Un algorithme d'arbre de jonction incrémental

Hamza Agli, Philippe Bonnard, Christophe Gonzales, Pierre-Henri Wuillemin

Un algorithme d'arbre de jonction incrémental

Hamza AGLI[1], **Philippe BONNARD**[1], **Christophe GONZALES**[2], **Pierre-Henri WUILLEMIN**[2]

1. *IBM France Lab,*
9 rue Verdun, Gentilly, France
{hamza.agli,philippe.bonnard}@fr.ibm.com

2. *Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6, UMR 7606*
4, place Jussieu, 75005, Paris, France
{christophe.gonzales,pierre-henri.wuillemin}@lip6.fr

RÉSUMÉ. *Cet article s'intéresse à l'inférence probabiliste dans les systèmes dynamiques multi-cibles. Dans un contexte où les cibles et/ou les observations peuvent évoluer, la plupart des algorithmes ne se servent pas complètement de l'incrémentalité du système pour optimiser les calculs. Cela induit des calculs inutiles et peut augmenter considérablement les temps d'inférence, particulièrement pour les systèmes de grande taille. Pour pallier ce problème, nous proposons un nouvel algorithme d'arbre de jonction utilisant une technique par envoi de messages. Étant donné une nouvelle requête, cet algorithme permet de minimiser les calculs en recalculant uniquement les messages différents des calculs précédents. Nous montrons l'efficacité de notre approche par des résultats expérimentaux.*

ABSTRACT. *This article addresses the question of probabilistic inference in multi-target dynamic systems. In a context where targets and/or evidence may evolve, most of the algorithms do not completely exploit the incrementality to optimize computations. This leads to useless computations and may increase significantly inference times. To cope with this problem, we propose a new junction tree-based message-passing inference algorithm that, given a new request, optimizes computations by only recomputing messages that are different from preceding computations. We highlight the efficiency of our approach by experimental results.*

MOTS-CLÉS : *Réseaux bayésiens, inférence incrémentale, arbre de jonction.*

KEYWORDS: *Bayesian networks, incremental inference, junction tree.*

1. Introduction

Pour gérer les incertitudes dans les systèmes experts, la communauté de l'intelligence artificielle se réfère couramment aux réseaux bayésiens (BN [1]) (Pearl, 1988 ; Koller, Friedman, 2009). Ces derniers sont largement utilisés dans des applications telles que le diagnostic médical, la gestion du risque et les systèmes d'aide à la décision. Un BN est une représentation graphique compacte d'une distribution de probabilité. On peut le voir comme une base de connaissance probabiliste dans laquelle le processus de requête s'appelle "inférence". En pratique, il existe plusieurs types de requêtes. À titre d'exemple, on peut citer le calcul de l'explication la plus probable (MPE) ou le calcul de probabilités *a posteriori* de certaines variables aléatoires (*cibles*). Dans cet article, nous nous intéressons à ce dernier. Bien que cela soit un problème PP-complet dans le cas général (Darwiche, 2009), plusieurs algorithmes d'inférence exacts ou approchés existent dans la littérature (Lauritzen, Spiegelhalter, 1988 ; Madsen, Jensen, 1999 ; Shenoy, Shafer, 1990), ainsi que des extensions pour manipuler les systèmes de grandes tailles (Koller, Pfeffer, 1998 ; Pfeffer, 2000 ; Torti *et al*., 2013) ou bien pour tenir compte d'aspects temporels (Dean, Kanazawa, 1989 ; Murphy, 2002 ; Robinson, Hartemink, 2009). La complexité des BNs n'a cessé de grandir et nécessite de nouveaux algorithmes d'inférence de plus en plus efficaces.

Par ailleurs, les systèmes à base de règles (RBS), qui sont à l'origine de notre recherche, sont aujourd'hui des outils très populaires pour l'automatisation de la prise de décision. Pour quantifier les incertitudes du domaine, ils utilisent fréquemment des modèles heuristiques, tels que *les facteurs de certitude* (Buchanan, Shortliffe, 1984). Ces derniers souffrent de limitations pratiques et théoriques (Heckerman, Shortliffe, 1992) qu'on peut pallier par l'exploitation des probabilités. Dans ce contexte, les BNs pourraient s'avérer utiles. Leurs mécanismes efficaces d'inférence, en particulier les algorithmes d'arbre de jonction (JT [2]) (Jensen *et al*., 1990 ; Shenoy, Shafer, 1990 ; Madsen, Jensen, 1999), semblent être de bons candidats pour réaliser des inférences dans les RBS. Or, ces derniers sont des environnements de nature incrémentale et multi-*cibles*. Par conséquent, pour être efficace, l'inférence des BN devrait être aussi incrémentale. Certains algorithmes exploitent cette caractéristique (Madsen, Jensen, 1999), mais le résultat est loin d'être optimal lorsque les *cibles* changent dynamiquement et forment un sous-ensemble strict de l'ensemble des variables aléatoires. La situation est pire lorsque la structure du JT évolue elle-même dans le temps. C'est une problématique pour les RBS, où les changements de la structure du BN, des observations et des *cibles* sont fréquents.

Dans D'Ambrosio (1993), quatre critères d'incrementalité relatifs à l'inférence probabiliste ont été présentés : incrementalité par rapport aux ressources de calculs, aux requêtes, aux observations et au modèle graphique. Dans cet article, on s'intéresse à tous ces critères, particulièrement aux trois derniers. Étonnamment, très peu d'algo-

1. Bayesian Networks
2. Junction Tree, structure construite à partir du BN initial, voir la section 2.

rithmes d'inférence abordent tous ces aspects. Ainsi, Darwiche (1998) traite l'incrémentatlité du point de vue des requêtes en reconfigurant dynamiquement certains JT quand ces requêtes changent. Mais la structure du BN est supposée statique, ce qui n'est pas forcément le cas dans les RBS. Dans Lin et Druzdzel (1998), les auteurs exploitent un raisonnement fondé sur la *pertinence* pour identifier les parties du réseau qui sont pertinentes pour les calculs, puis ils mettent à jour plusieurs sous-réseaux dont l'union couvre le réseau d'origine. Cet algorithme ne prend pas en compte les calculs effectués précédemment. Dans Madsen et Jensen (1999), un algorithme d'inférence par JT incrémental a été proposé. Il exploite les indépendances induites par les mises à jour des observations incrémentales. La structure du JT est toutefois statique et tous les nœuds sont considérés comme *cibles*, ce qui n'est pas optimal dans notre contexte. Dans Flores *et al*. (2003), seule l'incrémentalité de la structure du JT est prise en compte, mais pas celle des requêtes et les calculs probabilistes précédents ne sont pas non plus exploités. Dans la même perspective, Li *et al*. (2006) proposent une compilation du *BN* d'origine en une forme normale conjonctive couplée avec des techniques de cache, ce qui améliore l'inférence quand la structure du réseau est a mise à jour. Cette méthode ne prend pas en considération de façon optimale les requêtes et les évidences.

Dans cet article, nous examinons une nouvelle approche pour surmonter les limites précitées. Cette approche vise à améliorer l'efficacité de l'inférence dans les systèmes dynamiques de très grande taille. L'idée clé de notre algorithme, appelé « Inférence d'Arbre de Jonction Incrémentale » (*IJTI*[3]), consiste à limiter les calculs seulement aux parties du JT qui sont pertinentes pour les *cibles* et qui ont a été invalidées par les changements incrémentaux. En conséquence, *IJTI* optimise les calculs d'inférence probabiliste.

Cet article est organisé comme suit. Dans la section suivante, nous présentons le contexte théorique nécessaire. Dans la Section 3, nous présentons notre approche et la justifions mathématiquement. Les démonstrations sont fournies en appendice. Ensuite, nous mettons expérimentalement en évidence l'efficacité de notre contribution. Enfin, quelques conclusions et perspectives sont fournies dans la section 5.

2. Préliminaires et notations

Un BN est représenté par un couple $(\mathcal{G}, \Theta)$, où $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ est un graphe orienté sans circuit (DAG[4]). $\mathcal{V}$ est un ensemble de nœuds représentant des variables aléatoires[5]. $\mathcal{A}$ est un ensemble d'arcs et $\Theta = \{P(X|\mathrm{Pa}(X)) : X \in \mathcal{V}\}$ est l'ensemble des tables de probabilités (CPT) des variables de $\mathcal{V}$ conditionnellement à leurs parents dans $\mathcal{G}$. Le BN encode la loi de probabilité jointe sur $\mathcal{V}$ comme le produit de ces CPTs. Dans cet article, l'inférence probabiliste est fondée sur un algorithme par envoi

3. Incremental Junction Tree Inference.
4. Directed Acyclic Graph.
5. Par abus, nous ne faisons pas de distinction entre les nœuds et les variables aléatoires qu'ils représentent.

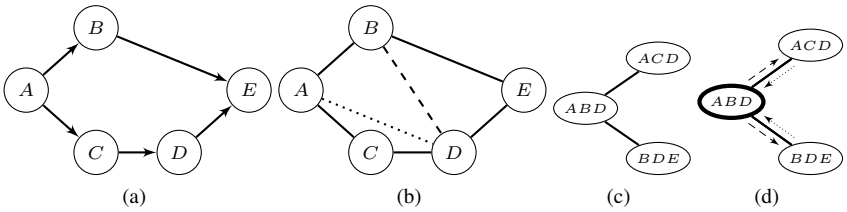

Figure 1. Construction d'un JT

de messages dans le JT. La construction de ce dernier consiste d'abord à convertir le DAG $\mathcal{G}$ en un graphe non orienté en ajoutant, pour chaque nœud dans $\mathcal{V}$, des arêtes entre tous ses parents (*moralisation*), en enlevant les orientations des arcs restants et en ajoutant par la suite des arêtes de telle sorte que, dans tout cycle d'au moins quatre nœuds, deux nœuds non adjacents sont reliés entre eux (*triangulation*). Les nœuds du JT correspondent alors à des sous-graphes complets et maximaux (des cliques) du graphe résultant. Ces cliques sont reliées par des arêtes de telle sorte que: $i)$ le JT ne contient aucun cycle; et $ii)$ tout couple de cliques d'intersection non vide sont reliées par une chaîne, sur laquelle toutes les cliques contiennent cette intersection. La figure 1a montre un exemple d'un DAG, son graphe moral et triangulé est donné dans la figure 1b, où les arêtes avec des traits discontinus et celles en pointillés représentent les arêtes introduites respectivement pendant la *moralisation* et la *triangulation*. Enfin, la figure 1c est une représentation possible du JT correspondant. Notons qu'un JT peut être une forêt, par exemple, lorsque le DAG $\mathcal{G}$ n'est pas connexe. Dans notre approche, traiter une forêt est équivalent à réitérer le même processus sur ses composantes connexes. D'où, sans perte de généralité, nous considérerons dans la suite que le JT sur lequel nous travaillons est un arbre $\mathcal{T}$. Dans ce qui suit, pour n'importe quel JT $\mathcal{T}$, nous dénoterons par $\mathcal{V}(\mathcal{T})$ et $\mathcal{E}(\mathcal{T})$ l'ensemble de ses cliques et de ses arêtes respectivement. Les algorithmes par envoi de messages consistent à appliquer une *collecte* et une *distribution* à partir d'une racine prédéterminée $r \in \mathcal{V}(\mathcal{T})$. Pendant la *collecte*, les messages sont envoyés à partir des feuilles vers r et, pendant la *distribution*, ils sont envoyés dans la direction opposée. Pour garantir la justesse des calculs, pour toute arête $(i, j) \in \mathcal{E}(\mathcal{T})$, le message envoyé de i à j, noté $\psi_{i\to j}$, est calculé seulement quand la clique i a reçu les messages de tous ses voisins sauf j. La figure 1d montre un exemple de passage de messages avec $r = ABD$ (la clique en traits épais), les arcs en pointillés et en traits discontinus représentant respectivement les messages de *collecte* et de *distribution*. Le calcul de ces messages n'est pas le sujet de cet article, mais peut être trouvé dans (Madsen, Jensen, 1999). Toutefois, il est important de se rappeler que la complexité de l'algorithme d'inférence réside principalement dans le calcul de ces messages (multi-dimensionnels). Il est donc pertinent de s'intéresser à la minimisation du nombre de calculs de tels messages.

Dans un environnement incrémental, tous les messages n'ont pas besoin d'être recalculés chaque fois qu'une modification arrive, parce que, en pratique, beaucoup resteront inchangés. Comme nous le verrons plus tard, avec l'utilisation des définitions

suivantes, nous pouvons caractériser précisément ceux qui ont besoin d'une mise à jour due à l'ajout de nouvelles observations, des changements de structure ou/et des *cibles*.

DÉFINITION 1 (Chaîne). — *Soit $\mathcal{T} = (\mathcal{V}(\mathcal{T}), \mathcal{E}(\mathcal{T}))$ un JT. $i_1, \ldots, i_{n+1}$ est appelé une chaîne dans $\mathcal{T}$ si $(i_\alpha, i_{\alpha+1}) \in \mathcal{E}(\mathcal{T})$ pour tout $\alpha \in \{1, \ldots, n\}$. Pour simplifier, cette chaîne sera notée $i_1 - i_{n+1}$ et sa longueur $len(i_1 - i_{n+1})$ (qui est ici égale à n).*

DÉFINITION 2 (Adjacence). — *Soit $i, j \in \mathcal{V}(\mathcal{T})$, $i \neq j$. i et j sont adjacents dans $\mathcal{T}$ ssi $(i, j) \in \mathcal{E}(\mathcal{T})$. L'ensemble des cliques adjacentes à i est noté* $\text{Adj}(i)$*, c-à-d.,* $\text{Adj}(i) := \big\{k \in \mathcal{V}(\mathcal{T}) : (i, k) \in \mathcal{E}(\mathcal{T})\big\}$*. Soit $r \in \mathcal{V}(\mathcal{T})$, $r \neq i$, alors $Adj_r(i)$ dénote le singleton contenant la clique adjacente à i située sur la chaîne reliant i et r, c-à-d., $Adj_r(i) := \big\{k \in \text{Adj}(i) : k \in i - r\big\}$. On définit aussi $Adj_r(r) := \emptyset$. Finalement, soit* $\text{Adj}_{\text{-}j}(i) := \text{Adj}(i) \setminus \{j\}$.

Par exemple, dans la figure 2a, $Adj_r(i) = \{k_3\}$ et $Adj_r(k_3) = \{r\}$. On notera $\mathcal{V}_{\text{-}j}(i)$ l'ensemble des nœuds du sous-arbre maximal dans $\mathcal{T}$ qui contient i et non pas $Adj_j(i)$. Enfin, on notera $\mathcal{V}_j(i) = \mathcal{V}_{\text{-}j}(i) \cup \{j\}$ (voir la partie grisée dans la figure 2a).

2.1. Caractérisation des messages

Un message $\psi_{i \to j}$ transitant dans $\mathcal{T}$, est orienté par définition. Il propage vers j (et, par induction, vers $\mathcal{V}_{\text{-}i}(j)$, voir la figure 2a) toutes les informations pertinentes provenant des cliques dans $\mathcal{V}_{\text{-}j}(i)$, notamment toutes les observations reçues. Par conséquent, $\psi_{i \to j}$ a besoin d'être recalculé dès lors qu'une nouvelle observation ou des changements de structure apparaissent dans $\mathcal{V}_{\text{-}j}(i)$ [6]. Cependant, même si $\mathcal{V}_{\text{-}j}(i)$ a reçu des observations, $\psi_{i \to j}$ n'a pas pas besoin d'être recalculé si $\mathcal{V}_{\text{-}i}(j)$ ne contient aucune *cible*. Dans ce cas, l'état de $\psi_{i \to j}$ devient « invalide » puisque son contenu est maintenant incorrect. Ceci ne pose pas de problème pour l'inférence actuelle, mais, pour les futures, nous devons prendre en compte cet état pour recalculer $\psi_{i \to j}$ s'il doit être utilisé par la suite.

La première contribution de ce travail est de proposer un algorithme permettant d'identifier les messages à recalculer.

Soit $\mathcal{A}(\mathcal{T})$ l'ensemble de tous les arcs obtenus de $\mathcal{E}(\mathcal{T})$ en considérant toutes les orientations possibles, c-à-d, $\mathcal{A}(\mathcal{T}) := \bigcup_{(i,j) \in \mathcal{E}(\mathcal{T})} \{(i, j)\} \cup \{(j, i)\}$. Pour formaliser les conditions ci-dessus, nous commençons par caractériser les informations qui sont « locales » à i et j par:

6. Par abus, nous disons qu'une clique reçoit une observation quand au moins une de ses variables aléatoires a été observée. Étant donné que le comportement de l'inférence est identique au regard de ces deux transformations (observations et changements de structure), on va les identifier et appeler cela des observations tout court dans la suite.

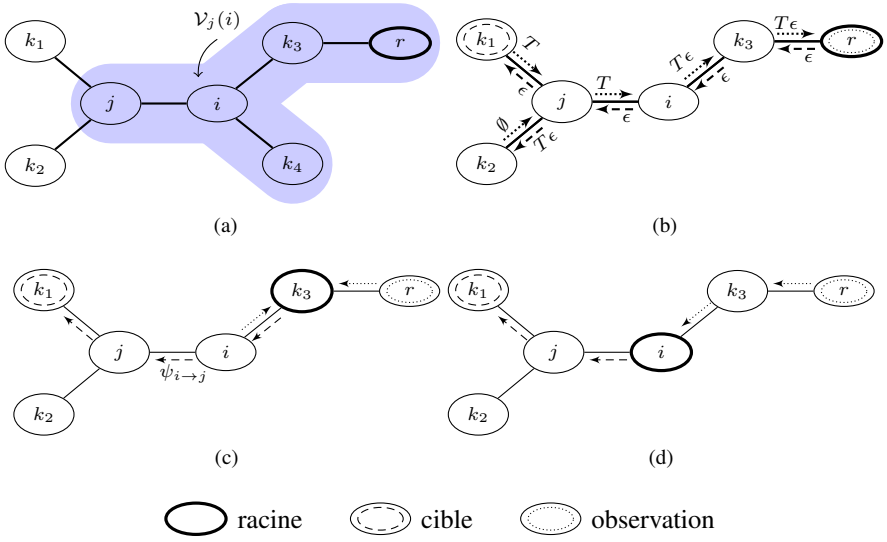

Figure 2. Envoi de messages dans un JT $\mathcal{T}$.

DÉFINITION 3 (Label-message local λ). — $\lambda : \mathcal{A}(\mathcal{T}) \mapsto 2^{\{\epsilon,T\}}$ *est une fonction telle que:*

$$(i,j) \longmapsto \lambda_{i\to j} := \begin{cases} \{\epsilon\} & \textit{si l'état de } \psi_{i\to j} \textit{ est « invalide » ou si de nouvelles observations ou des changements de structure affectent } i \ (1) \\ \{T\} & \textit{si } i \textit{ contient des cibles (2)} \\ \{T,\epsilon\} & \textit{si (1) et (2)} \\ \emptyset & \textit{sinon} \end{cases}$$

Pour simplifier la notation dans la suite, on note $\{T,\epsilon\}$ par $T\epsilon$. L'idée de notre algorithme consiste à marquer chaque arc (i,j) dans $\mathcal{A}(\mathcal{T})$ par des labels $\mu_{i\to j}$ exprimant l'ensemble des informations contenues dans $\mathcal{V}_{\text{-}j}(i)$.

DÉFINITION 4 (Label-message μ). — *Pour $(i,j) \in \mathcal{A}(\mathcal{T})$, le label-message envoyé de i à j est la fonction $\mu : \mathcal{A}(\mathcal{T}) \mapsto 2^{\{\epsilon,T\}}$ telle que* $\mu_{i\to j} := \bigcup_{\substack{k'\in\mathcal{V}_{\text{-}j}(i) \\ \{k\}=Adj_j(k')}} \lambda_{k'\to k}$.

Pour illustrer la discussion précédente, imaginons une première mise à jour incrémentale du DAG initial et, par conséquent, le JT initial dans la figure 2a. Il s'agit, par exemple, de la suppression de k_4, l'insertion d'une observation dans r et une nouvelle *cible* dans k_1. La figure 2b montre l'émission des μ-messages au sein du JT $\mathcal{T}$ après cette mise à jour, où les ellipses pointillées et en traits discontinus représentent respectivement les cliques contenant des observations et celles contenant des *cibles*. On

peut facilement voir que, par exemple, $\mu_{i\to j} = \epsilon$, $\mu_{j\to i} = T$ et $\mu_{j\to k_2} = T\epsilon$. La proposition suivante permet de construire récursivement les messages μ:

PROPOSITION 5 (Construction de μ). — *Soit* $(i,j) \in \mathcal{A}(\mathcal{T})$*, alors on a :* $\mu_{i\to j} = \lambda_{i\to j} \cup \bigcup_{k\in \text{Adj}_{\text{-}j}(i)} \mu_{k\to i}$.

3. Optimisation de l'inférence : *IJTI*

Rappelons que $\psi_{i\to j}$ dénote le message échangé entre les cliques i et j pendant un calcul d'inférence, à ne pas confondre avec le label-message $\mu_{i\to j}$ de la Définition 4.

3.1. Détermination de(s) la racine(s) optimale(s)

En général, le nombre de calculs effectués par un algorithme par envoi de messages dans un JT ne dépend pas de la clique racine choisie pour la *collecte/distribution*, car tous les messages $\psi_{i\to j}$ sont calculés *sur toutes* les arêtes du JT, c-à-d. que $2 \times |\mathcal{E}|$ messages sont systématiquement calculés. Dans notre cas, la situation est différente. En effet, notre algorithme calcule et envoie *seulement* les messages $\psi_{i\to j}$ qui sont nécessaires pour le calcul des distributions *a posteriori* des *cibles*. Sur certaines arêtes, IJTI ne calculera donc pas certains $\psi_{i\to j}$ parce qu'ils ne sont pas pertinents pour le calcul de ces distributions. En conséquence, dans IJTI, le nombre de calculs effectués est sensible à la sélection de la racine: par exemple, dans le JT de la figure 2a, si la clique i est observée et j est la seule *cible*, seulement le message $\psi_{i\to j}$ de i vers j est nécessaire, ce qui est précisément envoyé si la clique i est choisie comme racine (ici, seulement une distribution est nécessaire). Par contre, si la clique k_4 est choisie comme racine, le message $\psi_{i\to k_4}$ doit être envoyé pendant la *collecte* et les messages $\psi_{k_4\to i}$ et $\psi_{i\to j}$ doivent être envoyés pendant la *distribution*. On voit clairement que cela n'est pas optimal.

Pour déterminer les racines optimales, définissons $\delta_{i\to j}(r)$ comme l'indicateur exprimant si le message $\psi_{i\to j}$ doit être recalculé (dans ce cas, $\delta_{i\to j}(r) = 1$) ou non ($\delta_{i\to j}(r) = 0$) lorsque r est choisi comme racine. Dans IJTI, nous cherchons donc à minimiser $\delta(r) = \sum_{(k',k)\in\mathcal{A}(\mathcal{T})} \delta_{k'\to k}(r)$, qui correspond au nombre total de messages recalculés et envoyés. En se fondant sur la discussion de la section précédente, on peut écrire:

$$\delta_{i\to j}(r) = \begin{cases} 1 \text{ si } (\epsilon \in \mu_{i\to j} \text{ et } \{j\} = Adj_r(i)) \text{ ou} \\ \quad (\epsilon \in \mu_{i\to j} \text{ et } \{i\} = Adj_r(j) \text{ et } T \in \mu_{j\to i}) \\ 0 \text{ sinon} \end{cases} \tag{1}$$

La première ligne de l'équation (1) concerne les messages de *collecte* ($\{j\} = Adj_r(i)$). Elle stipule qu'un message $\psi_{i\to j}$ doit être recalculé seulement s'il est actuellement dans un « état invalide » ou si une nouvelle évidence ou des changements de structure affectent $\mathcal{V}_{\text{-}j}(i)$ ($\epsilon \in \mu_{i\to j}$). Quand ce n'est pas le cas, ce message est à jour et n'a pas besoin d'être recalculé. La deuxième ligne de l'équation (1) concerne les messages de *distribution* ($\{i\} = Adj_r(j)$)). Elle stipule que $\psi_{i\to j}$ doit être recalculé seulement

s'il existe une cible plus loin vers les feuilles du JT ($T \in \mu_{j\to i}$) et si des évidences ont été reçues dans $\mathcal{V}_{\text{-}j}(i)$ ou si certains messages venant de $\mathcal{V}_{\text{-}j}(i)$ ont été mis à jour ($\epsilon \in \mu_{i\to j}$). L'équation (1) peut être alors réécrite d'une façon plus compacte comme suit :

$$\delta_{i\to j}(r) = \begin{cases} 1 \text{ si } \epsilon \in \mu_{i\to j} \text{ et } (\{j\} = Adj_r(i) \\ \qquad \text{ou } (\{i\} = Adj_r(j) \text{ et } T \in \mu_{j\to i})) \\ 0 \text{ sinon} \end{cases} \tag{2}$$

Dans les figures 2c et 2d, $\delta(k_3) = 5$ et $\delta(i) = 4$. Dans ce cas, il vaut mieux choisir i comme racine plutôt que k_3, puisque cela nous épargne le calcul d'un message non nécessaire. Le théorème suivant montre l'existence des racines optimales et les caractérise.

THÉORÈME 6 (Racines optimales). — *Supposons que l'on a effectué le calcul des messages μ dans $\mathcal{T}$. Alors il existe $r \in \mathcal{V}(\mathcal{T})$ satisfaisant l'une des propriétés exhaustives et mutuellement exclusives suivantes:*

1. $(\mathcal{V}(\mathcal{T}), \mathcal{E}(\mathcal{T})) = (\{r\}, \emptyset)$
2. $\exists r' \in \mathcal{V}(\mathcal{T}) : \mu_{r'\to r} = \mu_{r\to r'} = T\epsilon$
3. $\forall k \in Adj(r) : \mu_{k\to r} \in \{T, \epsilon, \emptyset\}$

De plus, $r \in Argmin_{k\in\mathcal{V}(\mathcal{T})}\delta(k)$, c-à-d. que r est un nœud optimal par rapport aux calculs de l'inférence.

3.2. Une nouvelle inférence incrémentale

Dans ce paragraphe, nous proposons un nouvel algorithme conçu pour répondre au problème de l'incrémentalité dans l'inférence. On suppose qu'une première inférence par envoi de messages, a été déjà effectuée dans $\mathcal{T}$, en utilisant, par exemple, un algorithm de *collecte-distribution* dans une architecture de type *Lazy Propagation*. Des changements incrémentaux apparaissent par la suite et *IJTI* est appelé pour optimiser l'inférence. On rappelle que notre approche est dirigée par les *cibles*, ainsi, on recalcule seulement les messages de *collecte* invalides et on *distribue* les messages seulement vers les *cibles*. Ayant considéré ces hypothèses, l'algorithme proposé est décrit dans l'algorithme 1. Il s'agit d'un algorithme révisé d'envoi de messages pour calculer $\psi_{i\to j}$ seulement lorsque $\delta_{i\to j}(r) = 1$ pour tout i, j dans le JT modifié $\mathcal{T}$. Dans la ligne 5, une clique feuille i est telle que $|\text{Adj}(i)| = 1$. Nous signalons que le calcul des messages se fait de manière similaire aux algorithmes classiques basés sur l'utilisation du JT. La proposition suivante garantie la justesse de l'algorithme IJTI :

PROPOSITION 7. — *l'algorithme IJTI est valide, c-à-d. que le calcul des seuls messages $\psi_{i\to j}$ pour lesquels $\delta_{i\to j}(r) = 1$, pour tout $(i, j) \in \mathcal{A}(\mathcal{T})$, permet de calculer correctement les distributions* a posteriori *des variables* cibles.

Algorithme 1 : IJTI

input : $\mathcal{T}$ modifié, Q cliques des *cibles*
output : distributions *a posteriori* des *cibles*

```
// initialiser le nombre
// des voisins visités
// durant la collecte
```
pour $i \in \mathcal{V}(\mathcal{T})$ **faire**
  $i.nbVN \leftarrow 0$
Calculer les μ-labels de $\mathcal{T}$
Trouver r par le Théorème 6
$\mathbf{L} \leftarrow$ l'ensemble des feuilles de $\mathcal{T}$
```
// phase de collecte
```
pour chaque *clique* $i \in \mathbf{L}$ **faire**
  $p \leftarrow Adj_r(i)$
  si $\delta_{i \to p}(r) = 1$ **alors**
    Calculer $\psi_{i \to p}$
  $p.nbVN \leftarrow p.nbVN + 1$
  si $p \neq r$ **et** $p.nbVN = |Adj(p)| - 1$ **alors** $\mathbf{L} \leftarrow \mathbf{L} \cup \{p\}$
  $\mathbf{L} \leftarrow \mathbf{L} \setminus \{i\}$

```
// phase de distribution
```
$\mathbf{L} \leftarrow \{r\}$
pour chaque *clique* $i \in \mathbf{L}$ **faire**
  pour chaque $j \in Adj(i) \setminus Adj_r(i)$ **faire**
    si $\delta_{i \to j}(r) = 1$ **alors**
      Calculer $\psi_{i \to j}$
      $\mathbf{L} \leftarrow \mathbf{L} \cup \{j\}$
pour chaque *clique* $t \in Q$ **faire**
  Calculer les distributions *a posteriori* des *cibles* dans la clique t
retourner les distributions *a posteriori*

4. Résultats expérimentaux

Dans cette section, nous mettons en évidence l'efficacité de notre algorithme via la comparaison du gain obtenu avec son utilisation à la place d'un autre algorithme d'inférence de JT non-incrémental. Ce gain vaut $1-\delta(r)/(2|\mathcal{E}(\mathcal{T})|)$, c-à-d., le pourcentage des messages non nécessaires que IJTI évite de calculer par rapport aux algorithmes d'inférence classiques qui envoient des messages dans les deux directions sur toutes les arêtes [7]. Pour ce faire, nous avons effectué des tests en utilisant la librairie aGrUM [8] sur 9 BN réels de complexités différentes ainsi que sur des BN aléatoirement générés. Ces derniers contiennent $nbNodes$ variables aléatoires, ($6 \leq nbNodes \leq 900$, voir la figure 4) et, pour chaque valeur de $nbNodes$, 3 BN sont générés avec $nbArcs$ arcs, $nbArcs$ est choisi aléatoirement dans l'intervalle $[nbNodes - 1, 4/3 * nbNodes - 1]$. Pour simuler l'incrémentalité et induire des messages invalides dans $\mathcal{T}$, nous avons choisi aléatoirement, pour chaque inférence, un ensemble de *cibles* et d'observations. Les figures 3 et 4 montrent les gains moyens résultants et leurs écarts type (les barres

7. Il est à noter que nous effectuons des comparaisons en nombre de messages. Le gain "réel" nécessite des comparaisons en nombre de calculs, cette métrique est toutefois plus dispersée et donc difficile à interpréter.
8. http://agrum.lip6.fr

d'erreurs [9]) sur 20 requêtes d'inférence incrémentale. Notons que le comportement de l'algorithme est le même pour les BN réels et artificiels. Comme l'on pouvait s'y attendre, plus les modifications sont petites, plus le gain est grand. Notons également que le gain est très sensible à la taille du BN.

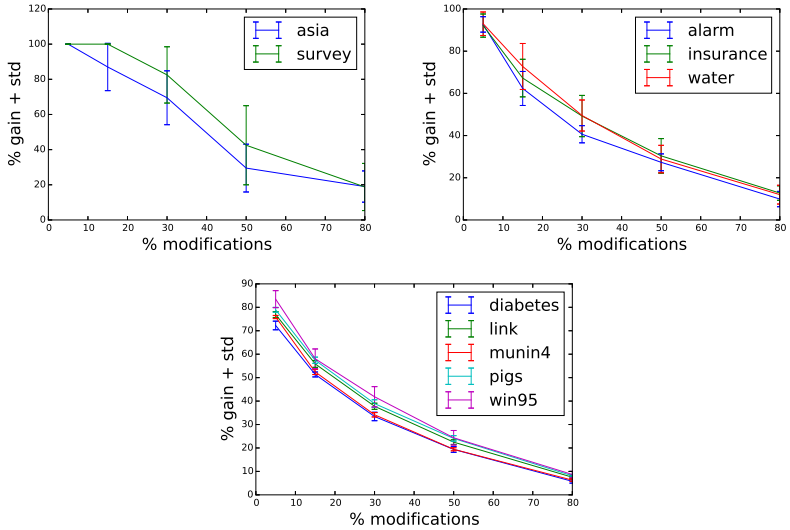

*Figure 3. le gain d'*IJTI *pour des BNs réels*

5. Conclusion

Dans cet article, nous avons introduit *IJTI*, un nouvel algorithme d'inférence incrémentale par JT qui est conçu pour les systèmes multi-cibles dynamiques. En supposant qu'une première et inférence complète a été effectuée, il extrait une racine optimale et optimise l'inférence en conséquence. La justesse mathématique de ces deux optimisations est prouvée et la partie expérimentale souligne des économies importantes de notre approche comparé aux méthodes classiques. Concernant de futurs travaux, nous visons l'amélioration de notre algorithme en considérant notamment des techniques de caches pour la détermination des racines. Nous envisageons également l'application d'IJTI aux Modèles Relationnels Probabilistes afin d'accélérer leur inférence et, in fine, garantir un couplage efficace de notre approche avec les RBS, où l'on estime apporter une amélioration de la gestion des incertitudes.

9. Dans la figure 3.a, le fait d'avoir un petit nombre de nœuds et d'arcs dans les BN implique l'absence de toute modification pour les pourcentages de modifications inférieurs à 10%, d'où l'absence des barres d'erreurs.

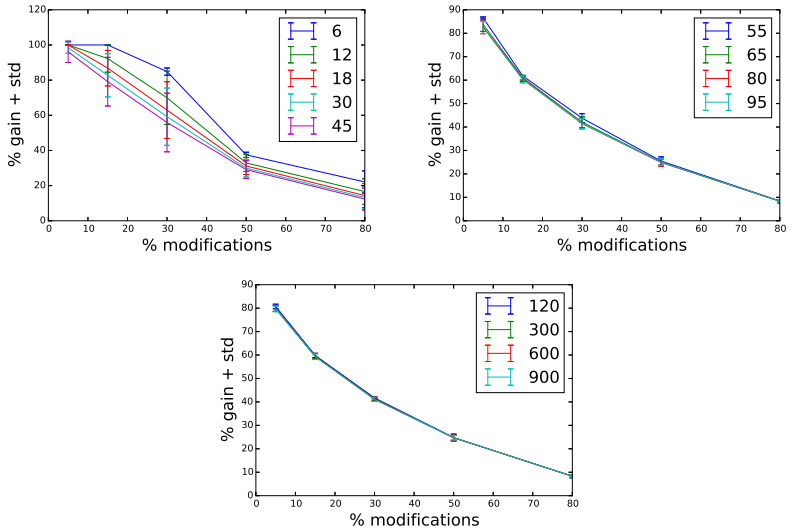

Figure 4. le gain d'IJTI sur des tailles différentes de BNs artificiels

Bibliographie

Buchanan B. G., Shortliffe E. H. (1984). *Rule based expert systems: The Mycin experiments of the Stanford heuristic programming project*. Addison-Wesley.

D'Ambrosio B. (1993). Incremental probabilistic inference. In *Proceedings of the 9th conference on uncertainty in artificial intelligence (uai)*, p. 301–308.

Darwiche A. (1998). Dynamic join trees. In *Proceedings of the fourteenth conference on uncertainty in artificial intelligence (uai)*, p. 97–104.

Darwiche A. (2009). *Modeling and Reasoning with Bayesian Networks*.

Dean T., Kanazawa K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, vol. 5, n° 2, p. 142–150.

Flores M. J., Gámez J. A., Olesen K. G. (2003). Incremental compilation of Bayesian networks. In *Proceedings of the nineteenth conference on uncertainty in artificial intelligence (uai)*, p. 233–240.

Heckerman D. E., Shortliffe E. H. (1992). From certainty factors to belief networks. *Artificial Intelligence in Medicine*, vol. 4, n° 1, p. 35–52.

Jensen F., Lauritzen S., Olesen K. (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, vol. 4, p. 269–282.

Koller D., Friedman N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.

Koller D., Pfeffer A. (1998). Probabilistic frame-based systems. In *Proceedings of the 15th national conference on artificial intelligence (aaai)*, p. 580–587.

Lauritzen S., Spiegelhalter D. J. (1988). Local computations with probabilities on graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society*, vol. 50, n° 2, p. 157–224.

Li W., Beek P. van, Poupart P. (2006). Performing incremental Bayesian inference by dynamic model counting. In *Proceedings of the national conference on artificial intelligence (aaai)*, p. 1173-1179.

Lin Y., Druzdzel M. J. (1998). Relevance-based sequential evidence processing in Bayesian networks. In *Proceedings of the eleventh international florida artificial intelligence research society conference (flairs)*, p. 446–450.

Madsen A. L., Jensen F. V. (1999). Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, vol. 113, n° 1–2, p. 203 - 245.

Murphy K. P. (2002). *Dynamic Bayesian networks: Representation, inference and learning*. Thèse de doctorat non publiée, UC Berkeley.

Pearl J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann.

Pfeffer A. J. (2000). *Probabilistic reasoning for complex systems*. Thèse de doctorat non publiée, Stanford University.

Robinson J., Hartemink A. (2009). Non-stationary dynamic Bayesian networks. In *Advances in neural information processing systems 21 - proceedings of the 2008 conference*, p. 1369–1376.

Shenoy P., Shafer G. (1990). Axioms for probability and belief-function propagation. In *Proceedings of the conference uncertainty in artificial intelligence*, vol. 4, p. 169—198.

Torti L., Gonzales C., Wuillemin P.-H. (2013). Speeding-up structured probabilistic inference using pattern mining. *International Journal of Approximate Reasoning*, vol. 54, n° 7, p. 900–918.

Remerciements

Ce travail a été partiellement financé par IBM France Lab, bourse CIFRE ANRT #2014/421.

Appendice: Démonstrations

Démonstration de la Proposition 5: Notons que $\mathcal{V}_{\text{-}j}(i) = \{i\} \cup \bigcup_{k \in \text{Adj}_{\text{-}j}(i)} \mathcal{V}_{\text{-}i}(k)$ et pour $k \in \text{Adj}_{\text{-}j}(i)$, $l' \in \mathcal{V}_{\text{-}i}(k)$, on a $Adj_j(l') = Adj_i(l')$. En utilisant la définition 4, on peut ainsi réécrire $\mu_{i \to j}$ comme suit:

$$\mu_{i\to j} = \bigcup_{\substack{l' \in \mathcal{V}_{\text{-}j}(i) \\ \{l\} = Adj_j(l')}} \lambda_{l'\to l} = \lambda_{i\to j} \cup \bigcup_{k \in \text{Adj}_{\text{-}j}(i)} \overbrace{\bigcup_{\substack{l' \in \mathcal{V}_{\text{-}i}(k) \\ \{l\} = Adj_j(l')}} \lambda_{l'\to l}}^{\mu_{k\to i}} = \lambda_{i\to j} \cup \bigcup_{k \in \text{Adj}_{\text{-}j}(i)} \mu_{k\to i}$$

■

Démonstration de l'exclusivité mutuelle des propriétés du Théorème 6: si la propriété 1) est vérifiée, alors $\mathcal{T}$ ne contient aucune arête, et par conséquent, les propriétés 2) et 3) ne peuvent pas être satisfaites.

Supposons maintenant qu'il existe r_1, r_1' tels que $\mu_{r_1' \to r_1} = \mu_{r_1 \to r_1'} = T\epsilon$ (propriété 2). soit r_2 une clique quelconque de $\mathcal{V}(\mathcal{T})$. Sans perte de généralité, supposons que r_1 appartienne à la chaîne $i_1 = r_2, i_2, \ldots, i_p = r_1'$ entre r_2 et r_1'. Alors, en appliquant la Proposition 5, $\mu_{i_2 \to r_2} \supseteq \mu_{i_3 \to i_2} \supseteq \cdots \supseteq \mu_{r_1' \to r_1} = T\epsilon$. Par conséquent, les propriétés 2) et 3) sont insatisfiables simultanément. ■

Démonstration de l'existence de r du Théorème 6: si $\mathcal{A}(\mathcal{T}) = \emptyset$, alors la propriété 1) est vraie et r est l'unique noœud de $\mathcal{T}$. Dans le reste de la démonstration, supposons que $\mathcal{A}(\mathcal{T}) \neq \emptyset$. S'il existe une arête $(i,j) \in \mathcal{E}(\mathcal{T})$ telle que $\mu_{i \to j} = \mu_{j \to i} = T\epsilon$, alors $r = i$ vérifie la propriété b). Sinon, ni les propriété 1) et 2) ne sont vraies. Supposons que la propriété 3) n'est pas vérifiée non plus. Alors pour toutes les arêtes (i,j), entre $\mu_{i \to j}$ et $\mu_{j \to i}$, un seul de ces messages est égal à $T\epsilon$ et l'autre appartient à $\{\emptyset, \epsilon, T\}$.

Soit (i_0, j_0) tel que $\mu_{i_0 \to j_0} = T\epsilon$ et $\mu_{j_0 \to i_0} \neq T\epsilon$. Alors, si $|\text{Adj}(i_0)| = 1$, la clique i_0 satisfait la propriété 3), ce qui est une contradiction. Comme nous avons aussi supposé que la propriété 2) n'est pas vérifiée, il existe $i_1 \in \text{Adj}(i_0)$ tel que $\mu_{i_1 \to i_0} = T\epsilon$ et $\mu_{i_0 \to i_1} \neq T\epsilon$. Le même raisonnement est toujours vrai pour i_1. Si i_1 est une feuille, on obtient une contradiction avec la propriété 3). Sinon, i_1 possède un voisin i_2 tel que $\mu_{i_2 \to i_1} = T\epsilon$ et $\mu_{i_1 \to i_2} \neq T\epsilon$. Par récurrence, on crée une chaîne $i_1, \ldots, i_n$ de taille maximale. Cette chaîne est forcément finie puisque $\mathcal{T}$ est lui même un arbre fini, ainsi la clique i_n est une feuille qui, de plus, satisfait la propriété, ceci est encore une fois une contradiction. En conséquence, lorsque les propriétés 1) et 2) ne sont pas satisfaites, la propriété 3) est forcément vérifiée. ■

Dorénavant, on peut prouver séparément l'optimalité de chaque propriété du Théorème 6, puisque ces propriétés sont mutuellement exclusives:

Démonstration de l'optimalité de la propriété 1) du Théorème 6: r est la seule racine à prendre. ■

LEMME 8. — *Soit $i, j \in \mathcal{V}(\mathcal{T})$ tels que $\epsilon \in \mu_{j \to i}$ et $\mu_{i \to j} = \emptyset$, alors $\forall l \in \mathcal{V}_{-j}(i)$: $\delta(l) = \delta(j) + len(l - j)$.*

Démonstration. Notons que lorsque $\epsilon \notin \mu_{j \to i}$, $\mathcal{T}$ est à jour dans l'inférence actuelle, on n'aura donc pas besoin d'effectuer des calculs. La preuve est établie par récurrence sur $n = len(l - j)$. Pour $n = 1$, on a $l = i$, donc par l'équation (2) et le fait que $\epsilon \in \mu_{j \to i}$ et $i \in Adj_i(j)$, on obtient $\delta_{j \to i}(i) = 1$. Par conséquent, $\delta(i) = \sum_{(k',k) \in \mathcal{A}(\mathcal{T}) \setminus \{(j,i)\}} \delta_{k' \to k}(i) + 1$. Or, puisque $T \notin \mu_{i \to j}$, on a $\delta_{j \to i}(j) = 0$; donc $\delta(j) = \sum_{(k',k) \in \mathcal{A}(\mathcal{T})) \setminus \{(j,i)\}} \delta_{k' \to k}(j)$. Puisque $\epsilon \notin \mu_{i \to j}$, $\delta_{i \to j}(i) = \delta_{i \to j}(j) = 0$. Pour $(k', k) \neq (i, j), (j, i)$, on a $Adj_i(k) = Adj_j(k)$ et $Adj_i(k') = Adj_j(k')$. Dans ce cas, il s'ensuit que $\delta_{k' \to k}(i) = \delta_{k' \to k}(j)$. On en conclue donc que $\delta(i) = \delta(j) + 1$.

Supposons maintenant que cette propriété est vraie jusqu'à l'ordre $n-1>1$, montrons qu'elle le restera pour n. soit l tel que $len(l-j)=n-1$ et $\{p\}=Adj_i(l)$. Alors $\delta(l)=1+\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(p,l)\}}\delta_{k'\to k}(l)$ car $\delta_{p\to l}(l)=1$ (puisque $\epsilon\in\mu_{p\to l}$ et $\{l\}=Adj_l(p)$). Étant donné que $T\notin\mu_{l\to p}$, on obtient $\delta_{p\to l}(p)=0$, ce qui implique que $\delta(p)=\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(p,l)\}}\delta_{k'\to k}(p)$. Maintenant, en utilisant le même raisonnement que pour le cas $n=1$ et en remarquant que $\delta_{l\to p}(p)=\delta_{l\to p}(l)=0$ car $\epsilon\notin\mu_{l\to p}$, on conclut que $\delta(l)=1+\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(p,l)\}}\delta_{k'\to k}(l)=1+\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(p,l)\}}\delta_{k'\to k}(p)=1+\delta(p)$. L'hypothèse du récurrence appliquée à l, où $len(l-j)=n-1$, assure que : $\delta(l)=1+\delta(p)=1+n-1+\delta(j)=\delta(j)+n$. ■

LEMME 9. — *Soit* $\mathcal{V}_1=\{r\in\mathcal{V}(\mathcal{T}):\exists k\in\text{Adj}(r),\mu_{r\to k}=\mu_{k\to r}=T\epsilon\}$*, alors pour tout* r,r' *in* $\mathcal{V}_1$ *on a* $\delta(r)=\delta(r')$.

Démonstration. Supposons que $|\mathcal{V}_1|>1$. D'après la Proposition 5, les nœuds dans $\mathcal{V}_1$ forment un sous graphe connexe. Soit $r,r'\in\mathcal{V}_1$ tels que $(r,r')\in\mathcal{E}(\mathcal{T})$. Enfin, soit $(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(r,r'),(r',r)\}$. Si $k'\notin\{r,r'\}$, alors soit $k=r$, $k=r'$ soit $k\notin\{r,r'\}$ et dans tous les cas on a: $Adj_r(k')=Adj_{r'}(k')$, ainsi $\delta_{k'\to k}(r)=\delta_{k'\to k}(r')$. Sinon, soit $k'=r'$ alors $k\neq r$ et on a aussi [10] $Adj_r(k)=Adj_{r'}(k)$ et encore une fois, $\delta_{k'\to k}(r)=\delta_{k'\to k}(r')$. Par conséquent, $\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(r,r'),(r',r)\}}\delta_{k'\to k}(r)=\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(r,r'),(r',r)\}}\delta_{k'\to k}(r')$. En utilisant l'équation (2), on obtient:
$\delta_{r\to r'}(r)+\delta_{r'\to r}(r)=\delta_{r\to r'}(r')+\delta_{r'\to r}(r')=2$. On en conclut que
$\delta(r)-\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(r,r'),(r',r)\}}\delta_{k'\to k}(r)=\delta(r')-\sum_{(k',k)\in\mathcal{A}(\mathcal{T})\setminus\{(r,r'),(r',r)\}}\delta_{k'\to k}(r')$.
Ainsi $\delta(r)=\delta(r')$. ■

Démonstration de l'optimalité de la propriété 2 du Théorème 6: Avec les notations de la propriété 2), il suffit de prouver que pour tout i n'appartenant pas à $\mathcal{V}_1,\delta(r)\leq\delta(i)$ [11]. Sans perte de généralité, supposons que $i\in\mathcal{V}_{\text{-}r'}(r)$. Soit $(k,k')\in\mathcal{A}(i-r)$, où $\mathcal{A}(i-r)$ est l'ensemble des arcs induits de $i-r$. Alors on a soit $\{k'\}=Adj_r(k)$ soit $\{k\}=Adj_r(k')$. Supposons par exemple que $\{k'\}=Adj_r(k)\,,k\neq r$, le deuxième cas est traité de la même façon. Alors $\mu_{k'\to k}=T\epsilon$, et par application de l'équation (2), on récapitule les résultats dans le tableau suivant:

$\mu_{k\to k'}$	$\delta_{k\to k'}(i)+\delta_{k'\to k}(i)$	$\delta_{k\to k'}(r)+\delta_{k'\to k}(r)$
$\emptyset$	1	0
T	1	1
ϵ	2	1

On conclut que $\sum_{(k',k)\in\mathcal{A}(i-r)}\delta_{k'\to k}(r)\leq\sum_{(k',k)\in\mathcal{A}(i-r)}\delta_{k'\to k}(i)$.(1)
Maintenant pour $(k,k')\notin\mathcal{A}(i-r)$, il est facile de voir que $\delta_{k\to k'}(i)=\delta_{k\to k'}(r)$

10. si $k'=r$ alors $k\neq r'$ et l'égalité est vérifiée.
11. Tous les nœuds sont équivalents en termes de calcul si $\forall i\in\mathcal{V}(\mathcal{T}),i\in\mathcal{V}_1$, puisque $\mathcal{V}(\mathcal{T})=\mathcal{V}_1$

et, par conséquent, $\sum_{(k,k')\in\mathcal{A}(\mathcal{T})\setminus\mathcal{A}(i-r)} \delta_{k'\to k}(r) = \sum_{(k,k')\in\mathcal{A}(\mathcal{T})\setminus\mathcal{A}(i-r)} \delta_{k'\to k}(i)$. (2). En comparant (1) et (2), on obtient $\delta(r) \leq \delta(i)$ pour $i \notin \mathcal{V}_1$. Jusqu'ici, nous avons obtenu, par le lemme 9, que pour tout i dans $\mathcal{V}_1$, $\delta(r) = \delta(i)$ et pour tout i n'appartenant pas à $\mathcal{V}_1$, on a $\delta(r) \leq \delta(i)$. On en déduit donc que $r \in Argmin_{i\in\mathcal{V}(\mathcal{T})}$ $\delta(i)$. ■

Démonstration de l'optimalité de la propriété 3) du Théorème 6: soit $i \in \mathcal{V}(\mathcal{T})$ tel que $i \neq r$.
premier cas: $\mu_{Adj_i(r)\to r} = \emptyset$. Supposons que $T, \epsilon \in \mathcal{V}_{\text{-}i}(r)$, car sinon il n'y a aucun besoin d'effectuer un quelconque calcul d'inférence, puisque il s'agit d'une absence de requête ou de modifications dans $\mathcal{T}$; donc par le lemme 8, on a $\delta(i) = \delta(r) + len(i–r)$ car $i \in \mathcal{V}_{\text{-}r}(Adj_i(r))$. Par conséquenct, $\delta(r) < \delta(i)$.
deuxième cas: Nous omettons le cas $\mu_{Adj_i(r)\to r} \in \{T, \epsilon\}$, mais le lecteur pourra utiliser la même méthode que dans la preuve de la propriété 2) et le fait que pour tout $k, k' \in i-r$ tels que $\{k'\} = Adj_r(k) : \mu_{k\to k'} = \mu_{i\to Adj_r(i)}$ et examiner $\delta_{k'\to k}(r)$ et $\delta_{k'\to k}(i)$. ■

Démonstration de la Proposition 7 : Étant donné une racine r, $\delta_{i\to j}(r)$ correspond, par construction, au fait que $\psi_{i\to j}$ est nécessaire durant l'inférence actuelle et a été invalidé durant la précédente. En conséquence, l'inférence actuelle a besoin uniquement de recalculer un tel message pour tout i, j dans $\mathcal{V}(\mathcal{T})$. ■