



**HAL**  
open science

## Nouvelles méthodes pour l'inférence basées sur la compilation des réseaux possibilistes

Raouia Ayachi, Nahla Ben Amor, Salem Benferhat

► **To cite this version:**

Raouia Ayachi, Nahla Ben Amor, Salem Benferhat. Nouvelles méthodes pour l'inférence basées sur la compilation des réseaux possibilistes. 5èmes Journées Francophones sur les Réseaux Bayésiens (JFRB2010), May 2010, Nantes, France. hal-00467585

**HAL Id: hal-00467585**

**<https://hal.science/hal-00467585v1>**

Submitted on 19 Apr 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Nouvelles méthodes pour l'inférence basées sur la compilation des réseaux possibilistes

**Raouia Ayachi\*** — **Nahla Ben Amor\*** — **Salem Benferhat\*\***

\* *Laboratoire de Recherche Opérationnelle, de Décision et de Contrôle de Processus (LARODEC) ISG de Tunis, 41 rue de la liberté 2000 Le Bardo, Tunisie*

\*\* *Centre de Recherche en Informatique de Lens(CRIL) Université d'Artois, Rue Jean Souveraz, SP18 62300 Lens, France*

---

*RÉSUMÉ. Les réseaux possibilistes sont des outils importants de traitement des informations incertaines dans le cadre de la théorie des possibilités. Malgré leur importance, seule l'adaptation des arbres de jonction permet d'assurer un raisonnement exact. Dans cet article, nous présentons trois alternatives basées sur des techniques de compilation. Nous proposons en premier des adaptations possibilistes de méthodes probabilistes basées sur la compilation. Ensuite, nous développons une nouvelle méthode, purement possibiliste, basée sur la transformation du réseau initial en une base possibiliste. Nous montrons que notre approche offre de meilleures performances qu'une simple adaptation des méthodes probabilistes.*

*ABSTRACT. Qualitative possibilistic networks, also known as min-based possibilistic networks, are important tools for handling uncertain information in the possibility theory framework. Despite their importance, only the junction tree adaptation has been proposed for exact reasoning with such networks. This paper explores alternative algorithms using compilation techniques. We first propose possibilistic adaptations of standard compilation-based probabilistic methods. Then, we develop a new, purely possibilistic, method based on the transformation of the initial network into a possibilistic base. A comparative study shows that this latter performs better than the possibilistic adaptations of probabilistic methods. This result is also confirmed by experimental results.*

*MOTS-CLÉS : Inférence, réseaux possibilistes basés sur le conditionnement ordinal, compilation.*

*KEYWORDS: Inference, qualitative possibilistic networks, compilation.*

---

## 1. Introduction

En théorie des possibilités il existe deux manières pour définir une contrepartie des réseaux Bayésiens selon que l'on utilise le conditionnement ordinal ou celui basé sur l'opérateur produit (Dubois *et al.*, 1988). L'utilisation de l'opérateur produit génère des réseaux très semblables, de point de vue théorique et algorithmique, aux réseaux Bayésiens. Ceci n'est pas le cas avec les réseaux possibilistes basés sur le conditionnement ordinal. Dans cet article, nous nous intéressons au problème d'inférence dans ces derniers. Le problème de décision *Est ce que*  $\Pi(A = a|E = e) = \alpha$  où  $\Pi$  est une mesure de possibilité est connu comme étant NP-difficile (Cooper, 1990).

L'idée donc est d'utiliser les techniques de compilation. En effet, la compilation des connaissances offre une alternative à la difficulté calculatoire de beaucoup de problèmes en intelligence artificielle. Elle consiste à transformer un problème en utilisant les langages cibles de compilation de façon à faciliter sa résolution pour certaines applications. Il existe des langages cibles de compilation, tels que les *Impliqués Premiers* (PI<sup>1</sup>), la *Forme Normale Négative Décomposable* (DNNF<sup>2</sup>), etc. Le choix d'un bon langage cible de compilation repose sur son efficacité à assurer des transformations et à résoudre des requêtes tout en trouvant un compromis entre efficacité et concision (pour plus de détails voir (Darwiche *et al.*, 2002)).

Récemment, plusieurs chercheurs ont proposé divers types d'algorithmes d'inférence exacts et approximatifs pour les réseaux Bayésiens en utilisant des techniques de compilation (Darwiche, 2003) (Chavira *et al.*, 2005) (Wachter *et al.*, 2007) (Chavira *et al.*, 2007). Néanmoins, de telles techniques n'ont pas été explorées dans le cadre des réseaux possibilistes.

Cet article propose trois méthodes de compilation de réseaux possibilistes. Dans un premier temps, nous proposons des adaptations de deux approches d'inférence probabilistes standards, à savoir la méthode des circuits arithmétiques (Darwiche, 2003) et une de ses variantes qui est la compilation logique des réseaux bayésiens (Wachter *et al.*, 2007). Nous proposons ensuite une nouvelle approche qui se base sur la transformation du réseau en une base possibiliste qui sera compilée par la suite pour répondre aux requêtes. Cette méthode regroupe les résultats obtenus dans (Benferhat *et al.*, 2002) et (Benferhat *et al.*, 2007).

Cet article est organisé comme suit : la section 2 rappelle les différents concepts de base de la théorie des possibilités, la logique possibiliste, les réseaux possibilistes et quelques notions sur la compilation. Dans la section 3, nous proposons les adaptations possibilistes des méthodes d'inférence probabilistes. La section 4 présente notre nouvelle méthode d'inférence en utilisant les bases possibilistes compilées. Dans la section 5, nous proposons les résultats expérimentaux des différents algorithmes proposés dans cet article.

---

1. Prime Implicates

2. Decomposable Negation Normal Form

## 2. Concepts de base

### 2.1. Théorie des possibilités

Cette sous section donne un bref aperçu de la théorie des possibilités (pour plus de détails voir (Dubois *et al.*, 1988)). Soit  $V = \{X_1, X_2, \dots, X_N\}$  un ensemble de variables.  $D_{X_i} = \{x_1, \dots, x_n\}$  est le domaine associé à la variable  $X_i$ .  $x_i$  dénote une instance quelconque de  $X_i$ .  $\Omega$  dénote l'univers de discours qui est le produit cartésien des domaines de toutes les variables dans  $V$ . Chaque élément  $\omega \in \Omega$  dit interprétation correspond à un état de  $\Omega$ .

Un élément de base de la théorie des possibilités est la notion de distribution de possibilité  $\pi$  qui est une application de  $\Omega$  vers l'intervalle  $[0,1]$ . L'échelle possibiliste peut être interprétée de deux manières différentes :

- d'une manière ordinale, si les valeurs affectées ne reflètent que l'ordre entre les différents mondes possibles.
  - d'une manière numérique, si les valeurs affectées prennent un sens sur l'intervalle.
- Dans le reste de cet article, nous considérons uniquement la première interprétation.

Etant donnée une distribution de possibilité  $\pi$  définie sur l'univers de discours  $\Omega$ , on peut définir deux mesures duales pour chaque événement  $\phi \subseteq \Omega$  : la *mesure de possibilité* qui représente à quel niveau  $\phi$  est cohérent avec nos connaissances représentées par  $\pi$  :  $\Pi(\phi) = \max_{\omega \in \phi} \pi(\omega)$  et la *mesure de nécessité* qui correspond au degré de certitude associé à  $\phi$  à partir des informations codées par  $\pi$  :  $N(\phi) = 1 - \Pi(\neg\phi)$ .

Le conditionnement possibiliste correspond à la révision d'une distribution de possibilité initiale  $\pi$  par une nouvelle information certaine  $\phi \subseteq \Omega$ . La distribution initiale est alors transformée en une nouvelle distribution. Le conditionnement ordinal basé sur l'opérateur minimum est défini comme suit :

$$\Pi(\psi | \phi) = \begin{cases} \Pi(\psi \wedge \phi) & \text{si } \Pi(\psi \wedge \phi) < \Pi(\phi) \\ 1 & \text{sinon} \end{cases} \quad [1]$$

Une distribution de possibilité peut être représentée compactement par différents formats de représentation de connaissances tels que les réseaux possibilistes, ainsi que la logique possibiliste.

### 2.2. Logique possibiliste

La logique possibiliste (Dubois *et al.*, 1994) est un outil de raisonnement en présence d'informations incertaines basé sur la relation de préférence entre les formules. Une formule logique est une paire :  $(\xi, a)$  où  $\xi$  est une formule *propositionnelle* et  $a$  son niveau d'incertitude qui estime à quel degré il est certain que  $\xi$  est vrai. Autrement dit,  $a$  est la borne inférieure du degré de nécessité de  $\xi$  ( $N(\xi)$ ). Plus une formule a un poids important plus elle est certaine. L'ensemble de formules est appelé une base de connaissances possibiliste  $\Sigma$  qui est un ensemble fini de  $n$  formules pondérées. Les

formules avec un degré de nécessité égal à 0 ne sont pas explicitement représentées dans  $\Sigma$  :

$$\Sigma = \{(\xi_i, a_i), i = 1, \dots, n\} \quad [2]$$

### 2.3. Réseaux possibilistes basés sur le conditionnement ordinal

Un réseau possibiliste basé sur le conditionnement ordinal, noté  $\Pi G_{min}$ , possède :

- un composant graphique qui est représenté par un graphe orienté acyclique (DAG : Directed Acyclic Graph). Chaque noeud représente une variable. Les liens permettent de représenter les relations d'indépendance entre ces variables. L'ensemble des parents d'un noeud  $X_i$  est noté par  $U_i = \{U_{i1}, U_{i2}, \dots, U_{im}\}$ . Dans ce qui suit,  $x_i, u_i, u_{ij}$  sont utilisés pour dénoter, respectivement, les instances possibles de  $X_i, U_i$  et  $U_{ij}$ .

- un composant numérique qui permet de quantifier les différents liens dans le DAG en utilisant les distributions de possibilités de chaque noeud  $X_i$  dans le contexte de ses parents  $U_i$ . Ces distributions de possibilité doivent satisfaire les conditions de normalisation suivantes : si  $U_i = \emptyset$  (c.a.d  $X_i$  est une racine), alors la distribution a priori relative à  $X_i$  doit satisfaire  $\max_{x_i} \Pi(x_i) = 1$ . Si ( $U_i \neq \emptyset$ ), alors la distribution conditionnelle relative à  $X_i$  doit satisfaire  $\max_{x_i} \Pi(x_i | u_i) = 1$ , pour chaque  $u_i$ . Étant donné toutes les distributions conditionnelles et a priori, la distribution jointe relative à l'ensemble de toutes les variables, est exprimée par la règle de chaînage suivante basée sur l'opérateur minimum :

$$\pi_{min}(X_1, \dots, X_N) = \min_{i=1..N} \Pi(X_i | U_i) \quad [3]$$

### 2.4. Concepts de compilation

Un langage cible de compilation est une classe de formules, caractérisé par sa *concision*, et par l'ensemble de *transformations* et de *requêtes* polynomiales admises (Darwiche *et al.*, 2002). Parmi les langages de compilation les plus efficaces, nous citons la *forme normale négative décomposable (DNMF)* (Darwiche, 2001). Ce langage est universel, il présente un certain nombre de propriétés lui donnant un grand intérêt et supporte un ensemble d'opérations logiques. Pour définir la DNMF, le point de départ est la *Forme Normale Négative (NNF)* qui est un ensemble de formules propositionnelles où les connecteurs possibles sont des conjonctions, des disjonctions et des négations. La NNF peut être caractérisée par un certain nombre de propriétés qui permettent de définir quelques uns de ses sous ensembles. Nous citons en particulier :

- *Décomposabilité* : les opérandes de chaque conjonction d'une NNF ne se partagent pas de variables.

- *Déterminisme* : les opérandes de chaque disjonction d'une NNF sont logiquement contradictoires.

– *Régularité*<sup>3</sup> : les opérands de chaque disjonction d'une NNF mentionnent les mêmes variables.

Parmi les sous ensembles de NNF les plus utilisés, nous distinguons : le langage DNNF (Darwiche, 2001) qui supporte la décomposabilité. Il joue un rôle primordial dans la compilation des bases de connaissances propositionnelles. En effet, à l'exception de PI, DNNF est le langage le plus concis parmi tous les langages cibles de compilation (Darwiche *et al.*, 2002). Il a été utilisé pour compiler des réseaux probabilistes. Aussi nous citons les variantes de DNNF à savoir : le langage d-DNNF qui satisfait le déterminisme et le langage sd-DNNF qui satisfait la régularité et le déterminisme.

Chaque langage de compilation supporte des *requêtes* et des *transformations* en un temps polynomial. Dans ce qui suit nous sommes, en particulier, intéressés par le conditionnement et l'oubli définis comme suit :

– Le *conditionnement* d'une formule propositionnelle  $\xi$  en un terme consistant  $\lambda$ , dénommé par  $\xi|\lambda$  génère une nouvelle formule en substituant chaque littéral de  $\lambda$  en  $\xi$  par  $\top$  s'il est consistant avec  $\lambda$ , sinon par  $\perp$  où  $\perp$  et  $\top$  indiquent respectivement une contradiction et une tautologie.

– L'*oubli* d'une variable  $X_i$  d'une formule propositionnelle  $\xi$  consiste à éliminer chaque référence de  $X_i$  dans  $\xi$ .

### 3. Adaptations possibilistes de méthodes d'inférence probabilistes basées sur des techniques de compilation

Il existe une variété de méthodes de compilation qui traitent le problème d'inférence dans les modèles graphiques probabilistes. Dans cette section, nous proposons tout d'abord une adaptation de la méthode des circuits arithmétiques de Darwiche (Darwiche, 2003), puis nous étudions l'une de ses variantes proposée dans (Wachter *et al.*, 2007).

La théorie des possibilités qualitative se base sur deux principaux opérateurs : Max et Min. Le premier est utilisé pour la marginalisation et le second pour le calcul de la distribution jointe. Nous proposons ainsi des langages possibilistes qui supportent ces deux opérateurs. Nous nous intéressons en particulier aux adaptations possibilistes de trois langages largement utilisés en probabilité, notamment  $\Pi$ -DNNF,  $\Pi$ -d-DNNF et  $\Pi$ -sd-DNNF. Le préfixe  $\Pi$  sera rajouté afin de signifier la version possibiliste de chacun de ces langages.

**Definition 1.** - Une expression en  $\Pi$ -DNNF est un graphe acyclique dirigé (DAG) avec une seule racine où chaque noeud feuille est étiqueté par vrai, faux (ou de manière générale les instances des variables) et où chaque noeud interne est étiqueté par les opérateurs *max* ou *min* et peut avoir arbitrairement plusieurs enfants. Autrement

---

3. Smoothness

dit,  $\Pi$ -DNNF est définie de la même manière que la DNNF en remplaçant  $\vee$  et  $\wedge$  par  $\max$  et  $\min$ , respectivement.

- Une expression en  $\Pi$ -d-DNNF est une expression en  $\Pi$ -DNNF qui satisfait la décomposabilité et le déterminisme.

- Une expression en  $\Pi$ -sd-DNNF est une expression en  $\Pi$ -DNNF qui satisfait la décomposabilité, le déterminisme et la régularité.

### 3.1. Inférence utilisant les circuits arithmétiques possibilistes

Dans (Darwiche, 2003) les auteurs se sont focalisés sur l'étude des algorithmes d'inférence dans des réseaux bayésiens compilés. L'idée principale est de représenter chaque réseau Bayésien par un *polynôme* qui sera ensuite différencié et évalué afin de répondre aux différentes requêtes. Ce dernier est de taille exponentielle, donc il est représenté d'une manière plus efficace à travers un *circuit arithmétique* extrait à partir d'une expression sd-DNNF, c'est à dire d'une DNNF qui satisfait le déterminisme et la régularité. Ce circuit peut être évalué et différencié en un temps et un espace linéaire par rapport à sa taille.

Dans ce qui suit, nous proposons une adaptation possibiliste de cette méthode. Considérons un réseau possibiliste basé sur le conditionnement ordinal, le principe est de le coder en une *fonction multi-linéaire MLF* dénommée  $f_{min}$  (équation 4) définie par deux types de variables :

- *Indicateurs d'évidence* : pour chaque variable  $X_i$ , une variable  $\lambda_{x_i}$  est associée à chaque instance  $x_i \in D_{X_i}$ .

- *Paramètres du réseau* : pour chaque variable  $X_i$  et ses parents  $U_i$ , une variable  $\theta_{x_i|u_i}$  est associée à chaque instance  $x_i \in D_{X_i}$  et  $u_i \in D_{U_i}$ .

$$f_{min} = \max_{\mathbf{x}} \min_{x_i, u_i \sim \mathbf{x}} \lambda_{x_i} \theta_{x_i|u_i} \quad [4]$$

où  $\mathbf{x}$  représente les instanciations de toutes les variables du réseau et  $u_i \sim \mathbf{x}$  indique la relation de compatibilité entre  $u_i$  et  $\mathbf{x}$ .

**Proposition 1.** Soit  $\Pi G_{min}$  un réseau possibiliste,  $\pi$  sa distribution jointe et  $f_{min}$  son polynôme. Etant donnée une évidence  $e$ , la valeur du polynôme  $f_{min}$ , dénotée  $f_{min(e)}$ , obtenue en remplaçant chaque indicateur d'évidence  $\lambda_{x_i}$  dans  $f_{min}$  par 1 si  $x_i$  est consistant avec  $e$  et par 0 sinon, est égale à  $\Pi(e)$ .

Le polynôme  $f_{min}$  est ensuite codé en une théorie propositionnelle CNF (*Conjunctive Normal Form*), dénotée par  $K_{MLF}$  comme suit :

– pour chaque variable  $X_i$ , le codage contient la clause suivante :  $\lambda_{x_i} \vee \lambda_{\neg x_i}$

– pour chaque variable propositionnelle  $\theta_{x_i|u_i}$ , le codage contient la clause :

$$\lambda_{x_i} \wedge \lambda_{u_{i1}} \wedge \dots \wedge \lambda_{u_{im}} \leftrightarrow \theta_{x_i|u_i}$$

Après l'élaboration du codage CNF, une étape de compilation en  $\Pi$ -sd-DNNF est nécessaire afin d'extraire le circuit arithmétique possibiliste, noté  $\zeta_\pi$ , qui implémente

$f_{min}$ . Un tel circuit est un DAG où les neuds feuilles représentent les entrées du circuit, les noeuds internes sont les opérateurs  $max$  et  $min$  et la racine correspond à la sortie du circuit.

Le calcul du degré de possibilité d'un événement donné en utilisant  $\zeta_\pi$  consiste à l'évaluer en affectant à chaque  $\lambda_x$  la valeur 1 si l'événement est compatible avec  $x$  et 0 sinon et en appliquant les opérateurs de  $\zeta_\pi$ . Ce calcul se fait en partant des feuilles vers la racine et il est assuré d'une manière linéaire de la même manière que le cas probabiliste (Darwiche, 2003).

En supposant que les distributions de possibilités conditionnelles liées au réseau initial montrent quelques *structures locales* dans le sens qu'elles contiennent des degrés égaux (0 ou 1 ou autres), alors nous pouvons exploiter ces structures pour réduire la taille du circuit arithmétique. Par ailleurs, comme indiqué dans la section 2.3, la condition de normalisation implique que la valeur 1 doit apparaître plusieurs fois dans chaque table (à part les tables liées aux noeuds racines), ceci signifie que les circuits arithmétiques possibilistes ont une représentation plus compacte que leur contrepartie probabiliste comme le montre la proposition suivante :

**Proposition 2.** Soient  $Nb_{poss}$  et  $Nb_{proba}$  le nombre de clauses des codages CNF possibilistes et probabilistes, respectivement, alors  $Nb_{poss} \leq Nb_{proba}$ .

Notons que dans le cas particulier où les réseaux probabilistes et possibilistes ne contiennent que des degrés égaux à 0 ou 1, on peut avoir  $Nb_{poss} = Nb_{proba}$  sinon  $Nb_{poss} \prec Nb_{proba}$ .

L'adaptation possibiliste de la méthode des circuits arithmétiques (Darwiche, 2003) peut donc être résumée par l'algorithme suivant :

**Algorithme 1:** Inférence basée sur une compilation  $\Pi$ -sd-DNNF

Données:  $\Pi G_{min}$ , instance d'intérêt  $x$ , évidence  $e$

Résultat:  $\Pi(x|e)$

**début**

1. *Compilation en  $\Pi$ -sd-DNNF*  
 Codage de  $\Pi G_{min}$  en *MLF* en utilisant équation 4  
 Codage CNF de la *MLF* en  $\xi$   
 Compilation de  $\xi$  en  $\xi^c$   
 $\zeta_\pi \leftarrow$  Circuit Arithmétique de  $\xi^c$
2. *Inférence*  
 Application des Opérateurs en  $\zeta_\pi$   
 $\Pi(x, e) \leftarrow$  Valeur de la Racine ( $\zeta_\pi$ ;  $(x, e)$ )  
 $\Pi(e) \leftarrow$  Valeur de la Racine ( $\zeta_\pi$ ;  $e$ )  
**si**  $\Pi(x, e) \prec \Pi(e)$  **alors**  $\Pi(x|e) \leftarrow \Pi(x, e)$  **sinon**  $\Pi(x|e) \leftarrow 1$   
**retourner**  $\Pi(x|e)$

**fin**



### 3.2. Inférence utilisant des représentations possibilistes compilées

Dans (Wachter *et al.*, 2007) les auteurs ont proposé une variante de (Darwiche, 2003) permettant d'avoir un codage CNF plus simplifié de la distribution de probabilité, puis une compilation de la CNF produite à une d-DNNF à partir de laquelle l'inférence est assurée d'une manière polynomiale. Dans cette section, nous proposons d'adapter ce codage au cas possibiliste.

Soient  $\Delta$  les propositions liées aux variables du réseau et  $\theta$  les propositions liées aux entrées de la distribution de possibilité égales à 1, la représentation logique d'une variable  $X_i$  est définie par :

$$\psi_{X_i} = \bigwedge_{u_i} \left( \bigwedge_{\theta_{x_i|u_i} \in \Omega_{\theta_{X_i|u_i}}} (u_{i1} \wedge \dots \wedge u_{im} \wedge \theta_{x_i|u_i} \rightarrow x_i) \right) \quad [5]$$

La représentation du réseau, dénotée  $\psi$ , est obtenue en prenant la conjonction des représentations logiques de toutes les variables comme suit :

$$\psi = \bigwedge_{X_i \in \Delta} \psi_{X_i} \quad [6]$$

La représentation logique  $\psi$  est ensuite écrite sous forme CNF, dénotée par  $K_\psi$ .

Ce codage affecte une variable par degré différent, tandis que le codage probabiliste utilise une variable par paramètre. En d'autres termes, pour chaque clause dans  $K_\psi$  il existe une clause ayant la même taille que le codage probabiliste mais l'inverse est faux. Ceci implique que le codage possibiliste est plus compact que le codage probabiliste proposé dans (Wachter *et al.*, 2007) puisque nous prenons en considération l'aspect structure locale.

**Proposition 3.** Soient  $NbV_{poss}$  et  $NbV_{proba}$  le nombre de variables des codages CNF possibilistes et probabilistes, respectivement, alors  $NbV_{poss} \prec NbV_{proba}$ .

Le codage  $K_\psi$  est ensuite compilé en langage II-d-DNNF qui supporte les transformations *conditionnement* et *oubli de variables* ainsi que les *calculs possibilistes*. Cette compilation permet de calculer les degrés de possibilité a posteriori en un temps polynomial. Cette approche peut être résumée par l'algorithme 2.

## 4. Nouvel algorithme d'inférence possibiliste

Dans cette section nous nous intéressons à la proposition d'une nouvelle méthode purement possibiliste. L'idée ici est de transformer le réseau initial en base possibiliste ensuite la compiler pour assurer l'inférence en un temps polynomial. En effet, dans (Benferhat *et al.*, 2002), les auteurs ont étudié la transformation des réseaux possibilistes en bases possibilistes. En effet, la base possibiliste associée à un réseau possibiliste est le résultat de la fusion de plusieurs bases élémentaires. Chaque base

**Algorithme 2:** Inférence basée sur une compilation  $\Pi$ -d-DNNFDonnées:  $\Pi G_{min}$ , instance d'intérêt  $x$ , évidence  $e$ Résultat:  $\Pi(x|e)$ **début**

```

1. Compilation en  $\Pi$ -d-DNNF
   Codage CNF de  $\Pi G_{min}$  en  $\psi$  en utilisant équation 6
   Compilation de  $\psi$  en  $\psi_p^c$ 
2. Inférence
    $v_1 \leftarrow$  Explorer  $\Pi$ -d-DNNF( $x \wedge e, \psi_p^c$ )
    $v_2 \leftarrow$  Explorer  $\Pi$ -d-DNNF( $e, \psi_p^c$ )
   si  $v_1 \prec v_2$  alors  $\Pi(x|e) \leftarrow v_1$  sinon  $\Pi(x|e) \leftarrow 1$ 
   retourner  $\Pi(x|e)$ 

```

**fin****Algorithme 3:** Explorer  $\Pi$ -d-DNNFDonnées: un ensemble d'instances  $x$ , la représentation compilée  $\psi_p^c$ Résultat:  $\Pi(x)$ **début**

```

si  $\forall x_i \in x, \theta_{x_i|U_i}$  n'est pas une feuille alors  $\Pi(x) \leftarrow 1$ 
sinon
    $y = \{x_i \mid \forall, \theta_{x_i|U_i} \text{ est une feuille } \forall U_i \subseteq x\}$ 
    $\psi_{p|y}^c \leftarrow$  Conditionnement de  $\psi_p^c$  en  $y$ 
    $\psi_{p\downarrow|y}^c \leftarrow$  Oubli  $\Delta$  de  $\psi_{p|y}^c$ 
   Application des Opérateurs en  $\psi_{p\downarrow|y}^c$ 
    $\Pi(x) \leftarrow$  Valeur de la Racine de  $\psi_{p\downarrow|y}^c$ 
retourner  $\Pi(x)$ 

```

**fin**

élémentaire est composée de formules associées aux degrés de possibilités a priori et conditionnels attachés à chaque noeud du réseau. Cette transformation est définie formellement comme suit :

**Definition 2.** La base de connaissances possibiliste d'un réseau basé sur le conditionnement ordinal  $\Pi G_{min}$  peut être exprimée par :  $\Sigma_{min} = \Sigma_{X_1} \cup \Sigma_{X_2} \cup \dots \cup \Sigma_{X_n}$  où  $\Sigma_{X_i} = \{(\neg x_i \vee \neg u_i, \alpha_i) : \alpha_i = 1 - \pi(x_i|u_i) \neq 0\}$  est la base locale d'une variable binaire  $X_i$  de  $\Pi G_{min}$ .

Dans (Benferhat *et al.*, 2007), les auteurs ont étudié la compilation des bases de connaissances possibilistes pour assurer une inférence polynomiale. Donc, notre idée est de transformer dans un premier temps le réseau possibiliste en une base possibiliste  $\Sigma_{min}$  et la coder en une base propositionnelle CNF comme suit : soit  $A = \{a_1, \dots, a_n\}$  tels que  $a_1 \succ \dots \succ a_n$  les différents poids utilisés dans la base de connaissances  $\Sigma_{min}$  (obtenue par application de la définition 2), un ensemble de variables propositionnelles additionnelles  $A_i$  correspondant exactement au nombre de poids différents est

introduit, et une formule propositionnelle  $\phi_i \vee A_i$  est associée à chaque formule  $\phi_i, a_i$ . Le codage propositionnel de  $\Sigma_{min}$ , dénoté par  $K_\Sigma$  est ainsi défini par :

$$K_\Sigma = \{\phi_i \vee A_i : (\phi_i, a_i) \in \Sigma_{min}\} \quad [7]$$

Une fois le codage CNF  $K_\Sigma$  est assuré, l'idée est de compiler la base de connaissance générée afin d'assurer l'inférence en un temps polynomial. Pour assurer cette phase de compilation nous allons utiliser les résultats de (Benferhat *et al.*, 2007) où plusieurs requêtes ont été étudiées. Dans notre contexte, nous allons nous limiter à une requête particulière utile aux réseaux possibilistes, à savoir, *Quel est le degré de possibilité d'un événement  $A = a$  étant donnée une évidence  $E = e$  ?*

Cet algorithme d'inférence peut donc être résumé comme suit :

**Algorithme 4:** Inférence basée sur une compilation DNNF

Données:  $\Pi G_{min}$ , instance d'intérêt  $x$ , évidence  $e$

Résultat:  $\Pi(x|e)$

**début**

```

1. Transformation en  $K_\Sigma$ 
Transformation de  $\Pi G_{min}$  en  $\Sigma_{min}$  en utilisant définition 2
Transformation de  $\Sigma_{min}$  en  $K_\Sigma$  en utilisant équation 7
2. Inférence
 $K_\Sigma^c \leftarrow Cible(K_\Sigma)$ ;  $K \leftarrow K_\Sigma^c$ 
StopCalcul  $\leftarrow$  faux;  $i \leftarrow 1$ ;  $\Pi(x|e) \leftarrow 1$ 
tant que ( $K \neq A_i \vee \neg e$ ) et ( $i \leq k$ ) et (StopCalcul=faux) faire
    K  $\leftarrow$  conditionnement (K,  $\neg A_i$ )
    si  $K \models \neg x$  alors
        StopCalcul  $\leftarrow$  vrai
         $\Pi(x|e) \leftarrow 1 - degree(i)$ 
    sinon
         $i \leftarrow i + 1$ 
retourner  $\Pi(x|e)$ 

```

**fin**

Notons que cette approche est flexible puisqu'elle supporte tous les langages cibles de compilation (Benferhat *et al.*, 2007).

## 5. Etude comparative et expérimentale

Notre objectif dans cette section est de comparer les trois méthodes de compilation que nous avons proposés. Pour des raisons de clarté, nous allons référer chacune d'elle par le nom du langage de compilation utilisé (II-sd-DNNF, II-d-DNNF, DNNF-PKB).

Le nombre de variables et de clauses des codages CNF est considéré la première dimension qui différencie les trois approches proposées. En effet, le nombre de variables additionnelles dans DNNF-PKB (resp. II-d-DNNF) est le nombre de poids

différents (resp. le nombre des degrés de possibilité non-redondants et différents de 1). En termes de clauses, la CNF de DNNF-PKB (resp.  $\Pi$ -d-DNNF) est basée sur le codage des instances  $\neg x_i$  ayant un degré de possibilité différent de 1 (resp. le codage des instances  $x_i$  ayant comme degré 1). Si on considère une variable  $A$  tel que  $\Pi(a_1) = 1$  and  $\Pi(a_2) = 0.6$ . Selon DNNF-PKB, le codage CNF génère  $(A_1 \vee a_1)$  où  $A_1$  est la variable propositionnelle associée au degré de nécessité 0.4. Alors que dans  $\Pi$ -d-DNNF, la clause produite est  $(A_2 \vee a_1)$  où  $A_2$  est la variable propositionnelle associée au degré de possibilité 0.6.

Dans  $\Pi$ -sd-DNNF, non seulement des variables propositionnelles sont associées aux degrés de possibilité (paramètres), mais aussi pour chaque valeur  $x_i$  de  $X_i$ , alors que dans DNNF-PKB le nombre de variables additionnelles correspond au nombre de degrés différents dans la base possibiliste. Ces interprétations nous conduit à la proposition suivante :

**Proposition 4.** - Soient  $NbV_{DNNF}$ ,  $NbV_{\Pi-d-DNNF}$ ,  $NbC_{DNNF}$ ,  $NbC_{\Pi-d-DNNF}$  le nombre de variables et de clauses du codage CNF des deux méthodes DNNF-PKB et  $\Pi$ -d-DNNF, respectivement, alors  $NbV_{DNNF} = NbV_{\Pi-d-DNNF}$  et  $NbC_{DNNF} = NbC_{\Pi-d-DNNF}$ .  
- Soient  $NbV_{\Pi-sd-DNNF}$  et  $NbC_{\Pi-sd-DNNF}$  le nombre de variables et de clauses de  $\Pi$ -sd-DNNF, respectivement, alors  $NbV_{\Pi-sd-DNNF} \succ NbV_{DNNF}$  et  $NbC_{\Pi-sd-DNNF} \succ NbC_{DNNF}$ .

Il est important de mentionner que même si les codages CNF de DNNF-PKB et  $\Pi$ -d-DNNF ont le même nombre de variables et de clauses, l'inférence n'est pas effectuée de la même manière puisque DNNF-PKB traite des degrés de nécessité et  $\Pi$ -d-DNNF traite des degrés de possibilité.

Analysons maintenant les deux approches DNNF-PKB et  $\Pi$ -sd-DNNF d'un point de vue expérimental afin de confirmer les résultats théoriques obtenus. Notre expérimentation est effectuée sur des réseaux possibilistes générés aléatoirement. En effet, nous avons comparé les CNF et les DNNF de DNNF-PKB et  $\Pi$ -sd-DNNF sur 100 réseaux possibilistes ayant de 10 à 50 noeuds.

Nous proposons, tout d'abord, de comparer les codages CNF sur la base de deux critères qui sont le nombre de variables et le nombre de clauses. La figure 1 montre les résultats de cette expérimentation. Il est évident que plus le nombre de noeuds du réseau est élevé, plus le nombre de variables et de clauses est élevé. Cette figure montre aussi que DNNF-PKB a le nombre le plus réduit de variables et de clauses comparé à  $\Pi$ -sd-DNNF, ce qui confirme les résultats théoriques détaillés ci-dessus.

Il est aussi important de comparer le nombre de noeuds et d'arcs de chaque base compilée comme le montre la figure 2. On remarque que ce nombre dépend fortement des paramètres CNF. En effet, le nombre de noeuds et d'arcs relatif aux bases compilées dans l'approche DNNF-PKB est plus petit que celui lié à  $\Pi$ -sd-DNNF. Ceci peut être expliqué par le nombre réduit de variables et de clauses des CNFs (voir proposition 4), ainsi que la structure locale qui rétrécit la taille des bases compilées.

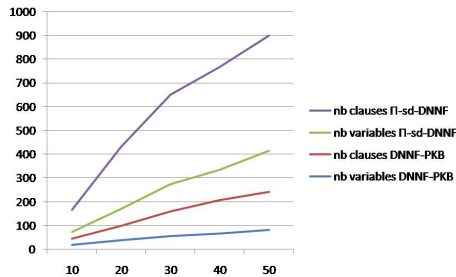


Figure 1. Les paramètres des CNFs.

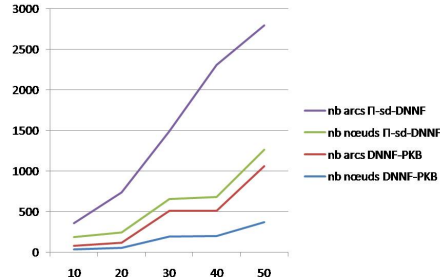


Figure 2. Les paramètres des DNNFs.

## 6. Conclusion

Cet article propose des algorithmes d'inférence dans des réseaux possibilistes en utilisant des techniques de compilation. Nous avons proposé en premier  $\Pi$ -sd-DNNF et  $\Pi$ -d-DNNF comme des adaptations possibilistes de la méthode des circuits arithmétiques (Darwiche, 2003) et la compilation logique des réseaux bayésiens (Wachter *et al.*, 2007), respectivement. Puis nous avons développé une nouvelle méthode possibiliste DNNF-PKB qui se base sur deux phases : dans la première nous avons transformé le réseau en une base possibiliste et dans la seconde nous avons compilé cette dernière pour inférer en un temps polynomial. Nous montrons théoriquement que DNNF-PKB et  $\Pi$ -d-DNNF ont le même nombre de variables et de clauses même si elles sont basées sur des procédés de calculs d'inférence différents. Nous avons également montré que DNNF-PKB est plus compacte que  $\Pi$ -sd-DNNF. Tous ces résultats ont été confirmés par une étude expérimentale. Le travail effectué dans cet article ouvre de nouvelles pistes à explorer, entre autres l'étude des différentes propriétés à savoir le déterminisme et la régularité, qui peuvent alléger la taille des bases compilées dans le cadre possibiliste, ainsi qu'une comparaison des différents algorithmes aux arbres de jonction possibilistes.

## 7. Bibliographie

- Benferhat S., Dubois D., Garcia L., Prade H., « On the transformation between possibilistic logic bases and possibilistic causal networks », *International Journal of Approximate Reasoning*, vol. 29, n° 2, p. 135-173, 2002.
- Benferhat S., Yahi S., Drias H., « On the Compilation of Stratified Belief Bases under Linear and Possibilistic Logic Policies », *International Joint Conference on Artificial Intelligence*, p. 2425-2430, 2007.
- Chavira M., Darwiche A., « Compiling Bayesian Networks with Local Structure », *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, p. 1306-1312, 2005.

- Chavira M., Darwiche A., « Compiling Bayesian Networks Using Variable Elimination », *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, p. 2443-2449, 2007.
- Cooper G. F., « The computational complexity of probabilistic inference using Bayesian belief networks (research note) », *Artif. Intell.*, vol. 42, n° 2-3, p. 393-405, 1990.
- Darwiche A., « Decomposable Negation Normal Form », *Journal of the ACM*, vol. 48, n° 4, p. 608-647, 2001.
- Darwiche A., « A Differential Approach to Inference in Bayesian Networks », *Journal of the ACM*, vol. 50, n° 3, p. 280-305, 2003.
- Darwiche A., Marquis P., « A knowledge compilation map », *Journal of Artificial Intelligence Research*, vol. 17, p. 229-264, 2002.
- Dubois D., Lang J., Prade H., « Possibilistic logic », *Handbook on Logic in Artificial Intelligence and Logic Programming*, vol. 3, Oxford University press, p. 439-513, 1994.
- Dubois D., Prade H., *Possibility theory :An approach to computerized, Processing of uncertainty*, Plenum Press, New York, 1988.
- Wachter M., Haenni R., « Logical Compilation of Bayesian Networks with Discrete Variables », *European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, p. 536-547, 2007.