Hybrid Possibilistic Networks

Salem Benferhat and **Salma Smaoui**
CRIL,Universite d'Artois , Rue Jean Souvraz SP 18, 62300 Lens, France
{benferhat,smaoui}@cril.univ-artois.fr

Abstract

Possibilistic networks are important tools for dealing with uncertain pieces of information. For multiply-connected networks, it is well known that the inference process is a hard problem. This paper studies a new representation of possibilistic networks, called hybrid possibilistic networks. The uncertainty is no longer represented by local conditional possibility distributions, but by their compact representations which are possibilistic knowledge bases. We show that the inference algorithm in hybrid networks is strictly more efficient than the ones of standard propagation algorithm.

Introduction

Probabilistic networks (Pearl 1988; Jensen 1996; Lauritzen & Spiegelhalter 1988) and possibilistic networks (Fonck 1994; Borgelt, Gebhardt, & Kruse 1998) are important tools proposed for an efficient representation and analysis of uncertain information. Their success is due to their simplicity and their capacity of representing and handling independence relationships which are important for an efficient management of uncertain pieces of information. Possibilistic networks and possibilistic logic have been successively applied in many domains such as fault detection (Cayrac *et al.* 1996).

Possibilistic networks are directed acyclic graphs (DAG), where each node encodes a variable and every edge represents a relationship between two variables. Uncertainties are expressed by means of conditional possibility distributions for each node in the context of its parents.

The inference in possibilistic graphs, as in probabilistic graphs, depends on the structure of a DAG. For instance, for simply connected graphs, the inference process can be achieved in a polynomial time. However, for multiply connected graphs (where between two nodes, it may exists more than one path), the propagation algorithm is expensive and generally requires a graphical transformation from the initial graph into another tree structure such as a junction tree. Nodes in this tree are sets of variables called clusters. The propagation algorithm efficiency depends on clusters' size, and the space complexity is function of cartesian product of clusters variables' domains.

This paper first proposes a new representation of uncertain information in possibilistic networks, called hybrid possibilistic graphs. Local uncertainty is no longer represented by conditional possibility distributions but by possibilistic knowledge bases. This representation generalizes the two well-known standard representations of uncertainty in possibility theory: possibilistic knowledge bases and possibilistic networks. Then we propose an extension to the junction tree algorithm for hybrid possibilistic networks using an extension of the notion of forgetting variables, proposed in (Lang & Marquis 1998; Darwiche & Marquis 2004) for computing marginal distributions.

The main advantage of this propagation algorithm concerns space complexity. Namely, during the junction tree construction, it may happen that the size of clusters can be very large. In such case, in standard possibilistic networks, it may be impossible to produce local possibility distributions associated with clusters. Our algorithm enables us to propagate uncertainty even with large clusters.

The rest of this paper is organized as follows: first, we give a brief background on possibilistic logic and propagation algorithms for standard possibilistic multiply-connected networks (section 2). Then, we present our new representation of possibilistic networks (Section 3). Section 4 introduces the prioritized forgetting variable. The adaptation of the propagation algorithm for multiply connected graphs is proposed in Section 5. Experimental results are presented in Section 6.

Possibilistic logic and possibilistic networks

Possibility distributions

Let $V = \{A_1, A_2, ..., A_n\}$ be a set of variables. D_{A_i} denotes the finite domain associated with the variable A_i. For the sake of simplicity, and without lost of generality, variables considered here are assumed to be binary. a_i denotes any of the two instances of A_i and $\neg a_i$ represents the other instance of A_i. $\varphi, \psi, ..$ denote propositional formulas obtained from V and logical connectors $\wedge$ (conjunction),$\vee$ (disjunction),$\neg$ (propositional negation). $\top$ and $\bot$, respectively, denote tautologies and contradictions.

$\Omega = \times_{A_i \in V} D_{A_i}$ represents the universe of discourse and ω, an element of Ω, is called an *interpretation*. It is either denoted by tuples $(a_1, ..., a_n)$ or by conjunctions $(a_1 \wedge ... \wedge a_n)$, where a_i's are respectively instance of A_i's. In the following, $\models$ denotes the propositional logic satisfaction. $\omega \models \varphi$ means that ω is a model of φ.

A possibility distribution π (Zadeh 1975) is a mapping $\Omega \rightarrow$

$[0,1]$. $\pi(\omega)$ denotes the compatibility degree of an interpretation ω with available pieces of information. By convention, $\pi(\omega)=0$ means that ω is impossible. $\pi(\omega)=1$ means that ω is totally possible. $\pi(\omega)>\pi(\omega')$ means that ω is preferred to ω'. Given a possibility distribution π, two dual measures are defined:

- A possibility measure of a formula φ:

$$\Pi(\varphi)=\max\{\pi(\omega):\omega\models\varphi\}$$

which represents the compatibility degree of φ with available pieces of information encoded by π.

- A necessity measure of a formula φ:

$$N(\varphi)=1-\Pi(\neg\varphi)$$

which corresponds to the certainty degree associated with φ from available pieces of information encoded by π.

Lastly, conditioning (Hisdal 1978) is defined by :

$$\pi(\omega\mid\phi)=\begin{cases}1 & if\ \pi(\omega)=\Pi(\phi)\text{ and }\omega\models\phi\\ \pi(\omega) & if\ \pi(\omega)<\Pi(\phi)\text{ and }\omega\models\phi\\ 0 & otherwise.\end{cases}\qquad(1)$$

Possibilistic knowledge bases

A possibilistic knowledge base (Dubois, Lang, & Prade 1994) is a finite set of weighted formulas $\Sigma=\{(\varphi_i,\alpha_i),i=1,...,m\}$, where φ_i is a propositional formula and $\alpha_i\in[0,1]$. (φ_i,α_i) can be viewed as a constraint stating that the certainty degree of φ_i is at least equal to α_i, namely $N(\varphi_i)\geq\alpha_i$.

Possibilistic knowledge bases are compact representations of possibility distributions. Namely, each possibilistic knowledge base induces a unique possibility distribution, defined by (Dubois, Lang, & Prade 1994): $\forall\omega\in\Omega$,

$$\pi_\Sigma(\omega)=\begin{cases}1 & if\ \forall(\varphi_i,\alpha_i)\in\Sigma,\ \omega\models\varphi_i,\\ 1-max\{\alpha_i:\omega\not\models\varphi_i\} & otherwise.\end{cases}\qquad(2)$$

The following definitions are useful for the rest of the paper:

Definition 1 *Two possibilistic knowledge bases Σ_1 and Σ_2 are said to be equivalent if their associated possibility distributions are equal, namely :*

$$\forall\omega\in\Omega,\quad\pi_{\Sigma_1}(\omega)=\pi_{\Sigma_2}(\omega)$$

Subsumption definition is as follows :

Definition 2 *Let (φ,α) a formula in Σ. Then (φ,α) is said to be subsumed by Σ if Σ and $\Sigma\backslash\{(\varphi,\alpha)\}$ are equivalent knowledge bases.*

Namely, subsumed formulas are redundant formulas that can be removed without changing possibility distributions.

Standard possibilistic networks

Possibilistic networks (Fonck 1994; Borgelt, Gebhardt, & Kruse 1998), denoted by ΠG, are directed acyclic graphs (DAG). Nodes correspond to variables and edges encode relationships between variables. A node A_j is said to be a parent of A_i if there exists an edge from the node A_j to the node A_i. Parents of A_i are denoted by U_{A_i}.

Uncertainty is represented at each node by local conditional possibility distributions. More precisely, for each variable A:

If A is a root, namely $U_A=\emptyset$, then we provide $\Pi(a)$ and $\Pi(\neg a)$ with $\max(\pi(a),\pi(\neg a))=1$.

If A has parents, namely $U_A\neq\emptyset$, then we provide $\Pi(a\mid u_A)$ and $\Pi(\neg a\mid u_A)$, with $\max(\pi(a|U_A),\pi(\neg a|U_A))=1$, for each $u_A\in D_{U_A}$, where D_{U_A} is the cartesian product of domains of variables which are parents of A.

Possibilistic networks are also compact representations of possibility distributions. More precisely, joint possibility distributions associated with possibilistic network are computed using a so-called possibilistic chain rule similar to the probabilistic one, namely :

$$\pi_{\Pi G}(a_1,...,a_n)=\min_{i=1..n}\Pi(a_i\mid u_{A_i}),\qquad(3)$$

where a_i is an instance of A_i and $u_{A_i}\subseteq\{a_1,...,a_n\}$ is an element of the cartesian product of domains associated with variables U_{A_i} which are parents of A_i.

Example 1 *Figure 1 gives an example of possibilistic networks. Table 1 provides local conditional possibility distributions of each node given its parents.*

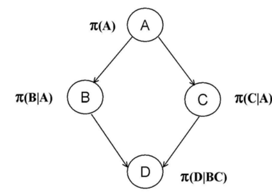

Figure 1: Example of possibilistic causal network ΠG

a	$\frac{1}{4}$
$\neg a$	1

$B\|A$	a	$\neg a$
b	$\frac{1}{4}$	$\frac{1}{4}$
$\neg b$	1	1

$C\|A$	a	$\neg a$
c	1	$\frac{1}{2}$
$\neg c$	$\frac{3}{4}$	1

$D\|BC$	bc	$\neg bc$	else
d	1	$\frac{1}{4}$	1
$\neg d$	$\frac{1}{2}$	1	1

Table 1: Local conditional possibility distributions associated with DAG of Figure 1

Using possibilistic chain rule, the possibility degree of $\pi(\neg ab\neg cd)$ is computed as follows :

$$\begin{aligned}\pi(\neg ab\neg cd)&=\min(\pi(\neg a),\pi(b|\neg a),\pi(\neg c|\neg a),\pi(d|b\neg c))\\&=\min(1,\tfrac{1}{4},1,1)=\tfrac{1}{4}\end{aligned}$$

Propagation in possibilistic multiply connected networks

Propagation algorithms aim to establish a posteriori possibility distributions of each node A given some evidence on a set of variables E. When DAGs are singly connected then the propagation algorithm is polynomial. In this section, we only focus on multiply connected graphs.

A well-known algorithm for multiply connected graphs proceeds to a transformation of the initial graph into a junction tree. The main idea is to delete loops from the initial graph gathering some variables in a same node. The resulting graph is a tree where each node, called cluster, is a set

of variables. Common variables of two adjacent clusters are grouped into another type of node, called a separator.
Figure 2 gives an example of a junction tree associated with the DAG of figure 1.
The propagation algorithm is then applied on this resulting

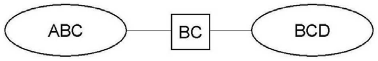

Figure 2: Junction tree associated with graph ΠG of figure 1

structure. The idea is to require that adjacent clusters sharing common variables should have the same marginal distributions associated with these common variables namely on their separator. The main steps of the junction tree propagation algorithm are (For more details see (Fonck 1994; Borgelt, Gebhardt, & Kruse 1998)):

- **Step S1 :** ***Standard initialization***
  This step initializes possibility distributions associated with clusters and separators using local possibility distributions in the initial DAG.
  - For each cluster C_i : $\pi^I_{C_i} \leftarrow \mathtt{1}$,
  where 1 is a possibility distribution where all elements have a highest possibility degree $\mathtt{1}$.
  - For each separator S_{ij} : $\pi^I_{S_{ij}} \leftarrow \mathtt{1}$,
  - For each variable A, select a cluster C_i containing $A \cup U_A$ and update its possibility distributions as follows :
  $\pi^I_{C_i}$: $\pi^I_{C_i} \leftarrow min(\pi^I_{C_i}, \Pi(A \mid U_A))$.
- **Step S2 :** ***Standard handling of evidence***
  If $A = a$ (evidence), then select a cluster C_i containing A, and update its possibility distribution as follows:
  $$\pi^I(\omega) = \min(\pi^I(\omega), \pi_a(\omega))$$
  where $\pi_a(\omega)$ is defined :
  $$\pi_a(\omega) = \begin{cases} 1 & if\ \omega \models a \\ 0 & otherwise. \end{cases} \quad (4)$$
- **Step S3 :** ***Standard updating of separators***
  Each cluster computes its possibility distribution and send it to the adjacent separator. The separator's distribution, denoted $\pi^{t+1}_{S_{ij}}$, is then updated as follows:
  $$\pi^{t+1}_{S_{ij}} \leftarrow \max_{C_i \setminus S_{ij}} \pi^t_{C_i}. \quad (5)$$
- **Step S4 :** ***Standard updating of clusters***
  Each cluster updates its possibility distribution, denoted $\pi^{t+1}_{C_j}$, when receiving a message from its adjacent separator as follows :
  $$\pi^{t+1}_{C_j} \leftarrow min(\pi^t_{C_j}, \pi^t_{S_{ij}}). \quad (6)$$
  Steps S3 and S4 are repeated until the junction tree is globally consistent, namely adjacent clusters should have same marginal distributions over common variables.
- **Step S5 :** ***Computing queries***
  When the junction tree is consistent, computing $\Pi(A = a)$ consists in selecting any cluster containing A and marginalizing Π_{C_i} on A:
  $$\Pi(A = a) = \Pi_{C_i}(A = a)$$

Possibilistic networks with local knowledge bases

Definition of hybrid graphs

Pieces of information can be either provided in terms of possibilistic knowledge bases or in terms of conditional possibilities. They can also be either represented using graphical structures or logic-based structures. The aim of the new representation is to take advantage of these two possible representation formats (as it has for instance been done in other frameworks (Wilson & Mengin 2001)). A graphical representation is used to take advantage of independence relations, and a logic-based representation is used to have compact representation of possibility distributions.
In this paper, we propose a new structure called hybrid possibilistic graphs. More precisely, hybrid possibilistic causal networks, denoted HG, are characterized by (see figure 3):

- *A graphical component* which is represented by a DAG (like standard possibilistic causal networks) that allows to represent independence relationships.
- *A quantitative component* which encodes uncertainties. It associates to each node a local bases instead of a conditional possibility distribution. Namely, at each node A_i, one provides a possibilistic knowledge base Σ_{A_i} which represents local knowledge base on A and its parents.

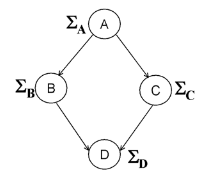

Figure 3: Hybrid graph HG with local knowledge bases

Hybrid graphs are also compact representation of joint possibility distributions. A possibility distribution associated with a hybrid possibilistic network HG is defined by:
$$\forall \omega,\ \pi_{HG}(\omega) = \min_{A_i \in V} \pi_{\Sigma_{A_i}}(\omega) \quad (7)$$
where $\pi_{\Sigma_{A_i}}$ is the possibility distributions associated with Σ_{A_i} obtained using equation (2).

From possibilistic bases and standard possibilistic networks to hybrid possibilistic networks

A hybrid representation is a general framework which records the standard ones recalled in the previous section. Namely, any possibilistic network ΠG (where local uncertainty is represented by a possibility distribution) or any possibilistic knowledge base, can be represented by hybrid networks HG. In (Benferhat & Smaoui 2005), an immediate encoding of possibilistic logic base into hybrid networks is provided. In this section, we give the encoding of hybrid networks.
Let ΠG be a standard possibilistic causal networks. Let A be a variable, and $\pi(a_i|u_i)$ be a local possibility degree associated with A. Then the hybrid possibilistic network HG associated with ΠG is obtained in the following way: for each A, define
$$\Sigma_A = \{(\neg a_i \vee \neg u_i, \alpha_i) : \alpha_i = 1 - \pi(a_i|u_i) \neq 0\}. \quad (8)$$

Then,

Proposition 1 *Let ΠG be a standard possibilistic network. Let HG be a hybrid network, having a same DAG, and where Σ_{A_i}'s are obtained using equation (8), then,*

$$\pi_{\Pi G}(\omega) = \pi_{HG}(\omega) \tag{9}$$

where $\pi_{\Pi G}$ and π_{HG} are obtained by using (3) and (7).

Example 2 *Let us build a hybrid possibilistic causal networks HG from standard possibilistic causal network ΠG of example 1 by associating knowledge bases to each node using 8. Uncertainty at the level of nodes A, B, C and D (binary variables) is represented by possibilistic knowledge bases Σ_A, Σ_B, Σ_C and Σ_D as follows:*
$\Sigma_A = \{(\neg a, \frac{3}{4})\}$
$\Sigma_B = \{(\neg a \vee \neg b, \frac{3}{4}), (a \vee \neg b, \frac{3}{4})\}$
$\Sigma_C = \{(a \vee \neg c, \frac{1}{2}), (\neg a \vee c, \frac{1}{4})\}$
$\Sigma_D = \{(b \vee \neg c \vee \neg d, \frac{3}{4}), (\neg b \vee \neg c \vee d, \frac{1}{2})\}$
We can check that, $\forall \omega$, $\pi_{\Pi G}(\omega) = \pi_{HG}(\omega)$. For instance,
$\pi_{HG}(\neg ab\neg cd) = \min(\pi_{\Sigma_A}(\neg ab\neg cd), \pi_{\Sigma_B}(\neg ab\neg cd),$
$\pi_{\Sigma_C}(\neg ab\neg cd), \pi_{\Sigma_D}(\neg ab\neg cd)) = \min(1, \frac{1}{4}, 1, 1) = \frac{1}{4}$.
which is the same as the one given in example 1.

Prioritized forgetting : A syntactic computation of marginalization

Lin and Reiter (1994) proposed an approach allowing variable domain restriction in propositional knowledge bases (see (Lang & Marquis 1998; Darwiche & Marquis 2004) for details). Variable forgetting (also known as projection or marginalization) is defined as:

Definition 3 *Let K be a propositional knowledge base and X be a propositional variable set. The forgetting of X in K, noted $forget(K, X)$, is equivalent to a propositional formula that can be inductively defined as follows :*
• $forget(K, \emptyset) = K$.
• $forget(K, \{x\}) = K_{x\leftarrow\bot} \vee K_{x\leftarrow\top}$.
• $forget(K, X \cup \{x\}) = forget(forget(K, X), \{x\})$.
where $K_{x\leftarrow\bot}$ (resp. $K_{x\leftarrow\top}$) refers to K in which we affect false (resp. true) value to each occurrence of x (instance of X). By $K_i \vee K_j$ we mean the set $\{(\varphi_i \vee \psi_j) : \varphi_i \in K_i$ and $\psi_j \in K_j\}$.

This approach is defined for classical propositional logic. We present an extension of this definition, called prioritized forgetting, which deals with possibilistic knowledge bases.
Let Σ_1 and Σ_2 be two possibilistic knowledge bases. The disjunction of two bases in possibilistic framework, denoted by $\oslash$, is defined as follows :
$\Sigma_1 \oslash \Sigma_2 = \{(\varphi_i \vee \psi_j, \min(\alpha_i, \beta_j)) : (\varphi_i, \alpha_i) \in \Sigma_1$ and $(\psi_j, \beta_j) \in \Sigma_2\}$
Prioritized forgetting, denoted $pforget$, can then be defined as follows:

Definition 4 *Let Σ be a possibilistic knowledge base and X be a variable set. The prioritized forgetting of X in Σ, denoted $pforget(\Sigma, X)$, is equivalent to a possibilistic formula defined as follows :*
• $pforget(\Sigma, \emptyset) = \Sigma$,
• $pforget(\Sigma, \{x\}) = K_{x\leftarrow\bot} \oslash K_{x\leftarrow\top}$
• $pforget(\Sigma, X \cup \{x\}) = pforget(pforget(\Sigma, X), \{x\})$.

Prioritized forgetting allows to syntacticly capture the base associated with marginal distributions. More precisely:

Proposition 2 *Let Σ be a possibilistic knowledge base and π its associated distribution. Let X be a set of variables. Then the possibility distribution associated with $pforget(\Sigma, X)$ is :*

$$\pi_{pforget(\Sigma,X)} = \max_{V \setminus X} \pi_\Sigma \tag{10}$$

Propagation in multiply connected graphs

One of the limits of junction tree algorithm is that the transformation step of initial multiply connected graphs can produce clusters with a great number of variables. In that case, it may be impossible to get local joint possibility distributions on clusters.
The aim of this section is to propose an alternative propagation algorithm in junction trees. For propagation algorithms on hybrid singly-connected networks see (Benferhat & Smaoui 2005).
We call a hybrid junction tree, denoted HJT, a junction tree where uncertainty is represented over clusters by possibilistic knowledge bases, instead of possibility distributions.
The main steps of the new junction tree algorithm are summerized in the following figure:

```
Procedure Hybrid junction tree propagation
Begin
  - Junction tree construction from the initial graph
  - Apply step S1 : Standard initialization
  If (Standard initialization succeeds) then
     - Apply step S2 : Standard handling of evidence,
     While (Junction tree is not consistent) do
        - Apply step S3 : Standard updating separators,
        - Apply step S4 : Standard updating clusters.
     done
  else
     - For each variable A, compute local knowledge base π_{Σ_A},
     - Apply step H1 : Hybrid initialization,
     While (Junction tree is not consistent) do
        - Apply step H2 : Hybrid handling of evidence,
        - Apply step H3 : Hybrid updating separators,
        - Apply step H4 : Hybrid updating clusters.
     done
     Steps H1, H2, H3, H4 and H5 are detailed below.
  end If
End
```

The idea of the algorithm is the following : if in the initialization step, the size of cluster is not very large then standard propagation steps ($S1 - S5$) described in the previous section are used. Now, if it is impossible to represent distribution over clusters, then we use alternative steps ($H1 - H5$) described below. These steps give the counterparts of ($S1 - S5$) for possibilistic knowledge bases.

Step H1 : Hybrid initialization

This step consists of initializing the junction tree by assigning knowledge bases to clusters and separators.

- An empty knowledge base Σ_{C_i} is first assigned to each cluster C_i.

$$\Sigma_{C_i} \leftarrow \emptyset$$

- An empty knowledge base $\Sigma_{S_{ij}}$ is also assigned to each separator S_{ij}.
$$\Sigma_{S_{ij}} \leftarrow \emptyset$$
- For each variable A, select a cluster C_i containing $\{A\} \cup U_A$ and add to the knowledge base Σ_{C_i} the possibilistic base Σ_A associated with A.
$$\Sigma_{C_i} \leftarrow \Sigma_{C_i} \cup \Sigma_A$$

Proposition 3 *Let HG be an hybrid possibilistic causal network. Let HJT be the junction tree associated with HG. Let $\{\Sigma_{C_i} : i = 1, ..., n\}$ be the knowledges bases associated with clusters $\{C_i : i = 1, ..., n\}$ at the end of the initialization step. Then we have:*
$$\pi_{HG} = \min_{C_i} \pi_{\Sigma_{C_i}}$$

Example 3 *Given the junction tree of Figure 4, local knowledge bases on clusters after the initialization step are:*
$\Sigma_{C_1} = \Sigma_A \cup \Sigma_B \cup \Sigma_C = \{(\neg a, \frac{3}{4}), (\neg a \vee \neg b, \frac{3}{4}),$
$(a \vee \neg b, \frac{3}{4}), (a \vee \neg c, \frac{1}{2}), (\neg a \vee c, \frac{1}{4})\}$
$\Sigma_{C_2} = \Sigma_D = \{(b \vee \neg c \vee \neg d, \frac{3}{4}), (\neg b \vee \neg c \vee d, \frac{1}{2})\}$
Let us consider the interpretation $\omega = \neg ab \neg cd$. We have :
$\pi_{HG}(\neg ab \neg cd) = \min(\pi_{\Sigma_{C_1}}(\neg ab \neg c), \pi_{\Sigma_{C_2}}(b \neg cd)) = \frac{1}{4}$
which is the same as the one obtained from example 1.

After the initialization step, messages are sent between clusters in order to guarantee the consistency conditions. If, for instance, for a given two clusters C_i and C_j, we have :
$$\max_{C_i \backslash S_{ij}} \pi_{C_i} \neq \max_{C_j \backslash S_{ij}} \pi_{C_j},$$
then C_i and C_j should update their knowledge bases iteratively. The following two elementary steps are repeated until reaching consistency:
- A separator S_{ij} computes its knowledge base from C_i (resp. C_j).
- A cluster C_j (resp. C_i) updates its knowledge base taking into account the knowledge base of the separator previously computed.

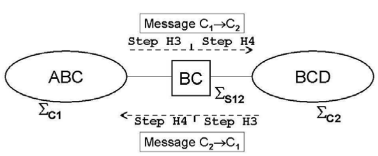

Figure 4: Message passing in the junction tree HJT

Step H2 : Hybrid handling evidence

If there are some observations (evidence), then for any observed variable $A = a$ select a cluster containing the variable A, and add the possibilistic formula $(a, 1)$ to the knowledge base associated with this cluster.

Step H3 : Hybrid updating separators

The knowledge base $\Sigma_{S_{ij}}$, associated with a separator S_{ij}, represents the restriction (marginalization) of the base Σ_{C_i} (resp. Σ_{C_j}) on common variables in the separator S_{ij}. This knowledge base is immediately obtained thanks to Proposition 2.
Let V' be the set of variables in $C_i \backslash S_{ij}$. Then,
$$\Sigma_{S_{ij}} = pforget(\Sigma_{C_i}, V')$$

Example 4 *Let us compute the knowledge base $\Sigma_{S_{12}}$, associated with the separator S_{12} from Σ_{C_1}. This leads to forgetting the variable A. Let us apply the definition of pforget:*
- $\Sigma_{a \leftarrow \perp} = \{(\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$
- $\Sigma_{a \leftarrow \top} = \{(\perp, \frac{3}{4}), (\neg b, \frac{3}{4}), (c, \frac{1}{4})\}$

$$\begin{aligned} \Sigma_{S_{12}} &= pforget(\Sigma_{C_1}, \{A\}) \\ &= \{(\neg b, \tfrac{3}{4}), (\neg c, \tfrac{1}{2}), (\neg b \vee \neg c, \tfrac{1}{2}), (\neg b \vee c, \tfrac{1}{4})\} \\ &= \{(\neg b, \tfrac{3}{4}), (\neg c, \tfrac{1}{2})\}. \end{aligned}$$
$(\neg b \vee \neg c, \frac{1}{2})$ *and* $(\neg b \vee c, \frac{1}{4})$ *are subsumed formulas.*

Step H4 : Hybrid updating clusters

When receiving messages from separator S_{ij}, the cluster C_i updates it's knowledge base as follows :
$$\Sigma_{C_j} \leftarrow \Sigma_{S_{ij}} \cup \Sigma_{C_j} \quad (11)$$
This is justified by the following proposition (Benferhat *et al.* 1999) :

Proposition 4 *Let HG be an hybrid possibilistic causal network. Let HJT be a junction tree associated with HG. Let $\{\Sigma_{C_i} : i = 1, ..., n\}$ be the knowledge bases associated with clusters $\{C_i : i = 1, ..., n\}$ after each updating step. Then, we have : $\forall \omega$,*
$$\pi_{HG}(\omega) = min_{C_i} \pi_{\Sigma_{C_i}}(\omega)$$

The steps of updating separators and clusters knowledge bases are repeated until reaching stability (global consistency) in the junction tree. Formally, HJT is consistent if $\forall i, j$, we have:
$$\pi_{\Sigma_{S_{ij}}} = \max_{C_i \backslash S_{ij}} \pi_{\Sigma_{C_i}} = \max_{C_j \backslash S_{ij}} \pi_{\Sigma_{C_j}} \quad (12)$$

Example 5 *The knowledge base Σ_{C_2} associated with the cluster C_2 after receiving $\Sigma_{S_{12}}$ is :*
$\Sigma_{C_2} = \Sigma_{C_2} \cup \Sigma_{S_{12}} = \{(b \vee \neg c \vee \neg d, \frac{3}{4}), (\neg b \vee \neg c \vee d, \frac{1}{2}), (\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$ *which is equivalent to* $\Sigma_{C_2} = \{(b \vee \neg c \vee \neg d, \frac{3}{4}), (\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$.
At the end of propagation process, we obtain the following local knowledge bases:
- $\Sigma_{C_1} = \{(\neg a, \frac{3}{4}), (\neg b, \frac{3}{4}), (\neg c, \frac{1}{2})\}$.
- $\Sigma_{C_2} = \{(\neg b, \frac{3}{4}), (\neg c, \frac{1}{2}), (b \vee \neg c \vee \neg d, \frac{3}{4})\}$.

It can be checked that HJT is consistent.

H5 : Hybrid computing queries

When the junction tree is consistent, computing $\Pi(A)$ is done syntacticly using possibilistic inference:

Proposition 5 *Let Σ be a possibilistic knowledge base. Let a be an instance of A. Then,*
$$\pi(a) = 1 - Inc(\Sigma \cup \{(a, 1)\})$$
where $Inc(\Sigma \cup \{(a, 1)\})$ is the inconsistency degree of $\Sigma \cup \{(a, 1)\}$. For computing the inconsistency degree Inc see (Dubois, Lang, & Prade 1994).

Experiments

In the previous section, it is clear that our algorithm is an improvement of standard junction tree propagation, since steps $H1 - H5$ are run only if it is not possible to initialize the junction tree with explicit local conditional possibility distributions. In this section, we present experimental results for the proposed possibilistic propagation algorithm. These experimentations show that our algorithm is a real improvement, since we identify several examples where standard junction tree blocks, while our algorithm provides answers. The experimentation was conducted on sets of possibilistic networks randomly-generated. DAGs are generated randomly by varying number of nodes and the maximum number of parents. We define links ratio to be the average number of links per node in the graph. Local conditional distributions on each node in the context of its parents are also generated randomly respecting the normalization constraints. It is well-known that the performance of standard junction tree does not depend on numerical degrees assigned to interpretations. In hybrid networks, the performance of the propagation algorithms depend on possibility distributions. The smaller is the number of interpretations having possibility degrees different from 0 and 1, the more efficient is the algorithm. In our experimentation, the number of intepretations having possibility degree different of 0 and 1 is around 25%. The experimentations show that with networks containing 35 (resp $40, 50, 60$) nodes, it begins to be impossible to initialize local distributions at the level of clusters with links ratio around 4.45 (resp. $3.55, 2.72, 1.78$). Results in Table 2 are obtained by fixing the maximum number of parents to 10. In most cases, we observe that hybrid junction tree algorithm provides a result. Our new algorithm can only be limited by the running-time but never blocks. We chose to set a time-limit equal to 10000 seconds. Clearly, in

nb nodes	avg ratio of links	JT algo error	avg time hybrid	Hybrid algo error
30	4.32	0%	0.91 s	0%
35	4.42	8%	126.45 s	0%
40	4.58	55%	240.97 s	2%
45	4.55	87%	393.37 s	2%
50	4.67	100%	1535.48 s	15%

Table 2: Experimental results

many examples when standard possibilistic networks blocks our algorithm provide answers. In particularly for networks with 50 nodes, standard junction tree algorithm blocks for basically each generated networks.

Conclusion

This paper provides a new representation of possibility networks, where conditional possibility distributions are compactly represented by local possibilistic knowledge bases. We have shown that standard possibilistic graphs can be equivalently encoded in hybrid possibilistic graphs.

We then extended the notion of forgetting variables introduced in (Lin & Reiter 1994; Lang & Marquis 1998; Darwiche & Marquis 2004), and showed that this extension indeed allows the computation of marginalized knowledge base.

An adaptation of junction tree algorithm is provided. When uncertainty on clusters are described by possibilistic knowledge bases, our algorithm improves standard junction tree propagation algorithm.

Lastly, we provide experimental studies where examples, which are blocked by standard junction tree algorithm, are solved using our algorithm based on hybrid representation of possibilistic networks.

References

Benferhat, S., and Smaoui, S. 2005. Possibilistic singly-connected networks with locally weighted knowledge bases. In *proceedings of the fourth international symposium on imprecise probabilitues and their applications (to appear)*. Pittsburgh, USA: Brightdocs.

Benferhat, S.; Dubois, D.; Garcia, L.; and Prade, H. 1999. Directed possibilistic graphs and possibilistic logic . In Bouchon-Meunier, B.; Yager, R. R.; and Zadeh, L. A., eds., *Information, Uncertainty and Fusion*. Boston, Mass.: Kluwer. 365–379.

Borgelt, C.; Gebhardt, J.; and Kruse, R. 1998. Possibilistic graphical models. In *Proceedings of International School for the Synthesis of Expert Knowledge (ISSEK'98)*, 51–68.

Cayrac, D.; Dubois, D.; Haziza, M.; and Prade, H. 1996. Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application. *IEEE Trans. on Fuzzy Systems* 4(3):251–269.

Darwiche, A., and Marquis, P. 2004. Compiling propositional weighted bases. *Artif. Intell.* 157(1-2):81–113.

Dubois, D.; Lang, J.; and Prade, H. 1994. Possibilistic logic. In *Handbook on Logic in Artificial Intelligence and Logic Programming*, volume 3. Oxford University press. 439–513.

Fonck, P. 1994. *Réseaux d'inférence pour le raisonnement possibiliste*. Ph.D. Dissertation, Université de Liège, Faculté des Sciences.

Hisdal, E. 1978. Conditional possibilities independence and non interaction. *Fuzzy Sets and Systems* 1:283–297.

Jensen, F. V. 1996. *Introduction to Bayesien networks*. University college, London: UCL Press.

Lang, J., and Marquis, P. 1998. Complexity results for independence and definability. *Proceeding of the 6th International Conference on Knowledge Representation and Reasoning (KR'98)* 356–367.

Lauritzen, S. L., and Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society* 50:157–224.

Lin, F., and Reiter, R. 1994. Forget it! *Proceeding of AAAI Fall Symposium on Relevance* 154–159.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco (California): Morgan Kaufmman.

Wilson, N., and Mengin, J. 2001. Embedding logics in the local computation framework. *Journal of Applied Non-Classical Logics* 11(3-4):239–267.

Zadeh, L. A. 1975. The concept of a linguistic variable and its application to approximate reasoning. *Information science* 9:43–80.