}$ does not appear in the (already processed) table of C. Now, if the processing of a feature X results in $T_X = \emptyset$, then Ω is not satisfiable. Otherwise, any assignment to $\mathbf{F}$ consistent with the processed tables will be undominated with respect to Ω. □

Note that, for sets of preference statements with cyclic dependency graphs, ELIGIBILITY remains hard even if the in-degree of each node is bounded by $k \geq 6$, since 3-SAT remains hard even if each variable participates in at most three clauses of the formula in the proof of Theorem 1 (Garey and Johnson, 1978). However, when at most one condition is allowed in each preference statement, and the features are Boolean, then ELIGIBILITY can be reduced to 2-SAT, and thus tested in polynomial time.

4. Comparing soft constraints and CP-nets

The last section showed that hard constraints are sufficient if we are interested only in the undominated outcomes of a set of cp statements. We will now consider the case where we are also interested in the induced ordering. We will show that in this respect CP-nets and constraints (hard or soft) are incomparable formalisms. More precisely, we will compare the expressive power of soft constraints and CP-nets with respect to the induced ordering. Consider the following definition of equivalence.

Definition 2 (equivalence). Consider a CP-net N and soft constraint problem P defined on the same set of variables V. P and N are *equivalent* if and only if they induce they same ordering on the set of assignments of the variables in V.

For consistent CP-nets, it is always possible to find an equivalent SCSP, since consistent CP-nets induce a partial order and SCSPs can model any partial order over the solutions.

Theorem 4. *Consider a consistent CP-net. Then there exists an equivalent SCSP.*

Proof: If the CP-net is consistent, it induces a partial order O over the outcomes. Then we consider a lattice containing a partial order O' of preferences which is isomorphic to O. We then build an SCSP with one variable and as many values in its domain as the number of outcomes of the CP-net, and we add a unary constraint over this variable, which associates to each value in the domain the corresponding preference in the partial order O'. □

However, the converse is not true, since CP-nets cannot induce an arbitrary partial order.

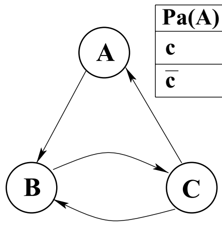
We will now show that there are CP-nets from which it is not possible to build an equivalent SCSP, and vice-versa. In other words, the two formalisms are incomparable with respect to the above notion of equivalence.

Theorem 5. *There are CP-nets for which it is not possible to build an equivalent SCSP. Conversely, there are SCSPs for which it is not possible to build an equivalent CP-net.*

Proof: To see that there are CP-nets for which it is not possible to build an equivalent SCSP, it is enough to consider any CP-net whose induced ordering is a preorder but not a partial order. The CP-net of Figure 3, whose induced ordering can be seen in Figure 4, is one example.

To see that there are SCSPs for which it is not possible to build an equivalent CP-net, it is sufficient to consider a fuzzy SCSP with three binary-valued variables A, B and C, and the following soft constraints:

Fig. 3 A CP-net



Pa(A)	A
c	$a > \bar{a}$
$\bar{c}$	$\bar{a} > a$

Pa(C)	C
b	$c > \bar{c}$
$\bar{b}$	$\bar{c} > c$

Pa(B)	B
$a \wedge c$	$b > \bar{b}$
$\bar{a} \wedge \bar{c}$	$b > \bar{b}$
$\bar{a} \wedge c$	$\bar{b} > b$
$a \wedge \bar{c}$	$\bar{b} > b$

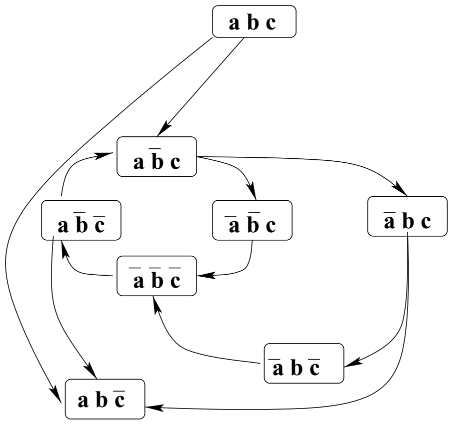


Fig. 4 The induced ordering of the CP-net in Figure 3

- a constraint connecting all three variables, and stating that $\overline{a}b\overline{c}$ has preference 0.9, $a\overline{b}c$ has preference 0.8, $ab\overline{c}$ has preference 0.7, and all other assignments have preference 1;
- a constraint over variables A and B such that ab has preference 0.9 and all other assignments have preference 1.

The induced ordering is total and has $a\overline{b}c$ with preference 0.8, $ab\overline{c}$ with preference 0.7, and all other assignments are above in the ordering (they have preference 0.9 or 1). Let us now assume that there is an equivalent CP-net, that is a CP-net with this ordering as induced ordering. Then there must be a worsening path from $a\overline{b}c$ to $ab\overline{c}$. However, since the two assignments differ for more than one flip, this is possible only if there is at least another assignment which is strictly worse than $a\overline{b}c$ and strictly better than $ab\overline{c}$. This is not true given the ordering, so there cannot be such a CP-net. □

One could say that preorders and partial orders do not really encode different information, because any preorder can be directly mapped onto a partial order simply by transforming each cycle into tied elements of the partial order. Thus we could consider a more tolerant notion of equivalence, where a CP-net inducing a preorder O and an SCSP inducing a partial order O' are considered equivalent if O' can be obtained from O as stated above. Let us call this notion *pre-equivalence*.

It may appear that if we allow this notion of equivalence then the SCSP framework is more expressive than CP-nets. In fact, SCSPs can induce any partial order, while (as noted in the proof of the above theorem) CP-nets do not. But this is true only if we allow for exponential-time mappings from CP-nets to SCSPs. If we restrict ourselves to polynomial time mappings then the following theorem holds (Meseguer et al., 2004).

Theorem 6. *Assuming $P \neq NP$, there are CP-nets for which it is not possible to build in polynomial time a pre-equivalent SCSP, and there are SCSPs for which it is not possible to build an pre-equivalent CP-net.*

Proof: Assume that, for all CP-nets, it is possible to build in polynomial time a pre-equivalent SCSP. Then, dominance testing in the given CP-net would become a polynomial problem since it would be sufficient to map in polynomial time the CP-net into the SCSP and then to perform the dominance test in the SCSP, which can be done in polynomial time. Since dominance testing in CP-nets is known to be NP-hard (Boutilier et al., 2004), this would contradict the hypothesis that $P \neq NP$. The other direction of the theorem follows directly from Theorem 5. In fact, in this direction we never start from a preorder but always from a partial or total order. □

We have just proved that CP-nets and soft constraints are incomparable with respect to the induced ordering. This means that, if we want the soft constraint machinery to reason about CP-net preferences efficiently, in general we must approximate the ordering of the given CP-net. In the next section we will propose two such approximations.

5. Approximating acyclic CP-nets with soft constraints

In addition to testing consistency and determining preferentially optimal outcomes, we may be interested in the *preferential comparison* of two outcomes. Comparison is essential in preference-based optimization when faced with hard constraints on the variables, and for sorting a predefined set of outcomes (e.g., the content of a database relation). Unfortunately, determining dominance between a pair of outcomes with respect to a set of qualitative preferential statements under the *ceteris paribus* assumption is PSPACE-complete in general (Lang, 2002), and is NP-hard even for acyclic CP-nets (Boutilier et al., 2004).

However, given a set Ω of preference statements, instead of using a preference relation $\succ$ induced by Ω one can use an approximation $\gg$ of $\succ$, achieving tractability of comparison while sacrificing precision to some degree. Clearly, different approximations $\gg$ of $\succ$ are not equally good, as they can be characterized by their precision with respect to $\succ$, the time complexity of generating $\gg$, and the time complexity of comparing outcomes with respect to $\gg$. In addition, it is vital that $\gg$ faithfully extends $\succ$ (i.e. $\alpha \succ \beta$ should entail $\alpha \gg \beta$). We call this property *order preserving*.

Definition 3 (order preserving). Consider any set S, and an ordering $\succ$ defined on the elements of S. An approximation of $\succ$, $\gg$, is *order preserving* if and only if $\forall \alpha, \beta \in S\ \alpha \succ \beta \Rightarrow \alpha \gg \beta$.

Another desirable property of approximations is that of preserving the *ceteris paribus* property.

Definition 4 (cp-condition). Consider a set of ceteris paribus statements defined on a set of attributes $\{X_1, \ldots, X_n\}$ and let Ω be the corresponding space of outcomes, with induced ordering $\succ$. An approximation of $\succ$, $\gg$, is *ceteris paribus preserving* if for each variable $X_i \in N$, each assignment $\mathbf{u}$ to $Pa(X)$, and each pair of values $x_1, x_2 \in D(X)$, if the CP-net specifies that $\mathbf{u} : x_1 \succ x_2$, then we have $x_1\mathbf{uy} \gg x_2\mathbf{uy}$, for all assignments $\mathbf{y}$ of $\mathbf{Y} = \mathbf{V} - \{\{X\} \cup Pa(X)\}$.

Here we study approximating CP-nets via soft constraints (SCSPs). This allows us to use the rich machinery underlying SCSPs to answer comparison queries in linear time. Moreover, it provides us with a uniform framework for combining user preferences with both hard and soft constraints. Given an acyclic CP-net, we construct a corresponding SCSP in two steps. First we build a constraint graph, which we call the *SC-net*. Second, we compute the preferences and weights for the constraints in the SC-net. This computation depends on the actual semi-ring framework being used. Here we present and discuss two alternative semi-ring

Pseudocode for CPnetToSCSP

1\. **input** CP-net N;
2\. Initialize SCSP N_c as follows:
3\. **for** every feature X of N add variable V_X to N_c;
4\. **for** every variable $V_X \in N_c$, $D(V_X) \leftarrow D(X)$;
5\. **for** every feature X of N such that $|Pa(X)| \geq 2$, add variable $V_{Pa(X)}$ to N_c;
6\. **for** every variable $V_{Pa(X)} \in N_c$, $D(V_{Pa(X)}) \leftarrow \prod_{Y \in Pa(X)} D(Y)$;
7\. **for** every variable $V_X \in N_c$:
8\.   **if** ($|Pa(X)| = 0$) add a soft unary constraint on V_X;
9\.   **if** ($|Pa(X)| = 1$) add a soft binary constraint on $V_{P(X)}$ and V_X in N_c;
10\.   **if** ($|Pa(X)| \geq 2$):
11\.   **for** each $Y \in Pa(X)$ add a hard binary constraint on V_Y and $V_{Pa(X)}$;
12\.   add a soft binary constraint on $V_{Pa(X)}$ and V_X;
13\. **output** SCSP N_c;

Fig. 5 Algorithm CPnetToSCSP given a CP-net as input produces a generic SCSP as output

frameworks, based on *min*+ and *SLO* (Soft constraint Lexicographic Ordering) semi-rings. In both cases, our computation of preferences and weights ensures order preserving and satisfies the cp-condition.

Figure 5 shows the pseudocode of algorithm CPnetToSCSP. Given a CP-net N (line 1), the corresponding SC-net N_c has two types of node. First, each feature $X \in N$ is represented in N_c by a node V_X that stands for a SCSP variable with $\mathcal{D}(V_X) = \mathcal{D}(X)$ (lines 3-4). Second, for each feature $X \in N$, such that $|Pa(X)| \geq 2$, we have a node $V_{Pa(X)} \in N_c$, with $\mathcal{D}(V_{Pa(X)}) = \Pi_{Y \in Pa(X)} \mathcal{D}(Y)$ (lines 5-6) (notice that Π stands for Cartesian product). Edges in N_c correspond to hard and soft constraints, where the latter are annotated with weights. Notice that in the SCSP framework weights on constraints are not allowed. The weights we impose here are merely a working tool for ensuring that the preferences on soft constraints will satisfy order preserving and the cp-condition (see Theorem 7). Once the semiring is chosen, the weights will be combined (not necessarily using the semiring multiplicative operator) with initial preferences, allowing us to obtain the final preferences in the SCSP (see the following paragraph). Each node V_X corresponding to an "independent feature" $X \in N$ has an incoming (source-less) soft constraint edge (line 8). For each node V_X corresponding to a "single-parent" feature $X \in N$ with $Pa(X) = \{Y\}$, we have a soft constraint edge between X and Y (line 9). Finally, for each node V_X such that $|Pa(X)| \geq 2$, we have (i) hard constraint edges between $V_{Pa(X)}$ and each $Y \in Pa(X)$ to ensure consistency, and (ii) a soft constraint edge between $V_{Pa(X)}$ and V_X (lines 10-12). The output is a generic (i.e. not instantiated to any semiring) SCSP N_c.

To assign preferences to variable assignments in each soft constraint, each soft constraint c (between $V_{Pa(X)}$ and V_X) is associated with two items: w_c, a real number which can be interpreted as a weight (defined in the next section), and $P_c = \{p_1, \ldots, p_{|\mathcal{D}(V_X)|}\}$, a set of reals which can be interpreted as "quantitative levels of preference". We will see in the next section how to generate the preference for each assignment to the variables of c, depending on the chosen semiring. In any case, each preference will be obtained by combining (via multiplication over reals) the weight of the constraint w_c and one of the elements of P_c. If $P_c = \{p_1, \ldots, p_n\}$, then such preferences will be denoted by $p'_1, \ldots, p'_n$.

As an example let us consider the SC-net N_c shown in Figure 6, obtained by applying CPnetToSCSP to the CP-net in Figure 1. As may be seen, there is a variable for every feature of N: V_A, V_B, V_C, and V_D. Since the features in N are binary-valued, so are the corresponding

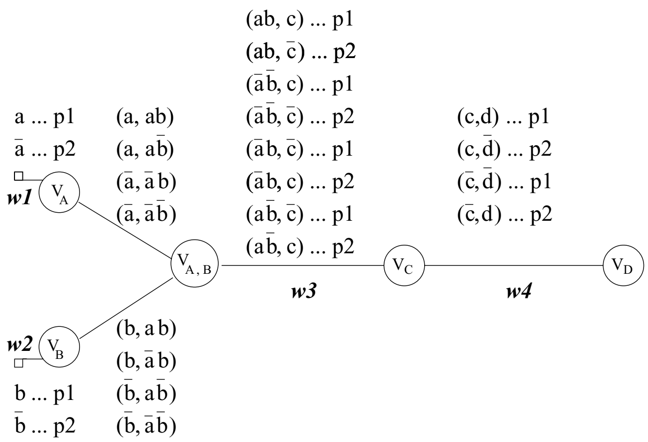


Fig. 6 The SCSP corresponding to the SC-net of Figure 1

variables in N_c. In addition there is variable $V_{A,B}$ for the parents of C, which is the only feature in N with more than one parent, and its domain is $D(A) \times D(B)$. Features A and B in N are independent (i.e. $Pa(A) = Pa(B) = \emptyset$. This means that the preference on their values do not depend on any assignment to other variables. Thus, a unary soft constraint is defined over their corresponding variables in N_c, assigning to each value of their domain a preference. If instead we consider feature C, the preferences on its domain depend on the assignments of its parents A and B. Thus in N_c there is a soft constraint between the variable representing the parents, $V_{A,B}$, and V_C, that assigns to each possible triple of values $v_A v_B v_C$ its preference. However, additional hard constraints are needed between variables V_A and V_B and $V_{A,B}$. Their role is to ensure that if V_A is assigned a value, e.g. a, and V_B is assigned another value, e.g. $\bar{b}$, then $V_{A,B}$ can only be assigned pair $(a, \bar{b})$. Finally there is a soft constraint between variable V_D and its only parent V_C assigning preferences to tuples $v_C v_D$. Each soft constraint has an additional weight attached to it, the meaning of which will become clear below.

5.1. Weighted soft constraints

The first approximation we propose applies in scenarios where soft constraints are used to model cost (or penalty) minimization. We will generate a weighted SCSP, based on the *min*+ semi-ring $S_{WCSP} = \langle R_+, min, +, +\infty, 0\rangle$. We assign preferences using real positive numbers (or penalties) and prefer assignments with smaller total penalty (i.e. the sum of all local penalties). In a soft constraint c on $V_{Pa(X)}$ and V_X there are $|\mathcal{D}(V_X)|$ penalties. Without loss of generality we assume that they range between 0 and $|\mathcal{D}(V_X)| - 1$, that is $p_1 = 0, \ldots, p_{\mathcal{D}(V_X)|} = |\mathcal{D}(V_X)| - 1$. In our example, since all variables are binary there are only two penalties, i.e. $p1 = 0$ and $p2 = 1$, in all the constraints.

We ensure that the cp-condition is satisfied in a similar way to that proposed in (Boutilier et al., 2001) in the context of UCP-nets. For now we just say that UCP-nets are also a quantitative approximation of CP-nets in a max + scenario, that is for maximizing the sum of utilities. To ensure the cp-condition, we will impose that each variable dominates its children. We rewrite this property, defined in (Boutilier et al., 2001), in our context.

Definition 5. Consider a CP-net N and the corresponding SCSP N_c, obtained applying CPnetToSCSP to N. Consider variable V_X and $V_{Pa(X)}$in N_c and let $\mathbf{Y}$ be the variables in $\mathbf{V} - \{X \cup Pa(X)\}$ in N. Denote the children of X by $\mathcal{B} = \{V_{B_1}, \cdots, V_{B_h}\}$. V_X dominates its children if and only if for any two values $x_1, x_2 \in D(X)$ such that $x_1\mathbf{u} \succ x_2\mathbf{u}$, then for any assignment $\mathbf{y}$ to $\mathcal{B}$,

$$p'\big((x_1\mathbf{uy})_{|_c}\big) - p'\big((x_2\mathbf{uy})_{|_c}\big) < \sum_{t_i \in T} p'\big((x_1\mathbf{uy})_{|_{t_i}}\big) - \sum_{t_i \in T} p'\big((x_2\mathbf{uy})_{|_{t_i}}\big)$$

where S is the set of soft constraints of N_c and notation $(x_1\mathbf{uy})_{|_s}$ stands for the projection on the outcome on constraint s, constraint c is that on $V_{Pa(X)}$ and V_X, and constraints $t_i \in T$ are on $V_{Pa(B_i)}$ and V_{B_i} such that $X \in Pa(B_i)$.

We will show that, as in (Boutilier et al., 2001), this property is sufficient for the cp-condition (Theorem 7). First we describe an algorithm which sets the weights on N_C in such a way that every variable will dominate its children. This is achieved by setting the minimum penalty on a variable to be greater than the sum of the maximum penalties of the children. In Figure 7 we show the pseudocode for algorithm Min+weights that computes such weights. In this code, $w(V_X)$ represents the weight of the soft constraint c between $V_{Pa(X)}$ and V_X. As shown by Figure 7 the algorithm considers the features of N in reverse topological order. Hence a successor of X in Min+weights is a parent of X in N. When it considers a feature X it sets the weight of the soft constraint (which by construction is single) defined on V_X and $V_{Pa(X)}$ in N_c. The value assigned is found by multiplying the weight of the corresponding constraints for all the children of X in N (which is known due to the reverse topological order) by the corresponding size of the domains, and summing the quantities obtained.

Considering the example in Figure 6, let $\{D, C, B, A\}$ be the reverse topological ordering obtained in line 2. Then the first soft constraint to be processed is the one between V_C and V_D. Since D has no children in N, in line 5 we assign $w(V_D)$ to 1. Next, we process the soft constraint between $V_{A,B}$ and V_C: V_D is the only child of V_C, hence $w(V_C) = w(V_D) \times \mathcal{D}(V_D) = 1 \times 2 = 2$. Subsequently, since V_C is the only child of both V_A and V_B, we assign $w(V_A) = w(V_B) = w(V_C) \times |\mathcal{D}(V_C)| = 2 \times 2 = 4$. Now, consider two outcomes $o_1 = abcd$ and $o_2 = a\bar{b}cd$. The total penalty of o_1 is $(w(V_A) \times p_1) + (w(V_B) \times p_1) + (w(V_C) \times p_1) + (w(V_D) \times p_1) = 0$, since $p_1 = 0$, while the total penalty of o_2 is $(w(V_A) \times p_1) + (w(V_B) \times p_2) + (w(V_C) \times p_2) + (w(V_D) \times p_1) = (4 \times 1) + (2 \times 1) = 6$ since $p_2 = 1$. Therefore, we can conclude that o_1 is better than o_2 since $min(0, 6) = 0$. Figure 8 shows the result of applying Min+weights and the final SCSP defined of semiring S_{WCSP}.

Fig. 7 Algorithm Min+weights computes weights on constraints when the semiring is S_{WCSP}

Pseudocode of Min+weights

```
1.Input CP-net N and SCSP N_c=CPnetToSCSP(N)
2. Order variables of N in a reverse topological ordering
3. for each X ∈ N do
4.    if X has no successors in N then
5.       w(V_X) ← 1
6.    else
7.       w(V_X) ← Σ_{Y s.t. X∈Pa(Y)} w(V_Y) · |D(V_Y)|
8. return N_c
```

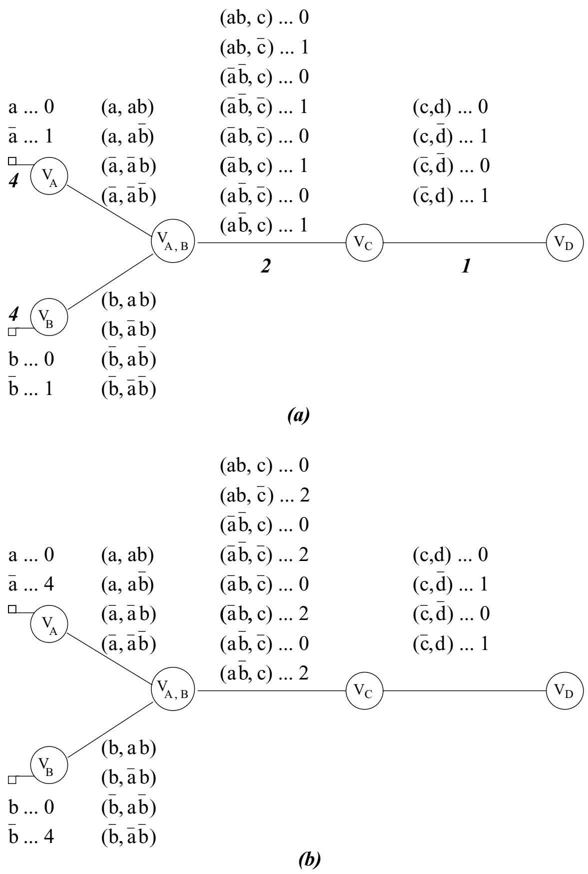


Fig. 8 The SC-net obtained by applying Min+weights to the CP-net in Figure 1 (part a) and the corresponding weighted SCSP (part b)

We now prove that our algorithm for weight computation ensures the cp-condition on the resulting set of soft constraints, and this also implies preserving the ordering information with respect to the original CP-net.

Theorem 7. *The SC-net based weighted SCSP N_c, generated from an acyclic CP-net N, is an approximation of N which satisfies the cp-condition and is order preserving, i.e. for each pair of outcomes α, β we have $\alpha \succ \beta \;\Rightarrow\; \alpha >_{min+} \beta$.*

Proof: Due to the CP-net semantics it is enough to show that, for each variable $X \in N$, each assignment $\mathbf{u}$ on $Pa(X)$, and each pair of values $x_1, x_2 \in \mathcal{D}(X)$, if CP-net specifies that $u : x_1 \succ x_2$, then we have $x_1\mathbf{uy} >_{min+} x_2\mathbf{uy}$, for all assignments $\mathbf{y}$ on $\mathbf{Y} = \mathbf{V} - \{\{X\} \cup Pa(X)\}$. By definition, $x_1\mathbf{uy} >_{min+} x_2\mathbf{uy}$ if and only if

$$\sum_{s \in S} p'\big((x_1uy)_{|_s}\big) < \sum_{s \in S} p'\big((x_2uy)_{|_s}\big),$$

where S is the set of soft constraints of N_c and notation $(x_1uy)_{|_s}$ stands for the projection on the outcome on constraint s. The constraints on which x_1uy differs from x_2uy are: constraint

c on $V_{Pa(X)}$ and V_X, and all the constraints $t_i \in T$ on $V_{Pa(B_i)}$ and V_{B_i} such that $X \in Pa(B_i)$ (in what follows, we denote the children of X by $\mathcal{B} = \{V_{B_1}, \cdots, V_{B_h}\}$). Thus, we can rewrite the above inequality as

$$p'\big((x_1uy)_{|_c}\big) + \sum_{t_i \in T} p'\big((x_1uy)_{|_{t_i}}\big) < p'\big((x_2uy)_{|_c}\big) + \sum_{t_i \in T} p'\big((x_2uy)_{|_{t_i}}\big)$$

By construction of N_c we have

$$p'\big(\pi_c(x_1uy)_{|_c}\big) = w_c \times p(x_1u) < p'\big((x_2uy)_{|_c}\big) = w_c \times p(x_2u)$$

and thus $x_1\mathbf{uy} >_{min+} x_2\mathbf{uy}$ if and only if

$$w_c p(x_2u) - w_c p(x_1u) > \sum_{t_i \in T} p'\big((x_1uy)_{|_{t_i}}\big) - \sum_{t_i \in T} p'\big((x_2uy)_{|_{t_i}}\big)$$

In particular, this will hold if

$$w_c(min_{x,x' \in \mathcal{D}(X)}|p(xu) - p(x'u)|) > \sum_{t_i \in T} w_{t_i}(max_{x,x',z,b}|p(x'zb) - p(xzb)|)$$

where z is the assignment to all parents of $\mathcal{B}$ other than X. Observe that the maximum value of the right term is obtained when $p(x'zb) = |\mathcal{D}(\mathcal{B})| - 1$ and $p(xzb) = 0$. On the other hand, $min_{x,x' \in \mathcal{D}(X)}|p(x'u) - p(xu)| = 1$. In other words $w_c > \sum_{t_i \in T} w_{t_i}(|\mathcal{D}(B_i)| - 1)$ must hold. But this is ensured by the algorithm, setting (in line 7) $w_c = \sum_{t_i \in T} w_{t_i}(|\mathcal{D}(B_i)|$. □

We will now prove that the complexity of the mapping we propose is polynomial in the size of the CP-nets which is the number of its features.

Theorem 8 (complexity). *Given an acyclic CP-net N with the node in-degree bounded by a constant, the construction of the corresponding SC-net based weighted SCSP N_c is polynomial in the size of N.*

Proof: If the CP-net has n nodes then the number of vertices V of the derived SC-net is at most $2n$. In fact, in the SC-net a node representing a feature appears at most once and there is at most one node representing its parents. If the number of edges of the CP-net is e, then the number of edges E in the SC-net (including hard and soft edges) is at most $e + n$, since each edge in the CP-net corresponds to at most one constraint, and each feature in the CP-net generates at most one new soft constraints. A topological sort can be performed in $O(V + E)$, that is, $O(2n + e + n) = O(e + n)$. Then for each node, that is $O(V)$ times, at most V children must be checked to compute the new weight value, leading to a number of checks which is $O(V^2) = O(n^2)$. Each check involves checking a number of assignments which is exponential in the number of parents of a node. Since we assume that the number of parents of a node is limited by a constant, this exponential is still a constant. Thus the total time complexity is $O(V^2)$ (or $O(n^2)$ if we consider the size of the CP-net). □

Let us compare in more detail the original preference relation induced by the CP-net and that induced by its min+ semi-ring based SC-net. The comparison is summarized in the following table, where $\sim$ denotes incomparability.

CP-nets $\Rightarrow$ min+	
$\prec$	$<$
$\succ$	$>$
$\sim$	$<, >, =$

By Theorem 7 we know that the pairs that are ordered in some way by $\succ$ remain ordered in the same way by $>_{min+}$. In the following corollary we prove that whatever pair is equally ranked by min+ cannot be (strictly) ordered in the CP-net semantics.

Corollary 2. *Given a CP-net N and its corresponding SCSP N_c with underlying semiring S_{WCSP}, if two outcomes $o_1, o_2 \in O$ are such that $o_1 =_{min+} o_2$ then either $o_1 = o_2$, that is they are the same outcome, or $o_1 \sim o_2$ in the CP-net.*

Proof: Assume, for the sake of contradiction that $o_1 \succ o_2$. Then by Theorem 7 it must be that $o_1 >_{min+} o_2$. But this contradicts the hypothesis that $o_1 =_{min+} o_2$. □

Thus, we have proved the first two implications of the above table.

Clearly, what is incomparable in the CP-net semantics is ordered in min+ since the ordering induced is total. If we consider our example (Figures 1 and 8) outcome $o_1 = a\bar{b}cd$ is incomparable to outcome $o_2 = \bar{a}bcd$. However since they both have penalty 6, $o_1 =_{min+} o_2$. This is an example of a pair of outcomes that are incomparable in the CP-net ordering and become equally ranked in min+. Instead, outcomes $o_3 = ab\bar{c}d$ and $o_4 = \bar{a}b\bar{c}\bar{d}$ are incomparable in the CP-net but are ordered by min+. In fact, $o_3 >_{min+} o_4$ since the cost associated to o_3 is 3 while that associated to o_4 is 4. These examples, together with Theorem 7, prove the last implication of the table.

In summary, by mapping a CP-net into a weighted SCSP we linearize the ordering of the CP-net, which can be a partial order or even a preorder. In other words, we are deciding an ordering on pairs of outcomes that the CP-net regarded as incomparable. On the other hand, the gain in tractability is considerable. In fact, preferential comparison is now achievable in linear time in the number of constraints, which are the same in number as the original cp statements of the CP-net: given any two outcomes, it is sufficient to compute their penalties and then compare them.

5.2. SLO soft constraints

We now consider a different semi-ring to approximate CP-nets via soft constraints. The SLO c-semi-ring is defined as follows: $S_{SLO} = \langle A, max_s, min_s, \mathbf{MAX}, \mathbf{0} \rangle$, where A is the set of sequences of n integers from 0 to MAX, **MAX** is the sequence of n elements all equal to MAX, and **0** is the sequence of n elements all equal to 0. The additive operator, max_s and the multiplicative operator, min_s are defined as follows: given $s = s_1 \cdots s_n$ and $t = t_1 \cdots t_n$, $s_i = t_i, i = 1 \leq k$ and $s_{k+1} \neq t_{k+1}$, then $max_s(s, t) = s$ if $s_{k+1} \succ t_{k+1}$ else $max_s(s, t) = t$; on the contrary, $min_s(s, t) = s$ if $s_{k+1} \prec t_{k+1}$ else $min_s(s, t) = t$. In the following theorem we prove that the algebraic structure defined above is a c-semiring.

Theorem 9. *$S_{SLO} = \langle A, max_s, min_s, \mathbf{MAX}, \mathbf{0} \rangle$ is a c-semiring.*

Proof: The result of applying max_s to two sequences of A is uniquely determined by the result of applying max on the values contained in the first component on which the two sequences differ. This allow us to derive the that max_s is commutative, associative and

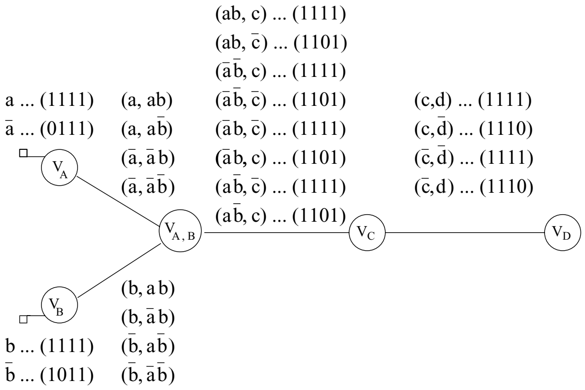


Fig. 9 The SLO SCSP for the SC-net of Figure 6

idempotent since max satisfies all these properties on the set of integers. It is also easy to see that the **0** is the unit element of max_s, since being defined as the string of n elements all equal to 0, given any other string $s = s_1 \cdots s_n$ since $s_i \in [0,\mathbf{MAX}]$, $s_i \leq 0\, \forall i$. The same reasoning can be applied to min_s, since its result is uniquely defined by that of min. Thus, min_s is associative and commutative. Moreover, **0** is the absorbing element of min_s since, as said before, all other strings contain all elements which greater or equal to 0. Since we know that for any string $s = s_1 \cdots s_n$, $s_i \in [0,\mathbf{MAX}]$, then applying min_s to s and **MAX** will always returns s as a result, thus **MAX** is the unit element of min_s. For the same reason **MAX** is the absorbing element of max_s. Finally, we have that min_s distributes over max_s since min distributes over max. □

The ordering induced by max_s on A is a lexicographic ordering (Fargier et al., 1993). To model a CP-net as a soft constraint problem based on S_{SLO}, we set MAX equal to the cardinality of the largest domain - 1, and n equal to the number of soft constraints of the SC net. All the weights of the edges are set to 1. Considering the binary soft constraint on $Pa(X) = \{U_1 \ldots U_h\}$ and X, a tuple of assignments $(u_1, \ldots, u_h, x)$ will be assigned, as preference, the sequence of n integers: $(MAX, MAX, \ldots, MAX - i + 1, \ldots, MAX)$. In this sequence, each element corresponds to a soft constraint. The element corresponding to the constraint on $Pa(X)$ and X is $MAX - i + 1$, where i is the distance from the top of the total order of the value x (i.e. we have a preference statement of the form $u : x_1 \succ x_2 \succ \ldots x_i = x \succ x_{|D(X)|}$).

In the SC-net shown in Figure 6, all the preferences are lists of four integers (0 and 1), where position i corresponds to constraint with weight w_i. For example, in constraint weighted w_3, $p_1 = (1, 1, 1, 1)$ and $p_2 = (1, 1, 0, 1)$. The resulting SLO SCSP is shown in Figure 9.

Given the pair of outcomes $o_1 = abcd$ and $o_2 = a\bar{b}cd$, the global preference associated with o_1 is (1, 1, 1, 1), since it does not violate any constraint, while the preference associated with o_2 is $min_S\{(1, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 1)\} = (1, 0, 1, 1)$. We can conclude that o_1 is better than o_2.

In the following theorem we prove that the SLO model both preserves the order information and ensures the cp-condition.

Theorem 10. *The SC-net based SLO SCSP N_c, generated from an acyclic CP-net N, is an approximation of N which respects the cp-condition and is order preserving.*

Proof: It is enough to prove that $>_{SLO}$ satisfies the cp-condition, that is, for each variable $X \in N$, each assignment $\mathbf{u}$ on $Pa(X)$, and each pair of values $x_1, x_2 \in \mathcal{D}(X)$, if CP-net specifies that $u : x_1 \succ x_2$, then we have $x_1\mathbf{uy} >_{SLO} x_2\mathbf{uy}$, for all assignments $\mathbf{y}$ on $\mathbf{Y} = \mathbf{V} - \{\{X\} \cup Pa(X)\}$. In the SLO model the preference associated to $x_1\mathbf{uy}$ and that associated to $x_2\mathbf{uy}$ are strings of positive integers each corresponding to a constraint. The constraints on which the two outcomes differ on are: constraint c on $V_{Pa(X)}$ and V_X, and all the constraints $t_i \in T$ on $V_{Pa(B_i)}$ and V_{B_i} such that $X \in Pa(B_i)$ (in what follows, we denote the children of X by $\mathcal{B} = \{V_{B_1}, \cdots, V_{B_h}\}$). By construction we have that since $u : x_1 \succ x_2$, the preference associated to the projection of $x_1\mathbf{uy}$ on constraint c is (**MAX MAX** $\cdots h \cdots$ **MAX MAX**) while for outcome $x_2\mathbf{uy}$ it is (**MAX MAX** $\cdots h - 1 \cdots$ **MAX MAX**). Since, by definition the position of constraint c precedes that of any constraint defined on the children of V_X, the first component on which the global preference of the outcomes will differ is that corresponding to c. Thus, applying max_s will return as a result that $x_1\mathbf{uy} >_{SLO} x_2\mathbf{uy}$ since $h > h - 1$. Moreover, it cannot be that two different outcomes have the same preference in SLO since by construction the preference string differ in at least one position. □

As in the previous case, we can easily see that the complexity of the mapping is polynomial in the size of the CP-net.

Theorem 11 (complexity). *Given an acyclic CP-net N with the size of the largest domain bounded by a constant d, the construction of the corresponding SC-net based SLO SCSP N_c is polynomial in the size of N.*

Proof: As in Theorem 11 we know that the SC-net has at most $2n$ and $E = e + n$ edges, if e is the number of edges of CP-net. A topological sort can be performed in $O(V + E)$, that is $O(2n + e + n) = O(e + n)$. Once each constraint is assigned a component in the string which is long at most $e + n$, the complexity of computing the preferences on each constraint is linear in the size of the domains $O(d)$. Thus, the complexity is $O(d(e + n)) = O(dn^2)$. □

In a similar way to the comparison performed for the min+ semi-ring, the following table compares the preference relation induced by the SLO semiring and that induced by the CP-net. Let us consider in detail the results shown in the table (starting from the left side):

CP-nets $\Rightarrow$ SLO	
$\prec$	$<$
$\succ$	$>$
$\sim$	$<, >$

– From Theorem 10 we know that $\succ$ implies $>_{SLO}$ and the symmetric result. This proves the first two implications of the table.
– If two outcomes are incomparable in the CP-net then they are ordered in the SLO, since SLO induces a total order. However, since they differ at least on the value assigned to one feature, say X, they also must have two different strings as SLO preference which differ on the component corresponding to the constraint defined on $Pa(V_X)$ and V_X. Going back to our example, outcome $o_1 = ab\bar{c}d$ is incomparable to outcome $o_2 = \bar{a}b\bar{c}\bar{d}$ in the CP-net, but $o_1 >_{SLO} o_2$ since it wins on the first constraint defined on V_A.

The SLO model, like the weighted model, is very useful for answering dominance queries, as it inherits the linear complexity of its semi-ring structure. In addition, the sequences of integers show directly the "goodness" of an assignment, i.e., where it actually satisfies the preference and where it violates it.

5.3. Comparing and combining the two approximations

Given an acyclic CP-net N, let N_c^{min+} and N_c^{SLO} stand for the corresponding min+ and SLO based SC-nets respectively. From the results in the previous section, we can see that pairs of outcomes ordered by N remain ordered the same way by both N_c^{min+} and N_c^{SLO}. On the other hand, pairs of outcomes incomparable in N are distributed among the three possibilities (equal or ordered in one the two ways) in N_c^{min+}, while being strictly ordered by N_c^{SLO}. Therefore, the (total) preference relation induced by N_c^{min+} is a less drastic linearization of the partial preference relation induced by N, compared to that induced by N_c^{SLO}. Mapping incomparability onto equality might seem more reasonable than mapping it onto an arbitrary strict ordering, since the choice is still left to the user. We might conclude that the min+ model is to be preferred to the SLO model, as far as approximation is concerned. However, maximizing the minimum reward, as in any fuzzy framework (Schiex, 1992), has proved its usefulness in problem representation. The user may therefore need to balance the linearization of the order and the suitability of the representation provided.

It is also possible to combine the two approximations and generate a third one which combines their advantages. In fact, let us consider the ordering over outcomes induced by the semiring obtained by performing the Cartesian product of the semirings of the two approximations. In Bistarelli et al. (1997) it has been shown that the Cartesian product of the kind of semirings used for soft constraints is still a semiring of the same kind. In this case, this amounts to associating to each outcome a pair of elements, one given by the min+ approach, and the other one by the SLO approach. Then, two outcomes are ordered if they are ordered in the same way on both elements of the pair, or if they are ordered in one of the elements and tied on the other. Otherwise, they are incomparable.

As the min+ and the SLO approximation, also this new approximation is order-preserving and respects the cp-condition. Moreover, some of the pairs of outcomes which are incomparable in the CP-net are left incomparable. Therefore, this approach allows for a better approximation of the CP-net ordering, since neither min+ nor SLO can model incomparability.

6. Related work

For acyclic CP-nets, two approximations that are order preserving have been introduced in the literature, both comparing outcomes in time linear in the number of features. The first is based on the relative position of the features in the CP-net dependency graph (Boutilier et al., 2004) and it is effectively similar to our SLO approximation. On the one hand, this approximation scheme in Boutilier et al. (2004) does not require any preprocessing of the CP-net. On the other hand, the SLO formalization has an advantage in unified treatment of both hard and soft constraints, and preference statements captured by CP-nets.

The second approximation, based on UCP-nets (Boutilier et al., 2001), can be used as a quantitative approximation of acyclic CP-nets. UCP-nets resemble weighted CSPs, and thus they can be used in constraint optimization using the soft constraints machinery. However, generating UCP-nets is exponential in the size of CP-net node's Markov family[2], and thus in the CP-net node out-degree.

An additional related work is described in McGeachie and Doyle (2002), where a numerical value function is constructed using graph-theoretic techniques by examining the graph of the preference relation induced by a set of preference statements. Note that this framework is also computationally hard, except for some special cases.

7. Conclusions and future work

We have proposed a unifying modelling and solving formalism in which both hard and soft constraints, as well as qualitative conditional preferences, can be handled efficiently. The framework consists of a soft constraint solver plus an algorithm for approximating the semantics of conditional preference statements by translating them into soft constraints. The translation requires some approximation but offers a computational gain. We have also studied the complexity of consistency checking for general sets of conditional preference statements.

We plan to develop this work in several ways. We will use our approach in a preference elicitation system in which we guarantee the consistency of the user preferences, and guide the user to a consistent scenario. We plan to exploit the use of partially ordered preferences, as allowed in soft constraints, to better approximate CP-nets. We will study the issue of abstracting one order with another one, which has been considered here in several instances. We also plan to study experimentally the phase transition in the satisfiability of conditional preference statements. Finally, we intend to use machine learning techniques to learn conditional preferences from comparisons of complete assignments.

References

Bistarelli, S., H. Fargier, U. Montanari, F. Rossi, T. Schiex, and G. Verfaillie. (1996). "Semiring-Based CSPs and Valued CSPs: Basic Properties and Comparison." In *Over-Constrained Systems*.

Bistarelli, S., U. Montanari, and F. Rossi. (1997). "Semiring-Based Constraint Solving and Optimization." *Journal of the ACM* 44(2), 201–236.

Boutilier, C., R. Brafman, H. Hoos, and D. Poole. (1999). "Reasoning with Conditional Ceteris Paribus Preference Statements." In *Proc. of UAI*, pp. 71–80.

Boutilier, C., F. Bacchus, and R.I. Brafman. (2001). "UCP-Networks: A Directed Graphical Representation of Conditional Utilities." In *Proc. of UAI*, pp. 56–64.

Boutilier, C., R. Brafman, C. Domshlak, H. Hoos, and D. Poole. (2004). "CP-nets: A Tool for Representing and Reasoning about Conditional. *ceteris paribus* preference statements." *Journal of Artificial Intelligence Research (JAIR)* 21, 135–191.

Boutilier, C., R. Brafman, C. Domshlak, H. Hoos, and D. Poole. (2004). "Preference-Based Constraint Optimization with CP-Nets." *Computational Intelligence* 20(2), 137–157.

Brafman, R. and Y. Dimopoulos. (2004). "Extended Semantics and Optimization Algorithms for CP-Networks." *Computational Intelligence* 20(2), 218–245.

Domshlak, C. and R. Brafman. (2002). "CP-Nets—Reasoning and Consistency Testing." In *Proc. of KR*, pp. 121–132.

Domshlak, C., F. Rossi, K.B. Venable, and T. Walsh. (2003). "Reasoning About Soft Constraints and Conditional Preferences: Complexity Results and Approximation Techniques." In *Proc. of IJCAI*, Acapulco, Mexico, August.

Doyle, J. and R.H. Thomason. (1999). "Background to Qualitative Decision Theory." *AI Magazine* 20(2), 55–68.

[2] The *Markov family* of a node X contains X, its parents and children, and the parents of its children.

Doyle, J. and M. Wellman. (1994). "Representing Preferences as Ceteris Paribus Comparatives." In *Proc. of AAAI Spring Symposium on Decision-Making Planning*, pp. 69–75.

Dubois, D., H. Fargier, and H. Prade. (1993). "The Calculus of Fuzzy Restrictions as A Basis for Flexible Constraint Satisfaction." In *Proc. of IEEE International Conference on Fuzzy Systems*.

Fargier, H., J. Lang, and T. Schiex. (1993). "Selecting Preferred Solutions in Fuzzy Constraint Satisfaction Problems." In *Proc. of 1st European Congress on Fuzzy and Intelligent Technologies (EUFIT)*, pp. 277–288.

Freuder, E.C. and R.J. Wallace. (1992). "Partial Constraint Satisfaction." *Artificial Intelligence* 58(1–3), 21–70.

Garey, M.R. and D.S. Johnson. (1978). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New-York.

Goldsmith, J., J. Lang, M. Truszczynski, and N. Wilson. (2005). "The Computational Complexity of Dominance and Consistency in CP-Nets." In *Proc. of IJCAI*.

Hansson, S.O. (2001). "Preference Logic." In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pp. 319–394. Kluwer.

Lang, J. (2002). "From Preference Representation to Combinatorial Vote." In *Proc. of KR*.

McGeachie, M. and J. Doyle. (2002). "Efficient Utility Functions for Ceteris Paribus Preferences." In *Proc. of AAAI*, pp. 279–284.

Meseguer, P., F. Rossi, T. Schiex, and K.B. Venable. (2004). "Private Communication." April.

Prestwich, S., F. Rossi, K.B. Venable, and T. Walsh. (2005). "Constraint-Based Preferential Optimization." In *Proc. of AAAI*, Pittsburgh, Pennsylvania, July 9–13, Morgan Kaufmann.

Sabin, D. and R. Weigel. (1998). "Product Configuration Frameworks—A Survey." *IEEE Intelligent Systems and their Applications* 13(4), 42–49.

Schiex, T., H. Fargier, and G. Verfaille. (1995). "Valued Constraint Satisfaction Problems: Hard and Easy Problems." In *Proc. of IJCAI*, pp. 631–637.

Schiex, T. (1992). "Possibilistic Constraint Satisfaction Problems," or "How to Handle Soft Constraints?." In *Proc. of UAI*, pp. 269–275.