



## Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>  
Eprints ID : 17239

The contribution was presented at SUM 2016 :  
<http://sum2016.unice.fr/>

**To cite this version** : Ben Amor, Nahla and Dubois, Didier and Gouider, Hela and Prade, Henri *Graphical Models for Preference Representation: An Overview*. (2016) In: 10th International Conference on Scalable Uncertainty Management (SUM 2016), 21 October 2016 - 23 October 2016 (Nice, France).

Any correspondence concerning this service should be sent to the repository administrator: [staff-oatao@listes-diff.inp-toulouse.fr](mailto:staff-oatao@listes-diff.inp-toulouse.fr)

# Graphical models for preference representation: An overview

Nahla Ben Amor<sup>1</sup>, Didier Dubois<sup>2</sup>, H ela Gouider<sup>1</sup>, and Henri Prade<sup>2</sup>

<sup>1</sup> LARODEC Laboratory, ISG de Tunis, 41 rue de la Libert e, 2000 Le Bardo, Tunisia

<sup>2</sup> IRIT – CNRS, 118, route de Narbonne, 31062 Toulouse Cedex 09, France

nahla.benamor@gmx.fr, {dubois,prade}@irit.fr, gouider.hela@gmail.com

**Abstract.** Representing preferences into a compact structure has become an important research topic. Graphical models are of special interest. Indeed, they facilitate elicitation, exhibit some form of independence, and serve as a basis for solving optimization and dominance queries about choices. The expressiveness of the representation setting and the complexity of answering queries are then central issues for each approach. This paper proposes an extensive overview of the main graphical models for preference representation and provides a comparative survey by emphasizing their main characteristics. We also indicate possible transformations between some of these models. We contrast qualitative models such as CP-nets and TCP-nets with quantitative ones such as GAI networks, UCP-nets, and Marginal utility nets, and advocate  $\pi$ -Pref nets, recently introduced by the authors, as an interesting compromise between the two types of models.

## 1 Introduction

Modeling preferences is essential in any decision analysis task. However, getting these preferences becomes non trivial as soon as alternatives are described by a Cartesian product of multiple features. Indeed, the direct assessment of a preference relation between these alternatives is usually not feasible due to its combinatorial nature. Fortunately, the decision maker can express *contextual* preferences that exhibit some independence relations, which allows us to be represent her/his preferences in a compact manner. Moreover, graphical representations facilitate preference elicitation, as well as the construction of an ordering from these contextual local preferences. This use of graphical preference representations has been inspired by the success of Bayesian networks as a computationally tractable knowledge representation device [20].

Various graphical models have been proposed in the literature in order to capture preferences in an intuitive manner. We may roughly distinguish two classes: (i) qualitative models where preferences are contextually expressed by comparisons between attributes values. Within these models, CP-nets [7] are the most popular and well-developed compact representation setting for preferences; (ii) quantitative models, where a numerical value function can be computed for comparing all possible choices, such as GAI networks [19], UCP-nets [5], or marginal utility nets [10]. In general, these models are mostly motivated by the easiness of elicitation. However, some of them still suffer from various limitations: their expressive power may be somehow restricted, elicitation may be complex, or answering queries may require costly reasoning algorithms.

This paper surveys most graphical models for preference representation. It enlarges the only existing past overview [21]. For each model, we emphasize the independence relation underlying it, study how it operates for defining an order between the choices from the expressed preferences, and recall the computational complexity of dominance and optimization algorithms. The paper is organized as follows. Sections 2 and 3 provide a presentation of the major qualitative or quantitative graphical models respectively, allowing for a local processing of elementary preferences by exploiting some structural independence relations carried by their graphical components. Section 4 presents a symbolic graphical model for preferences based on possibility theory and possibilistic networks. This recent approach recently introduced by the authors, is halfway between qualitative and quantitative models. Section 5 concludes with a summary and a thorough comparative discussion.

## 2 Graphical preferential qualitative models

Let  $V = \{A_1, \dots, A_N\}$  be a set of  $N$  variables. Each variable  $A_i$  has a domain  $D(A_i)$ ;  $a_i$  denotes any value of  $A_i$ .  $\Omega = \{\omega_1, \dots, \omega_{|\Omega|}\}$  denotes the universe of discourse, which is the Cartesian product of all variable domains in  $V$ . Each element  $\omega_i \in \Omega$  is called a *configuration*. It corresponds to a complete instantiation of the variables in  $V$ . If  $X \subseteq V$ , let  $D(X)$  refer to the Cartesian product of the domains of variables in  $X$  and  $\omega[X]$  denotes the restriction of variable  $\omega$  to variables in  $X$ .

Semantically, preferences are defined by an order between the configurations (or choices). Let  $\succeq$  be a binary relation on  $\Omega$  such that  $x \succeq y$  means that “ $x$  is at least as preferred as  $y$ ”. Other relations can be derived from  $\succeq$  as usual:  $\omega_i \sim \omega_j$  iff  $\omega_i \succeq \omega_j$  and  $\omega_j \succeq \omega_i$ ;  $\omega_i \succ \omega_j$  iff  $\omega_i \succeq \omega_j$  but not  $\omega_j \succeq \omega_i$ ;  $\omega_i \pm \omega_j$  iff neither  $\omega_i \succeq \omega_j$  nor  $\omega_j \succeq \omega_i$  (non comparability). Ordering relations may be total (i.e. we can compare any two configurations) or partial, strict (i.e. asymmetric) or weak. Preference relations between different configurations  $\omega_i \in \Omega$  can be expressed via some preference relations over subsets of variables, and take advantage of (in)dependencies that exist between the variables or subsets of variables. We denote by  $Pa(A_i)$  the set of parents of  $A_i$ ,  $u_i$  any instantiation of  $Pa(A_i)$  and  $\mathcal{Y}(A_i) = \{Y_1, \dots, Y_n\}$  the set of its children.  $Dn(A_i)$  denotes its descendants and  $Co(A_i) = V / (Dn(A_i) \cup Pa(A_i) \cup A_i)$  denotes the set of non-descendants. We will use these notations for the rest of the paper.

In a preference model, two types of queries are commonly used: namely, optimization queries for finding the optimal configuration(s) (i.e. those which are not dominated by others) and dominance queries for comparing configurations. Besides, another important task is the elicitation of the model which corresponds to constructing the graph and eliciting the user preferences. Most of practically used preferential graphical models are qualitative since they are easy to elicit. In the sequel, we detail two of the most important ones, namely, Conditional Preference networks (CP-nets) and their extension Tradeoffs-enhanced CP-nets.

### 2.1 Conditional Preference networks (CP-nets)

CP-nets, initially introduced in [7], are considered as an efficient model to manage qualitative preferences. They are based on a preferential independence relation often

referred to as a *Ceteris Paribus* assumption such that a partial configuration is preferred to another everything else being equal. Formally, it is defined as follows:

**Definition 1 (Preferential independence).** Let  $V$  be a set of variables and  $W$  be a subset of  $V$ .  $W$  is said to be preferentially independent from its complement  $Z = V \setminus W$  iff for any instantiations,  $z, z', w, w'$  we have:

$$(w, z) \succ (w', z) \Leftrightarrow (w, z') \succ (w', z') \quad (1)$$

Preferential independence is asymmetric. Indeed, it might happen, e.g., for disjoint sets  $X, Y$  and  $Z$  of variables that  $X$  is preferentially independent (Definition 1) from  $Y$  given  $Z$  without having  $Y$  preferentially independent from  $X$ . This independence is at a work in the graphical structure underlying CP-nets.

**Definition 2 (CP-nets).** A CP-net consists of a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  denotes the set of nodes and  $\mathcal{E}$  denotes the set of edges. A node corresponds to a variable. Edges represent the preference dependencies between the variables. To each variable  $A_i$  we associate a conditional preference table that corresponds to a total order between the values of  $A_i, \forall u_i$ .

Here, preferences over values of a variable depend only on the parent(s) context, and are preferentially independent from the rest of variables. In contrast with Bayesian nets, CP-nets may be cyclic (without necessarily encoding inconsistent preferences). Using the information in the CP-Tables and applying the *Ceteris Paribus* principle, when one *flips* one variable value in a configuration one may obtain either an improved configuration, or a worsened one. These swap pairs can be organized into a collection of worsening (directed) paths with a unique root corresponding to the best configuration and where the other path extremities are the worst ones. A CP-net is said to be satisfiable if there exists at least one partial order of configurations that satisfies it. Note that, every acyclic CP-net is satisfiable.

*Example 1.* Let us consider the simple CP-Net of Fig. 1(a), with 3 variables. The build-

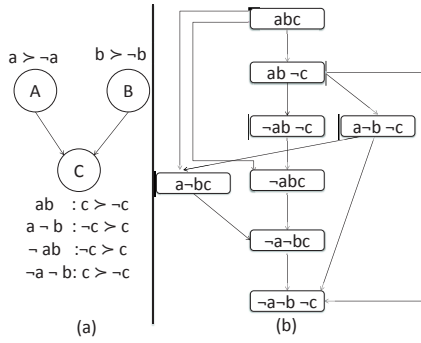


Fig. 1: An example of a CP-net (a) and its worsening flips graph (b)

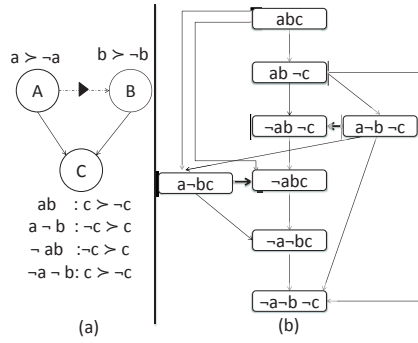


Fig. 2: An example of a TCP-net (a) and its worsening flips graph (b)

ing of the worsening flips graph (Figure 1(b)) leads to the partial ordering:  $abc \succ_{CP}$

$ab\bar{c} \succ_{CP} \bar{a}b\bar{c} \succ_{CP} \bar{a}bc \succ_{CP} \bar{a}\bar{b}c \succ_{CP} \bar{a}\bar{b}\bar{c}$ ,  $ab\bar{c} \succ_{CP} a\bar{b}\bar{c} \succ_{CP} \bar{a}\bar{b}\bar{c}$ ,  $abc \succ_{CP} a\bar{b}c \succ_{CP} \bar{a}\bar{b}c$ . The best configuration is  $abc$ .

CP-nets have a unique optimal configuration. Finding it amounts to look for a configuration where all the conditional preferences are best satisfied. This can be found by a simple forward sweeping procedure where, for each node, we assign the most preferred value according to the parents context. For acyclic CP-nets, this procedure is linear w.r.t. the number of variables [7]. In contrast, for cyclic ones answering this query needs an NP-hard algorithm and may lead to more than one optimal configuration [17]. Dominance queries are more complex. Using the information in the CP-Tables and applying the *Ceteris Paribus* principle, when one *flips* one variable value in a configuration one may obtain either an improved configuration, or a worsened one. These swap pairs can be organized into a collection of worsening (directed) paths with a unique root corresponding to the best configuration and where the other path extremities are the worst ones. Thus, a configuration is preferred to another if there exists a chain (directed path) of worsening flips between them [6]. Note that if for any variable  $A_i \in V$ ,  $A_i$  is preferentially independent from  $V \setminus A_i$ , then the CP-net graph is disconnected and many configurations cannot be compared. Testing dominance is PSPACE-complete for unrestricted CP-nets, NP-hard for acyclic ones, and quadratic for tree-structures [17]. In general, the ordering induced by a CP-net is strict and partial, since several configurations may remain non comparable (i.e. no worsening flips chain exists between them). Clearly, acyclic CP-nets cannot exhibit any ties. *Ceteris Paribus* makes the preference elicitation simple for CP-nets; the elicitation complexity is equal to  $O(N^k)$  such that  $N$  is the number of nodes and  $k$  is the maximal number of parents [22].

However, in CP-nets, a parent preference tends to be more important than a child one [6]. In other words, violating a preference associated with a father node is more important than violating a preference associated with a child one; this priority *implicitly* given by the application of *Ceteris Paribus* may be debatable. For instance, in the previous example, configuration  $ab\bar{c}$  is preferred to configuration  $\bar{a}bc$ . Moreover, this kind of priority is not transitive in the sense that CP-nets cannot always decide whether violating preferences of two children nodes is preferred to violating preferences associated with one child and one grandson node respectively (which might have been expected as being less damaging than violating two children preferences) [13]. This limitation is problematic. Generally, CP-nets cannot express any partial preference ordering, see [4] for counterexamples.

## 2.2 Tradeoffs-enhanced CP-nets (TCP-nets)

As mentioned above, the expressive power of CP-nets is limited. In particular, we are unable to specify importance relations between variables, beside those implicitly imposed between parents and children. Tradeoffs-enhanced CP-nets (TCP-nets) [8] are an extension of CP-nets that adds a notion of importance between the variables by enriching the network with new arcs. These arcs express importance relations for stating the priority of a node over another (i.e., “preference about the values of  $X$  is more important than preference about the values of  $Y$ ”). Such priority statements may be conditioned on the values of other variables, e.g., “if the variable  $Z$  has value  $z$ , the preference about

values of  $X$  is more important than the preference about the values of  $Y$ .” Formally, TCP-nets are annotated graphs with three types of edges and are defined as below.

**Definition 3 (TCP-nets).** A TCP-net  $\mathcal{G}'$  over a set  $V$  of variables is a CP-net  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  augmented with two types of arcs:

1. A set of directed  $i$ -arcs (where  $i$  stands for importance). An  $i$ -arc  $\langle \overrightarrow{A_i, A_j} \rangle$  belongs to  $\mathcal{G}'$  iff  $A_i$  is more important than  $A_j$ , which is denoted by  $A_i \triangleright A_j$ .
2. A set of undirected  $ci$ -arcs (where  $ci$  stands for conditional importance). A  $ci$ -arc  $(A_i, A_j)$  belongs to  $\mathcal{G}'$  iff the relative importance of  $A_i$  and  $A_j$  is conditioned on  $Z$  s.t.  $Z \subseteq V \setminus \{A_i, A_j\}$ . Each  $ci$ -arc  $(A_i, A_j)$  is associated with a mapping from a subset of  $D(Z)$  to total orders over the set  $\{A_i, A_j\}$ .

Let us turn to the expressive power of TCP-nets. TCP-nets are obey the preference statements induced by *Ceteris Paribus*, since the ordering obtained is a refinement of the CP-nets ordering. In fact, the refinement brought by TCP-nets cannot override the implicit priority in favor of parents nodes. Indeed, in case one would add a  $i$ –, or a  $ci$ – arc yielding a preference in favor of a son with respect to a parent (at least in some context), one would face an inconsistency between a worsening I-flip and a worsening CP-flip that act in opposite directions, thus we would have inconsistent TCP-nets.

The main issue for TCP-nets is the challenge of performing queries with this representation. Some first proposals are presented in [9]. For consistent TCP-nets, the optimization procedure works like CP-nets. Indeed, the relative importance relations do not play a role in this case. The dominance problem can be also be treated as a search for an improving flipping sequence, where the notion of flipping sequence is extended. In fact, a flip corresponds either to a CP-flip like CP-nets or to an I-flip (“importance flip”). Let  $\omega$  and  $\omega'$  be two configurations, such that  $\omega$  differs from  $\omega'$  in the value of exactly two variables  $A_j$  and  $A_k$ , and such that  $\omega[A_j] \succ \omega'[A_j]$  and  $\omega[A_k] \prec \omega'[A_k]$  (given the same values of  $Pa(A_j)$  and  $Pa(A_k)$  in  $\omega$  and  $\omega'$ ). Then, a worsening I-flip from  $\omega$  to  $\omega'$  takes place when there is a priority of  $A_j$  over  $A_k$  when conditioned (or not) on a subset of variables  $Z$  such that  $Z$  takes the same values in  $\omega$  and  $\omega'$ . However, no general algorithm is known for dominance query since results in the context of CP-nets do not seem to be immediately adaptable to TCP-nets.

*Example 2.* Let us consider the TCP-net in Figure 2(a). An unconditioned importance  $a \triangleright b$  is added. Indeed, a new arc  $i$ -arc  $\langle \overrightarrow{A, B} \rangle$  is added with respect to the CP-net in Figure 1(a). The ordering given by the worsening flips graph in Figure 2(b) is refined, compared to the CP-net. Indeed,  $a \neg b \neg c \succ_{TCP-net} \neg ab \neg c$  and  $a \neg bc \succ_{TCP-net} \neg abc$ , while these configurations comparable by I-flips, are not comparable in the CP-net, see Figure 1(b). In place of the previous unconditioned importance statement, one may exhibit an example of  $ci$ -arc  $(A, B)$  by stating that  $A$  is more important than  $B$  if  $C = c$ , and  $B$  is more important than  $A$  if  $C = \neg c$ . Then, we would have  $a \neg b \neg c \prec_{TCP-net} \neg ab \neg c$  and  $a \neg bc \succ_{TCP-net} \neg abc$ .

As CP-nets, TCP-nets generally yield partial orderings. Precisely, from the same preference statements, the orderings induced by TCP-nets are refinements of the ordering induced by CP-nets.

*Example 3.* Let us consider the following preferences over variables  $A$  and  $B$  with  $D(A) = \{a, \neg a\}$  and  $D(B) = \{b, \neg b\}$ : (i) In all cases  $a$  is preferred to  $\neg a$ ; (ii)  $b$  is preferred to  $\neg b$ . The CP-net view yields the order:  $ab \succ_{CP} a\neg b \pm_{CP} \neg ab \succ_{CP} \neg a\neg b$ . No CP-net yields the refined order  $ab \succ a\neg b \succ \neg ab \succ \neg a\neg b$ , while it can be represented with a TCP-net, with the additional information “ $A$  is more important than  $B$ ”.

### 3 Graphical preferential quantitative models

It is often convenient to have preferences expressed in numerical terms, since it enables an easy comparison of possible choices. It is therefore interesting to consider quantitative graphical models for preferences. These latter are generally based on utility functions corresponding to a mapping from the Cartesian product of variables domains to numerical values, namely  $u : \Omega \mapsto \mathbb{R}$ . These utilities corresponds to a total ordering s.t., for two configurations  $\omega$  and  $\omega'$ ,  $\omega \succ \omega'$  (respectively  $\omega \sim \omega'$ ) if and only if  $u(\omega) > u(\omega')$  (respectively  $u(\omega) = u(\omega')$ ). In this section, we review the most important quantitative graphical models based on these utilities.

#### 3.1 Generalized Additive Independence networks (GAI-nets)

GAI-networks [19] are one of the first graphical quantitative preference models. They rely on generalized additive independence decomposition (GAI decomposition, for short) [16]. This independence allows to represent the preferences by a utility separable into a sum of local utility functions. Each local utility pertains to a subset of variables and represents a total ordering between their possibles instantiations. Moreover, there may be some interactions between these local utilities since the subsets of variables pertaining to them can be non disjoint. Thus, these GAI-decompositions can express some general interactions between attributes while preserving some decomposability of the model.

**Definition 4 (GAI decomposition).** Let  $C_1, \dots, C_k$  be subsets of  $V$  s.t.  $V = \bigcup_{j=1}^k C_j$ . A utility function  $u(\cdot)$  representing  $\succeq$  over  $\Omega$  is GAI-decomposable w.r.t.  $C_1, \dots, C_k$  iff  $\forall j \in [1, k]$ , there exists a function  $u_j : D(C_j) \mapsto \mathbb{R}$  s.t.,  $\forall \omega \in \Omega$  :

$$u(\omega) = \sum_{j=1}^k u_j(\omega[C_j]) \quad (2)$$

These GAI decompositions can be represented by graphical structures called GAI networks. These latter are undirected graphs where each clique consists of a subset of variables. Between two cliques having some variables in common there exists a path linking them. Each edge in the network is labeled by the intersection between the nodes.

**Definition 5 (GAI-nets).** A GAI network is an undirected graph  $\mathcal{G} = (\mathcal{C}, \mathcal{E})$  where  $\mathcal{C}$  denotes the set of cliques and  $\mathcal{E}$  denotes the set of edges.  $\mathcal{G}$  has two components:

- *Graphical component:* Each clique  $C_j \in \mathcal{C}$ , is a set of variables such that  $C_j \subseteq V$  and  $\bigcup_{i=1}^k C_i = V$ ; For each edge  $(C_i, C_j) \in \mathcal{E}$ ,  $C_i \cap C_j \neq \emptyset$ . Each edge is labeled by  $C_i \cap C_j$ ;

- *Numerical component: To each clique  $C_j$  we associate a local utility function  $u_j$  that defines a complete preorder between the configurations in  $D(C_j)$ .*

The graphical structure of GAI-nets is similar to the notion of junction tree used for Bayesian networks [20, 23]. Indeed, even for a GAI-net with a more general graph structure, we can always construct a tree-structured network based on the triangulation of the Markov network corresponding to it [18] (This transformation is NP-complete [1]). Optimization queries look for the configurations having the maximal global utility value. A standard algorithm for finding the optimal configurations has been proposed for tree structured GAI networks. However, as mentioned above, this is not restrictive. Optimization for GAI-nets corresponds to an adaptation of the belief propagation algorithm used in Bayesian networks and its complexity is exponential to the number of variables of the biggest clique. To compare two configurations  $\omega$  and  $\omega'$  by a GAI-net, we compute their corresponding utilities and compare them. Thus, the dominance test for GAI is linear in the number of the cliques which is considered as an advantage compared to the other models.

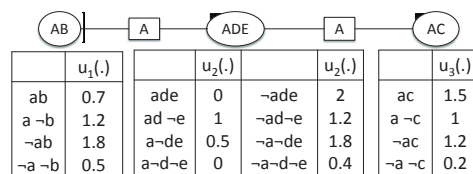


Fig. 3: An example of GAI network

*Example 4.* Let  $\omega_1 = abcde$  and  $\omega_2 = a \neg bc \neg d \neg e$  be two configurations. From the GAI-network  $\mathcal{G}$  of Figure 3, we can compute the utilities of the configurations:  $\omega_1$  is equal to  $u_{\mathcal{G}}(\omega_1) = u_1(ab) + u_3(ac) + u_2(ade) = 0.7 + 1.5 + 0 = 2.2$ ,  $\omega_2$  is equal to  $u_{\mathcal{G}}(\omega_2) = u_1(a \neg b) + u_3(ac) + u_2(a \neg d \neg e) = 1.2 + 1.5 + 0 = 2.7$ . Thus  $u_{\mathcal{G}}(\omega_2) > u_{\mathcal{G}}(\omega_1)$ , and  $\omega_2 \succ_{GAI} \omega_1$ .

GAI-nets rely on a weak form of symmetric independence which make the model flexible enough to be applied to many situations. GAI-nets are not limited to the expression of *Ceteris Paribus* preferences as CP-nets, TCP-nets, or their numerical counterpart, UCP-nets. Still there are cases of numerical preferences that are not representable by a GAI-net [15]. With regard to elicitation, there is no method to construct the GAI decompositions. In practice it is always assumed that an expert provided the GAI decomposition and only the utilities are elicited. One may take advantage of the GAI structure for designing an elicitation method based on “local” utility queries rather than global queries over full configurations [12].

### 3.2 Utility CP-nets (UCP-nets)

Utility CP-nets (UCP-nets), introduced in [5], are an extension of CP-nets that replaces the ordinal preference relations of CP-nets by utility factors. In fact, UCP-nets combine the aspects of two preference models, namely, CP-nets and GAI-nets. Like GAI-nets, utility is obtained from the sum of functions associated to groups of variables, defined



here by a variable and its parents. Similarly to CP-nets, UCP-nets are directed and arcs reflect the *Ceteris Paribus* independence.

**Definition 6 (UCP-nets).** A UCP-net is a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  denotes the set of nodes and  $\mathcal{E}$  denotes the set of edges. It has two components:

- Graphical component: Each variable  $A_i \in V$  is represented by a node and directed edges represent *Ceteris Paribus* dependencies;
- Numerical component: a set of factors  $f_i(a_i, u_i)$ ,  $\forall a_i \in D(A_i)$  and  $\forall u_i$ , such that the global utility of a configuration is defined by:

$$u_{\mathcal{G}}(a_1, \dots, a_N) = \sum_{i=1}^N f_i(a_i, u_i) \quad (3)$$

*Example 5.* The UCP-net  $\mathcal{G}$  presented in Figure 4 has 3 variables  $V = \{A, B, C\}$ . For instance, we can check that the configuration  $a \neg b \neg c$  is preferred to  $abc$  since  $u_{\mathcal{G}}(abc) = 5 + 2 + 2 = 9 < u_{\mathcal{G}}(a \neg b \neg c) = 5 + 10 + 6 = 21$ .

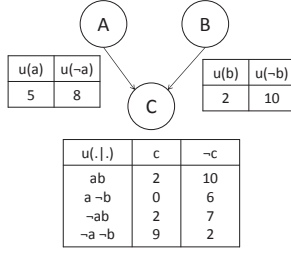


Fig. 4: An example of a UCP-net

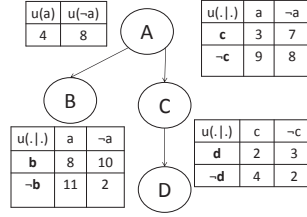


Fig. 5: An example of a marginal utility net

The UCP-net formalism has a number of computational advantages. In particular, dominance queries can be answered trivially since they amount to computing the global utilities and compare them, as in the above example. This can be done in linear time in the number of variables (this contrasts with CP-nets where dominance testing is computationally difficult). Optimization queries can also be answered directly, taking linear time in the network size, where each node is instantiated to its maximal value given the instantiation of its parents. This procedure, inherited from CP-nets, exploits the considerable power of *Ceteris Paribus* semantics. Thus, CP-nets are endowed with quantitative utility information, and then the expressive power is enhanced and dominance queries become computationally efficient. Moreover, when introducing directionality and the *Ceteris Paribus* semantics to GAI relations, we allow utility functions to be expressed more naturally and optimization queries to be answered more easily.

This model is intuitive to assess since, as CP-nets, it captures preference statements that are naturally expressed by the user. However, in order to remain consistent with CP-nets, utilities should be subject to constraints expressing the priority of father nodes over child nodes. More precisely, let  $A$  be a variable with parents  $Pa(A)$  and children  $\mathcal{Y}(A) = \{Y_1, \dots, Y_n\}$  and let  $\mathcal{Z}_i$  be the subset of parents of  $Y_i$  excluding  $A$  and any of its parents in  $Pa(A)$ . Let  $\mathcal{Z} = \bigcup \mathcal{Z}_i$  and  $P_i$  be the subset of variables in  $Pa(A)$  that are parents of  $Y_i$  and where  $p_i$  is an instantiation of  $P_i$ . The fact that the node corresponding to variable  $A$  dominates its children given any instantiation  $u$  of  $Pa(A)$  is expressed by

the requirement  $\forall a_1, a_2 \in D(A)$  such that  $f_A(a_1, u) \geq f_A(a_2, u)$ , we should have  $\forall z$  an instantiation of  $\mathcal{Z}$  and  $\forall y_i$  an instantiation of  $\mathcal{Y}(A)$ ,  $f_A(a_1, u) - f_A(a_2, u) \geq \sum_i f_{Y_i}(y_i, (a_2, p_i, z_i)) - f_{Y_i}(y_i, (a_1, p_i, z_i))$ . This expresses that for any variable  $A$ , given an instantiation of its parents, the utility gain in choosing in this context  $a_1$  rather than  $a_2$ , should be more important than the maximum value of the sum of the possible utility loss for its children over all possible instantiations of the other related variables.

This means that not every GAI decomposition can be represented by a UCP-net. Thus, beside the difficulty encountered for learning utilities, added constraints should be taken into account in order to remain consistent with the *Ceteris Paribus* principle.

### 3.3 Marginal utility networks

With the aim to define preference networks that resemble Bayesian networks, Brafman and Engel [10, 11] introduce a notion of conditional independence (denoted  $CDI_r$ ) using an arbitrarily fixed reference instantiation  $\omega^r$ . Indeed utility functions differ from probability distributions in the fact there is no obvious analogue of marginalization for utility; to cope with this difficulty, the authors propose to use reference instantiation for fixing the values of the independent variables. Then, the utility satisfies additive analogues of the Bayes and chain rules of Bayesian networks. Variables  $A_i$  and  $A_j$  are  $CDI_r$  if any difference in values among instantiations to  $A_i$  does not depend on the current instantiation of  $A_j$ , for any possible instantiation to the rest of the variables.

**Definition 7 (Reference configuration and the reference utility).** Let  $\omega^r = a_1^r, \dots, a_N^r \in \Omega$  be a predetermined configuration and,  $X$  and  $Y$  be subsets of  $V$ . The reference utility function  $u_r$  is defined by  $u_r(x) = u(x\bar{x}^r)$ , s.t.  $\bar{X} = V \setminus X$  is fixed on the values of the reference configuration  $\omega^r$ . Its conditional form is defined by  $u_r(X|Y) = u_r(XY) - u_r(Y)$ .

**Definition 8 (Difference utility independence).** Let  $Z$  and  $W$  be two subsets of  $V$ , s.t.  $Z \cap W = \emptyset$ .  $Z$  and  $W$  are  $CDI_r$  given  $X \subseteq V / (Z \cup W)$ , denoted by  $CDI_r(Z, W|X)$ , if for all assignments  $x, z', z'', w', w''$  we have:  $u_r(z'w') - u_r(z''w') = u_r(z'w'') - u_r(z''w'')$ .

This type of independence ( $CDI_r$ ) satisfies the foundations of graphoid theory [11], that is, each variable is independent from its non descendants in the context of its parents as for Bayesian nets. This leads to a preference representation by directed graphs.

**Definition 9 (Marginal utility network).** A marginal utility network is a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  is the set of nodes and  $\mathcal{E}$  is the set of edges.  $\mathcal{G}$  has two components:

- Graphical component: A node for each variable and edges correspond to conditional (in)dependencies between variables such that, given a fixed configuration  $\omega^r \in \Omega$ , for any  $A_i \in V$ ,  $CDI_r(A_i, Co(A_i) | Pa(A_i))$ .
- Numerical component: Each node  $A_i$  is associated to a conditional utility table (CUT) corresponding to the function  $u_r(a_i | u_j)$  such that  $u_j$  is an instantiation of the parents  $Pa(A_i)$  of  $A_i$ . containing  $\forall a_i \in D(A_i), \forall u_i, u_r(a_i | u_j)$ .

The utility of a configuration is then computed as  $u_G(a_1, \dots, a_N) = \sum_{i=1}^N u_r(a_i|u_i)$  where  $u_i$  is an instantiation of  $P(A_i)$ . This is now exemplified.

*Example 6.* Let us consider preferences over four binary variables  $A, B, C$  and  $D$  represented by the marginal utility network of Figure 5. Assume that  $\omega_r = abc\bar{d}$  is the reference configuration. Then,  $u_r(abc) - u_r(a\bar{b}c) = u_r(ab\bar{c}) - u_r(a\bar{b}\bar{c})$ . In fact,  $(4 + 8 + 3 + 4) - (4 + 11 + 3 + 4) = (4 + 8 + 9 + 4) - (4 + 11 + 9 + 4)$ . Thus,  $CDI_r(B, D|A)$ . The utility of a configuration is the summation of all the local utilities. For instance,  $u_G(abcd) = u_r(a) + u_r(b|a) + u_r(c|a) + u_r(d|c) = 4 + 8 + 3 + 2 = 17$  and  $u_G(a\bar{b}\bar{c}\bar{d}) = 4 + 11 + 9 + 2 = 26$ . Therefore, we have  $abcd \prec_{MU} a\bar{b}\bar{c}\bar{d}$  since  $u_G(abcd) < u_G(a\bar{b}\bar{c}\bar{d})$ .

Thanks to the strong similarity between Bayesian nets and marginal utility nets, adaptations of algorithms are possible. The authors in [11] have briefly mentioned two of them. First, finding the Most Probable Explanation which is used as an optimization query for finding the optimal configuration. Second, Constraint Belief Propagation for finding the best configuration when particular combinations between the variables are impossible. No method to answer dominance queries has been proposed, however the algorithm used in GAI nets seems to be applicable in this case. Elicitation may be inspired from Bayesian nets [10].

Following also the idea of keeping close to Bayesian nets, it has been recently proposed to use Ordinal Conditional Function networks (which are like Bayesian nets with infinitesimal probabilities: the value  $n$  of the OCF is like the probability  $10^{-n}$ ) for describing preferences [14]. OCF-nets satisfy the local directed Markov independence property. By enforcing the priority of father nodes over child nodes by suitable constraints, it is possible to build an OCF-net that induces a total order compatible with the partial order of a given CP-net [14]. Besides, note that UCP-nets can be viewed as particular cases of marginal utility nets where constraints should be added in order to make them consistent with *Ceteris Paribus*.

## 4 Conditional Preference Possibilistic networks

Marginal networks are inspired from Bayesian networks. Similarly, one may use possibilistic networks [2], a possibility theory counterpart to Bayes nets, for modeling preferences rather than uncertainty (understanding the possibility degrees as satisfaction levels). Possibility theory relies on the idea of a possibility distribution  $\pi$ , which is a mapping from a universe of discourse  $\Omega$  to the unit interval  $[0, 1]$ , or to any bounded totally ordered scale. Two forms of conditioning, respectively based on minimum and product, make sense in possibility theory, leading to two types of chain rules. We may then compute satisfaction values for configurations, taking advantage of Markov property, and obtain a total order between configurations in both cases. In the absence of available quantitative values, one may think of keeping the possibility degrees unspecified (which also preserves the ability of representing partial orders). This led us to propose a new graphical preference model based on possibilistic networks [3, 4], called  $\pi$ -Pref nets. In a  $\pi$ -Pref net, for each variable  $A_i \in V$ , for each instantiation  $u_i$  of  $Pa(A_i)$ , the preference order between the values of variable  $A_i$  is encoded by a local

conditional possibility distribution expressed by symbolic weights. A symbolic weight means a symbol representing a real number whose value is unspecified.

**Definition 10 ( $\pi$ -Pref nets).** A possibilistic preference network ( $\pi$ -Pref net) over a set of variables  $V = \{A_1, \dots, A_N\}$  is a possibilistic network, i.e., a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where each node  $A_i$  is associated with symbolic possibility distributions conditioned  $\forall u_i$ . It encodes the ordering between values  $a_i$  and  $a'_i$  in  $D(A_i)$  in each context  $u_i$ :

- If  $a_i \prec a'_i$  then  $\pi(a_i|u_i) = \alpha, \pi(a'_i|u_i) = \beta$  where  $\alpha$  and  $\beta$  are non-instantiated weights on  $(0, 1]$  called symbolic weights, and  $\alpha < \beta \leq 1$ ;
- If  $a_i \sim a'_i$  then  $\pi(a_i|u_i) = \pi(a'_i|u_i) = \alpha$  where  $\alpha \leq 1$ ;
- $\forall u_i, \exists a_i \in D(A_i)$  such that  $\pi(a_i|u_i) = 1$ .

In addition to the preferences encoded by a  $\pi$ -Pref net, additional a set  $\mathcal{C}$  of equality or inequality constraints between symbolic weights can be taken into account. Such constraints may represent, for instance, the relative strength of preferences associated to different instantiations of parent variables of the same variable. The satisfaction value of each configuration is computed as the product of symbolic weights using the chain rule associated with product-based conditioning, namely  $\pi(a_1, \dots, a_N) = \prod_{i=1}^N \pi(a_i|u_i)$  where  $u_i$  is an instantiation of  $Pa(A_i)$ . In spite of the symbolic nature of expressions just obtained, one may still compare some configurations thanks to properties of product and constraints (e.g.,  $\alpha < 1, \alpha \times \beta < \alpha$ , or, if  $\beta < \gamma \in \mathcal{C}, \alpha \times \beta < \alpha \times \gamma$ ). Obviously, some expressions may remain incomparable, then only a partial order is obtained.

*Example 7.* Let Figure 6 represent a  $\pi$ -Pref net over 3 variables  $V = \{A, B, C\}$  and  $\mathcal{C} = \{\delta_3 < \delta_1\}$  represent the set of constraints. Consider two configurations  $ab \neg c$  and  $\neg abc$ . Using the chain rule, we obtain their corresponding symbolic joint possibility expressions:  $\pi(ab \neg c) = 1 \times 1 \times \delta_1, \pi(\neg abc) = \alpha_1 \times 1 \times \delta_3$ . Since  $\delta_3 < \delta_1$ , we can deduce that  $ab \neg c \succ_{\pi Pref} \neg abc$ . However,  $ab \neg c \pm \neg ab \neg c$  since no constraint exists between  $\delta_1$  and  $\alpha$ . These two configurations remain non compared.

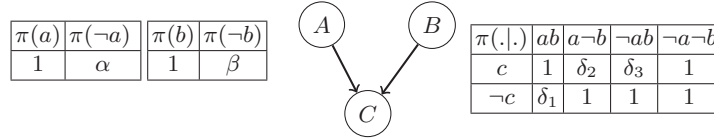


Fig. 6: An example of a  $\pi$ -Pref net

Each configuration  $\omega = a_1 \dots a_N$  can be associated with a vector  $\vec{\omega} = (\alpha_1, \dots, \alpha_N)$ , where  $\alpha_i = \pi(a_i|u_i)$  and  $u_i = \omega[Pa(A_i)]$ , e.g.,  $\neg abc = (\alpha, 1, \delta_3)$ . These vectors can be compared using symmetric Pareto and results are exactly as the product [4].

In  $\pi$ -Pref nets, it is clear that the best configurations are those having a joint possibility degree equal to 1, due to the normalization of conditional possibility distributions. We can always find an optimal configuration, starting from the root nodes where we choose each time the most or one of the most preferred value(s). At the end of the procedure, we get one or several configurations having a possibility equal to 1. This procedure is linear in the size of the network (using a forward sweep algorithm). Dominance

queries are answered by comparing the symbolic vectors. Indeed, each configuration is associated to a vector of symbolic weights where each component corresponds to the satisfaction degree of a node (as illustrated in the above example). Symbolic vectors are compared by symmetric Pareto ordering [4], which amounts to reordering them for applying the constraints between weights as much as possible. The order obtained this way is the same as the one yielded by the comparison of the product expressions. The complexity of dominance queries is  $O(N!)$ .

$\pi$ -Pref nets may be considered as being halfway between qualitative and quantitative models. This is due to the use symbolic weights. Indeed,  $\pi$ -Pref nets can be used in two ways: symbolically, or in an instantiated manner. The use of product, even in the symbolic case, adds a quantitative flavor. Moreover, symbolic possibilistic networks, using a logarithmic transformation, may be equivalently represented as symbolic OCF-nets [4]. Both  $\pi$ -Pref nets and OCF-nets share the same type of (Markovian) independence, and lead exactly to the same orderings.

Lastly, a  $\pi$ -Pref net can be equivalently represented by a possibilistic logic base [4]. In [13], attempts at representing a CP-net ordering using a possibilistic logic framework are reported. But, it may not be possible to build an exact logical representation due to the particular behavior of CP-nets (see Section 2.1). [13] suggests that symmetric Pareto and leximin orderings respectively lower and upper bound the CP-net ordering. It may have counterparts in graphical models based on the Markov property as OCF-nets.

## 5 Discussion and concluding remarks

Figure 7 presents a classification of the preferential graphical models surveyed. Roughly speaking, there are three classes: qualitative, quantitative and models that are halfway.

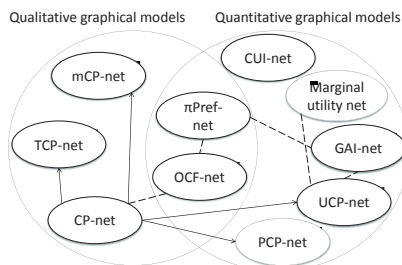


Fig. 7: Classification of preferential graphical models (Continuous arrows point to extensions of CP-nets and dashed lines are discussed later in the section)

A summary of the main differences and similarities between the models is given below. These models can be further compared in terms of the underlying independence relation (and expressiveness), and the ease of elicitation. Regarding the first issue, we distinguish three situations: (i) *Ceteris Paribus* independence shared by CP-nets, and its extensions. Models based on it are unable to express any possible ordering between configurations. UCP-nets can represent some total orderings, at the expense of constraints added on utilities; (ii) Generalized additive independence used in GAI-nets, is a weaker form of independence leading to an improved expressive power; (iii) Markov independence, used by  $\pi$ -Pref nets, OCF-nets and marginal utility nets. In contrast with

Model	CP-nets	TCP-nets	GAI-nets	UCP-nets	Marginal utility nets	$\pi$ Pref-nets
Graphical component	Variable	Variable	Cliques	Variable	Variable	Variable
Node:	Directed	Directed	Undirected	Directed	Directed	Directed
Edges:						
Preference table	Conditional pref. relation on variables	Cond. pref. relation + Importance relation	Utility functions	Conditional utility distribution	Conditional utility distribution	Conditional symbolic possibility distributions
Independence relation	<i>Ceteris Paribus</i>	<i>Ceteris Paribus</i>	Generalized Additive	<i>Ceteris Paribus</i> + GAI	Markovian	Markovian
Ordering	Partial	Partial	Total	Total	Total	Partial/ Total
<b>Queries Complexity</b>						
Optimization	Linear	Linear	Exponential	Linear	Unknown	Linear
Dominance	NP-complete to PSPACE	Unknown	Linear	Linear	Unknown	Linear to $O(N!)$

GAI, this kind of independence does not allow mutual dependencies between variables due to the acyclicity constraint. *Ceteris Paribus* and Markov independence lead to different completion principles. With *Ceteris Paribus*, pairs of compared partial configurations are completed with the same instantiation of the rest of the variables, while with Markov-based nets, at first one takes the best instantiation for all dependent variables, and, secondly, completes the other variables in the same manner in all possible ways.

Regarding elicitation, although quantitative models are convenient since providing total orderings, they are not easy to assess (any difference in values may lead to different orderings). In contrast, eliciting qualitative models is easier since it suffices to provide contextual preference ordering.  $\pi$ -Pref nets enable a progressive elicitation since we may add constraints between symbolic weights, or completely instantiate them.

Thanks to some resemblances between those models many transformations can be considered and are depicted by dashed lines in Figure 7. UCP-nets are a restriction of GAI-nets and a generalization of CP-nets. Indeed, a UCP-net structure can be transformed into a junction tree such that for each clique we sum up the local utilities of the variables belonging to it, just leading to a GAI net. However, due to the acyclic restriction of UCP-nets and the necessary, commitment with *Ceteris Paribus*, not any GAI-net can be represented by a UCP-net. Besides, when handled symbolically,  $\pi$ -Pref nets and marginal utility nets lead to the same orderings. Indeed comparing configurations is nothing but comparing vectors of weights. Therefore, product and addition make no difference on symbolic weights. Transformation from  $\pi$ -Pref nets to GAI-nets might also be considered since, as for Bayesian nets, possibilistic nets can be translated into junction trees. However, an important difference between these two settings lie in the meaning of values. Both utilities and possibility degrees express levels of satisfaction, but the latter are bounded. In GAI-nets, what really matters is the difference between utilities. Thus, representing the same information in  $\pi$ -Pref nets is not possible; one may only try to induce the same qualitative order between the configurations. The opposite transformation is not obvious. In fact, it requires a two level transformation. First, translating utilities to possibility degrees. Second, moving from a junction tree to a possibilistic network. This procedure was never studied in the literature.

As can be seen, the advantages of the different models are a matter of trade-offs. One may prefer one or another depending on the level of information available, the expressiveness needed for the situation at hand, and the time available for eliciting preferences. From a computational viewpoint, UCP-nets, instantiated  $\pi$ -Pref nets and OCF-nets are the less demanding. On the other hand, elicitation and construction might

be onerous for UCP-nets, GAI-nets and TCP-nets, while CP-nets and  $\pi$ -pref nets are easy to elicit. Getting a total order may also be considered as important. Thus, one may prefer models such as GAI-nets, OCF-nets and instantiated  $\pi$ -Pref nets in that respect.

## References

1. Arnborg, S., Corneil, D.G., Proskurowski, A.: Complexity of finding embeddings in ak-tree. *Journal on Algebraic Discrete Methods* 8(2), 277–284 (1987)
2. Ben Amor, N., Benferhat, S.: Graphoid properties of qualitative possibilistic independence relations. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Sys.* 13(1), 59–96 (2005)
3. Ben Amor, N., Dubois, D., Gouider, H., Prade, H.: Possibilistic networks: A new setting for modeling preferences. In: *Proc. SUM*. pp. 1–7 (2014)
4. Ben Amor, N., Dubois, D., Gouider, H., Prade, H.: Possibilistic conditional preference networks. In: *Proc. ESQUARU'13*. pp. 36–46 (2015)
5. Boutilier, C., Bacchus, F., Brafman, R.I.: UCP-networks: A directed graphical representation of conditional utilities. In: *Proc. UAI*. pp. 56–64. Morgan Kaufmann Publishers Inc. (2001)
6. Boutilier, C., Brafman, Ronen, I., Hoos, H., Poole, D.: Reasoning with conditional ceteris paribus preference statements. In: *Proc UAI*. pp. 71–80 (1999)
7. Boutilier, C., Brafman, R., Domshlak, C., Hoos, 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)
8. Brafman, R.I., Domshlak, C.: TCP-nets for preference-based product configuration. In: *The Forth Workshop on Configuration (ECAI)*. pp. 101–106 (2002)
9. Brafman, R.I., Domshlak, C., Shimonyl, S.: On graphical modeling of preference and importance. *J. Artif. Intell. Res.* pp. 389–424 (2006)
10. Brafman, R.I., Engel, Y.: Directional decomposition of multiattribute utility functions (2009)
11. Brafman, R.I., Engel, Y.: Decomposed utility functions and graphical models for reasoning about preferences. In: *Proc. AAAI* (2010)
12. Brazuinas, D., Boutilier, C.: Local utility elicitation in GAI models. *Proc. UAI* pp. 42–49 (2005)
13. Dubois, D., Prade, H., Touazi, F.: Conditional Preference-nets, possibilistic logic, and the transitivity of priorities. In: *Proc. SGAI*. pp. 175–184 (2013)
14. Eichhorn, C., Fey, M., Kern-Isberner, G.: CP-and OCF-networks—a comparison. *Fuzzy sets and Sys.* 298, 109–127 (2016)
15. Engel, Y., Wellman, M.P.: CUI networks: A graphical representation for conditional utility independence. *J. Artif. Intell. Res.* 31, 83–112 (2008)
16. Fishburn, P.C.: *Utility theory for decision making*. Tech. rep., DTIC Document (1970)
17. Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in CP-nets. *J. Artif. Intell. Res.* pp. 403–432 (2008)
18. Gonzales, C., Perny, P.: GAI networks for utility elicitation. *Proc. KR* 4, 224–234 (2004)
19. Gonzales, C., Perny, P.: GAI networks for decision making under certainty. In: *IJCAI–Workshop on Advances in Preference Handling* (2005)
20. Jensen, F.V., Lauritzen, S.L., Olesen, K.G.: Bayesian updating in causal probabilistic networks by local computations. *Proc. Comput. Stat. Quarterly* 4, 269–282 (1990)
21. Kaci, S., Lang, J., Perny, P.: Représentation des préférences. In: *Représentation des connaissances et formalisation des raisonnements*. pp. 181–214. Cépaduès éditions (2014)
22. Koriche, F., Zanuttini, B.: Learning conditional preference networks. *AI* 174(11), 685–703 (2010)
23. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmman (1988)