Towards a formalization of the Linguistic Conditional Preference networks

I. Truck*

EA4383 – LIASD,
Université Paris 8,
2, rue de la Liberté
93526 Saint-Denis, France
E-mail: truck@ai.univ-paris8.fr
*Corresponding author

J. Malenfant

UMR7606 CNRS – LIP6,
University Pierre et Marie Curie,
4, place Jussieu
75005 Paris, France
E-mail: jacques.malenfant@lip6.fr

Abstract: In recent works, we have proposed a graphical model to represent linguistic preferences called LCP-nets. LCP-nets have been implemented and used in a specific use case of industrial engineering. In this paper, we consolidate this contribution in formalizing it through a set of notations and computation rules in order to guarantee its durability and its reusability to other multi-criteria decision contexts. The paper formalizes the LCP-net structure, semantics, and validity. It also formalizes the dominance testing and optimization queries (for a discretized version of the problem in this latter case), in the line of previous CP-nets models.

Keywords: preference networks; fuzzy preferences; CP-nets; LCP-nets; multi-criteria decision making; dominance testing query; optimization query.

Reference to this paper should be made as follows: Truck, I. and Malenfant, J. (201x) 'Towards a formalization of the Linguistic Conditional Preference networks', *Int. J. of Applied Management Science*, Vol. x, No=; y, pp.xx–yy.

Biographical notes: I. Truck received her PhD in computer science from the Université de Reims (France) in 2002. She joined the laboratory EA4383 – LIASD at the Université Paris 8 in 2003 as an assistant professor. Her research interests include knowledge representation under uncertainty, especially by way of fuzzy logic and many-valued logic. She is also interested in music and computer science and, more recently to software architectures. She participates to projects in this field and co-supervises several PhD students.

J. Malenfant received his Ph.D. from the Université de Montréal in 1990 and his habilitation from the Université de Nantes (France) in 1997. He is full professor at the Université Pierre et Marie Curie. His research interests include the design, implementation and semantics of programming models for software architectures in autonomic computing, autonomous robotics and cyber-physical systems. More precisely, he addresses the compositionality of subsystems (*e.g.*, components) and the correctness of the resulting assemblies, as well as decision-making and large-scale coordination among decision-making software entities.

1 Introduction

Optimizing complex systems and decision-making processes are among the main points of industrial engineering and management. This optimization requires lots of tools from various disciplines such as mathematics, management science or artificial intelligence, and decision making is often seen as a central problem to be solved.

To address the multi-criteria characteristics of these problems, the preference modeling and their elicitation have attracted widespread attention for many years. Several formalisms have been proposed to express the choices or wishes. Among them are the *factored models* that decompose preferences. *Additive models* are a subclass of factored models. Their principle is that the preference values on subsets of attributes may be expressed independently of each others. It is the *ceteris paribus* principle (Braziunas and Boutilier, 2006). This avoids to ask the user for the comparison of all the attributes which would require to go through their entire joint instantiation.

The *generalized additive independence model* (GAI) allow for an additive decomposition (preferences are *added* instead of being multiplied, for example) of a utility function. In (Boutilier et al., 2004), the authors proposed a graphical and compact additive model called *CP-nets* (Conditional Preference networks) that links wished attributes to preferences. This model is a graph and is quite intuitive. It has three elements: *nodes* that represent the problem variables, *arcs* (or *cp-arcs*) that carry preferences among these variables for various given values, and *conditional preference tables* (CPTs) that express preferences on values taken by the variables.

The CP-nets permit to model wishes such as "for property X, I prefer the V_1 value instead of V_2 if the property Y equals to V_Y and the property Z equals to V_Z". There exists also a notion of *relative preference* between the preferences themselves: a CPT associated to a node has a higher priority than the CPTs of its descendants. This notion is taken into account when the complete outcomes are compared. A complete outcome is a tuple of values for all the variables of the graph. One of the interests of the CP-nets is their ability to approximate preferences easily with inference rules that will be nothing else than the CPTs. But CP-nets must obey some restrictions in order to allow ("algorithmically" speaking) the inference computations. The first restriction is that the graphs must be acyclic. The second one implies a "reasonable" use of the indifference relation between the preferences and so it implies total preorders in the CPTs for each parent node (Boutilier et al., 2004).

The *Utility CP-nets* (UCP-nets) (Boutilier et al., 2001) are inspired by the CP-nets but the definition of the binary relation $\succ$ ("is preferred to") between two values of nodes in the CPTs is replaced by numerical values (utility factors). Thus the CPTs contain numerical values. The recourse to values instead of order relations has been motivated by the fact that, in a CP-net, it was not possible to establish a comparison nor an order between the alternatives given as solutions to the problem (when there was more than a unique solution). By quantifying preferences, this problem becomes less important (Boubekeur and Tamine-Lechani, 2006). A utility factor is a real number associated to a node assignment given the assignment of its parent nodes. It expresses a preference degree between several assignments. There are *local* utility factors that indicate choices local to a node and *global* utility factors computed for each complete outcome that permit to order the solutions without any ambiguity. A UCP-net indeed defines a total order on the outcomes.

Another model inspired by the CP-nets is the *Tradeoffs-enhanced CP-nets* (TCP-nets). It allows to manage tradeoffs in the expression of the preferences (Brafman and Domshlak, 2002). TCP-nets deal with linguistic expressions such as:"a better assignment for X is more important than a better assignment for Y". These are called *relative importance preferences*. Moreover, TCP-nets deal with *conditional* relative importance preferences: "a better assignment for X is more important than a better assignment for Y if $Z = z$". Thus, new elements are introduced in the model: the *Conditional Importance Tables* (CITs) and two new kinds of arcs: *i-arcs* and *ci-arcs*. These arcs permit to model *basic* and *conditional* clauses of relative importance.

However these models (CP-nets, UCP-nets, TCP-nets) have two important restrictions. The first one concerns the continuity of the variable definition domains. Only discrete and finite domains are handled. The second one is about the difficulty to obtain precise utility values from the users. Indeed, there are many situations where the user is doubtful about his wishes which leads to imprecision in the preferences.

To overcome these limitations, we proposed in recent works an alternative to these models, the *linguistic CP-nets* (LCP-nets) that can deal with linguistic clauses and that can take into account variables defined over continuous domains (Châtel et al., 2008, 2010b). This model has been used in a specific use case to perform late binding between services consumers and service providers. But in order to make this model generic, a formalization is needed.

The LCP-net approach to express conditional preferences bridges the gap between GAI-based techniques towards the field of fuzzy preference elicitation. Examplary of this field, Curry and Lazzari (2009) elicit preferences from the ground up using raw data about choices of deciders exhibiting their preferences. Their approach classifies the choices among fuzzy subsets representing utility classes. Our work and theirs complement each other and will allow for a more thorough comparison between bottom-up approaches treating preferences in extension and the top-down ones that aim at capturing preferences in intension.

This paper lays the foundations of a formal definition of the LCP-nets. In Section 2, we recall our tool and then give two concrete examples. Section 3 exhibits the preliminary notations of the LCP-nets essential to the formal definitions of Section 4. Foundational properties are given in Section 5, especially regarding the CP-condition and the weights. Finally, we are interested in queries over LCP-nets in Section 6 (the dominance testing and the optimization query) while Section 7 concludes this study.

2 LCP-nets as a tool for expressing preferences

In LCP-nets, we partition the continuous domains using linguistic terms associated to fuzzy subsets (Zadeh, 1965) or to linguistic 2-tuples (Herrera and Martínez, 2000). Thus the utility factors are *words*. This allows for an easier way to capture the user wishes and for a better ordering of the outcomes that can be proposed to the user. Indeed if two outcomes exhibit more or less the same attributes, a use of discrete coarse-grained domains will prevent a ranking between them. Except if the granularity is increased and if the differences between the preferences are high enough — this is actually an explicit condition in the UCP-nets — to allow for a discrimination.

LCP-nets, as the other models from the "CP-net family", are acyclic graphs with nodes, arcs and preference tables. Linguistic descriptors must first be chosen to describe the term sets on each universe of discourse. As usual we take term sets with an odd cardinality (5, 7 or 9) (Delgado et al., 1993) in order to have a mid-term. For example, a term set T is: $T = \{s_0 : \textit{very low}, s_1 : \textit{low}, s_2 : \textit{medium}, s_3 : \textit{high}, s_4 : \textit{very high}\}$. It is also required to have the three following operators:

1. $\mathrm{Neg}(s_i) = s_j$ such that $j = g - i$ (with $g + 1$ being the cardinality),
2. $\max(s_i, s_j) = s_i$ if $s_i \geq s_j$,
3. $\min(s_i, s_j) = s_i$ if $s_i \leq s_j$.

In the linguistic 2-tuple model of Herrera and Martínez, trapezoidal or triangular fuzzy subsets are enough to express the imprecision of the clauses. Given a linguistic term, the 2-tuple formalism provides for a pair (fuzzy set, symbolic translation) $= (s_i, \alpha)$ with $\alpha \in [-0.5, 0.5[$ as can be seen in Figure 1 where the obtained 2-tuple is $(s_2, -0.3)$.

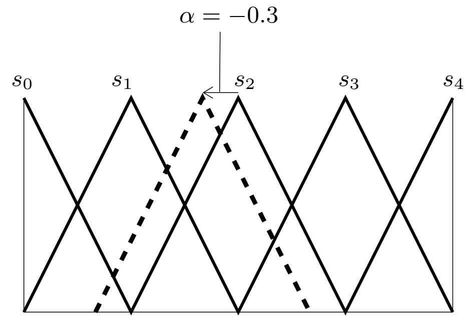


Figure 1 Lateral displacement of a linguistic label ⇒ $(s_2, -0.3)$ 2-tuple.

In this example, the α translation can be seen as a weakening modifier of the linguistic term s_2. Thus, using this model for the computations, one can give a result more or less equals to one of the elements *from the original term set*, *i.e.*, the same linguistic term set can be kept during the whole process.

Compared to the CP-, TCP- or UCP-nets, LCP-nets allow to deal with clauses such as "I tend to prefer the *more or less* V_1 value for property X over *exactly* V_2 if properties Y equals *approximately* V_Y and Z equals *a bit more than* V_Z". These statements that resemble improved fuzzy rules must be interpreted in a context where the global preference on X has to take into account each preference to be applied to Y to a certain degree.

Actually, this is equivalent to propose a flexible and intuitive model to express complicated sets of fuzzy rules that can be potentially interdependent.

The LCP-nets allow the users to express relative importances (conditional or not) and tradeoffs among the variables in using i-arcs or ci-arcs from the TCP-nets in addition to the cp-arcs from the CP-nets[1]. They include CPTs similar to those from the UCP-nets but with linguistic utility factors.

Let us now illustrate our LCP-nets with two examples.

Example 1 (Evening dress). This example is inspired by the one explained in (Brafman et al., 2006). Imagine a woman that has to choose an evening dress: she has to attend a formal evening and she would like to impress people with a long dress if she can find shoes going with it. She always prefers to optimize the length (L) over the color (C) of her dress (or skirt). Her preference about the color of her shoes (S) and about the height of heel (H) is conditioned by the color of the dress. And her preference between the optimization of the color of the shoes and the height of heel is conditioned by the length of the dress. (If the dress is long, she doesn't really care about the height of heel and prefers to take care about the color of shoes.) Figure 2 sums this example up: four CPTs, one CIT and the three kinds of arcs are used.

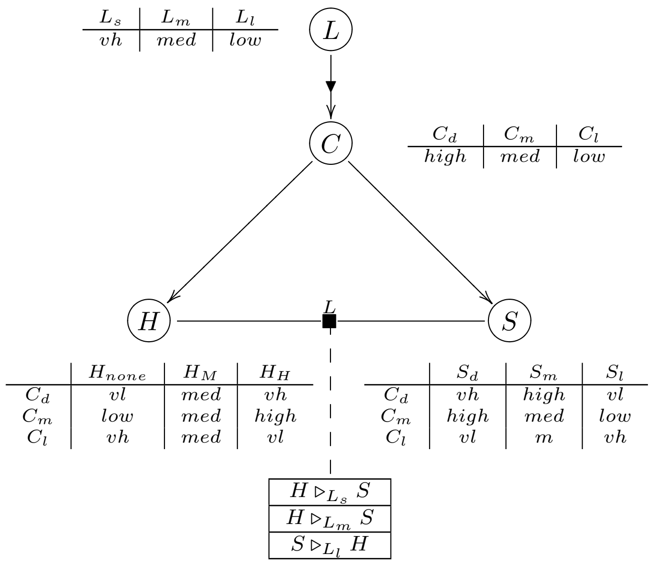


Figure 2 Example of preferences for an evening dress.

The glossary that is used is the following:

L_s	short dress
L_m	medium dress
L_l	long dress
C_d	dark color
C_m	medium color
C_l	light color
H_{none}	no heel
H_M	medium height of heel
H_H	high heels
S_d	dark color of shoes
S_m	medium color of shoes
S_l	light color of shoes
vvl	very very low utility
vl	very low utility
low	low utility
med	medium utility
high	high utility
vh	very high utility
vvh	very very high utility

Example 2 (Purchase). Imagine a person that has to purchase some good (any kind: a TV, a car, a computer, etc.). He wishes to receive his purchase as soon as possible, at the best price, unless he gets a really good deal. He always prefers to optimize the delivery time (D) on the quantity (Q) of options and on the price of the item (P). He also always prefers to optimize D over the rebate (R). And his preference between Q and P is conditioned by R: if the rebate is weak or medium, a good price is more important than the options. But if the rebate is high, he prefers to obtain as many options as possible. Figure 3 sums this example up: four CPTs, one CIT and the three kinds of arcs are used.

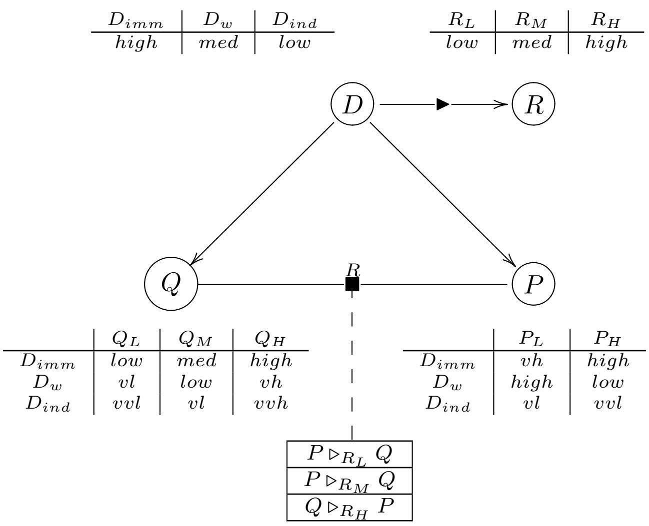


Figure 3 Example of preferences for a purchase.

The new terms added in this example are the following:

D_{imm}	immediate
D_w	week
D_{ind}	indefinite
R_L	low rebate
R_M	medium rebate
R_H	high rebate
Q_L	a few options
Q_M	a medium number of options
Q_H	a lot of options
P_L	low price
P_H	high price

LCP-nets have been implemented in Java using EMF (Eclipse Modeling Framework) to represent the LCP-net itself, and delegating the fuzzy inferences to the library *jfuzzylogic* (which has been improved with the consideration of linguistic 2-tuples by a member of our team (Abchir, 2011)). This work is available at the following Internet address: `http://code.google.com/p/lcp-nets/`. This implementation has already been used in the context of service-oriented computing to add a new programming abstraction to BPEL (Business Process Execution Language) that selects services to be called given their current quality of service (QoS). This work has been done in the context of the French ANR project SemEUsE (07-TLOG-018) (Châtel et al., 2010a).

The implementation proceeds in three steps. The first one is the elicitation process. It is performed *before* execution and creates the EMF model of the LCP-net. The second step is the translation of the preference model into an efficient representation that can be used *during* execution. Each CPT is translated into an inference system with a rule per table line. These inference systems are then translated to the *jfuzzylogic* format and loaded to be ready for computations. The last step is the preference model evaluation that corresponds to the valuation algorithm, see Subsection 4.3. At runtime, current attribute values are injected after a fuzzification phase (into fuzzy sets or into linguistic 2-tuples). Then the system computes local utilities for each node thanks to the CPTs with the inference systems. Finally the global utility is obtained by aggregating local utilities.

The aggregation operator cannot be a simple weighted mean. Indeed we must take into account the fact that the arcs give the relative importance of the nodes they interconnect. This *implicit* relative importance due to the node position in the graph, also called CP-condition (see §5.1), must be reflected in order to be considered while computing the global utility factor. For instance, for the purchase example, delivery time D that is the higher vertex of the graph is necessarily more important than rebate, quantity of options and price. Let us imagine that R equals R_H, so Q dominates P. Thus, R and Q have the same depth, *i.e.*, are of equal importance. A weight is thus attached to each node *i.e.*, to each utility factor (see figure 4).

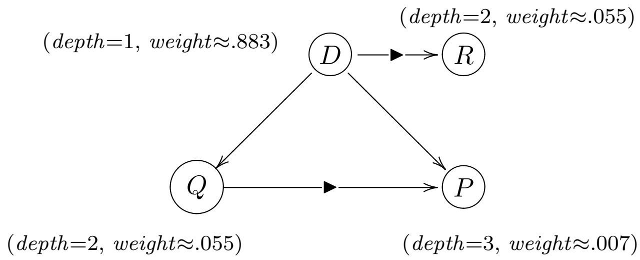


Figure 4 From node depth to node weights.

It has to be noted that UCP-nets don't use any weights on nodes but rather constrain the utility factors themselves to contain this information. Indeed, the UCP-nets formalism requires gaps between utility values big enough to carry this information implicitly. Because LCP-nets use linguistic terms instead of real numbers to express the utilities, it is much more difficult to introduce such gaps. Weights are thus given to the nodes of the LCP-net, taking into account their depth and beginning with the root node. The assignment of the weights needs a monotonous and strictly decreasing function according to depth. Weights are between 0 and 1 and sum to 1. This point is discussed and detailed in Subsection 5.2.

The resulting global utility allows for a simple and quick comparison (since we have real numbers after defuzzification) of the various outcomes for a given decision and permits a precise ranking. If we want to give the final proposed choices to a human, it can be useful to give a linguistic answer for their utilities (and give instead linguistic 2-tuples answers).

Now we introduce all the preliminary notations needed to define the LCP-nets formally and to guarantee their reusability in any context.

3 Preliminary notations

Following TCP-net notations, we define our LCP-nets in a formal manner. To illustrate this work, we take the purchase example. Let:

- V_i be a variable ($i \in \{1, \ldots, p\}$): *e.g.*, price,
- $\mathcal{D}(V_i)$ be the definition domain of V_i: *e.g.*, $[0, 100]$,
- T_{V_i} be the linguistic term set associated to V_i: *e.g.*, $\{P_L, P_H\}$,
- LV (a linguistic variable) be the following triplet: $LV = \langle V, \mathcal{D}(V), T_V \rangle$:
  e.g., $\langle$price,[0, 100],$\{P_L, P_H\}\rangle$,
- $Ker(t)$ be the kernel of the fuzzy set that represents the linguistic term t, *e.g.*, P_H,
- $\sup_{Ker}(t)$ be the maximum value of abscissa of the kernel of t,
- $\inf_{Ker}(t)$ be the minimum value of abscissa of the kernel of t,
- $\delta(t_1, t_2) = \inf_{Ker}(t_1) - \sup_{Ker}(t_2)$.

As in the UCP-net formalism, preferences are expressed through utilities in our framework. But they are expressed through linguistic variables, as the other variables. For all tables in the LCP-net, they take their values in the single triplet $\langle V_U, \mathcal{D}(V_U), T_{V_U} \rangle$ defined once for all (for each LCP-net) over a normalized domain $[0, 1]$: *e.g.*, $\langle$utility,$[0, 1]$, $\{very\_very\_low, very\_low, low, medium, high, very\_high, very\_very\_high\}\rangle$. This definition of linguistic utilities entail an order relation on the linguistic terms so that the first one (here, *very_very_low*) is the weakest.

One utility is a triplet $LV_U = \langle V_U, \mathcal{D}(V_U), S_{V_U} \rangle$, with $S_{V_U} \in T_{V_U}$, *e.g.*, $\langle$utility,$[0, 1], low\rangle$.

A conditional preference table $CPT(LV)$ associates preferences over $\mathcal{D}$ for every possible value assignment to the parents of LV denoted $Pa(LV)$. In addition, as in the TCP-nets formalism, each undirected ci-arc is annotated with a conditional importance table $CIT(LV)$. A CIT associated with such an edge (LV_i, LV_j) describes the relative importance of LV_i and LV_j given the value of the corresponding importance-conditioning linguistic variables $\mathbf{LV}_k$.

Graphically, a preference table (CPT or CIT) is a tuple of triplets, *i.e.*, a table with N dimensions. N is the number of the linguistic variables interrelated with LV, including LV ($N = |Pa(LV)| + 1$) and a utility S_{V_U} is defined in each case.

Thus a preference table may be represented by the tuple
$\langle LV_i, LV_{i'}, \ldots, LV_{i''\cdots'}, LV_{U_1}, LV_{U_2}, \ldots, LV_{U_\eta} \rangle$
with $\eta \in \{2N, \ldots, K\}$
and $K = |T_{V_i}| \times |T_{V_{i'}}| \times \ldots \times |T_{V_{i''\cdots'}}|$.

For example, a preference table is the tuple:
$\langle\langle$price,[0, 100],$\{P_L, P_H\}\rangle$,
$\langle$delivery_time,[0, 90],$\{D_{imm}, D_w, D_{ind}\}\rangle$,
$\langle$utility,[0, 1], *very_high*$\rangle$,
$\langle$utility,[0, 1], *high*$\rangle$,
$\langle$utility,[0, 1], *high*$\rangle$,
$\langle$utility,[0, 1], *low*$\rangle$,
$\langle$utility,[0, 1], *very_low*$\rangle$,
$\langle$utility,[0, 1], *very_very_low*$\rangle\rangle$.

More precisely, a preference table is equal to:

$$\begin{array}{l}\langle\ \langle S_{V_{i_1}}, S_{V_{i'_1}}, \ldots, S_{V_{i''\cdots'_1}}, S_{V_{U_1}} \rangle, \\ \quad \langle S_{V_{i_2}}, S_{V_{i'_2}}, \ldots, S_{V_{i''\cdots'_2}}, S_{V_{U_2}} \rangle, \\ \quad \ldots \\ \quad \langle S_{V_{i_\eta}}, S_{V_{i'_\eta}}, \ldots, S_{V_{i''\cdots'_\eta}}, S_{V_{U_\eta}} \rangle \\ \rangle\end{array}$$

So we get η tuples $\langle S_{V_i}, S_{V_{i'}}, \ldots, S_{V_{i''\cdots'}}, S_{V_U} \rangle$ with $\min(\eta) = 2N$ and $\max(\eta) = K$. The reason why the minimum is equal to $2N$ is because it is necessary that $|T_V| \geq 2$ in order to be able to express a preference (!).

Following the same example and knowing that price and delivery_time are interrelated, the associated preference table can be defined as these six ($\eta = 6$) tuples:
$\langle\langle P_L, D_{imm}, very\_high\rangle,$
$\langle P_L, D_w, high\rangle,$
$\langle P_L, D_{ind}, high\rangle,$
$\langle P_H, D_{imm}, low\rangle,$
$\langle P_H, D_w, very\_low\rangle,$
$\langle P_H, D_{ind}, very\_very\_low\rangle\rangle.$

This is to be read as six rules implying two different linguistic variables, *i.e.*, six triplets $\langle V, \mathcal{D}(V), S_V\rangle$ and six triplets $\langle V_U, \mathcal{D}(V_U), S_{V_U}\rangle$:

R1. If we have
⟨price,[0, 100],P_L⟩ and
⟨delivery_time,[0, 90],D_{imm}⟩
then we have
⟨utility,[0, 1],*very_high*⟩;

R2. ...

...

R6. If we have
⟨price,[0, 100],P_H⟩ and
⟨delivery_time,[0, 90],D_{ind}⟩
then we have ⟨utility,[0, 1],*very_very_low*⟩.

4 LCP-net formal definition

We now introduce some structural definitions to be able to define our LCP-nets formally.

4.1 Structural definitions

Definition. An LCP-net $\mathcal{L}$ over variables $\{LV_1, \ldots, LV_p\}$ is a directed graph over $\{LV_1, \ldots, LV_p\}$ whose nodes are annotated with conditional preference tables $CPT(LV_i)$ and with conditional importance tables $CIT(LV_i)$ for $i \in \{1, \ldots, p\}$.

Thus $\mathcal{L}$ is a tuple ⟨SL,cp,i,ci,cpt,cit,W⟩ where:

- SL is a set of linguistic variables $\{LV_1, \ldots, LV_p\}$, *e.g.* $SL = \{$⟨delivery_time,[0, 90],$\{D_{imm}, D_w, D_{ind}\}$⟩, ⟨rebate,[0, 100],$\{R_L, R_M, R_H\}$⟩, ⟨quantity,[0, 10],$\{Q_L, Q_M, Q_H\}$⟩, ⟨price,[0, 100],$\{P_L, P_H\}$⟩$\}$,
- cp is a set of directed cp-arcs. A cp-arc $\langle\overrightarrow{LV_i, LV_j}\rangle$ is in $\mathcal{L}$ iff the preferences over the values of LV_j depend on the actual value of LV_i. For each $LV \in SL$, $Pa(LV) = \{LV' | \langle\overrightarrow{LV', LV}\rangle \in \mathsf{cp}\}$,
- i is a set of directed i-arcs. An i-arc $(\overrightarrow{LV_i, LV_j})$ is in $\mathcal{L}$ iff $LV_i \rhd LV_j$, *i.e.*, iff LV_i is more important than LV_j (see Definition 3 in (Brafman et al., 2006)),
- ci is a set of undirected ci-arcs. A ci-arc (LV_i, LV_j) is in $\mathcal{L}$ iff we have $\mathcal{RI}(LV_i, LV_j | LV_k)$, *i.e.*, iff the relative importance of LV_i and LV_j is conditioned on $\mathbf{LV}_k$, with $\mathbf{LV}_k \subseteq SL \smallsetminus \{LV_i, LV_j\}$. We call $\mathbf{LV}_k$ the *selector set* of (LV_i, LV_j) and denote it by $\mathcal{S}(LV_i, LV_j)$,
- cpt associates a CPT with every linguistic variable $LV \in SL$, where $CPT(LV)$ is a mapping from $\mathcal{D}(Pa(LV)) \times \mathcal{D}(V)$ (i.e., assignments to LV's parent linguistic variables) to $\mathcal{D}(V_U)$,
- cit associates with every ci-arc between LV_i and LV_j a CIT from $\mathcal{D}(\mathcal{S}(LV_i, LV_j))$ to orders over the set $\{LV_i, LV_j\}$,
- W is a weight vector as defined in Section 4.3.

4.2 Structural invariants

When implementing the LCP-nets in EMF, we construct the graphs incrementally. So, it is very important to factorize the objects. In particular, we define LCP-nets *fragments* that are pieces of graphs (*e.g.*, only a node and its CPT), that are incrementally added to the LCP-net.

But this way of doing doesn't guarantee the obtention of structurally *valid* LCP-nets. Verifying the validity of LCP-nets *a posteriori* is far from trivial. Therefore, we propose to construct LCP-nets by gradually adding coherent fragments that, given a valid LCP-net, augment it to a larger valid one. There is a certain number of conditions that have to be fulfilled. We define an *atomic* valid LCP-net (a minimal LCP-net) as an object with only one node and its CPT (or CIT). The elementary operators to manipulate valid LCP-nets are: addition (of a node, of an arc, etc.), subtraction, etc. These operators have invariants, preconditions and postconditions.

In this paper we only focus on invariants.

Let consider the following objects:

- n is a node;
- SN is the set of nodes;
- an arc is denoted (s, t) with s the source node and t the sink node. In the ci-arcs, s can be exchanged with t;
- SA is the set of arcs (cp, i and ci) : $SA = \{$cp,i,ci$\}$.

Invariants that share all the operators on LCP-nets are the following:

- the total number of arcs (cp, i, ci) is not greater than the number of pairs (s, t) where $s, t \in SN$ and $s \neq t$;
- there is mutual exclusion between the kinds of arcs:
  - if $(s, t) \in \mathsf{cp}$ then $(s, t) \notin \mathsf{i}$ and $(s, t) \notin \mathsf{ci}$;
  - if $(s, t) \in \mathsf{i}$ then $(s, t) \notin \mathsf{cp}$ and $(s, t) \notin \mathsf{ci}$;
  - if $(s, t) \in \mathsf{ci}$ then $(s, t) \notin \mathsf{i}$ and $(s, t) \notin \mathsf{cp}$.

- the dimension of the CPT associated to s node is equal to 1 + the number of cp-arcs that are indegrees of s;
- each CPT has a dimension that is less or equal than the number of domain values of the associated node;
- there is no conditional cycles in the graph;
- there is at least one node (*i.e.*, at least a CPT) and there are from 0 to n arcs;
- there are at least as many CPTs than nodes, *i.e* there are exactly $\sharp nodes$ CPTs and $\sharp ci\text{-}arcs$ CITs.

Under these conditions, we construct structurally valid LCP-nets, *i.e.*, acyclic graphs, with a number of arcs less or equal to the number of nodes minus 1 ($\sharp nodes - 1$).

4.3 LCP-nets semantics

The LCP-net semantics defines how a structurally valid LCP-net is used to compute the global utility function it is expressing. The CPT (attached to an LV) supplies with a local utility for this LV. Let lu be this local utility which is also an object of LV type. It is computed thanks to an inference engine using the aforementioned rules.

So we get on the one hand an LCP-net that expresses the preferences and on the other hand p local utilities denoted by the tuple:

$$LU = \langle lu_1, lu_2, \ldots, lu_p \rangle$$

Then each node of $\mathcal{L}$ is associated with a weight w, *i.e.*, we obtain a weight vector W:

$$W = \langle w_1, w_2, \ldots, w_p \rangle$$

where w_i depends on the node depth (see Section 5).

W is combined with LU to obtain the global utility associated to an outcome o denoted GU_o.

$GU_o = Agg(LU_o, W)$, with Agg a weighted aggregator such as an OWA operator (for example a simple weighted average aggregation).

A local utility is either a linguistic term, or a linguistic 2-tuple, or a number corresponding to the defuzzification (through operator d) of the subset: lu equals $\langle V_U, \mathcal{D}(V_U), S_{V_U} \rangle$ and denotes either $\mu_{S_{V_U}}$, or (S_{V_U}, α), or $d(\mu_{S_{V_U}})$ with $\mu_{S_{V_U}}$ the membership function of S_{V_U} such that:

$$\mu_{S_{V_U}}(y) = \begin{cases} \bot(\mu_{S^1_{V_U}}(y), \ldots, \mu_{S^\eta_{V_U}}(y)) \\ \quad \text{if the } \eta \text{ rules are independent} \\ \top(\mu_{S^1_{V_U}}(y), \ldots, \mu_{S^\eta_{V_U}}(y)) \\ \quad \text{otherwise} \end{cases}$$

with $y \in \mathcal{D}(V_U)$, $\bot$ a triangular conorm and $\top$ a triangular norm.

For sake of simplicity, we assume that $lu_i = \mu_{S_{V_{U_i}}}(y)$. GU_o is thereof either a linguistic term or a number. We assume that in the case where it is a linguistic term, it is always possible to find a defuzzification operator d that provides for a number.

Considering that the rules are independent and applying the generalized modus ponens (GMP),

$$\mu_{S_{V_U}}(y) = \sup_{(x_1,\ldots,x_N) \in \mathcal{D}(V_1)\times\cdots\times\mathcal{D}(V_N)} \Big\{ \top \Big[g(\mu_{S_{V'_1}}(x_1), \ldots, \mu_{S_{V'_N}}(x_N)), \Phi(g(\mu_{S^1_{V_1}}(x_1), \ldots, \mu_{S^1_{V_N}}(x_N)), \mu_{S^1_{V_U}}(y)) \Big] \vee \ldots \vee \top \Big[g(\mu_{S_{V'_1}}(x_1), \ldots, \mu_{S_{V'_N}}(x_N)), \Phi(g(\mu_{S^\eta_{V_1}}(x_1), \ldots, \mu_{S^\eta_{V_N}}(x_N)), \mu_{S^\eta_{V_U}}(y)) \Big] \Big\}$$

with $\mu_X(x)$ the membership function of element $x \in X$, Φ any fuzzy implication, V' the real variables observed (retrieved, given by the user), $S_{V'_1}$ the linguistic term associated to the first variable (V'_1) observed and g an aggregation operator such as a triangular norm (min for example). Let us precise that the linguistic terms of the real variables observed $S_{V'}$ can be of two types. If the LCP-net deals with fuzzy sets, $S_{V'}$ are possibly *modified*[2] fuzzy sets. If the LCP-net deals with 2-tuples, $S_{V'}$ are 2-tuples (s_i, α) with α possibly different from zero.

Thus an outcome o is actually a tuple $\langle S_{V'_1}, \ldots, S_{V'_p} \rangle$.

The valuation algorithm that permits to compute the global utility factor for each outcome is defined as follows (see Algorithm 1). First thing to do is the computation of the node weights w_i, *i.e.*, the weight vector W. The observed values are then injected in the fuzzy inference system that gives local utilities, one per node. Each local utility is combined with its associated weight and a global utility is then obtained thanks to Agg.

Algorithm 1 Valuation algorithm

Require: o is a tuple $\langle S_{V'_1}, \ldots, S_{V'_p} \rangle$, SO is the set of o, $\mathcal{L}$ is valid

```
for i = 1 to p do
    compute w_i and add it to the weight vector W
end for
for each outcome o ∈ SO do
    for each table CPT(LV_i) or CIT(LV_i) do
        inject observed values of o and apply an inference such as the generalized modus ponens,
        compute and retrieve the set of lu_i for this o.
    end for
    compute and retrieve GU for this o
end for
return the set of GU, one per outcome
```

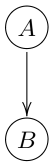
5 Properties for the LCP-nets

This section examines in more details conditions on utility values that make them compatible with the preferences expressed through the arcs, and then looks at ways to compute them.

5.1 Weights and the CP-condition

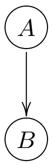
As in UCP-nets, the use of utilities in CPT, rather than order relations in other kinds of CP-nets, can come in contradiction with the preferences expressed through the arcs. The underlying concept is the one of dominance, which is better understood through an UCP-net example.

Example Consider the two following pseudo-UCP-nets that define two preference sets for the binary variables A, B:



f_A	a	$\bar{a}$
	4	5

f_B	a	$\bar{a}$
b	8	2
$\bar{b}$	9	1



f_A	a	$\bar{a}$
	5	10

f_B	a	$\bar{a}$
b	4	2
$\bar{b}$	3	1

UCP-nets tables are filled with utilities taken from $\mathbb{R}$, and the global utility function is computed by adding the values from the different tables that provide for its GAI decomposition. For instance, $GU_{ab} = f_A(a) + f_B(ab) = 4 + 8 = 12$ with $f_X(\cdot)$ the value of factor at the level of X variable. On the left side, the cp-arc from A to B makes a better assignment to A more preferable than a better assignment to B, so $\bar{a}$ is preferred to a given their respective utilities. Yet, when we compute the global utility of ab, $GU_{ab} = 12$, it is better than any one of $\bar{a}b$ or $\bar{a}\bar{b}$, *i.e.*, $GU_{\bar{a}b} = 7$ and $GU_{\bar{a}\bar{b}} = 6$. In this case, we say that the utilities defined in f_B invert the preference expressed by the cp-arc from A to B. One can see that the UCP-net on the right hand side does not exhibit such an inversion. □

The pseudo-UCP-net on the left side in the example is not a UCP-net, as it does not ensure the CP-condition (Boutilier et al., 2001), while the one on the right is. Intuitively, the utilities in the right UCP-net ensures that any potential gain given by changing the choice for A in f_B will exhibit a more important loss in f_A, thus refraining the decider to make this change. In the UCP-net terms, this property is called the dominance of a node over all of its children, a property that is adapted to the case of linguistic utilities of LCP-net as follows.

Definition (adapted from (Boutilier et al., 2001)) Let X be a variable with parents $\mathbf{U}$ and children $\mathbf{Y} = \{Y_1, ..., Y_n\}$ and let $\mathbf{Z}_i$ be the parents of Y_i excluding X and any element of $\mathbf{U}$. Let $\mathbf{Z} = \bigcup \mathbf{Z}_i$. Let $\mathbf{U}_i$ be the subset of variables in $\mathbf{U}$ that are the parents of Y_i. We say that X *dominates its children given* $\mathbf{u} \in \mathcal{D}(\mathbf{U})$ if, for all x_1, x_2 such that $f_X(x_1, \mathbf{u}) \geq f_X(x_2, \mathbf{u})$, then for all $\mathbf{z} \in \mathcal{D}(\mathbf{Z})$, for all $\langle y_1, ..., y_n \rangle \in \mathcal{D}(\mathbf{Y})$, we have:

$$\delta(f_X(x_1, \mathbf{u}), f_X(x_2, \mathbf{u})) \geq \sum_i^n \delta(f_{Y_i}(y_i, x_2, \mathbf{u}_i, \mathbf{z}_i), f_{Y_i}(y_i, x_1, \mathbf{u}_i, \mathbf{z}_i)) \quad (1)$$

X dominates its children if this relation holds for all $\mathbf{u} \in \mathcal{D}(\mathbf{U})$. □

Based on this definition, the following proposition establishes how a fully valid LCP-net respects the CP-condition.

Proposition (adapted from (Boutilier et al., 2001)) Let $\mathcal{L}$ be a DAG over $\{V_i\}$ whose CPT reflect the GAI-structure of the utility function it defines. Then $\mathcal{L}$ is a LCP-net *iff* each variable V_i dominates its children.
Proof: see (Boutilier et al., 2001). □

Observing this condition puts an annoying burden over the elicitation process. Boutilier *et al.* have shown that stronger but simpler conditions can be adopted to facilitate the elicitation. If the domain of utilities is normalized to $[0, 1]$, then, for each X and instantiation of its parents $\mathbf{u}$, they show that there exist a *multiplicative tradeoff weight* $\pi_X^{\mathbf{u}}$ and *additive tradeoff weights* $\sigma_X^{\mathbf{u}}$ such that the global utility function obtained by applying these weights to the values of the CPT respects the CP-condition and thus always gives a UCP-net. Using only multiplicative tradeoff weights gives an even stronger condition but allows to ensure the CP-condition on LCP-nets merely by a careful choice of the weights used to aggregate the local utilities, therefore freeing the users from this burden.

Proposition For any LCP-net, there exist weights $w_1, ..., w_p$ such that the global utility function respects the CP-condition.

Proof: The proof is constructive, exhibiting a set of weights that ensures the respect of the CP-condition. Consider the equation 1. Let A be

$$A = \min_{x_1, x_2} \delta(f_X(x_1, \mathbf{u}), f_X(x_2, \mathbf{u})) \quad (2)$$

then if each of the terms in the right hand side sum in equation 1 is affected of a weight A/n, observe that the inequation will always be verified. Indeed, as the utilities are normalized to $[0, 1]$, then 1 is an upper bound of $\delta(f_{Y_i}(y_i, x_2, \mathbf{u}_i, \mathbf{z}_i), f_{Y_i}(y_i, x_1, \mathbf{u}_i, \mathbf{z}_i))$, so

$$\sum_{i=1}^n \delta(f_{Y_i}(y_i, x_2, \mathbf{u}_i, \mathbf{z}_i), f_{Y_i}(y_i, x_1, \mathbf{u}_i, \mathbf{z}_i)) \leq \sum_i^n 1 = n \quad (3)$$

And if we multiply this result by the weight A/n, we get A as an upper bound for the right hand side of the inequation. As A is taken as the minimal value for the left hand side, the inequation will always be true.

Let $V_l, l = 1, ..., m$ be the partition of the set of variables V into the levels $1, ..., m$ of the LCP-net, then the weights that are the solution to (nodes on the same level have the same weight):

$$w_l = w_{l-1} \left(\frac{\min_{V \in V_l} \min_{v_1, v_2 \in T_V, \mathbf{u}} \delta(f_V(v_1, \mathbf{u}), f_V(v_2, \mathbf{u}))}{\max_{V \in V_l} \sharp\mathscr{C}(V)} \right) \quad (4)$$

for $l = 2, ..., m$, and

$$\sum_{1=1}^m (\sharp V_l) w_l = 1 \quad (5)$$

where $\mathscr{C}(V)$ is the set of children of V and $\sharp$ the cardinality operator. Then the CP-condition will hold on the LCP-net because the decrease in the weights over the levels just implements the above necessary condition for the weighted inequation to be always true. □

Indeed, this way to look at LCP-nets considers them as discretized into their linguistic terms. In general, outcomes are not restricted to take only their values among the linguistic terms of variable definitions, but rather observed values over the entire continuous domains to give utilities also covering the entire continuous domain $\mathcal{D}(U)$. In the continuous domain case, the CP-condition is more subtle. Intuitively, the dominance requires that for all values of the parent variable X and all values of its children Y_i, any changes in the value of X that would give better utilities in the tables of the children would give a more important loss in utility in the table of X. Such a condition can be expressed using directional derivative: in any direction changing the values of the outcome where the gradient is positive among the CPT of the children, if the gradient found for the projection of this direction onto the dimension X in its CPT is negative, then it is larger in absolute value than the former. The following therefore adapts the above definition to the continuous case of LCP-nets.

Definition (continuous domains case) Let X, $\mathbf{U}$, $\mathbf{Y} = \{Y_1, ..., Y_n\}$, $\mathbf{Z}_i$, $\mathbf{Z}$ and $\mathbf{U}_i$ as in the previous definition. For all variables V, let F_V be the functions from $\mathcal{D}(Pa(V)) \times \mathcal{D}(V) \to \mathcal{D}(U)$, defined from their associated CPT given the underlying fuzzy inference system, which are derivable everywhere, and $F_{\mathbf{Y}}(\mathbf{y}, x, \mathbf{u}, \mathbf{z}) = \sum_{i=1}^{n} F_{Y_i}(y_i, x, \mathbf{u}, \mathbf{z})$.

We say that X *dominates its children given* $\mathbf{u} \in \mathcal{D}(\mathbf{U})$ if, for all x, for all $\mathbf{z} \in \mathcal{D}(\mathbf{Z})$, for all $\mathbf{y} = \langle y_1, ..., y_n \rangle \in \mathcal{D}(\mathbf{Y})$, for all vectors $\mathbf{b}$ in the space $\mathcal{D}(\mathbf{Y}) \times \mathcal{D}(X) \times \mathcal{D}(\mathbf{Z})$ and its projection $\hat{\mathbf{b}}$ onto the dimension $\mathcal{D}(X)$, then if $\hat{\mathbf{b}} \cdot \nabla f_X(x, \mathbf{u}) < 0$, we have

$$|\, \hat{\mathbf{b}} \cdot \nabla f_X(x, \mathbf{u})\, | > \mathbf{b} \cdot \nabla F_{\mathbf{Y}}(\mathbf{y}, x, \mathbf{u}, \mathbf{z}) \qquad (6)$$

where ∇ is the usual gradient operator, and the directional derivative obtained by the scalar mutiplication of the direction vector $\mathbf{b}$ and the gradient vector.

As before, X *dominates its children* if this relation holds for all $\mathbf{u} \in \mathcal{D}(\mathbf{U})$. □

Unfortunately, this condition is very difficult to verify. The form of the functions F_V depends upon the underlying GMP used to do the fuzzy inference and it also depends upon the shapes of the fuzzy subsets that are adopted. Cases that do not respect it are easy to construct. It is also very difficult to get an analytical definition of these functions, that would be required to verify the condition. Moreover, expressing simpler sufficient conditions that would allow for the assignment of weights to guarantee the CP-condition, as it has been done in the discrete case, remains an open problem. In practice, when using for example Mamdani-style GMP over well-spaced triangular fuzzy subsets, the weights computed for the discretized case in the previous definition appears to also respect the CP-condition for the continuous case.

5.2 Fuzzy interpretation of weights

The algorithm for computing W can be based on a BUM (Basic Unit-interval Monotonic) family function (Yager, 2007). A BUM function f_{BUM} is a mapping from $[0, 1]$ to $[0, 1]$ and assumes the following properties:

- $f_{BUM}(0) = 0$
- $f_{BUM}(1) = 1$
- f_{BUM} is increasing
(*i.e.*, if $x > y$ then $f_{BUM}(x) \geq f_{BUM}(y)$)

So weights W are computed thanks to f_{BUM} in the following manner:

$$w_i = f_{BUM}(i/p) - f_{BUM}((i-1)/p)$$

The chosen f_{BUM} function can be $f_{BUM}(x) = x$ (in this case, all weights equal $(1/p)$ with p the number of nodes) ; or $f_{BUM}(x) = x^3$ (in that case, w_1 is very small compared to w_p) ; or $f_{BUM}(x) = \sqrt{x}$ (in that case, w_1 is the greater weight). To be able to analyze the choice of f_{BUM}, we can compute a measure of *orness* on this weight vector (Yager, 1988):

$$\mathit{orness}(W) = \frac{1}{p-1} \sum_{i=1}^{p} (p-i) w_i$$

This measure, between 0 and 1, allows us to express to which extent the aggregator using these weights resembles an OR. For example, when $f_{BUM}(x) = x$, $\mathit{orness}(W) = 0.5$. But when w_1 is much bigger than the "following" weights, $\mathit{orness}(W)$ tends towards 1.

As in the CP-nets, the deeper we go, the smaller the weights: we will then choose a vector W whose measure *orness* is between 0.5 and 1^3, *i.e.*, $f_{BUM}(x) = \sqrt{x}$ or $\sqrt[3]{x}$, etc.

Assigning weights to nodes of a graph is slightly different from a classical weight assignment to values. The difference is in the order of the values. In an LCP-net graph several nodes can have the same depth, so the order is not total. That is why assigning w only thanks to a BUM function, even appropriately chosen, doesn't permit to completely answer our problem, since nodes of the same depth would be discriminated.

We apply a BUM function such as the associated w be decreasing ($w_i > w_{i+1}$, with $i \in [1, p]$). Then for every node of the same depth, we sum their associated weights and make an equirepartition of the obtained sum between these nodes.

Thus, every constraint is fulfilled, by constructing weights through f_{BUM}:

- $\sum_{i=1}^{p} w_{i,l_i} = 1$ with l_i the depth of node i, $l_i \in [1, L]$ and $L \leq p$
- $\forall i \in [1, p], \forall l_i \in [1, L], \begin{cases} w_{i,l_i} > w_{i+1,l_{i+1}} & \text{if } l_i \neq l_{i+1} \\ w_{i,l_i} = w_{i+1,l_{i+1}} & \text{otherwise} \end{cases}$

6 Queries over LCP-nets

The two major uses of preference networks are to compare two outcomes and to look for an optimal outcome. The LCP-nets algorithms for these two kinds of queries are now presented.

6.1 Dominance testing

A basic query with respect to the LCP-net model is preferential comparison between outcomes. In order to perform the dominance testing, we shall prove that an outcome o_1 can be found as being preferred to another outcome o_2.

In the theorem that follows, the notations used are those from the sequent calculus (Gentzen, 1935). Sequents are expressions of the form $\Gamma \vdash \Delta$, where Γ and Δ are (possibly empty) sequences of logical formulas. A statement Δ follows semantically from a set of premises Γ ($\Gamma \models \Delta$) iff the sequent $\Gamma \vdash \Delta$ can be derived by the above rules. And with the horizontal lines, we proceed to sequential derivations.

Theorem.
Given an LCP-net $\mathcal{L}$ and a pair of outcomes o_1 and o_2, we have that $\mathcal{L} \models o_1 \preccurlyeq o_2$ iff GU_{o_1} is weaker than GU_{o_2}. We say that o_2 is preferred o_1 and that o_2 dominates o_1 with respect to $\mathcal{L}$.

$$\frac{\mathcal{L} \vdash lu_1^o = \mu_{S_{V_{U,1}^o}} \quad \ldots \quad lu_p^o = \mu_{S_{V_{U,p}^o}}}{\mathcal{L} \vdash LU_o = \langle lu_1^o, \ldots, lu_p^o \rangle} \tag{7}$$

$$\frac{\mathcal{L} \vdash LU_{o_1} = \langle lu_1^{o_1}, \ldots, lu_p^{o_1} \rangle \quad \mathcal{L} \vdash LU_{o_2} = \langle lu_1^{o_2}, \ldots, lu_p^{o_2} \rangle}{\mathcal{L} \vdash \Delta(LU_{o_1}, W) \leq \Delta(LU_{o_2}, W)} \tag{8}$$

$$\frac{\mathcal{L} \vdash \Delta(LU_{o_1}, W) \leq \Delta(LU_{o_2}, W)}{\mathcal{L} \vdash d(GU_{o_1}) \leq d(GU_{o_2})} \tag{9}$$

$$\frac{\mathcal{L} \vdash d(GU_{o_1}) \leq d(GU_{o_2})}{\mathcal{L} \vdash GU_{o_1} \preccurlyeq GU_{o_2}} \tag{10}$$

$$\frac{\mathcal{L} \vdash GU_{o_1} \preccurlyeq GU_{o_2}}{\mathcal{L} \models o_1 \preccurlyeq o_2} \tag{11}$$

This means that starting with a *well-formed* LCP-net, *i.e.*, a valid LCP-net, it is always possible to infer whether an outcome is preferred to another one. Of course, that doesn't mean that in all situations, indifference is impossible. Indeed, if two outcomes are very close (granularity would be too coarse to distinguish between them) then both will be equally chosen as the best ones.

6.2 Optimization query

After stating precisely the problem, the optimization algorithm is presented in two steps, global optimization and per CPT local optimizations, before looking forward to remove the current underlying hypothesis.

6.2.1 Problem statement

The outcome optimization query on a LCP-net defined over a set of variables $\mathbf{V} = \{V_1, ..., V_p\}$ consists in finding the outcome $o = \langle v_1, ..., v_p \rangle$ such that $\forall o' \neq o, o \not\succcurlyeq o'$. In CP-nets, where the domains are discrete and finite, this amounts to select the most preferable tuple of values among the combinatorial set of possible tuples. As LCP-nets are defined over linguistic variables which themselves have continuous domains, optimization queries can take one of two flavors:

- a *linguistic flavor*, consisting in finding the most preferable outcome defined over the linguistic term sets, or
- a *fuzzy logic flavor*, consisting in finding the most preferable outcome defined over the (infinite and continuous) set of fuzzy subsets for every variable.

Note that finding the optimal crisp outcome is just a special case of the second flavor, albeit a bit simpler as crisp values are represented by singleton fuzzy subsets.

As LCP-nets are defined over linguistic variables and foster a qualitative assessment of preferences, the optimization queries tackled in this section is from the first flavor. Hence, the problem statement is: given a LCP-net $\mathcal{L}$, find the optimal outcome $o = \langle v_1, ..., v_p \rangle$ such that $v_1 \in T_{V_1}, \ldots, v_p \in T_{V_p}$.

6.2.2 Forward sweeping over LCP-nets

Performing optimization queries over LCP-nets inherits much of the properties of UCP-nets. As for UCP-nets, the fact that LCP-nets satisfy the CP-condition enables a forward sweep procedure to optimize each variable in turn from the outmost to the inmost level in the DAG. In UCP-nets, a simple topological sort of the nodes in the DAG produces an order in which variables can be optimized. In LCP-nets, it is not as simple, because of the ci-arcs, which are undirected and becomes directed i-arcs only when the variables they depend upon receive values. Such values will only be known during the forward sweep, so the topological sort must be done in parallel with the optimization.

Thus the forward sweep algorithm for LCP-nets becomes the following. Let $\mathbf{LV}, \mathbf{cp}, \mathbf{i}, \mathbf{ci}$ be initially the whole set of variables, cp-arcs, i-arcs and ci-arcs respectively in the LCP-net, then

1. Extract a variable LV_i from $\mathbf{LV}$ such that for all $LV_j \in \mathbf{LV} \setminus \{LV_i\}$
   - $\overrightarrow{(LV_j, LV_i)} \notin \mathbf{cp}$

- $\overrightarrow{(LV_j, LV_i)} \notin \mathbf{i}$
- $(LV_j, LV_i) \notin \mathbf{ci}$

2. Let $LV_i = \langle V_i, \mathcal{D}(V_i), T_{V_i} \rangle$, find $v_i \in T_{V_i}$ that gives the highest utility in the CPT of V_i given the partial assignment to the previous variables $LV_1, ..., LV_{i-1}$.

3. For each ci-arc in **ci** which selector set $\mathbf{LV}_k \subseteq \{LV_1, ..., LV_i\}$ of variables already optimized, convert the ci-arc to the i-arc selected from the optimal assignment to $\mathbf{LV}_k$.

4. Delete from **cp** and **i** all arcs originating from LV_i.

5. Repeat steps 1 to 4 with $\mathbf{LV} \setminus \{LV_i\}$ until it becomes empty.

This algorithm gives the optimal outcome $\langle v_1, ..., v_p \rangle$, subject to the proper implementation of step 2, which is tackled in the next subsection.

6.2.3 Abductive reasoning over CPT

The step 2 in the forward sweep algorithm calls for finding the values that will give the best utility. Because of the fuzzy inference systems, this turns out to be a complex task. To do this, we have to reverse the inferences in order to obtain the linguistic term defined by $\mu_{S_{V_{m+1}}}(x)$ that gives the best possible conclusion, knowing that the m first nodes have been already assigned. This depends on the way (*i.e.*, with a triangular conorm or norm) the aggregation of the rules is performed.

In our running example, it should be a purchase that would propose P_L as price, given that delivery_time equals to D_{imm}.

This reverse inference is an abduction problem. Peirce (1839 – 1914), a famous logician, defined the abduction this way: in case "C is true if A is true", and C is observed (C is called the *manifestation*), then there is some reasons to think A may be true.

Since, many works have focused on this problem. Miyata *et al.* define cause-and-effect relationships. They try to give a definition of maximum and minimum fuzzy sets which can explain the manifestations (Miyata et al., 1995). In our case, the manifestation is the best outcome. Revault d'Allonnes *et al.* also tried to construct a set of likely explanations for a manifestation (Revault d'Allonnes et al., 2007), but they noticed that it is very hard to extend formal fuzzy abductive results to different classes of implications. A set of explanations can be constructed only for 'deduction-coherent' implications, not for all the implications (Revault d'Allonnes et al., 2009).

All these studies show us that it is not possible to prove the outcome optimization query without fixing some conditions on different entities, such as:

- the shapes of all the fuzzy sets (considering only linguistic 2-tuples shall be a great simplification),
- the implication operators,
- the operators used to aggregate the rule conclusions,
- the operator Agg that aggregates the local utilities lu thanks to vector W.

The conditions to transform this general abduction problem into a simpler one to solve the optimization queries are the following:

(C1) As in the CP-nets, we always want clear preferences, *i.e.*, no indifference in the CPTs (the highest preference among the different values of the variable is always *unique*, for any partial assignment of its parents).

(C2) All the observable values are bounded and the whole set of values they can take is represented in the CPTs. The union of the whole values they can take covers the entire universe, instead of being strictly included in it.

(C3) All the variables are expressed through fuzzy sets or linguistic 2-tuples that fulfill Ruspini condition (Ruspini, 1969), *i.e.*, that get into a well-formed partition.

Under these conditions, the *implications* of the rules become *equivalences* in the case where conclusions are equal to the highest preferences (and only in this case). So the optimization query problem becomes: for each table, look for the tuple of fuzzy-sets (or linguistic 2-tuples) among the premises that maximize the user preferences. Inside the CPT of node X, knowing the (best) values taken by $Pa(X)$, we only have to search for the highest preference (we recall that it is unique, according to (C1)) in one dimension (the one of X). This permits to abduce the (best) associated value for X (we recall that the implications can be considered as equivalences according to (C2)) and to save this value.

7 Conclusion

In previous works, we have proposed the LCP-nets, a new model to express conditional preferences over variables of continuous domains in a qualitative way, thanks to the linguistic modeling of both the problem variables and the utilities expressing user preferences. In this paper we have established LCP-nets on a firmer ground by formally defining their structure, their semantics, and their validity. We have also formalized the dominance testing and optimization queries (for a discretized version of the problem in this latter case), in the line of previous CP-nets models.

For LCP-nets themselves, future work will essentially have to address the optimization query and the hypothesis that we have put on it. First, we will need to complete the current assessment of the conditions to be put on the inference process used to map outcomes to local utilities through their conditional preference tables in order to guarantee correctness of the global optimization. Also, we will address the extension of the optimization query to the continuous outcome case.

Another line of work is the comparison of LCP-nets with other models for expressing conditional preferences, especially at the frontier between the CP-net family of models and the fuzzy preference models. As LCP-nets bridges the gap between the two worlds, we conjecture that they will allow for a more precise comparison between the two, in order to better characterize their respective limitations, advantages and disadvantages.

References

Abchir, M. (2011). A jFuzzyLogic Extension to Deal With Unbalanced Linguistic Term Sets. In *Proc. of the 12th International Student Conference on Applied Mathematics and Informatics (ISCAMI'11)*, pages 54–55.

Boubekeur, F. and Tamine-Lechani, L. (2006). Recherche d'information flexible basée CP-Nets. In *Proc. Conference on Recherche d'Information et Applications (CORIA'06)*, pages 161–167.

Boutilier, C., Bacchus, F., and Brafman, R. I. (2001). UCP-Networks: A directed graphical representation of conditional utilities. In *Proc. of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., and Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional *Ceteris Paribus* Preference Statements. *Journal of Artificial Intelligence Research*, 21:135–191.

Brafman, R. I. and Domshlak, C. (2002). Introducing variable importance tradeoffs into CP-nets. In *Proc. of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI'02)*, pages 69–76.

Brafman, R. I., Domshlak, C., and Shimony, S. E. (2006). On graphical modeling of preference and importance. *J. Artif. Intell. Res. (JAIR)*, 25:389–424.

Braziunas, D. and Boutilier, C. (2006). Preference elicitation and generalized additive utility. In *Proc. of the Twenty-First National Conference on Artificial Intelligence (AAAI'06)*, pages 1573–1576, Boston, MA.

Châtel, P., Malenfant, J., and Truck, I. (2010a). QoS-based Late-Binding of Service Invocations in Adaptive Business Processes. In *The 8th International Conference on Web Services (ICWS'10)*, pages 227–234.

Châtel, P., Truck, I., and Malenfant, J. (2008). A linguistic approach for non-functional preferences in a semantic SOA environment. In *The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 889–894.

Châtel, P., Truck, I., and Malenfant, J. (2010b). LCP-nets: A linguistic approach for non-functional preferences in a semantic SOA environment. *Journal of Universal Computer Science*, 16(1):198–217.

Curry, B. and Lazzari, L. (2009). Fuzzy consideration sets: a new approach based on direct use of consumer preferences. *International Journal of Applied Management Science*, 1(4):420–436.

Delgado, M., Verdegay, J., and Vila, M. (1993). On Aggregation Operations of Linguistic Labels. *International Journal of Intelligent Systems*, 8:351–370.

Gentzen, G. (1935). Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210.

Herrera, F. and Martínez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans. Fuzzy Systems*, 8(6):746–752.

Miyata, Y., Furuhashi, T., and Uchikawa, Y. (1995). A study on fuzzy abductive inference. In *Proc. of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, pages 337–342.

Revault d'Allonnes, A., Akdag, H., and Bouchon-Meunier, B. (2007). Selecting implications in fuzzy abductive problems. In *IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 597–602.

Revault d'Allonnes, A., Akdag, H., and Bouchon-Meunier, B. (2009). For a data-driven interpretation of rules, wrt gma conclusions, in abductive problems. *Journal of Uncertain Systems*, 3(4):280–297.

Ruspini, H. (1969). A New Approach to Clustering. *Information and Control*, 15:22–32.

Yager, R. (1988). On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190.

Yager, R. (2007). Using Stress Functions to Obtain OWA Operators. *IEEE Trans. on Fuzzy Systems*, 15(6):1122–1129.

Zadeh, L. A. (1965). Fuzzy sets. *Information Control*, 8:338–353.

Note

[1]Cp-arcs are symbolized by a simple arrow, i-arcs by an arrow with a black triangle ▶ on it and ci-arcs by a line with a black square ■ on it.

[2]modified through linguistic modifiers.

[3]In our implementation, the obtained weight vector verifies this criterion.