Preference Modeling with Possibilistic Networks and Symbolic Weights: A Theoretical Study

Nahla Ben Amor[1] and **Didier Dubois**[2] and **Héla Gouider**[3] and **Henri Prade**[4]

Abstract.
The use of possibilistic networks for representing conditional preference statements on discrete variables has been proposed only recently. The approach uses non-instantiated possibility weights to define conditional preference tables. Moreover, additional information about the relative strengths of these symbolic weights can be taken into account. The fact that at best we have some information about the relative values of these weights acknowledges the qualitative nature of preference specification. These conditional preference tables give birth to vectors of symbolic weights that reflect the preferences that are satisfied and those that are violated in a considered situation. The comparison of such vectors may rely on different orderings: the ones induced by the product-based, or the minimum-based chain rule underlying the possibilistic network, the discrimin, or leximin refinements of the minimum-based ordering, as well as Pareto ordering, and the symmetric Pareto ordering that refines it. A thorough study of the relations between these orderings in presence of vector components that are symbolic rather numerical is presented. In particular, we establish that the product-based ordering and the symmetric Pareto ordering coincide in presence of constraints comparing pairs of symbolic weights. This ordering agrees in the Boolean case with the inclusion between the sets of preference statements that are violated. The symmetric Pareto ordering may be itself refined by the leximin ordering. The paper highlights the merits of product-based possibilistic networks for representing preferences and provides a comparative discussion with CP-nets and OCF-networks.

1 Introduction

For more than a decade, representing preferences has attracted much interest in Artificial Intelligence. Preference models having a graphical basis are particularly appealing since they offer a compact representation, fit quite well with preference elicitation, and offer a basis for computation. Roughly speaking, one may distinguish between qualitative and quantitative settings. In quantitative models, such as Generalized Additive Independence networks (GAI nets) [19, 20, 21], representing preferences comes down to constructing a value function that enables us to compare all possible situations. However, decision-makers are rarely able to express their preferences directly in terms of numerical local value functions due to the considerable cognitive burden of determining accurate numerical values. Instead, qualitative models such as Conditional Preference networks (*CP-nets*) [5, 6] allow the representation of partially specified and contextually expressed preference relations. The problem is then to reconstruct, if not a value function, at least an order relation between all possible situations.

Recently, a new graphical model for preferences representation, called π-Pref nets, based on possibilistic networks, has been briefly outlined and studied in [3, 2]. Possibilistic networks use a graphical structure similar to the one of Bayesian nets, where conditioning either relies on a minimum-based, or on a product-based equation, depending on whether we are in a qualitative or in a numerical setting. Since their introduction, possibilistic networks have been applied to reasoning under uncertainty only. Both types of such networks have a direct logical counterpart in terms of possibilistic logic knowledge bases [4], where formulas are assigned degrees of necessity [13]. In contrast, sets of formulas weighted by probability values are generally not equivalent to a single probability distribution, hence cannot encode Bayesian nets. Logical encodings of Bayesian nets require other methods [8]. Conditional preference statements can be represented using product-based possibilistic networks on discrete variables. They use *non-instantiated* numerical possibility degrees (we call them *symbolic weights*) in the conditional preference tables, and possibly additional information about the relative strengths of symbolic weights can be taken into account. Such a symbolic model is situated halfway between quantitative and qualitative models. Indeed, it can be handled qualitatively if the symbolic weights remain non-instantiated, or quantitatively when instantiating them.

π-Pref nets and CP-nets share the same graphical structure and conceptual simplicity. CP-nets rely on the *Ceteris Paribus* principle, and may induce debatable priorities between decision variables [15]. π-Pref nets do not suffer from such a questionable behavior and leave complete freedom for stating relative priorities between variables. Ordering two possible situations in π-Pref nets amounts to comparing two vectors of symbolic weights reflecting the user's specifications that are satisfied and those that are violated.

One may think of different orderings for comparing these vectors, starting with the ones induced by the product-based, or the minimum-based chain rules underlying possibilistic networks, as well as the Pareto ordering, and refinements of previous orderings, such as the discrimin, the leximin, or the symmetric Pareto orderings. In this paper, we provide a thorough comparative study between these orderings in the case of symbolic weights, which departs from the numerical situation, in presence, or not, of additional constraints between symbolic weights.

Moreover, we also compare π-Pref nets with OCF-networks [17, 18, 22] that use integer additive penalty costs to define conditional preference tables. They appear to be similar to product-

[1] LARODEC Laboratory, University of Tunis, ISG, Tunisia, email: nahla.benamor@gmx.fr
[2] IRIT, University of Toulouse, France, email: dubois@irit.fr
[3] LARODEC Laboratory, University of Tunis, ISG, Tunisia, email: gouider.hela@gmail.com
[4] IRIT, University of Toulouse, France, & QCIS, University of Technology, Sydney, Australia, email: prade@irit.fr

based possibilistic networks. In [3], it was proved that such costs can be expressed in terms of possibility degrees. This suggests that the two graphical network representations are closely related. Numerical OCF-networks have been recently proposed to mimic the ordering of CP-nets in [16]. In this paper, we also discuss the possibility to induce CP-nets ordering using numerical, or symbolic π-Pref nets.

The paper is organized as follows. Section 2 provides a brief background on possibilistic networks, while Section 3 introduces possibilistic networks with symbolic weights as a way of representing preferences. Section 4 defines the different possible orderings we may think of for comparing vectors with symbolic components, and establishes that the product-based and the symmetric Pareto orderings always coincide in the presence of non-zero symbolic weights. Section 5 presents a thorough comparison of different possible orderings between symbolic vectors, including the case of additional constraints between the symbolic weights. Lastly, Section 6 provides a compative discussion between π-Pref nets, OCF-nets, and CP-nets.

2 Background on possibility theory

Let $V = \{A_1, \ldots, A_N\}$ be a set of N variables. Each variable A_i has a value domain $D(A_i)$. Elements $a_i \in D(A_i)$ denote values of A_i. $\Omega = \{\omega_1, \ldots, \omega_{|\Omega|}\}$ 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 (or a solution). It corresponds to a complete instantiation of the variables in V. If $U \subseteq V$, then $\omega[U]$ denotes the restriction of solution ω to variables in U.

We start by a brief refresher on possibility theory [11, 27] in the setting of uncertainty representation. It relies on the idea of a possibility distribution π, which is a mapping from a universe of discourse Ω to the unit interval $[0, 1]$, or to any bounded totally ordered scale. This possibilistic scale could be interpreted at least in two ways: a numerical interpretation when values must be the result of a clear measurement procedure, and an ordinal one when values only reflect a total preorder between the different interpretations. $\pi(\omega_i) = 0$ means that ω_i is fully impossible, while $\pi(\omega_i) = 1$ means that ω_i is fully possible. The possibility distribution π is normalized if $\exists\ \omega_i \in \Omega$ s.t. $\pi(\omega_i) = 1$. Given a normalized possibility distribution π, we can describe the uncertainty about the occurrence of an event $F \subseteq \Omega$ via a possibility measure $\Pi(F) = \sup_{\omega_i \in F} \pi(\omega_i)$ and its dual necessity measure $N(F) = 1 - \Pi(\bar{F}) = 1 - \sup_{\omega_i \notin F} \pi(\omega_i)$. Measure $\Pi(F)$ evaluates to which extent F is *consistent* with the knowledge represented by π while $N(F)$ evaluates at which level F is certainly implied by π. Conditioning in possibility theory is defined from the Bayesian-like equation $\Pi(F \cap E) = \Pi(F|E) \otimes \Pi(E)$, where $\otimes$ stands for the product in a quantitative setting (numerical), or for the minimum in a qualitative setting (ordinal).

Possibilistic networks [4, 1] are defined as counterparts of Bayesian networks [25] in the context of possibility theory. They share the same basic components, namely: (i) a *graphical component* which is a DAG (Directed Acyclic Graph) $\mathcal{G}= (V, E)$ where V is a set of nodes representing variables and E a set of edges encoding conditional (in)dependencies between them. (ii) a *valued component* associating a local normalized conditional possibility distribution to each variable $A_i \in V$ in the context of its parents (denoted by $Pa(A_i)$). The two definitions of possibilistic conditioning lead to two variants of possibilistic networks: in the numerical context, we can express product-based networks, while in the qualitative context, we only have min-based networks (also known as qualitative possibilistic networks). Given a possibilistic network, we can compute its encoded joint possibility distribution using the following chain rule:

$$\pi(A_1, \ldots, A_N) = \otimes_{i=1..N}\ \Pi(A_i \mid Pa(A_i)) \qquad (1)$$

where $\otimes$ is either the minimum, or the product operator $*$, depending on the semantic underlying it.

In the following, we advocate possibilistic networks for representing knowledge about preferences (rather than uncertain knowledge as it has been the case until now). Thus, $\pi(\omega)$ defines the level of satisfaction of ω, $\Pi(F)$ evaluates to what extent satisfying a constraint modeled by F is satisfactory and $N(F)$ evaluates to what extent this constraint is imperative.

3 Possibilistic networks for handling preferences

This section provides a generic definition of conditional preference possibilistic networks and, shows various ranking procedures to induce an ordering between the solutions. Then, we propose a comparison between these distinct induced orderings. Conditional preference statements can be associated to a graphical structure. In this approach, this graphical structure is understood as a possibilistic network where each node A_i is associated with a conditional possibility table used for representing the preferences.

Definition 1 (Preference Network) *A* preference network G *over a set* $V = \{A_1, \ldots, A_N\}$ *of decision variables is a DAG* $\mathcal{G}$ *where each node* $A_i \in V$ *is associated to local preference relations (preference table for short), such that to each instantiation* u_i *of its parents* $Pa(A_i)$ *is associated a complete preordering* $\succeq_{u_i}$ *of* $D(A_i)$*, directly provided by the user.*

In a possibilistic preference network, for each particular instantiation u_i of $Pa(A_i)$, the preference order between the values of A_i stated by the user will be encoded by a local conditional possibility distribution expressed by symbolic weights. By a symbolic weight, we mean a symbol representing a real number whose value is unspecified. We rely on symbolic weights in the absence of available numerical values. These weights may be instantiated totally or partially when possible.

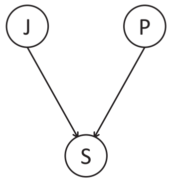
Definition 2 (Conditional Preference Possibilistic network) *A* possibilistic preference network (π-Pref net) ΠG *over a set* $V = \{A_1, \ldots, A_N\}$ *of decision variables is a preference network where each local preference relation at node* A_i *is associated with a* symbolic *conditional possibility distribution (π_i-table for short), encoding the ordering between the values of* A_i *such that:*

- *If* $a_i \prec_{u_i} a'_i$ *then* $\pi(a_i|u_i) = \alpha, \pi(a'_i|u_i) = \beta$ *where* α *and* β *are non-instantiated variables on (0, 1] we call* symbolic weights*, and* $\alpha < \beta \leq 1$*;*
- *If* $a_i \sim_{u_i} a'_i$ *then* $\pi(a_i|u_i) = \pi(a'_i|u_i) = \alpha$ *where* α *is a symbolic weight such that* $\alpha \leq 1$*;*
- *For each instantiation* u_i *of* $Pa(A_i)$, $\exists\ a_i \in D(A_i)$ *such that* $\pi(a_i|u_i) = 1$.

Let $\mathcal{C}_0$ *be the set storing the constraints existing between the symbolic weights introduced as above.*

In addition to the preferences encoded by a π-Pref net, additional constraints in $\mathcal{C}_1$ can be taken into account. Such constraints may represent, in particular, the relative strength of preferences associated to different instantiations of parent variables of the same variable. Let $\mathcal{C} = \mathcal{C}_0 \bigcup \mathcal{C}_1$. In the case one cannot infer any relation between two weights by transitivity, we consider them as incomparable.

Example 1 *Consider a preference specification about an evening dress over 3 decision variables* $V = \{J, P, S\}$ *standing for jacket, pants and shirt respectively, with values in* $D(J) = \{Red\ (j_r),\ Black\ (j_b)\}$, $D(P) = \{White\ (p_w),\ Black\ (p_b)\}$ *and* $D(S) = \{Black\ (s_b),\ Red\ (s_r),\ White\ (s_w)\}$. *The DAG is given by Figure 1 and the conditional possibility weights are given in Table 1. Preference statements* (s_1) *and* (s_2) *are unconditional. Note that the user is indifferent between the values of variable* S *in context* $u_j = j_r p_w$. *The constraints between symbolic weights inherent from the preference specification are represented by the set* $\mathcal{C}_0$ *such that* $\mathcal{C}_0 = \{(\delta_1 > \delta_2), (\theta_1 > \theta_2), (\lambda_1 > \lambda_2)\}$.



$(s_1)\ j_b \succ j_r$
$(s_2)\ p_b \succ p_w$
$(s_3)\ j_b p_b\!: s_b \succ s_r \succ s_w$
$(s_4)\ j_b p_w\!: s_w \succ s_b \succ s_r$
$(s_5)\ j_r p_b\!: s_r \succ s_b \succ s_w$
$(s_6)\ j_r p_w\!: s_b \sim s_r \sim s_w$

Figure 1: A preference network and its corresponding preference specification

$\pi(j_b)$	$\pi(j_r)$
1	α

$\pi(p_b)$	$\pi(p_w)$
1	β

$\pi(.\|.)$	$j_b p_b$	$j_b p_w$	$j_r p_b$	$j_r p_w$
s_b	1	θ_1	λ_1	1
s_r	δ_1	θ_2	1	1
s_w	δ_2	1	λ_2	1

Table 1: Possibilistic conditional preference tables

From now on, we assume the complete specification of conditional preferences, i.e., in each possible context, the user provides a complete preordering of the values of the considered variable in terms of strict preference or indifference.

These particular possibilistic networks have a logical counterpart [3, 4], namely symbolic possibilistic logic bases. In fact, in previous works [10, 15], possibilistic logic was advocated to represent symbolic preferences. However, beside the lack of a graphical structure, it is much less flexible than this symbolic graphical approach. Indeed, it supports only binary variables and associates to each variable exactly one symbolic weight. Precisely, a possibilistic logic encoding uses at most one symbolic weight per π_i-Table.

The ultimate aim of graphical representations of preference is to efficiently compare all possible solutions in Ω. Each possibility degree of a solution, computed from the possibilistic chain rule (1), expresses its satisfaction level. This leads to the following definition of the induced ordering.

Definition 3 (Preference ordering) *Given a symbolic possibilistic preference network* ΠG *and a set* $\mathcal{C}$ *of constraints between the symbolic weights, given two solutions* ω_i *and* ω_j *in* Ω, *let* $\pi_{\Pi G}(\omega_i)$ *(resp.* $\pi_{\Pi G}(\omega_j)$*) be the symbolic possibility degree of* ω_i *(resp.* ω_j*) computed by (1). Then,* $\omega_i \succeq_\otimes \omega_j$ *iff* $\pi_{\Pi G}(\omega_i) \geq \pi_{\Pi G}(\omega_j)$.

Other relations can be derived from $\succeq_\otimes$ as usual: $\omega_i \sim_\otimes \omega_j$ if and only if $\omega_i \succeq_\otimes \omega_j$ and $\omega_j \succeq_\otimes \omega_i$ (indifference); $\omega_i \succ_\otimes \omega_j$ if and only if $\omega_i \succeq_\otimes \omega_j$ but not $\omega_j \succeq_\otimes \omega_i$ (strict preference); $\omega_i \pm_\otimes \omega_j$ iff neither $\omega_i \succeq_\otimes \omega_j$ nor $\omega_j \succeq_\otimes \omega_i$ (non comparability). $\otimes$ stands for either the $\min$ or the *product* operator $*$. Note that $\otimes$ is associative, monotonic in the wide sense and 1 represents the identity element such that $1 \otimes \alpha = \alpha$.

Since we use symbolic weights, preference between some solutions cannot be established (as long as we do not instantiate all the symbolic weights). To each solution $\omega = a_1 \ldots a_N$ can be associated with a vector $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$, where $\alpha_i = \pi(a_i|u_i)$ and $u_i = \omega[Pa(A_i)]$. Vectors associated to the preference possibilistic network of Example 1 are represented by Table 2. Thus, comparing solutions amounts to comparing those vectors of symbolic weights, and the use of the chain rule is just one way of comparing solutions, among other ones as discussed in the next section.

	Symbolic vectors		
configurations	J	P	S
$j_b p_b s_b$	1	1	1
$j_b p_b s_r$	1	1	δ_1
$j_b p_b s_w$	1	1	δ_2
$j_b p_w s_b$	1	β	θ_1
$j_b p_w s_r$	1	β	θ_2
$j_b p_w s_w$	1	β	1
$j_r p_b s_b$	α	1	λ_1
$j_r p_b s_r$	α	1	1
$j_r p_b s_w$	α	1	λ_2
$j_r p_w s_b$	α	β	1
$j_r p_w s_r$	α	β	1
$j_r p_w s_w$	α	β	1

Table 2: Vectors associated to each configuration of Example 1

4 Symbolic weights

In Section 3, we have shown how to encode the preference specifications in a possibilistic network format. In this section we will present a number of partial order relations with the purpose to use them to generate a particular ordering over configurations.

Vectors of these weights, $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$ for instance, can be compared using different ordering procedures namely, Product, symmetric Pareto, Minimum or Leximin orderings. These procedures can be defined for partially ordered symbolic weights. They are defined as follows:

Definition 4 (Product) $\vec{\omega} \succeq_{prod} \vec{\omega'}$ *iff* $prod(\vec{\omega}) \geq prod(\vec{\omega'})$ *where* $prod(\vec{\omega}) = \prod_{i=1}^{N} \alpha_i$.

Definition 5 (Minimum) $\vec{\omega} \succeq_{\min} \vec{\omega'}$ *iff* $\min(\vec{\omega}) \geq \min(\vec{\omega'})$, *where* $\min(\vec{\omega}) = \min_{i=1}^{N} \alpha_i$.

Definition 6 (Pareto) $\vec{\omega} \succeq_{Pareto} \vec{\omega'}$ *iff* $\forall\, k,\ \alpha_k \geq \beta_k$.

Definition 7 (Symmetric Pareto) $\vec{\omega} \succeq_{SP} \vec{\omega'}$ *iff there exists a permutation* σ *of the components of* $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$, *yielding a vector* $\vec{\omega_\sigma} = (\alpha_1, \ldots, \alpha_N)$, *s.t.* $\vec{\omega_\sigma} \succeq_{Pareto} \vec{\omega'}$.

Definition 8 (Discrimin) *First, delete all pairs* (α_i, β_i) *from* $\vec{\omega}$ *and* $\vec{\omega'}$ *such that* $\alpha_i = \beta_i$. *Let* D *be the set of indices of components not deleted. Then,* $\vec{\omega} \succ_{discrimin} \vec{\omega'}$ *iff* $\min_{i \in D} \alpha_i > \min_{i \in D} \beta_i$.

Definition 9 (Leximin) *Let* $\vec{\omega_\sigma}$ *be the reordered vector* $\vec{\omega}$ *by permutation* σ *of its components.* $\vec{\omega} \succ_{leximin} \vec{\omega'}$ *iff* $\exists\, \sigma, \vec{\omega_\sigma} \succ_{discrimin} \vec{\omega'}$.

In the standard case of a totally ordered scale, the leximin order is defined by first reordering the vectors in an increasing way, and then applying the min order to the sub-parts of the reordered vectors without consideration of identical components. However, here we deal with symbolic possibility degrees between which the ordering can be unknown. In the extreme case, we may just know that $\alpha < 1$ for a weight α. Thus, reordering the vectors is no longer possible, and leximin must be defined as proposed above.

We need the concept of refinement of a strict preference relation.

Definition 10 (Refinement) *[9]. Let $\succ$ and $\succ$ be any two strict order relations on Ω. Then, we say that $\succ'$ refines $\succ$ iff:*

$$\forall\, \omega,\ \omega' \in \Omega,\ \omega \succ \omega' \Rightarrow \omega \succ' \omega'.$$

As shown in [9], in the instantiated case, leximin is a refinement of the symmetric ordering of Pareto. As well, the discrimin ordering refines the ordering induced by the minimum as well as the Pareto ordering, and is itself refined by leximin. Leximin refinements of min-based orderings can be very discriminant, as they would solve cases left pending by minimum orders. Moreover, product-based orderings refine symmetric Pareto orderings of vectors containing no zero components. As for product-based orderings, they can strongly differ (including preference reversal) from the min-based orderings. These refinement relationships are illustrated by Figure 2(a) (where each arrow $a \rightarrow b$ expresses that a refines b).

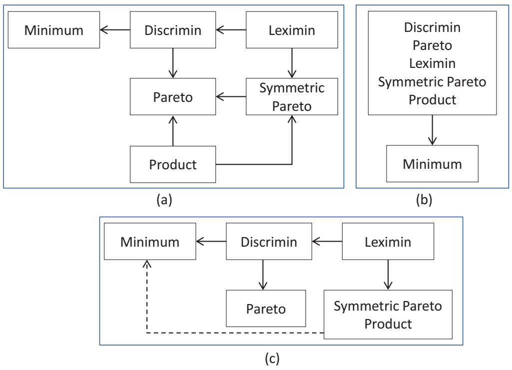


Figure 2: Refinements between orderings in numerical setting (a), symbolic setting without constraints (b), or with constraints (c)

The orderings do not behave in the same manner in the numerical case and in the symbolic case, as exemplified in the following.

Example 2 *Consider the vectors (α, β) and (γ, δ) where $\alpha < \beta$ and $\gamma < \delta$. If $\alpha < \gamma$ then $(\alpha, \beta) < (\gamma, \delta)$ for leximin and min, while according to symmetric Pareto these vectors are still incomparable. If $\alpha = \gamma < \delta < \beta$ then min considers (α, β) and (γ, δ) as equal, while we have $(\alpha, \beta) > (\gamma, \delta)$ with the product and the Leximin is unable to compare them. However, if $\alpha < \gamma < \delta < \beta$ then $(\alpha, \beta) < (\gamma, \delta)$ with the min while the product operator fails to order them.*

In the symbolic framework, it has been suggested in [3] that, when there is no constraint between symbolic weights in the vectors, the ordering induced by the product-based chain rule corresponds exactly to the a symmetric Pareto ordering. This result actually holds even in the presence of inequality constraints between symbolic weights.

Proposition 1 *Given any set of constraints $\mathcal{C}$ of the form $\alpha_i \geq \beta_j$ or $\alpha_i > \beta_j$ between symbolic weights:*
$\vec{\omega} \succ_{SP} \vec{\omega'}$ iff $\vec{\omega} \succ_{prod} \vec{\omega'}$ and $\vec{\omega} \succeq_{SP} \vec{\omega'}$ iff $\vec{\omega} \succeq_{prod} \vec{\omega'}$.

Proof: The proof is not straightforward. We proceed in several steps. First notice that the implications:
$\vec{\omega} \succ_{SP} \vec{\omega'}$ implies $\vec{\omega} \succ_{prod} \vec{\omega'}$
and $\vec{\omega} \succeq_{SP} \vec{\omega'}$ implies $\vec{\omega} \succeq_{prod} \vec{\omega'}$
are obvious since the product is symmetric and monotonically increasing. For the converse, we must basically show that if $\vec{\omega}$ is SP-incomparable with $\vec{\omega'}$ then they are also incomparable wrt the product ordering. We use several lemmas.

Lemma 1 *If Proposition 1 holds for a set of constraints $\mathcal{C}$, it holds a fortiori for any subset of constraints in $\mathcal{C}$.*

Indeed taking away constraints from $\mathcal{C}$ yields more freedom to the choice of values for the coefficients in order to ensure the non comparability of the symbolic product expressions associated to each vector. As a consequence of this lemma, the result should be established with the maximal amount of constraints, namely assuming a (non-trivial) complete pre-ordering of the symbolic coefficients appearing in the two vectors.

Lemma 2 ([7]) *Consider two symbolic vectors $\vec{\omega} = (\alpha_1, \ldots \alpha_N)$ such that $\mathcal{C}$ enforces $\alpha_1 \leq \ldots \leq \alpha_N$ and $\vec{\omega'} = (\beta_1, \ldots \beta_N)$. Let τ be a permutation of the components of $\vec{\omega'}$ such that $\beta_{\tau(1)} \leq \ldots \leq \beta_{\tau(N)}$ and $\vec{\omega'_\tau}$ the corresponding reordered vector. Then $\vec{\omega} \succ_{SP} \vec{\omega'}$ if and only if $\vec{\omega} \succ_{P} \vec{\omega'_\tau}$.*

In the totally ordered setting it gives a constructive way of expressing the SP ordering by applying the Pareto ordering to the increasingly reordered vectors.

Without loss of generality, due to Lemma 2, we can assume that vectors are increasingly ordered. Now we can try to prove that if $\vec{\omega}$ and $\vec{\omega'}$ are SP-incomparable then they are so for product. If they are SP-incomparable then there are $i \neq j$ such that $\alpha_i > \beta_i$ and $\alpha_j < \beta_j$. The most constrained case is when there is one constraint of the form $\alpha_i > \beta_i$, while all the other ones are of the form $\alpha_j < \beta_j$. In that case $\prod_{j \neq i} \beta_j > \prod_{j \neq i} \alpha_j$ and denoting by $\vec{\alpha_{-i}}$ the vector $\vec{\alpha}$ deprived of component i, we also have $\vec{\omega_{-i}} \succ_{SP} \vec{\omega'_{-i}}$.

Let us show that this strong prerequisite does not enforce an inequality between $\prod_{j=1}^N \beta_i$ and $\prod_{j=1}^N \alpha_j$. First, if α_i and β_j are very close, then $\prod_{j=1}^N \beta_j > \prod_{j=1}^N \alpha_j$. Now, for the reversed inequality, replace α_j by α_1 for $j < i$, and by α_i for $j > i$, β_j by β_i for $j < i$ and by β_n for $j > i$. The inequality $(\alpha_1)^{i-1} \cdot (\alpha_i)^{N-i+1} > (\beta_i)^i \cdot (\beta_N)^{N-i}$ is more demanding than the inequality $\prod_{j=1}^N \alpha_j > \prod_{j=1}^N \beta_j$. Let us show we can satisfy the former because $\alpha_i > \beta_i$, even if $\alpha_1 < \alpha_i, \beta_i < \beta_N$. To see it, we can write $\alpha_1 = \alpha_i / p$ and $\beta_n = q\beta_i$ with $p, q > 1$. It is easy to see that the inequality now writes $\frac{\alpha_i}{\beta_i} > p^{\frac{i-1}{N}} q^{\frac{N-i}{N}} > 1$. It is clear that we can set $p, q > 1$ and find $\alpha_i > \beta_i \in [0, 1]$ that verifies this inequality. $\square$

The consequence of this result is that the use of product of symbolic values in the approach is just one way of implementing the SP-ordering, whose essence is qualitative. In particular the compensatory effect, usually present in product of numbers (whereby, e.g. 0.5×0.5 is the same as 0.25×1) is absent from the SP-ordering, which creates cases for incomparability.

5 Comparison of orderings between vectors of symbolic weights

We pursue the comparison of the different orderings defined in the previous section, first in the absence, and then in the presence of additional constraint on symbolic weights.

5.1 Comparison of orderings without additional constraints on symbolic weights

In this section we will study the possible relations between the different orderings in the particular case where the constraints known between the symbolic weights are only the ones relative to the expression of conditional preferences, as allowed by Definition 2. Thus, a constraint of this kind focuses only on a unique vector component, and we have $\mathcal{C}_1 = \emptyset$.

Under this assumption, Pareto ordering and symmetric Pareto yield the same ordering. Indeed, for two vectors $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$ each symbolic weight $\alpha_i \neq 1$ of $\vec{\omega}$ can be only compared to the symbolic weight $\beta_i \neq 1$ of $\vec{\omega'}$. Thus, there is no need to permute components as the result would definitely be non comparable with another component weight since $\mathcal{C}_1 = \emptyset$. This is as well true for leximin and discrimin orderings since they coincide in this case. In fact, with this hypothesis, the difference between leximin and discrimin is that leximin deletes some components with value 1 which cannot affect the result of the final application of the min.

We first compare the different orderings induced by the use of product or minimum, depending on the chain rule applied to the possibilistic network. We will evaluate how well each option uses the information given to rank-order alternative solutions. We keep in mind that the product-based ordering and the symmetric Pareto ordering are the same.

Example 3 presents the different orderings induced by min-based and product-based chain rule for Example 1.

Example 3 *Let us consider the possibilistic preference network of Example 1. Using the chain rule, we obtain the symbolic vectors presented in Table 2. The product-based induced ordering without additional inequality constraint is represented by Figure 3.*

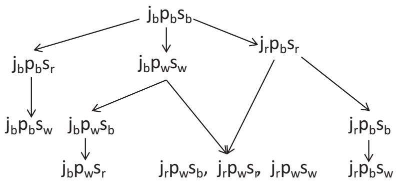


Figure 3: Possibilistic product-based order relative to Example 1

Now, if we use the min-based chain rule, we will not be able to compare many configurations as long as no other constraint is added. In fact, the only strict ordering information we can get at that stage is that $j_bp_bs_b > j_bp_bs_r > j_bp_bs_w$, $j_bp_bs_b > j_bp_ws_w$ and $j_bp_bs_b > j_rp_bs_b$. Otherwise, we only get at best weak inequalities ; for example $j_bp_ws_w$ and $j_rp_ws_b$, since $\pi_{\min}(j_bp_ws_b) = \min(\alpha, \beta) \leq \pi_{\min}(j_rp_ws_w) = \beta$. Figure 4 depicts this min-based ordering.

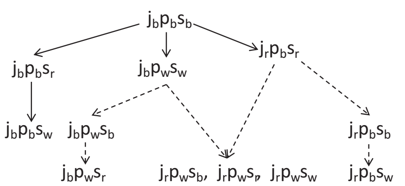


Figure 4: Possibilistic minimum-based order relative to Example 1

Symmetric Pareto on symbolic vectors may have a discriminating power greater than the one of the minimum operator, in the sense that $\alpha \cdot \beta < \alpha$, while we only have $\min(\alpha, \beta) \leq \alpha$. Clearly, when dealing with instantiated numerical values both product and minimum lead to total orders that can contradict each other: for instance $0.1 \cdot 0.9 > 0.2 \cdot 0.2$, while with the min we get $\min(0.1, 0.9) < \min(0.2, 0.2)$.

Proposition 2 $\omega \sim_{SP} \omega' \Leftrightarrow \omega \sim_{min} \omega'$.

Proof: Trivial as it compares the same sets of weights. □

Indeed, if equalities are found between every pair of the same index then the two vectors contain the same symbolic weights. For instance, we have $j_rp_ws_b \sim_{SP} j_rp_ws_r$ (resp . $j_rp_ws_b \sim_{\min} j_rp_ws_r$) where $\overrightarrow{j_rp_ws_b} = \overrightarrow{j_rp_ws_r} = (\alpha, \beta, 1)$.

Proposition 3 $\omega \pm_{SP} \omega' \Leftrightarrow \omega \pm_{\min} \omega'$.

Proof: Let us consider two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. If $\omega \pm_{SP} \omega'$ which means $\omega \pm_{Pareto} \omega'$, then we are in one of the following cases: either $\exists\, i$, s.t. $\alpha_i \pm \beta_i$ or $\exists\, i$, s.t. $\alpha_i < \beta_i$ and $\exists\, j$, s.t. $\alpha_j > \beta_j$. Besides, the only case where minimum is able to compare is when $\forall\, i$, $\alpha_i \geq \beta_i$. It is not the case here, then $\omega \pm_{SP} \omega' \Rightarrow \omega \pm_{\min} \omega'$. For the converse, if $\vec{\omega} \pm_{\min} \vec{\omega}$ then obviously $\vec{\omega} \pm_{SP} \vec{\omega}$ since we would not have $\forall\, i$, $\alpha_i \geq \beta_i$. □

Many cases can be identified on Example 1 and the min-based (resp. product-based) ordering presented by Figure 4 (resp. Figure 3). For instance, we have $j_rp_bs_r \pm_{SP} j_bp_ws_w$ (resp . $j_rp_bs_r \pm_{\min} j_bp_ws_w$). Moreover, using symbolic weights, symmetric Pareto and minimum provide consistent orderings in the sense that:

Proposition 4 *If* $\omega \succ_{SP} \omega' \Rightarrow \omega \succeq_{min} \omega'$.

Proof: Let us consider two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. If $\omega \succ_{SP} \omega'$ then $\vec{\omega} \succ_{Pareto} \vec{\omega'}$ more precisely $\forall\, i$, $\alpha_i \geq \beta_i$ and $\exists\, j$, $\alpha_j > \beta_j$ where $i, j \in [1, \ldots, N]$. From Proposition 3 we can deduce that it is impossible to have $\vec{\omega} \pm_{min} \vec{\omega'}$ since it is equivalent to having $\vec{\omega} \pm_{SP} \vec{\omega'}$. Moreover, it is impossible to have $\vec{\omega} \sim_{min} \vec{\omega'}$ because according to Proposition 2 this would mean that $\vec{\omega} \sim_{SP} \vec{\omega'}$. Consequently, since $\forall i$ we have $\alpha_i \geq \beta_i$, we can only have $\min(\alpha_1, \ldots, \alpha_N) \geq \min(\beta_1, \ldots, \beta_N)$. This is due the fact that the min function is a non decreasing function. □

For instance, one strict preference pattern is $j_bp_bs_b > j_bp_bs_r > j_bp_bs_w$ induced by the min chain rule on Example 1 is found on Figure 4 that depicts the preference relation induced by the product chain rule. The rest of preferences are preferences in the wide sense ($\succeq$), for example, $j_rp_bs_r \succeq j_rp_bs_b$.

Proposition 5 $\omega \succ_{\min} \omega' \Rightarrow \omega \succ_{SP} \omega'$.

Proof: Let us consider two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. If $\omega \succ_{\min} \omega'$ then $\forall\, i$ such that $\alpha_i \neq 1$ or $\beta_i \neq 1$, we have $\alpha_i > \beta_i$. Thus clearly $\omega \succ_{SP} \omega'$. □

This indicates that $\succ_{\min}$ is a strong form of Pareto, namely, $\omega \succ_{\min} \omega' \Leftrightarrow \forall i$, either $\beta_i \neq 1$ and $\alpha_i > \beta_i$ or $\alpha_i = \beta_i = 1$. Thus, the symmetric Pareto is a refinement of the minimum-based ordering. In Example 1 we can see that $j_bp_ws_b \succeq_{\min} j_bp_ws_r$, while we have a strict order with the symmetric Pareto (equivalently, the product-based) ordering $j_bp_ws_b \succ_{SP} j_bp_ws_r$ and we have $j_bp_bs_r \succ_{SP|\min} j_bp_bs_w$.

Proposition 6 $\omega \succeq_{\min} \omega' \Rightarrow \omega \succeq_{SP} \omega'$.

Proof: It immediately follows from $\omega \succ_{\min} \omega' \Rightarrow \omega \succ_{SP} \omega'$ (Prop. 5) and from $\omega \sim_{min} \omega' \Rightarrow \omega \sim_{SP} \omega'$ (by Prop. 2). □

When there is no additional constraint, i.e., $\mathcal{C}_1 = \emptyset$, Pareto and discrimin orderings yield the same ordering:

Proposition 7 *Pareto and discrimin coincide on vectors when* $\mathcal{C}_1 = \emptyset$

Proof: Let us consider two vectors $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$, where a symbolic weight α_i of $\vec{\omega}$ may only be compared to the corresponding symbolic weight β_i of $\vec{\omega'}$ (if there is a relevant constraint in $\mathcal{C}_0$). Three cases arise:

- $\vec{\omega} \succ_{Pareto} \vec{\omega'}$ iff $\vec{\omega} \succ_{discrimin} \vec{\omega'}$: ($\Rightarrow$) if $\vec{\omega} \succ_{Pareto} \vec{\omega'}$ then $\vec{\omega} \succeq_{\min} \vec{\omega'}$. This means that $\min(\vec{\omega}) \geq \min(\vec{\omega'})$ and since discrimin deletes all equalities $\alpha_i = \beta_i$. Thus we will have $\forall i \in D$ $\min_{i\in D} \alpha_i > \min_{i\in D} \beta_i$ s.t. D is the set of component indexes not deleted. Therefore, $\vec{\omega} \succ_{discrimin} \vec{\omega'}$. ($\Leftarrow$) Since discrimin deletes only weights where $\alpha_i = \beta_i$ and never strict comparisons, then after the deletion process we only have constraints such that $\alpha_j > \beta_j$, which means that the strict order is the same as Pareto ordering.
- $\vec{\omega} \sim_{Pareto} \vec{\omega'}$ iff $\vec{\omega} \sim_{discrimin} \vec{\omega'}$. Obvious.
- $\vec{\omega} \pm_{Pareto} \vec{\omega'}$ iff $\vec{\omega} \pm_{discrimin} \vec{\omega'}$: ($\Rightarrow$) if $\vec{\omega} \pm_{Pareto} \vec{\omega'}$ we have $\min(\vec{\omega}) \pm \min(\vec{\omega'})$ (by Proposition 3), and discrimin can only delete equalities, then the vectors remain non comparable with discrimin. Thus $\vec{\omega} \pm_{discrimin} \vec{\omega'}$. ($\Leftarrow$) if $\vec{\omega} \pm_{discrimin} \vec{\omega'}$ then we have not $\forall i \in D$ $\alpha_i < \beta_i$ where D is the set of component indexes not deleted. Thus we have $\vec{\omega} \pm_{Pareto} \vec{\omega'}$. □

Consequently from Proposition 7, we can derive that $\succ_{leximin} \Leftrightarrow \succ_{discrimin} \Leftrightarrow \succ_{Pareto} \Leftrightarrow \succ_{SP}$. Relations between the different orderings are depicted by Figure 2(b) (inside each box, relations are equivalent). This indicates a collapse of many notions when no additional constraints between symbolic weights applicable to different components exist.

5.2 Comparison of orderings with additional constraint on symbolic weights

As already mentioned, constraints between symbolic weights, beside those induced from the preference specification, can be added when available. In this section we will study the relations between the different ordering relations in the presence of such constraints. First, we will see that the refinement relations that exist in the case of numerical values remain valid.

Proposition 8 $\vec{\omega} \succ_{Pareto} \vec{\omega'} \Rightarrow \vec{\omega} \succ_{SP} \vec{\omega'}$.

Proof: It suffices to consider the permutation σ as the identity. □

Minimum-based ordering suffer from a "drowning" effect. Ways to overcome this problem are the discrimin or leximin orderings. In the numerical case, these latter are refinements of the min-based ordering [9]. We will prove that this is still the case in the symbolic framework. More formally:

Proposition 9 $\vec{\omega} \succ_{\min} \vec{\omega'} \Rightarrow \vec{\omega} \succ_{disrimin} \vec{\omega'}$,
$\vec{\omega} \succ_{disrimin} \vec{\omega'} \Rightarrow \vec{\omega} \succ_{leximin} \vec{\omega'}$.

Proof: Let us consider two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. If $\min(\vec{\omega'}) < \min(\vec{\omega})$, then $\exists\ \beta_i$ s.t. $\beta_i < \alpha_1, \ldots, \alpha_N$ and $\beta_i \neq \alpha_1, \ldots, \alpha_N$. This symbolic weight β_i cannot be eliminated in the deletion process of discrimin nor leximin. Thus, $\vec{\omega} \succ_{\min} \vec{\omega'} \Rightarrow \vec{\omega} \succ_{disrimin} \vec{\omega'}$ and $\vec{\omega} \succ_{\min} \vec{\omega'} \Rightarrow \vec{\omega} \succ_{leximin} \vec{\omega'}$. Besides, it is clear that leximin still refines discrimin since it suffices to consider the permutation σ processed by leximin as the identity. □

Example 4 *Let us consider the possibilistic preference network of Example 1. Let us consider some additional constraints such that* $\mathcal{C}_1$ *includes* $(\alpha < \beta)$, $(\beta = \lambda_1)$, $(\lambda_1 < \theta_1)$. *Thus,* $\mathcal{C} = \{(\delta_1 > \delta_2), (\theta_1 > \theta_2), (\lambda_1 > \lambda_2), (\alpha < \beta), (\beta = \lambda_1), (\lambda_1 < \theta_1)\}$. *Let us take the two configurations* $j_b p_w s_b$ *and* $j_r p_b s_b$ *such that* $\overrightarrow{j_b p_w s_b} = (\beta, \theta_1)$ *and* $\overrightarrow{j_r p_b s_b} = (\alpha, \lambda_1)$. *We can see that* $\overrightarrow{j_b p_w s_b} \succ_{\min|discrimin|leximin} \overrightarrow{j_r p_b s_b}$.

If we consider only partially ordered symbolic weights, leximin may lead to non comparability when discrimin or minimum considers two configurations equal. Thus, leximin ordering will sometimes lead to a partial ordering. This can be illustrated by the following example:

Example 5 *Let us consider the same two vectors of example 4,* $\overrightarrow{j_b p_w s_b} = (\beta, \theta_1)$ *and* $\overrightarrow{j_r p_b s_b} = (\alpha, \lambda_1)$. *We assume the set of constraints* $\mathcal{C} = \{(\delta_1 > \delta_2), (\theta_1 > \theta_2), (\lambda_1 > \lambda_2)(\lambda_1 < \alpha), (\beta = \lambda_1), (\beta < \theta_1)\}$, *then* $\overrightarrow{j_b p_w s_b} \sim_{\min|discrimin} \overrightarrow{j_r p_b s_b}$ *while* $\overrightarrow{j_b p_w s_b} \pm_{leximin} \overrightarrow{j_r p_b s_b}$. *Now if we suppose that* $\overrightarrow{j_b p_w s_b} = (\theta_1, \beta)$, *then* $\overrightarrow{j_b p_w s_b} \sim_{\min} \overrightarrow{j_r p_b s_b}$ *while* $\overrightarrow{j_b p_w s_b} \pm_{discrimin|leximin} \overrightarrow{j_r p_b s_b}$.

Let us now compare the discrimin and the Pareto orderings. Then the leximin and the symmetric Pareto orderings will be in a similar relation. Besides, there is no relation between the discrimin and the symmetric Pareto orderings when $\mathcal{C}_1 \neq \emptyset$. Indeed there are situations where discrimin can compare two vectors and the symmetric Pareto cannot (e.g., if we only know that the component of one vector is smaller than all the other components of the two vectors), and situations where symmetric Pareto can compare and discrimin cannot (e.g., $\vec{\omega} = (\alpha_1, \alpha_2, \alpha_3)$ and $\vec{\omega'} = (\beta_1, \beta_2, \beta_N)$ and $\mathcal{C} = \{\alpha_1 > \beta_1, \alpha_2 > \beta_3, \alpha_3 > \beta_2\}$.

Proposition 10 $\vec{\omega} \succ_{Pareto} \vec{\omega'} \Rightarrow \vec{\omega} \succ_{discrimin} \vec{\omega'}$.

Proof: Let us consider two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. By definition, if $\vec{\omega} \succ_{Pareto} \vec{\omega'}$ then $\forall i,\ \alpha_i \geq \beta_i$ and $\exists j,\ \alpha_i > \beta_i$. Let $\vec{\omega_*}$ (resp. $\vec{\omega'_*}$) denote the vector induced after deleting all vector components such that $\alpha_i = \beta_i, \forall i \in N$. Then, $\forall i \in D$, such that D is the set of the remaining vector component indexes, we have $\alpha_i > \beta_i$. This means that $\exists \beta_j \in \vec{\omega'_*}$ such that $\beta_j < \min(\vec{\omega_*})$. Therefore, $\vec{\omega} \succ_{\min} \vec{\omega'}$. Since discrimin refines minimum (Proposition 9), hence Proposition 10. □

From Proposition 10 we can derive that symmetric Pareto and leximin lead to consistent orderings. Moreover, each time when symmetric Pareto succeeds to order two configurations, discrimin will induce, if not the same ranking, at most non comparability.

Let us compare the minimum based-ordering and the product-based ordering (equivalently, SP). It is clear that we have:

Proposition 11 $\omega \sim_{SP} \omega' \Rightarrow \omega \sim_{min} \omega'$.

Proof: Assume two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. If $\omega \sim_{SP} \omega'$ then $\omega \sim_{Pareto} \omega'_\sigma$. Thus, $\forall\ i, \alpha_i = \beta_{i\sigma}$, where $i \in [1 \ldots N]$. Therefore, $\min(\beta_1, \ldots, \beta_N) = \min(\alpha_1, \ldots, \alpha_N)$. Hence the product ordering equalities are always found in min-based ordering. □

Equalities between solutions in product-based ordering may appear when one assumes equalities between symbolic weights associated to the same nodes and the same context or to symbolic weights of different nodes. This is unlike min-based ordering where it always considers the most important constraint violated, more precisely, having the smallest symbolic weight. Hence, in min-based orderings equalities appear when two solutions violate the same preference with the highest priority compared to the set of the other violated preferences.

Proposition 12 shows that symmetric Pareto is a special kind of refinement of the min-based ordering. Indeed:

Proposition 12 *If* $\omega \succ_{\min} \omega'$ *we may either have* $\omega \pm_{SP} \omega'$ *or* $\omega \succ_{SP} \omega'$.

Proof: Let us consider two solutions ω and ω' such that $\vec{\omega} = (\alpha_1, \ldots, \alpha_N)$ and $\vec{\omega'} = (\beta_1, \ldots, \beta_N)$. Indeed, from Proposition 11, if $\omega \sim_{SP} \omega'$ then $\omega \sim_{\min} \omega'$. Moreover, if $\omega \prec_{SP} \omega'$ then by the definition we have $\forall i$, $\alpha_i \leq \beta_i$, thus $\min(\alpha_1, \ldots, \alpha_N, \beta_i, \ldots, \beta_N) \in \{\alpha_i, \ldots, \alpha_N\}$, this proves that we cannot have $\omega \succ_{\min} \omega'$ in this case. Hence a contradiction, and Proposition 12 follows. □

Relations between the different orderings can be illustrated by Figure 2(c). Leximin ordering refines symmetric Pareto ordering, which in its turn refines Pareto ordering. Moreover, leximin refines discrimin and both are refinements of minimum ordering. It is important to notice that, in contrast with the numerical setting, minimum and leximin orderings may lead to non comparability and thus to partial orderings. Besides, symmetric Pareto still refines the minimum ordering, but in a wider sense since symmetric Pareto may yield non comparability when minimum succeeds in comparing (this relation is represented in Figure 2(c) by a dotted line).

One extreme case is when assuming a total preorder between the symbolic weights. In that case, leximin and minimum orderings are total. However, in the presence of such constraints, symmetric Pareto may still lead to non comparability. Indeed, the only case, where symmetric Pareto leads to a total ordering is when there are constraints between *subsets* of symbolic weights (corresponding to the comparison of subproducts). Thus, the relationships between the different orderings are the same as in the numerical setting except for the product and symmetric Pareto orderings, which are the same as previously proved.

6 π-Pref nets vs. other preference graphical models

In this section we compare and discuss existing relationships between π-Pref nets and two models that are related to them in some sense, namely, CP-nets and OCF-nets. The first one shares the same preference specification and graphical structure, while the second is based on an additive structure which parallels the one of π-Pref nets.

6.1 π-Pref nets vs. CP-nets

CP-nets can be viewed as a qualitative counterpart of Bayesian networks based on the *Ceteris Paribus* preferential independence relation. To each variable we associate a table representing the local preferences on its domain values in the context of its parents. The induced order is often referred to as a *Ceteris Paribus* preference order i.e. one partial outcome is preferred to another everything else being equal. Formally, this preference independence is defined as follows:

Definition 11 (Preference Independence) *Let V be the set of variables and W be a subset of V. We say that W is preferentially independent of its complement $Z = V \setminus W$ iff for any instantiations, z, z', w, w' we have:*

$$(w, z) \succ (w', z) \Leftrightarrow (w, z') \succ (w', z') \qquad (2)$$

This form of independence clearly simplifies the preference elicitation process [23, 24]. However, it can only represent a part of the preferences that a user may express. The following Example 6 represents a simple preference problem that CP-nets fail to represent.

Example 6 *Let us consider two binary variables A and B standing respectively for "vacations" and "good weather". Suppose that we have the following preference ordering: $ab \succ \neg a\neg b \succ a\neg b \succ \neg ab$. We observe that this complete preorder cannot be represented by a CP-net. In fact, given two variables we can define two possible structures: either A depends on B or conversely, both of them are unable to capture this order in the CP-net setting. This is due to the fact that in both structures we have a reversal of the Ceteris Paribus preferences. However, such preferences can be represented by a joint possibility distribution such that: $\pi(ab) > \pi(\neg a\neg b) > \pi(a\neg b) > \pi(\neg ab)$. Thus, we have $\top : a \succ \neg a$, $a : b \succ \neg b$ and $\neg a : \neg b \succ b$. It corresponds to a network with two nodes with their corresponding conditional possibility distributions: $\Pi(a) = 1$, $\Pi(\neg a) = \alpha$, $\Pi(b|a) = 1$, $\Pi(b|\neg a) = \gamma$, $\Pi(\neg b|a) = \beta$ and $\Pi(\neg b|\neg a) = 1$. This yields $\pi(ab) = 1 > \pi(\neg a\neg b) = \alpha > \pi(a\neg b) = \beta > \pi(\neg ab) = \alpha\gamma$ taking $\alpha > \beta$ and $\beta = \gamma$.*

From this simple example, we can see that π-Pref nets and CP-nets do not share the same form of preference independence. Although both graphical networks are syntactically based on the same preference statements, they are semantically handled in different ways. More precisely, orderings in CP-nets are induced from *Ceteris Paribus* and transitivity, while orderings in π-Pref nets are built using the chain rule and conditional preferences.

Indeed, if we consider the two preference (in)dependencies closely, we can notice that both have somehow contrasting properties. Let $desc(A)$ be the set of node A descendants and let $ndesc(A) = V \setminus desc(A) \setminus Pa(A)$ be the set of A non descendants, their possible instantiations are denoted by d and n respectively. The conjunction of instantiations is denoted by xy such that $X \cap Y = \emptyset$ and $X, Y \subseteq V$. Let us consider a preference statement: $u : a_1 \succ a_2$ where u an instantiation of $Pa(A)$ and $D(A) = \{a_1, a_2\}$. In CP-nets setting and based on the *Ceteris Paribus* independence, we can deduce that $ua_1dn \succ ua_2dn$. Aside the instantiations of A, the rest of the variables have the same instantiation. In contrast, the same preference statement is handled differently by possibilistic networks and means that $\pi(a_1|u) > \pi(a_2|u)$. Therefore, we have $\pi(a_1|un) > \pi(a_2|un')$ thanks to the Markov properties of possibilistic networks, namely, each node is independent from its non-descendants in the context of it parents. Thus, in contrast with CP-nets, the preference is still preserved even if some variables, precisely, $ndesc(A)$ are configured differently. Moreover, we can see that based on *Ceteris Paribus* independence we have $desc(A)$ instantiated similarly in both configurations, thus independently of A, which cannot be the case with possibilistic networks since $desc(A)$ depends on the instantiation of A. This is illustrated by the next example.

Example 7 *Figure 5 represents a possibilistic network (1) and a CP-net (2) induced from the same preference specification. If we consider the preference statement at node C, based on (2) and from the preference statement $\neg a : \neg c \succ c$, we can deduce that $\neg a\neg b\neg cd \succ_{CP} \neg a\neg bcd$, $\neg a\neg b\neg c\neg d \succ_{CP} \neg a\neg bc\neg d$, $\neg ab\neg c\neg d \succ_{CP} \neg abc\neg d$ and $\neg ab\neg cd \succ_{CP} \neg abcd$. Indeed, we can deduce as many comparisons as the number of possible configurations of the variables other than C and A, namely B and D. However, from the same statement, based on (1), we can deduce that $\neg a\neg b\neg c\neg d \succ_{\pi} \neg a\neg bcd$ and $\neg ab\neg c\neg d \succ_{\pi} \neg abcd$. This is due the fact that node B is independent of node C in the context of its parent A. Thus, the preference relation holds no matter the instantiation of B. Node D depends on C, thus, based on the context, we choose each time the best values for C.*

Therefore, we can deduce that the main difference between the two frameworks is in the completion principle underlying them. In fact, CP-nets complete a partial preference statement with the same instantiation of the rest of the variables. However, a π-Pref net, in a

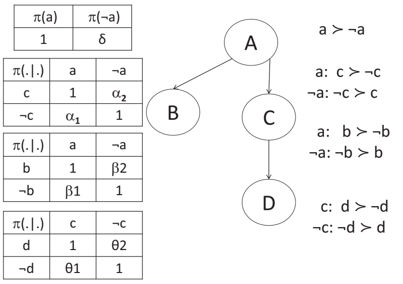


Figure 5: A π-Pref net (1) and a CP-net (2) modeling of the same preference specification

first step considers the best instantiation for all the dependent variables, and, in a second step, completes the rest in all possible ways.

In [10, 15], attempts at representing a CP-net ordering using a possibilistic logic framework are reported. However, authors indicate that it may not be possible to build an exact logical representation due some paradoxical behavior of CP-nets. Besides, they show that Symmetric Pareto and Leximin orderings respectively lower and upper bound the CP-net ordering. These results can be exploited with a π-Pref-net since it is the graphical counterpart of a symbolic possibilistic logic base [3].

6.2 π-Pref nets vs. OCF-nets

Ordinal Conditional Functions (OCF) [26] are an uncertainty representation framework very close to possibility theory [14]. OCF-nets may also offer a semi-quantitative graphical model for preference modeling [16]. OCF-nets obey Markov property as Bayesian networks (and possibilistic networks). Indeed, they have the same structure and carry the same conditional independence namely, each node is independent from its descendant in the context of its parents. This strong resemblance raises the question of a possible transformation between OCF-nets and π-Pref nets.

Formally, an OCF-net κG has two components: (i) a graphical component, a directed graph $\mathcal{G} = (V, E)$ where V denotes the set of nodes and E denotes the set of edges representing the preferential dependencies; (ii) a quantitative component: each variable $A_i \in V$ is associated to a normalized[5] conditional rank, a non-negative integer $\kappa(A_i|u_i)$, where u_i is an instantiation of the parents $Pa(A_i)$ of A_i.

The OCF relative to a solution ω, denoted by $\kappa(\omega)$ is the sum of the elementary ranks of the conditional rank tables such that:

$$\kappa(\omega) = \sum_{i=1}^{N} \kappa(A_i|u_i) \qquad (3)$$

This expression parallels the product-based possibilistic chain rule, where weights are combined by the product operator. The best solution in an OCF-net has a cost equal to 0, while in the possibilistic framework it has a possibility degree equal to 1. However, this preference network with a rank interpretation has a close relationship with product-based π-Pref nets. Indeed, the cost of a solution induced by an OCF-net corresponds actually to a transformation of the possibility degree computed from a π-Pref net.

In [12], it was pointed out that the set-function $\pi_\kappa(A_i) = 2^{-\kappa(A_i)}$ is a possibility measure. The converse holds to some extent insofar as if $\pi(A_i) = \alpha$, the values $\kappa(A_i) = -\log_2(\alpha)$ are integer rank weights. However, we can also extend the OCF framework to positive reals. Up to this proviso, the ordering induced by the product-based chain rule of π-pref nets is the same as the order induced by the corresponding rank function. In [3], it was proposed to use this transformation at the symbolic level, yielding a symbolic additive counterpart to π-pref nets.

Clearly, $\pi_{\Pi G}(\omega) = \alpha_1 \cdot \ldots \cdot \alpha_N \Rightarrow \kappa_{\mathcal{K}G}(\omega) = -(\log_2(\alpha_1) + \ldots + \log_2(\alpha_N))$ and $\kappa_{\kappa G}(\omega) = (\alpha_1 + \ldots + \alpha_N) \Rightarrow \pi_{\Pi G}(\omega) = 2^{-\alpha_1} \cdot \ldots \cdot 2^{-\alpha_N}$). Thus, after the logarithmic transformation, OCF-nets yield the same ordering on configurations as π-Pref nets. Note that π-pref nets with products cannot always be turned into OCF-nets with integer values. However, OCF-nets can be turned into π-pref nets with products.

Example 8 *Let us consider the following conditional rank tables corresponding to an OCF-net of two binary variables A and B: $\kappa(a) = 3$, $\kappa(\neg a) = 0$, $\kappa(b|a) = 0$, $\kappa(b|\neg a) = 2$, $\kappa(\neg b|a) = 1$ and $\kappa(\neg b|\neg a) = 0$. This yields $\kappa(\neg a \neg b) = 0 < \kappa(\neg ab) = 2 < \kappa(ab) = 3 < \kappa(a \neg b) = 4$. Thus we have $\neg a \neg b \succ_{\kappa G} \neg ab \succ_{\kappa G} ab \succ_{\kappa G} a \neg b$. The transformation from this OCF-net to a numeric π-Pref net leads to the following possibilistic conditional tables: $\pi(a) = 0.125$, $\pi(\neg a) = 1$, $\pi(b|a) = 1$, $\pi(b|\neg a) = 0.25$, $\pi(\neg b|a) = 0.5$ and $\pi(\neg b|\neg a) = 1$ which yields $\pi(\neg a \neg b) = 1 > \pi(\neg ab) = 0.25 > \pi(ab) = 0.125 > \pi(a \neg b) = 0.0625$. Clearly the two models lead to the same ordering after this transformation.*

Until now, OCF-nets have been used for dealing with numerical values only. However, the transformation of a *symbolic* possibilistic network leads to a *symbolic* OCF-net. Thus, the application of the different ordering relations defined above lead exactly to the same orderings induced by possibilistic networks. In fact, summation and product are handled similarly when we work in a symbolic setting.

Recently, numerical OCF-nets have been shown to "mimic" the CP-net ordering [16]. The proposed generation of an OCF-net from a CP-net leads to a total ordering, which contrasts with CP-nets. However, they proved that such an ordering is always consistent with the one induced by the corresponding CP-net. They also showed that the CP-net formalism is able to represent only a subclass of OCF-nets, which proves that OCF-nets are more expressive than CP-nets. These remarks can be immediately applied to *numerical* π-Pref nets as well.

7 Conclusion

This paper proposes a detailed study of π-Pref nets, which, if based on the product chain rule, are closer to Bayesian nets than CP-nets. This model proves to be flexible enough to support different readings leading to different orderings of solutions, and establishes the main relationships between them. π-Pref nets correctly reflect the elicited information in the sense that no further implicit priority is enforced like with CP-nets (e.g., in favor of parent nodes). π-Pref nets also offer a cautious way of modeling preferences without requiring numerical values, which should make them attractive for the same class of applications as CP-nets. In fact, precise numerical assessments are hard to get for conditional preferences that are qualitative in nature. Moreover, we have shown that symbolic possibilistic networks can handle additional qualitative information when available. Beside the fact that π-Pref nets can be put under an equivalent possibilistic logic form suitable for inference, they have another additive graphical counterpart under the form of OCF-nets.

[5] $\forall u_i \in Pa(A_i)$, $\exists j$ such that $\kappa(a_j|u_i) = 0$

REFERENCES

[1] N. Ben Amor, S. Benferhat, and K. Mellouli, 'Anytime propagation algorithm for min-based possibilistic graphs', *Soft Computing*, **8**(2), 150–161, (2003).

[2] N. Ben Amor, D. Dubois, H. Gouider, and H. Prade, 'Possibilistic networks: A new setting for modeling preferences', in*8th Int.Conf. SUM*, pp. 1–7. Springer, (2014).

[3] N. Ben Amor, D. Dubois, H. Gouider, and H. Prade, 'Possibilistic conditional preference networks', in *Proc.ECSQARU'2015*, pp. 36–46, (2015).

[4] S. Benferhat, D. Dubois, L. Garcia, and H. Prade, 'On the transformation between possibilistic logic bases and possibilistic causal networks', *Int. J. of Approximate Reasoning*, **29**(2), 135–173, (2002).

[5] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole, 'Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements', *J. Artif. Intell. Res.(JAIR)*, **21**, 135–191, (2004).

[6] G. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole, 'Preference-based constrained optimization with CP-nets', *Computational Intelligence*, **20**(2), 137–157, (2004).

[7] C. Cayrol, D. Dubois, and F. Touazi, 'Ordres Partiels entre Sous-Ensembles d'un Ensemble Partiellement Ordonné', Research report RR–2014-02–FR, IRIT, Université Paul Sabatier, Toulouse, (february 2014).

[8] L. De Raedt, K. Kersting, S. Natarajan, and D. Poole, *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2016.

[9] D. Dubois, H. Fargier, and H. Prade, 'Refinements of the maximin approach to decision-making in fuzzy environment', *Fuzzy Sets and Systems*, **81**, 103–122, (1996).

[10] D. Dubois, S. Kaci, and H. Prade, 'Approximation of conditional preferences networks "CP-nets" in possibilistic logic', in *Proc. 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE), Vancouver, July 16-21*, (2006).

[11] D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, 1988.

[12] D. Dubois and H. Prade, 'Epistemic entrenchment and possibilistic logic', *Artif. Intell.*, **50**(2), 223–239, (1991).

[13] D. Dubois and H. Prade, 'Possibilistic logic: a retrospective and prospective view', *Fuzzy Sets and Systems*, **144**(1), 3–23, (2004).

[14] D. Dubois and H. Prade, 'Qualitative and semi-quanlitative modeling of uncertain knowledge-a discussion', in *Computational Models of Rationality Essays Dedicated to Gabriele Kern-Isberner on the occasion of Her 60th Birthday*, pp. 280–292. College Publications, (2016).

[15] D. Dubois, H. Prade, and F. Touazi, 'Conditional Preference-nets, possibilistic logic, and the transitivity of priorities', in *Proc. 33rd SGAI Int. Conf., Cambridge*, pp. 175–184. Springer, (2013).

[16] C. Eichhorn, M. Fey, and G. Kern-Isberner, 'CP-and OCF-networks–a comparison', *Fuzzy sets and Systems*, **298**, 109–127, (2016).

[17] C. Eichhorn and G. Kern-Isberner, 'Using inductive reasoning for completing OCF-networks', *J. of Applied Logic*, **13**(4), 605–627, (2015).

[18] M. Goldszmidt and J. Pearl, 'Qualitative probabilities for default reasoning, belief revision, and causal modeling', *Artificial Intelligence*, **84**(1), 57–112, (1996).

[19] C. Gonzales and P. Perny, 'GAI networks for utility elicitation', in *Proc. 9th int. conf. Principles of Knowledge Representation and Reasoning*, pp. 224–234, (2004).

[20] C. Gonzales and P. Perny, 'GAI networks for decision making under certainty', in *IJCAI05–Workshop on Advances in Preference Handling*. Citeseer, (2005).

[21] C. Gonzales, P. Perny, and S. Queiroz, 'Réseaux GAI pour la prise de décision', *Revue d'intelligence artificielle*, **21**(4), 555–587, (2007).

[22] G. Kern-Isberner and C. Eichhorn, 'OCF-networks with missing values', in *Proceedings of the 4th Workshop on Dynamics of Knowledge and Belief*, pp. 46–60, (2013).

[23] F. Koriche and B. Zanuttini, 'Learning conditional preference networks', *Artificial Intelligence*, **174**(11), 685–703, (2010).

[24] J. Lang and J. Mengin, 'The complexity of learning separable ceteris paribus preferences.', in *IJCAI*, pp. 848–853, (2009).

[25] J. Pearl, 'Bayesian networks: A model of self-activated memory for evidential reasoning', in *Proc. of Cognitive Science Society (CSS-7)*, (1985).

[26] W. Spohn, 'Ordinal conditional functions: a dynamic theory of epistemic states', in *Causation in Decision, Belief Change, and Statistics*, eds., W. L. Harper and B. Skyrms, volume 2, 105–134, D. Reidel, (1988).

[27] L. A. Zadeh, 'Fuzzy sets as a basis for a theory of possibility', *Fuzzy Sets and Systems*, **1**, 3–28, (1978).